CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Fluctuations in Parkinson's disease symptoms and effectiveness of medications over the menstrual cycle
Authors	Moore, Sarah C.
Publication date	2026-01-31
Original Citation	Moore, S. 2026. Fluctuations in Parkinson's disease symptoms and effectiveness of medications over the menstrual cycle. MRes Thesis, University College Cork.
Type of publication	Masters thesis (Research)
Rights	© 2026, Sarah Moore. - https://creativecommons.org/licenses/by/4.0/
Download date	2026-10-26 11:59:34
Item downloaded from	https://hdl.handle.net/10468/18912

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Ollscoil na hÉireann, Corcaigh

National University of Ireland, Cork



Fluctuations in Parkinson's disease symptoms and effectiveness of medications over the menstrual cycle

Thesis presented by

Sarah Moore, BSc

[ORCID ID: 0009-0003-7807-3277]

for the degree of

Master of Research in Medical and Health Science

Department of Pharmacology and Therapeutics,

School of Medicine,

College of Medicine and Health

University College Cork

Lead Supervisor: Prof. Aideen Sullivan

Co-Supervisors: Dr. Lucy Collins Stack and Dr. Aonghus Lavelle

2026

Table of Contents

TABLE OF CONTENTS **1**

ABSTRACT **5**

ABBREVIATIONS **7**

1. INTRODUCTION **9**

Figure 1: Age and sex dependent incidence and prevalence rates of PD 9

1.1. Sex-Specific Differences in PD 10

1.1.1. Sex-Specific Differences in Symptom Presentation *10*

1.1.1.1. Motor Symptom Presentation 10

Figure 2: Sex Differences in PD Symptom Prevalence 11

1.1.1.2. Non-Motor Symptom Presentation 12

1.1.2. Sex-Specific Differences in Responses to Medication *12*

1.1.3. Sex Differences in Parkinson's Disease Pathophysiology *15*

1.1.3.1. Sex-Specific Differences in Dopaminergic neuronal function and protection 15

1.1.3.2. Sex-Specific Differences in PD-linked Genetic Factors 16

1.1.3.3. Sex-Specific Differences in Biomarkers 16

1.1.3.4. Sex-Specific Differences in Neuroinflammation 16

1.1.3.4.1 Sex-Specific Differences in Prostaglandin Pathways 17

Table 1: Summary of Sex-Specific Differences in Parkinson's Disease Pathophysiology 0

1.2. PD and Hormonal Events 0

1.2.1. Menstruation *0*

Table 2: Summary of studies investigating menstrual cycle effects on PD symptoms 3

Figure 3: Relationship Between Menstrual Cycle Hormone Levels and PD Symptoms 5

1.2.1.1. Estrogen 5

1.2.1.1.1 Neuroprotective Evidence for Estrogen in PD 5

1.2.1.1.2 Mechanisms of Estrogen-Mediated Neuroprotection in PD 6

1.2.1.1.3 Selective Estrogen Receptor Modulators as Therapeutic Targets in PD 7

1.2.1.2. Progesterone 7

1.2.1.2.1 Neuroprotective Evidence for Progesterone in PD 7

1.2.1.2.2 Mechanisms of Progesterone-Mediated Neuroprotection in PD 7

1.2.1.2.3 Translational Challenges and Therapeutic Potential of Progesterone in PD 8

1.2.2. Menopause *9*

1.2.2.1. Perimenopause 10

1.2.2.2. HRT 10

1.2.3. Pregnancy *11*

1.3. Apps for PD 13

2. METHODS. **14**

2.1. Ethical Approval. 14

2.2. Participant Recruitment. 15

2.2.1. Recruitment. *15*

2.2.2. Inclusion Criteria. *15*

2.3. Survey Data Collection. 15

2.3.1. Survey Form Creation. *15*

Follow-Up Survey 16

2.3.2. Survey Question Formatting. *16*

Table 3: PD Medications 17

2.3.3 User Research. *19*

2.4. Survey Data Cleaning and Coding 19
2.5. Data Analysis 19
2.5.1. Initial Survey *19*
2.5.1.1. Data preparation and analysis (Script 03) 19
2.5.1.2. Demographic characterisation by menstruation status (Script 04) 20
2.5.1.3. Demographic count visualisation by menstruation status (Script 04.1) 20
2.5.1.4. Symptom and medication comparison by menstruation status (Scripts 03.3 & 05) 20
2.5.1.5. HRT usage in postmenopausal women (Script 03.2.1) 21
2.5.1.5.1. Age-stratified HRT analysis (03.2.1.1) 21
2.5.1.6. Contraceptive usage in menstruating women (Script 03.2.2) 21
2.5.2. MMM App *22*
2.5.2.1. App data processing and preparation (Script 1) 22
2.5.2.2. Mini data table creation (Script 2) 22
Table 4: Example individual participant data table 24
2.5.2.3. Participant flow chart statistics (Script 14.3) 28
2.5.2.4. Matching app IDs to initial survey IDs (Script 2.4) 28
2.5.2.5. App symptom prevalence and severity (Script 14.1) 28
2.5.2.6. Multi-cycle participant data export (Script 2.5) 29
2.5.2.7. Bleeding vs non-bleeding population analysis (Script 8) 29
2.5.2.8. Motor vs non-motor between period tracking groups (Script 8.1) 30
2.5.2.9. Menstruating vs non-menstruating days analysis (Script 8.2) 30
2.5.2.10. Post-menstrual symptom burden analysis (Script 9) 30
2.5.2.11. Selected participant timeline visualisation (Script 10) 31
2.5.2.12. Menstrual phase symptom burden analysis (Script 11 & 11.1) 31
2.5.2.13. Symptom entry analysis across menstrual phases (Script 6) 31
2.5.2.14. Individual symptom boxplot analysis across menstrual phases (Script 4.2) 32
2.5.2.15. Medication categorisation (Script 2.6) 32
2.5.2.16. L-DOPA dose comparison between menstruating and postmenopausal participants (Script 15) . 33
2.5.2.17. Follow-Up cohort selection (Script 03) 33
2.5.3. Follow- Up Survey *34*
2.5.3.1. Follow-up survey data preparation and analysis (Script 103 &102.0.) 34
2.5.3.2. Follow-up survey symptom and medication comparison by menstruation status (Script 105) 34
2.5.3.3. App experience response extraction (Script 103.2) 34
2.5.3.4. App usability visualisations (Script 103.3) 34
Table 5. App Usability Survey Questions Thematic Categorisation 35

3. RESULTS 37

3.1. Participant Flow Chart Statistics *37*
Figure 4: Participant Flow Chart from Initial Survey to App Engagement 37
3.2 Survey Data *38*
3.2.1. Demographics by Menstruation Status 38
Table 6: Demographics by Menstruation Status 39
3.2.2. Symptom and PD Medication Prevalence by Menstruation Status 40
Figure 5A-D: Symptoms and PD Medication by Menstruation Status 42
Table 7: Significant Differences between Menstruation Status and Clinical Variables 45
3.2.3. Age-Stratified HRT Analysis in Postmenopausal Women 46
Figure 6A-F: Motor and non-motor symptoms by HRT status split by age groups 47
3.2.4. Contraceptive Usage Analysis in Menstruating Women 53
Figure 7: Symptom Prevalence by Hormonal Contraceptive Status in Menstruating Women 54
3.3 App Data *56*
3.3.1 App Symptom and Medication Analysis *56*
3.3.1.1. L-DOPA Dose Comparison Between Menstruating and Postmenopausal Women 56
Figure 8: Daily L-DOPA Dose Comparison Between Menstruating and Postmenopausal Women 56

3.3.1.2. App Symptom Reporting by Menstruation Status 56
Table 8: Motor and non-motor symptoms by menstruation status among app participants 57
3.3.1.3. Motor vs Non-Motor Symptom Burden by Menstruation Tracking Status 58
Figure 9: Motor and Non-Motor Symptom Burden by Menstruation Tracking Status 58
3.3.1.4. Motor vs Non-Motor Symptoms: Menstruating vs Non-Menstruating Days 58
Figure 10: Motor vs Non-Motor Symptoms: Menstruating vs Non-Menstruating Days 59
3.3.1.5. Post-Menstrual Symptom Burden Analysis 60
Figure 11: Motor and Non-Motor Symptom Burden During and After Menstruation 60
3.3.1.6. Menstrual phase symptom burden analysis 61
Figure 12: Symptom burden across menstrual cycle phases 61
3.3.1.7. Dyskinesia severity analysis 62
Figure 13: Dyskinesia severity across menstrual cycle phases 62
3.3.2. App Usage and Engagement *63*
3.3.2.1. Symptom entry analysis across menstrual phases 63
Figure 14: Average no. of symptoms per day by phase 63
3.3.2.2. App Usability Survey Results 64
Figure 15: App Usability Survey Thematic Summary Analysis 64
3.3.2.3. App Tracking Experience 64

4. DISCUSSION 65

4.1. Study Population 65
4.2. Differences between menstruating and postmenopausal participants 66
4.2.1. Medications *66*
4.2.2. Symptoms *66*
Rigidity 66
Postural Instability 67
Hyposmia 67
Constipation 68
PMS-related 69
Hot Flushes 69
Excessive daytime sleepiness 70
Dyskinesia 70
Depression/Anxiety 71
4.3. Hormonal Medication 72
4.3.1. HRT *72*
4.3.2. Contraceptives *72*
Imbalance 72
Skin 73
4.4. Menstrual Cycle Tracking Findings 74
Menstruating vs non-menstruating *74*
Menstrual Cycle Findings *74*
Motor and non-motor symptoms across menstrual cycle 74
App Usage Timing 75
Dyskinesias across the menstrual cycle 75
4.5. App Usability and Tracking Experience 76
4.5.1. MMM App Survey Analysis *76*
4.5.1.1. Future Directions for the MMM App 77
4.5.2. MMM Tracking Thematic Analysis *78*
4.6. Limitations of the study 79
4.7. Conclusion and Future Directions 81

5. BIBLIOGRAPHY 83

6. ACKNOWLEDGEMENTS **109**

7. SUPPLEMENTARY MATERIAL **110**

Figure S1: Initial Survey Completion Message 110
Table S1: Extended Demographics by Menstruation Status 112
Figure S2: Demographic Counts by Menstruation Status 114
Figure S3: HRT Analysis of Symptom in Users vs Non-Users 121
Table S2: MMM App Motor Symptoms: Menstruating vs Postmenopausal Comparison 123
Table S3: MMM App Non-Motor Symptoms: Menstruating vs Postmenopausal Comparison 124
Table S4: MMM App Medication Categories: Menstruating vs Postmenopausal Comparison 126
Figure S4(A-Q): Individual Participant Menstrual Cycle and Symptom Timelines 128
Table S5: Motor Symptom Abbreviations in the Initial Survey 139
Table S6: Non-motor Symptom Abbreviations in the Initial Survey 140
Table S7: Motor Symptom Options and Abbreviations in the MMM app 142
Table S8: Non-motor Symptom Options and Abbreviations in the MMM App 143
Figure S5: App Usability Follow-up Survey Analysis 148
Figure S6: Copy of Initial Survey 149
Figure S7: Copy of Follow-up Survey 158
Initial Survey Scripts *166*
Script 03 166
Script 04 185
Script 04.1 192
Script 03.3. 198
Script 05 204
Script 03.2.1. 211
Script 03.2.1.1. 219
Script 03.2.2. 227
App Scripts *236*
Script 1 236
Script 2 254
Script 14.3 282
Script 2.4 286
Script 14.1 298
Script 2.5 326
Script 8 328
Script 8.1 338
Script 8.2 350
Script 9 369
Script 10 384
Script 11 400
Script 11.1 418
Script 6 437
Script 4.2 453
Script 2.6 463
Script 15 476
Follow-up Survey Scripts *488*
Script 102. 488
Script 103. 491
Script 105. 505
Script 103.2. 512
Script 103.3. 517

Abstract

INTRODUCTION: The sex of a person with Parkinson's disease (PD) impacts the degree and features of disability they experience, yet women with PD (WwPD) remain understudied. Published literature on menstruation, pregnancy, and menopause in WwPD suggests a pattern where estrogen decline seems to correlate with worsened PD symptoms or higher disease risk. Current understanding of menstrual-related PD symptom fluctuations is limited, highlighting the need for prospective longitudinal research.

AIMS: The primary aim was to compare PD symptoms and medication patterns between menstruating and postmenopausal women, to investigate the effects of HRT and contraceptives on these, and to examine whether PD symptoms fluctuate across menstrual cycle phases, through prospective longitudinal data collection.
A secondary aim was to assess the usability and user experience of a mobile application designed for WwPD to track symptoms, medications, and menstrual cycles.

METHODS: Women aged >18 years with PD diagnoses were recruited through PD associations and support groups to complete REDCap surveys capturing demographics, symptoms, and medications. Of n=493 participants completing the initial survey (n=125 menstruating, n=283 postmenopausal), n=131 used the mobile app for longitudinal symptom, medication, and menstrual cycle tracking, with n=40 completing a follow-up survey. Statistical analysis was completed using R version 4.4.1.

RESULTS: Initial survey analysis of menstruating (n=125) and postmenopausal (n=283) women revealed significant differences in symptoms between the groups. For motor symptoms, rigidity was higher ($p<0.001$), whilst postural instability was lower ($p=0.007$) in menstruating women. For non-motor symptoms, depression/anxiety ($p=0.026$), pain ($p=0.016$), and mood swings ($p<0.001$) were higher, whilst hyposmia ($p=0.003$) and constipation ($p=0.042$) were lower in menstruating women. No differences in medication usage were observed between the groups. HRT users aged 50-54 years had lower tremor prevalence ($p=0.006$) than those who did not use HRT, and contraceptive users had fewer skin problems than non-users . App tracking data from 48 menstruating and 83 postmenopausal participants showed that dyskinesia was higher in menstruating women ($p=0.045$). Multiple premenstrual syndrome (PMS) related non-motor symptoms were higher in menstruating women, for example, irritability ($p=0.003$), crying spells ($p=0.014$), hot flushes ($p=0.007$), pelvic cramps ($p=0.001$), mood swings ($p=0.043$), increased appetite ($p=0.026$), and excessive daytime sleepiness ($p=0.026$). Analysis of menstrual phase in eight women showed no significant differences in motor and non-motor symptom burden across the cycle phases.

Assessment of app usability found that satisfaction that varied somewhat across usability categories, ranging from 4.31 to 5.35 on a seven-point scale.
CONCLUSIONS: This study demonstrates consistent differences in both motor and non-motor symptoms between menstruating and postmenopausal WwPD, suggesting hormonal influences on symptom presentation. There were no significant differences in motor and non-motor symptom burden across the phases of the menstrual cycle.

Abbreviations

ADL - Activities of Daily Living

AUC - Area under the curve

BDNF - Brain-derived neurotrophic factor

BMI - Body mass index

COMT - Catechol-O-Methyltransferase

CSF - Cerebrospinal fluid

DA - Dopamine

DAT - Dopamine transporter

DBS – Deep brain stimulation

DRD2 - Dopamine receptor D2

EDS – Excessive daytime sleepiness

ER – Estrogen receptor

FSH - Follicle-stimulating hormone

GFAP - Glial fibrillary acidic protein

GPER - G protein-coupled Estrogen Receptor

IL - Interleukin

iNOS - Inducible Nitric Oxide Synthase

LB - Lewy body

L-DOPA - Levodopa

LID - Levodopa-induced dyskinesia

LOPD - Late onset Parkinson's Disease

LPS - Lipopolysaccharide

LRRK2 - Leucine-rich repeat kinase 2

MAO-B - Monoamine oxidase B

MFN2 - Mitofusion 2

MMSE - Mini-Mental State Examination

MPTP - 1-methyl-4-phenyl-1,2,3,6-tetrahydropyridine

NMFs - Non-motor fluctuations

NMS – Non-motor symptom(s)

NSCs - Neural stem cells

NO - Nitric Oxide

OPA1 - Optic Atrophy 1

OR- Odds Ratio

PD - Parkinson's disease

PGC-1α - Peroxisome proliferator-activated receptor gamma co-activator 1 alpha

PMS - Premenstrual syndrome

PwP - People with Parkinson's disease

QoL – Quality of life

REM – Rapid Eye Movement

SABV - Sex as a biological variable

SEM - Standard error of the mean

SERM - Selective estrogen receptor modulator

SN – Substantia Nigra

TNF - Tumour necrosis factor

t1/2 – Half-life

UPDRS - Unified Parkinson's Disease Rating Scale

WwPD - Women with Parkinson's disease

YOPD - Young onset Parkinson's disease

6-OHDA - 6-hydroxydopamine

1. Introduction

The importance of delivering tailored healthcare that acknowledges and treats heterogeneity in treatments and support has been increasingly acknowledged and researched. Despite this growing recognition of patient diversity, women's health concerns have historically been underrepresented in medical research (Health Innovation Hub Ireland, 2025). The bias was so pronounced that it was not until 2014 that the US National Institutes of Health required researchers to include sex as a biological variable (SABV) in preclinical studies involving animals and cell cultures for funding applications (Clayton, 2016; Clayton and Collins, 2014). Current clinical care of people with Parkinson's Disease (PwP) does not incorporate this need for specialised care for women, and women with PD (WwPD) have been continuously overlooked regarding their specific care needs (Subramanian et al., 2022). WwPD do not account for as many PD cases as men, with current research indicating the male-to-female prevalence ratio is approximately 1.5:1 (Figure 1) (Willis et al., 2022; reviewed by Zirra et al., 2022). Interestingly, a French nationwide analysis, reported no significant difference in Parkinson's Disease (PD) incidence rates between men and women under the age of 50 (Moisan et al., 2016). However, this sex disparity in incidence rates widens substantially with age, with men over 80 years experiencing PD at more than 1.6 times the rate of women in the same age group (Figure 1) (Brakedal et al., 2022; Moisan et al., 2016; Willis et al., 2022).

Figure 1: Age and sex dependent incidence and prevalence rates of PD

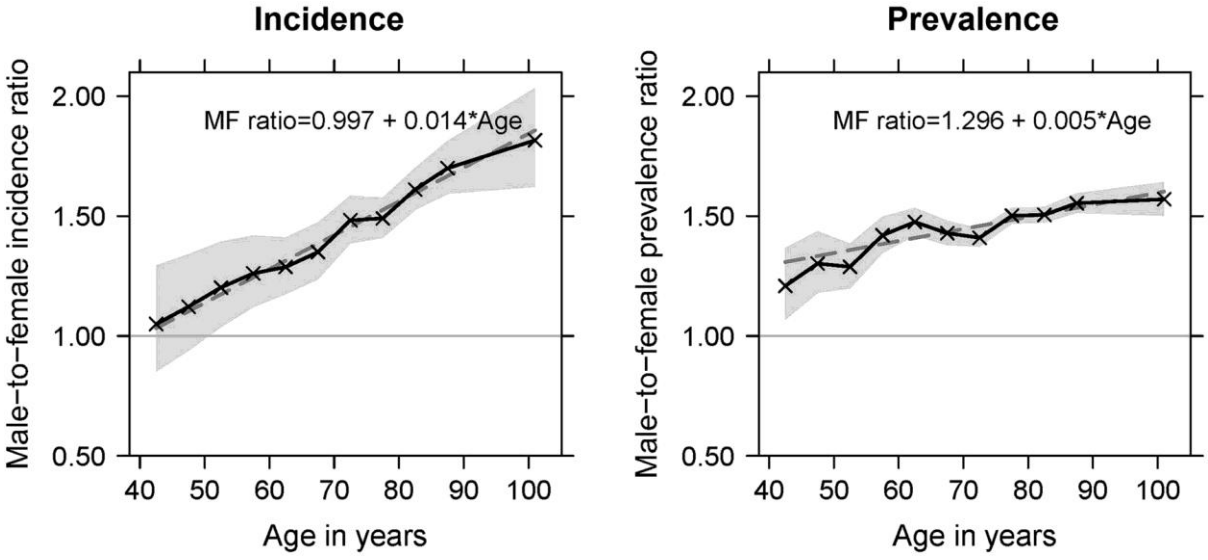


Figure 1: Male-to-female ratios of PD by age in France (2010). Left panel: incidence ratios; right panel: prevalence ratios. X marks indicate observed ratios at each age group. Black lines show fitted regression models with equations displayed. Grey shading represents 95% confidence intervals (taken from Moisan et al., 2016).

Sex refers to biological characteristics determined by anatomy and genetics, whereas gender represents the social and cultural constructs that shape identity and roles within society (Bates et al., 2022). The sex of a person with PwP impacts the degree and features of disability they experience (Abraham et al., 2019). A personalised approach to PD care must consider not only the individual's sex but also their age. Individuals with young onset Parkinson's Disease (YOPD), defined as those diagnosed before the age of 50, exhibit slower overall disease progression, a heightened incidence of motor fluctuations, dystonia (at onset and during treatment), and dyskinesias, compared to people with late onset Parkinson's Disease (LOPD) (Hely et al., 1995; reviewed by Wickremaratchi et al., 2009). Globally, YOPD exhibits an incidence rate of 1.3 new diagnoses per 100,000 people in the worldwide population (Nabizadeh et al., 2024). This number of YOPD has increased in recent years, emphasising the need to understand the clinical nuances involved in PD presentation at different ages (Dexter and Jenner, 2013).

1.1. Sex-Specific Differences in PD

1.1.1. Sex-Specific Differences in Symptom Presentation

All original research articles referenced in this section were conducted in human subjects.

1.1.1.1. Motor Symptom Presentation

To diagnose PD, clinicians look for the cardinal triad of motor symptoms which are bradykinesia, tremor, and rigidity (Bereczki, 2010; Jankovic, 2008). Understanding how these as well as other motor symptoms differ in timing and severity between men and women is crucial, as it can significantly impact both the accuracy of diagnosis and the effectiveness of treatment approaches.

In terms of initial presentation, women are more likely to present with tremor as the first motor symptom of PD compared to men (Haaxma et al., 2007) (Figure 2). Furthermore, some studies report that women present with slightly later onset of tremor and milder motor symptoms compared to men, indicated by lower UPDRS (Unified Parkinson's Disease Rating Scale) part III scores after controlling for demographic variables (De Pablo-Fernández et al., 2019; Haaxma et al., 2007; Kang et al., 2022).

Looking at specific motor manifestations, women experience less severe rigidity (Baba et al., 2005), fewer difficulties with speech, reduced hypomimia, less drooling, and fewer axial symptoms overall (Santos-García et al., 2023; Szewczyk-Krolikowski et al., 2014). However, women do face specific motor challenges, such as a higher risk of falling compared to men (Baba et al., 2005; Parashos et al., 2018).

Regarding disease progression, the patterns of how men and women experience motor impairment worsening and dopaminergic system dysfunction change over time. While some studies report that women have similar overall progression rates to men until the

late PD stages, there is evidence that after 20 years of diagnosis, women's PD seems to progress more rapidly (Abraham et al., 2019; Boccalini et al., 2022).
Sex disparities are also observed in treatment-related problems. Women have a significantly higher prevalence of wearing-off compared to men and female gender is associated with dyskinesia risk, though this may not be independent of other factors such as lower body weight (Colombo et al., 2015; Reviewed by Sharma et al., 2010; Reviewed by Warren Olanow et al., 2013). Particularly noteworthy is that WwPD develop levodopa-induced dyskinesias (LID) significantly faster after starting to take L-DOPA than men—typically in 4 years versus 6 years for men (Hassin-Baer et al., 2011). This sex difference remains significant regardless of other factors like age at diagnosis or treatment timing, suggesting biological differences in how women respond to dopaminergic therapy and the bioavailability of levodopa (L-DOPA) in women (Conti et al., 2022; Hassin-Baer et al., 2011).

Figure 2: Sex Differences in PD Symptom Prevalence

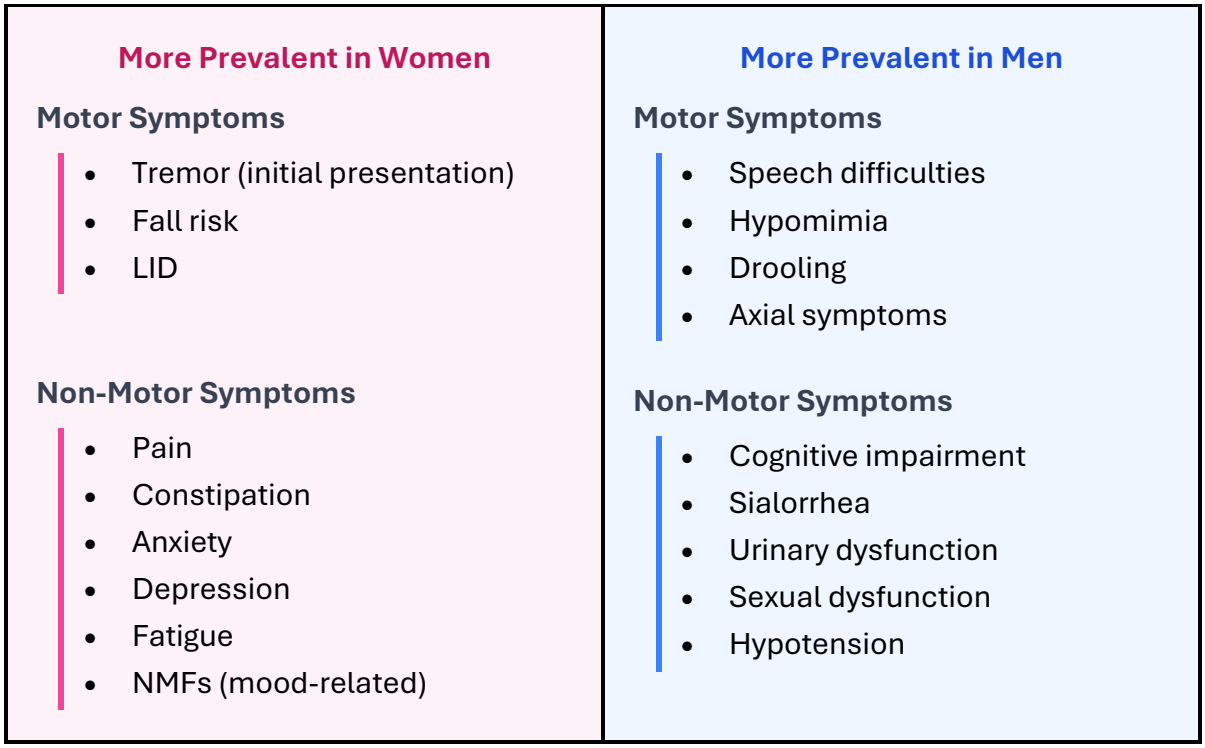


Figure 2: PD symptoms categorised by sex-specific prevalence patterns. Left panel shows symptoms more prevalent in women; right panel shows symptoms more prevalent in men.
(PD - Parkinson's disease, LID - Levodopa-induced dyskinesia, NMFs - Non-motor fluctuations).

1.1.1.2. Non-Motor Symptom Presentation

Non-motor symptoms (NMS) in PD are often experienced at the prodromal stage, many years before the onset of the cardinal PD symptoms (Alonso et al., 2009; Leentjens et al., 2003; Ross et al., 2008; Savica et al., 2009).

NMS demonstrate clear sex-based differences in their prevalence and severity with men experiencing more cognitive impairment, sialorrhea, urinary dysfunction, sexual dysfunction and hypotension (Boccalini et al., 2022; Kang et al., 2022; Maas et al., 2024; Martinez-Martin et al., 2012) (Figure 2). On the other hand, women suffer with more pain, constipation, anxiety, depression and fatigue (Boccalini et al., 2022; Gao et al., 2022; Kang et al., 2022; Kurlawala et al., 2021; Maas et al., 2024; Martinez-Martin et al., 2012). Adding to the picture, non-motor fluctuations (NMFs) in PD are episodic changes in non-motor symptoms that significantly impact the person's quality of life (QoL) (Reviewed by Classen et al., 2017; Prakash et al., 2016; Santos García et al., 2019). These NMFs are often concurrent with motor fluctuations, but can also occur as standalone episodes without accompanying motor symptoms (Del Prete et al., 2022; Nebe and Ebersbach, 2009; Storch et al., 2013). Research indicates that WwPD experience a higher burden of NMFs compared to men, with mood-related NMF being particularly prevalent in the female population, suggesting that WwPD require more targeted treatment approaches for these undertreated symptoms (Donzuso et al., 2023; Picillo et al., 2016).

These pronounced sex differences in NMS profiles indicate that clinicians should be more mindful of demographic differences and consider sex-specific treatment approaches when managing the complex non-motor manifestations of PD (Maas et al., 2024). approaches when managing the complex non-motor manifestations of PD (Maas et al., 2024).

1.1.2. Sex-Specific Differences in Responses to Medication

All original research articles referenced in this section were conducted in human subjects.

Sex differences significantly impact pharmacological treatments in PD. WwPD experience greater L-DOPA bioavailability and higher plasma concentrations compared to men, even when accounting for body weight differences (Conti et al., 2022; Kompoliti et al., 2002; Kumagai et al., 2014). This might occur due to a combination of biological factors that affect drug absorption and distribution rates: women have approximately 25% lower COMT (Catechol-O-Methyltransferase) enzyme activity, lower oral clearance and slower gastric emptying time (Boudíková et al., 1990; Datz et al., 1987; Reviewed by Marrinan et al., 2014; Martinelli et al., 2003).

When considering dosing metrics, body mass index (BMI) emerges as a considerably more valuable predictor than simple body weight, with research demonstrating that BMI significantly forecasts half-life (t1/2) in women but not in men (Conti et al., 2022).

Higher BMI correlates with reduced area under the curve (AUC) and extended t1/2 values—indicating that BMI might serve as a more effective biometric measurement for tailoring L-DOPA dosages than weight alone, especially for WwPD (Conti et al., 2022). WwPD are sometimes prescribed lower L-DOPA equivalent daily doses specifically because they have a higher predisposition to develop LID and motor fluctuations compared to male PwP (Iwaki et al., 2021a; Nyholm et al., 2010; Zappia et al., 2005). Moreover, a subgroup of PwP classified as having a "brittle response" to L-DOPA (defined by highly disabling dyskinesia after standard 100mg doses) is predominantly female (58%) with distinctive clinical characteristics including lower body weight, lower BMI, longer disease duration, and more years on L-DOPA therapy (Martinez-Ramirez et al., 2014). This literature search did not find any research comparing the effects of DA agonist usage between sexes.

Beyond pharmacokinetics, genetic factors also contribute to sex-specific treatment responses in PD. Male PwP carrying the monoamine oxidase B (MAO-B) G allele have a 2.84-fold increased risk of motor complications with high-dose L-DOPA, while the dopamine receptor D2 (DRD2) polymorphism provides protection against dyskinesia development only in men (dos Santos et al., 2019; Sampaio et al., 2018; Zappia et al., 2005).

Despite these pharmacokinetic differences between men and WwPD, the medical community has not fully translated this knowledge into comprehensive sex-specific treatment protocols. While clinicians do typically prescribe lower L-DOPA doses to women based on their increased susceptibility to complications, no formal sex-specific dosing guidelines or comprehensive treatment recommendations have been established (Ferreira et al., 2013; Subramanian et al., 2022).

Sex differences are also evident in other treatments. Safinamide, a dual-mechanism antiparkinsonian medication that inhibits MAO-B and modulates glutamate release, shows quantifiable sex differences in PD symptom improvement. While one study showed males experienced slightly higher improvement in total UPDRS scores (43.5% versus 39.1% in females), more recent research from the XINDI trial suggests females may experience greater benefits in several domains (Pellecchia et al., 2023, 2021). WwPD showed >50% higher improvements than males in total daily OFF time reduction (-1.15h versus -0.76h), total daily ON time increase (1.28h versus 0.44h), and total UPDRS score (-8.30 points versus -5.25 points). These contradictory findings highlight the need for further sex-specific research in PD pharmacology (Pellecchia et al., 2023, 2021).

Regarding surgical interventions, WwPD are significantly underrepresented in deep brain stimulation (DBS) referrals compared to the general PD population, with research showing they access surgical procedures later in their disease course (longer disease duration and more severe symptoms at time of surgery) despite being more likely to experience motor complications that would benefit from DBS (Hariz et al., 2013; Weaver et al., 2009; Willis et al., 2014). Notably, sex-specific responses to DBS in PD

reveal distinct patterns of improvement: male PwP experience greater benefits in posture abnormalities (particularly camptocormia), while female PwP achieve superior improvements in QoL measures, activities of daily living (ADL), mobility, stigma perception, and dyskinesia reduction (Hariz et al., 2013; Roediger et al., 2019). Whilst the disproportionate sex difference in DBS referral has been well documented, the reasons behind this disparity are complicated. One potential clinical explanation for the relatively reduced referral of WwPD may be their increased axial symptoms, which is a known risk factor for increased mortality post-operatively (Lau et al., 2019; Parashos et al., 2018). However, non-clinical factors also contribute to why women are less likely to ultimately receives DBS treatment, mainly due to less preoperative knowledge on the surgery and increased anxiety and depression in WwPD (Jost et al., 2022). (Ba et al., 2025; Shpiner et al., 2019). Ba et al, put forward that the increased rates of anxiety in WwPD may be a contributing factor, making women less likely to present to a clinician seeking surgical treatment and also making clinician discernment of the need for DBS treatment in WwPD more difficult (Ba et al., 2025; Kovács et al., 2016). This is further evidenced by women being more likely to be asleep during DBS implantation (Vinke et al., 2022). Indeed, women are reportedly more fearful and hesitant to undergo DBS and less likely to take initiative to receive this treatment in comparison to men, showing that both psychological and social determinants may be at play (Hamberg and Hariz, 2014). Furthermore, WwPD often have less social support and are more reluctant to lean or place burden on their support systems compared to men (Fullard et al., 2024). This influences their decision to undergo DBS and may also contribute to the delay seen between sexes (Fullard et al., 2024). Increased preoperative information and social support may help in moving towards equity for PWP eligible for DBS (Dinkelbach et al., 2017; Shpiner et al., 2019).

1.1.3. Sex Differences in Parkinson's Disease Pathophysiology

Naturally, the variations in presentation and progression between sexes call for interest into why, on a biochemical level, these differences emerge. Generally, the lower number of PD cases in women, as well as differences such as lower required L-DOPA dosages despite more prevalent dyskinesias, point toward some biologically protective element of female biology, which is generally attributed to sex hormones (Figure 1) (Beyer et al., 1991; Haaxma et al., 2007; Iwaki et al., 2021b; Martinez-Ramirez et al., 2014; Murray et al., 2003; Nyholm et al., 2010; Willis et al., 2022; Zirra et al., 2022). The following sections examine biological mechanisms including dopaminergic neuronal function, genetic mutations, biomarkers, and neuroinflammatory processes that contribute to these sex differences in PD and a summary of these mechanisms is presented in Table 1.

1.1.3.1. Sex-Specific Differences in Dopaminergic neuronal function and protection

It is well established, in research on both PwP and animal models, that females have higher striatal DA levels and neuronal reserves, that contribute to their general delay in symptom onset in PD (Beyer et al., 1991; Haaxma et al., 2007). Understanding the neurochemical and structural foundations of these differences is an important and ongoing area of research that informs us as to how to potentially maintain these neuroprotective effects in women, to prolong their effects, and regulate them where they may be contributing to fluctuation in PD.

Preclinically, at a cellular level, DA neurons in female mice demonstrate reduced vulnerability to degeneration in response to neurotoxins and cellular stress factors, with preservation of tyrosine hydroxylase expression and ganglioside levels not seen in DA neurons from male mice (Costa et al., 2019, 2018).

This resilience appears to be hormonally mediated, research in animal models shows that, as estrogen regulates multiple aspects of DA circuit function through direct modulation of DA synthesis, transport, and receptor dynamics, while simultaneously activating neuroprotective mechanisms that shield DA neurons from oxidative stress and lipid disruption (Cullity et al., 2019; Marin et al., 2013; Morissette and Di Paolo, 1993; Rivest et al., 1995; Song et al., 2019).

Structurally, the basal ganglia exhibit sex-specific molecular architecture, female mice are found to have approximately 20% higher D1:D2 DA receptor ratios in the striatum and increased DA transporter binding capacity (Cullity et al., 2019), and female primates have shown enhanced G-protein coupled estrogen receptor (GPER) signalling that directly reduces nigrostriatal vulnerability to oxidative stress (Al Sweidi et al., 2012).

These differences in DA availability, and structures related to PD, and their interaction in the female versus male brain give insight into some potential reasons for the delayed onset of symptoms that is generally seen in women, as well as the differential clinical course between the sexes, as outlined above.

1.1.3.2. Sex-Specific Differences in PD-linked Genetic Factors

Leucine-rich repeat kinase 2 (LRRK2) mutations show sex-specific prevalence, with higher occurrence in WwPD (Cilia et al., 2014; Orr-Urtreger et al., 2007; Shu et al., 2018).

Additionally, DA neurons from female postmortem human brain tissue express more neuroprotective genes for signal transduction and maturation of neurons, while male neurons overexpress SCNA, the gene that encode for α -synuclein, and PINK1, creating an inherently higher risk molecular environment for PD (Cantuti-Castelvetri et al., 2007; Simunovic et al., 2010).

1.1.3.3. Sex-Specific Differences in Biomarkers

Elevated serum uric acid provides male-specific neuroprotection in PD, with men having lower urate concentrations showing accelerated disease progression and increased severity, while women show no significant association between urate levels and PD risk or progression (Baik et al., 2020; Jesús et al., 2013; Lee et al., 2018; McFarland et al., 2013; Schwarzschild et al., 2011).

Additionally, sex differences in membrane lipid composition directly influence neurodegeneration patterns, with male PwP showing significant loss of neuronal gangliosides (particularly GD1a and GT1b) and phospholipids, in postmortem substantia nigra (SN) tissue, while WwPD maintain normal lipid content, potentially through estrogen's protective effects on membrane integrity (Marin and Diaz, 2018; Seyfried et al., 2018).

Other peripheral biomarkers exhibit sex-specific profiles in PwP, with homocysteine correlating with motor symptoms in men ($p<0.001$) but cognitive decline in women ($p=0.021$), DJ-1 protein elevated 1.7-fold exclusively in male PwP, and decreased brain-derived neurotrophic factor (BDNF) levels selectively associated with depression in WwPD (Bakeberg et al., 2019; Ho et al., 2014; Huang et al., 2021).

1.1.3.4. Sex-Specific Differences in Neuroinflammation

Evidence from preclinical mice models demonstrates that microglia exhibit sex-specific development and inflammatory responses, with male microglia showing a more mature inflammatory phenotype and enhanced reactivity to inflammatory stimuli due to early estrogen priming (Hanamsagar et al., 2017; Villa et al., 2018).

Estrogen exerts direct anti-inflammatory effects on microglia in mouse PD models, shifting microglial polarization toward neuroprotective phenotypes that protect DA neurons (Mendes-Oliveira et al., 2017; Siani et al., 2017).

Astrocytes derived from male and female mice exhibit sex-specific inflammatory and bioenergetic profiles in PD, with astrocytes from male animals producing significantly higher levels of pro-inflammatory cytokines interleukin 6 (IL-6), tumour necrosis factor alpha (TNF-α), and IL-1β when exposed to lipopolysaccharide (LPS), whilst astrocytes

from female mice preferentially produce interferon-inducible protein 10 (Santos-Galindo et al., 2011). Despite their inflammatory disadvantage, cortical astrocytes from male rats demonstrate greater maximal mitochondrial respiration capacity at physiologically relevant low oxygen tensions (Jaber et al., 2018). A 2018 review by Martin-Jiménez outlined how estrogen provides protection by acting directly on astrocytes in vitro (Martin-Jiménez et al., 2018). In vivo animal studies show that reactive astrocytes respond to brain injury by expressing aromatase, increasing local estrogen synthesis. This injury-induced estrogen synthesis provides neuroprotection (Littleton-Kearney et al., 2002; Martin-Jiménez et al., 2018; Pedersen et al., 2016).

Regarding iron metabolism, significant sex differences appear in PD pathogenesis in mice, with male striatum accumulating substantially more toxic iron than female striatum (Wang et al., 2015). Evidence from human studies suggests these patterns translate clinically; a human study found that at identical blood iron concentrations, women have significantly lower PD probability than men (Mariani et al., 2016). Additionally, an in vivo mouse study found that, estrogen provides sex-specific protection through distinct pathways: in males, it activates GPER1 receptors to prevent excessive iron-triggered autophagy that would lead to neuronal death, while in females, it works through estrogen receptor (ER) α to block iron-induced membrane lipid breakdown that would otherwise damage dopaminergic neurons (T.-Y. Chen et al., 2019).

1.1.3.4.1 Sex-Specific Differences in Prostaglandin Pathways

Prostaglandins are eicosanoids derived from arachidonic acid through the action of cyclooxygenase enzymes, and they play an important role in neuroinflammation and neurodegeneration (Tzeng et al., 2005). In PD, prostaglandin E2 (PGE2) is the main prostaglandin implicated in pathology (Ikeda-Matsuo et al., 2019). Studies in postmortem SN brain tissue have demonstrated that prostaglandin synthase is increased in people with PD compared to controls (Ikeda-Matsuo et al., 2019; Mattammal et al., 1995).

The mechanisms by which prostaglandins contribute to PD pathology operate through multiple pathways. Cyclooxygenase-2 (COX-2), the rate-limiting enzyme in PGE2 synthesis, is increased in the SN pars compacta in PD and has damaging effects on the tissue (Feng et al., 2002; Reviewed by Teismann, 2012).

Mattammal et al. demonstrated that increased prostaglandin H synthase activity results in co-oxidation of dopamine to reactive intermediates that lead to dopaminergic neuronal damage (Mattammal et al., 1995). Additionally, prostaglandins contribute to the protein aggregates found in Lewy bodies characteristic of PD (Koharudin et al., 2010). Prostaglandins also activate microglia, which contribute to neuronal loss in the substantia nigra in PD (Beach et al., 2007). The prostaglandin D2/J2 signalling pathway has been shown to mimic PD progression in rodents (Corwin et al., 2018). Furthermore,

in male mice, prostaglandin J2 (PGJ2) causes dose-dependent and selective neurodegenerative effects on the SN and microglial activation (Pierre et al., 2009). Given prostaglandins' involvement in neuroinflammation, they have been investigated as potential therapeutic targets in PD (Chaturvedi and Beal, 2008; Feng et al., 2002; Ikeda-Matsuo et al., 2019). The prostaglandin-producing enzyme prostaglandin-H2 D-isomerase has also been identified as a potential biomarker for PD detection (Hu et al., 2024).

Whilst prostaglandins play a role in PD pathology, investigation of sex-specific differences in prostaglandin metabolism in PD remains sparse. A recent plasma metabolomics study in PwP revealed that eicosanoid expression was predominantly different in men, with only a single eicosanoid, PGF2alpha methyl ether, differentially expressed in WwPD compared to controls (Hu et al., 2024). This sex-specific pattern of prostaglandin expression suggests that the aforementioned prostaglandin-mediated therapeutic targets may require sex-tailored approaches, given the sexually dimorphic nature of lipid metabolism in PD pathophysiology (Hu et al., 2024). However, no literature was found examining prostaglandin fluctuations in menstruating WwPD. This would be an interesting avenue for future research, as prostaglandins are known to be the cause of primary dysmenorrhea during menstruation (Guimarães and Póvoa, 2020; Powell et al., 1985). Investigation of the relationship between hormone-related prostaglandin elevations and PD symptoms and/or neurodegeneration in WwPD warrants further study.

Table 1: Summary of Sex-Specific Differences in Parkinson's Disease Pathophysiology

Pathophysiological Mechanism	Study Type	Finding	References
Dopaminergic Neuronal Function and Protection	**Animal**	• Females: ↑ striatal DA levels and neuronal reserves • Females: ↓ DA neuronal vulnerability to neurotoxins • Females: ↑ D1:D2 DA receptor ratios in striatum • Females: ↑ DA transporter binding capacity	*Beyer et al., 1991; Costa et al., 2019, 2018; Cullity et al., 2019*
PD-linked Genetic Factors	**Human**	• Females: ↑ LRRK2 mutations • Females: ↑ neuroprotective gene expression for signal transduction and neuronal maturation in DA neurons • Males: ↑ SCNA and PINK1 overexpression in DA neurons	*Cilia et al., 2014; Orr-Urtreger et al., 2007; Shu et al., 2018; Cantuti-Castelvetri et al., 2007; Simunovic et al., 2010*
Biomarkers	**Human**	• Males: ↑ serum uric acid associated with ↓ PD risk and slower progression; Females: no association • Males: ↑ loss of neuronal gangliosides (GD1a, GT1b) and phospholipids in SN; Females: normal SN lipid content maintained • Males: homocysteine correlates with motor symptoms; Females: homocysteine correlates with cognitive decline • Females: ↓ BDNF levels associated with depression	*Baik et al., 2020; Jesús et al., 2013; Lee et al., 2018; McFarland et al., 2013; Schwarzschild et al., 2011; Marin & Diaz, 2018; Seyfried et al., 2018; Bakeberg et al., 2019; Ho et al., 2014; Huang et al., 2021*
Neuroinflammation	**Human**	• Females: ↓ PD probability compared to males at identical blood iron concentrations	*Mariani et al., 2016*
	Animal	• Males: ↑ microglial inflammatory phenotype and reactivity • Females: estrogen shifts microglia toward neuroprotective phenotypes • Males: ↑ cortical astrocyte mitochondrial respiration capacity • Males: ↑ toxic iron accumulation in striatum	*Hanamsagar et al., 2017; Villa et al., 2018; Mendes-Oliveira et al., 2017; Siani et al., 2017; Jaber et al., 2018; Wang et al., 2015*

Table 1: Summary of sex-specific differences in Parkinson's disease pathophysiology as presented in section 1.1.3. Findings are categorised by evidence type: Human (yellow) and Animal/preclinical (blue). ↑ = increase; ↓ = decrease. A full explanation of findings is available in section 1.1.3 of this introduction.

1.2. PD and Hormonal Events

As mentioned above, historically research has overlooked women, and female animals, and attributed this to hormonal fluctuations associated with the menstrual cycle leading to 'variability', a point which has since been debunked, as hormone variations are not exclusive to females, and not always necessitate being considered in some research as reviewed by Beery (Beery, 2018). While this has now been resolved due to the rule mentioned above, it remains critical to recognise that hormones play a significant role in PD research specifically.

The importance of studying hormonal factors in PD becomes clear when examining their neurobiological effects. Research has demonstrated that sex hormones interact with various neurotransmitter systems throughout the brain, with direct implications for PD pathophysiology as reviewed by Bourque 2024 (Bourque et al., 2024). Estrogen is the primary focus of research regarding neuroprotection as testosterone has not demonstrated protective effects on DA neurons in animal models (Ekue et al., 2002; Gillies et al., 2004; Murray et al., 2003). Progesterone may also play a role, either independently or in conjunction with estrogen, as both hormones experience significant decline during menopause, a period hypothesised to be associated with increased PD risk in women (Bovenzi et al., 2024b).

These hormonal fluctuations throughout a woman's lifetime are, in the case of PD, inherently linked to the symptoms associated with the disease, its severity, progression and sometimes onset. The following studies examine women's experiences with PD during periods of substantial hormonal changes, exploring how these fluctuations may have impacted their condition. Although PD predominantly occurs in older people, roughly 5% of WwPD are diagnosed under the age of 40, raising important considerations about managing both menstrual cycles and potential pregnancies alongside their condition as reviewed by Mehanna and Jankovic, 2019 (Mehanna and Jankovic, 2019).

1.2.1. Menstruation

Many premenopausal WwPD experience a cyclical worsening of symptoms associated with their menstrual cycles (Figure 3). This phenomenon appears to be common, with symptom deterioration typically beginning several days before menstruation and improving either shortly after the onset of menses or a few days after the end of menses (Table 2). Aswell as the studies presented in Table 2, a 2021 survey conducted by PD advocates found that 86% of nearly 200 WwPD experienced symptom worsening in the week before menstruation (“PD Avengers Championing ‘Women and PD,’” 2021).

The relationship between menstrual hormones and PD symptoms is complex and not fully understood. Evidence from case studies suggests that low estrogen levels during menses might contribute to reduced responsiveness to dopaminergic treatment (Sprenger et al., 2014). During these perimenstrual periods, WwPD frequently report

reduced effectiveness of dopaminergic medications during the perimenstrual period, with some experiencing medication failure, shortened duration of action, and increased "off" periods (Factor, 1993; Quinn and Marsden, 1986; Tolson et al., 2002). Despite these consistent patient reports, when Kompoliti et al. conducted a prospective study measuring objective changes in PD symptoms across different phases of the menstrual cycle, they did not find significant correlations with hormonal levels (Table 2) (Kompoliti et al., 2000). This disparity highlights the need for more sophisticated research monitoring.

Specific motor symptoms that tend to worsen during the perimenstrual period include off-medication dystonia, tremor, gait disturbances, and bradykinesia (Castrioto et al., 2010; “PD Avengers Championing ‘Women and PD,’” 2021). In some cases, the deterioration is so severe that women require assistance with daily activities during their periods (Quinn and Marsden, 1986; Tolson et al., 2002). Notably, these menstrual-related symptom fluctuations may persist even in women who have undergone DBS surgery, despite this typically being an effective treatment (Castrioto et al., 2010).

Case studies provide interesting insights into potential hormonal mechanisms. Sprenger et al. (2014) reported that when two sisters with PD both took an estrogen supplement continuously, it ameliorated both of their symptoms (Sprenger et al., 2014). This estrogen supplement (dienogest-ethinylestradiol) improving the perimenstrual worsening supports the hypothesis that the steep drop in estrogen before the onset of menses could be the cause of these perimenstrual fluctuations. This finding contradicts a 1994 study by Session et al. which reported that leuprolide acetate, which induced a pharmacological menopause thereby reducing estrogen, reportedly improved worsening of symptoms premenstrually in one woman (Session et al., 1994). Session et al. believed estrogen to be antidopaminergic, whereas more recent research strongly supports estrogen's neuroprotective role in the dopaminergic system., as discussed earlier in this narrative literature review.

There may be an association between premenstrual syndrome (PMS) and menstrual-related PD symptom fluctuations. A study by Castrioto and colleagues found that the women who experienced PD symptom fluctuations during their menstrual cycle also reported PMS (Castrioto et al., 2010). Researchers face challenges in determining whether symptom fluctuations are due to PMS itself or specific interactions between menstruation and PD mechanisms (Castrioto et al., 2010; Kompoliti et al., 2002). The co-occurrence of PMS and menstrual-related PD fluctuations raises questions about shared underlying mechanisms or whether one condition might exacerbate the other.

In addition to symptom management, WwPD face unique and often humiliating challenges with self-care during menstruation, including difficulties managing sanitary protection due to dexterity problems and mobility impairments, which can lead to loss of independence and dignity (Tolson et al., 2002).

The clinical management of these menstrual-related fluctuations calls for improvement. There appears to be a lack of awareness and recognition among

healthcare providers about menstrual-related PD symptom fluctuations, with women's concerns sometimes being dismissed or inadequately addressed (“PD Avengers Championing ‘Women and PD,’” 2021; Tolson et al., 2002).

A significant challenge in this field lies in separating normal hormonal fluctuations that affect all women from the specific interactions between these hormonal changes and PD mechanisms. This complexity highlights the relatively unexplored nature of this relationship and emphasises the need for more prospective, longitudinal research to better understand these interactions and develop targeted therapeutic strategies (Subramanian et al., 2022).

Current understanding of menstrual-related PD symptom fluctuations is limited by small sample sizes, predominantly retrospective studies, and inconsistent methodologies, highlighting the need for more robust research in this area (Table 2).

Table 2: Summary of studies investigating menstrual cycle effects on PD symptoms

STUDY AUTHOR(S)	STUDY MODALITY	WOMEN WHO REPORTED FLUCTUATIONS N (% OF STUDY)	MEAN AGE OF PARTICIPANTS	MEAN AGE OF ONSET (YEARS)	PERIMENSTRUAL TIMING	KEY FINDINGS
SPRENGER ET AL., 2014	• Case Study • Retrospective	2 (100%)	45.5	35.5*	• Case 1: 5 days before, during, and 3 days after menstruation • Case 2: 2 days before and up to 2 days after menstruation	• Cyclical decline in medication response • Oral contraceptive therapy restored medication efficacy • OFF periods reduced from >75% to <25% of the day with hormone therapy
CASTRIOTO ET AL., 2010	• Survey • Retrospective	11 (64.7% of 17 PD patients)	50.5 ± 6.6	40.5 ± 7.9	• Week before menstruation • During menstruation	• Off-medication dystonia most common symptom (73%) • Walking and sleep disturbances (64%) • Balance and clumsiness issues (55%)
SESSION ET AL., 1994	• Case Study • Retrospective	1 (100%)	44	~34*	• 1 week before menstruation	• Regular worsening of movement • Deterioration in postural reflexes • Improvement with leuprolide acetate-induced pharmacological menopause
TOLSON ET AL., 2002	• Descriptive Case Study • Prospective	15/18 (83%)	47	~39	• Week before menstruation • During menstruation	• Menstrual problems worsened after PD diagnosis • Two cases required hysterectomy • Medication effectiveness declined during menstruation

						• Night-time care difficult as medication effects diminished
KOMPOLITI ET AL., 2000	• Observational • Prospective	7/10 (70%) reported worsening, not confirmed objectively	42.3 ± 5.8	~36.5*	Not objectively confirmed	• No correlation between motor symptoms and hormone levels • Three women showed lower UPDRS with higher estrogen • Subjective reports not confirmed by objective measures
QUINN AND MARSDEN, 1986	• Clinical Interview •Retrospective	11/12 (92%)	Not stated	37.8	5 days before until 2.2 days after menstruation onset	• 40-50% subjective symptom worsening • Increased freezing in L-DOPA treated patients • Handwriting deterioration and increased tremor • Initial documentation of menstrual-PD relationship
FACTOR, 1993	• Case Study • Retrospective	1 (100%)	38	30	• 3 days before menstruation • Lasting 5-6 days total	• Prolonged "off" periods and medication wearing off more quickly • Dramatic improvement with acetazolamide • First report of acetazolamide efficacy in menstrual-related PD

*Table 2: Summary of studies investigating menstrual cycle effects on PD symptoms. (*Age of onset was calculated from reported disease duration and age during study).*

(PD - Parkinson's disease, UPDRS - Unified Parkinson's Disease Rating Scale).

Figure 3: Relationship Between Menstrual Cycle Hormone Levels and PD Symptoms

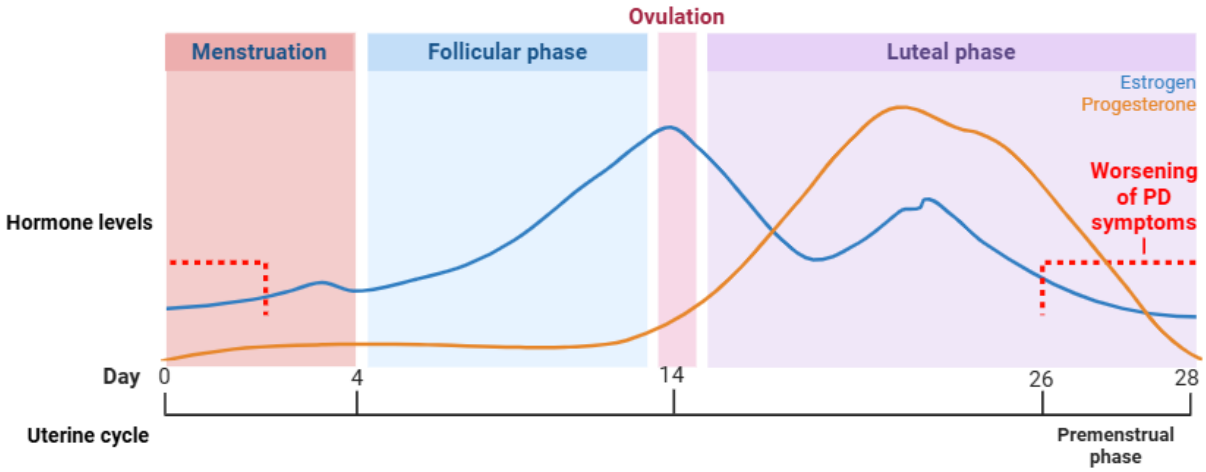


Figure 3: Graph showing estrogen (blue line) and progesterone (orange line) fluctuations across menstrual phases (menstruation, follicular, ovulation, luteal). Red dotted lines indicate periods of worsened PD symptoms. Adapted from "Uterine Cycle", by BioRender.com (2024). Retrieved from https://app.biorender.com/biorender-templates

1.2.1.1. Estrogen

1.2.1.1.1 Neuroprotective Evidence for Estrogen in PD

Evidence from animal models establishes the foundational neuroprotective role of estrogen in the dopaminergic system. Estrogen mediates sex-specific neuroprotection against dopaminergic neurotoxins, as demonstrated in multiple animal models (Dluzen et al., 1996a; Gomez-Mancilla and Bédard, 1992; Murray et al., 2003). Female mice, with naturally higher estrogen levels, show significantly less neurotoxin-induced striatal DA depletion than males (Brooks et al., 1989; Dluzen et al., 1996a). When ovariectomized to remove endogenous estrogen, female mice lose this protection and exhibit damage comparable to males, confirming estrogen's crucial protective role in the female nigrostriatal system (Dluzen et al., 1996b, 1996a).

This laboratory evidence translates to human epidemiology, as women with higher cumulative lifetime estrogen exposure show significantly reduced PD risk, despite there being some conflicting evidence (Gatto et al., 2014; Haaxma et al., 2007; Rugbjerg et al., 2013; Yoo et al., 2020). Gatto et al used a composite measure incorporating multiple estrogen-related factors (later menopause, greater parity, longer fertile lifespan) to demonstrate that women with high lifetime exposure have up to 43% lower risk of developing PD compared to those with minimal exposure (Gatto et al., 2014). The postmenopausal predominance of PD onset in women strongly suggests that the loss of estrogen's neuroprotective effects contributes to increased disease vulnerability (Cereda et al., 2013).

As mentioned previously, WwPD experience more dyskinesias than males. Promisingly, estrogen treatment in ovariectomized female monkeys significantly reduces LID, demonstrating estrogen's direct modulatory effects on motor complications of L-DOPA therapy (Gomez-Mancilla and Bédard, 1992).

1.2.1.1.2 Mechanisms of Estrogen-Mediated Neuroprotection in PD

Returning to mechanistic evidence, preclinical studies have further characterised the molecular pathways through which estrogen exerts these effects. Estrogen can prevent α-synuclein from forming toxic fibrils and can destabilize existing fibrils, potentially inhibiting lewy body (LB) development (Hirohata et al., 2009). Concurrently, estrogen regulates several processes in DA neurons: it promotes cell plasticity, enhances neurite growth and branching, modulates autophagy processes, and reduces neuronal damage during injury in mouse models (Küppers et al., 2000; Wang et al., 2015).

Not all estrogens provide equal protection, however. 17β-estradiol demonstrates compound-specific neuroprotection in rodent models of PD (Callier et al., 2001). While 17β-estradiol significantly preserves striatal DA levels against MPTP (1-methyl-4-phenyl-1,2,3,6-tetrahydropyridine), 6-OHDA (6-hydroxydopamine), and methamphetamine toxicity, 17α-estradiol shows no protective effects and estrone exhibits only weak neuroprotective activity in animal models (Dluzen, 1997; Dluzen et al., 2011; Gao and Dluzen, 2001; Grandbois et al., 2000; Miller et al., 1998; Yu and Liao, 2000).

One mechanism through which estrogen provides neuroprotection is by enhancing mitochondrial function in the brain, providing protection against the mitochondrial dysfunction observed in PD (Reviewed by Lejri et al., 2018). Brain tissues have high energy demands and contain mitochondrial ERs that, together with nuclear ERα and ERβ and their coactivators, protect against oxidative stress and regulate apoptosis (Chen et al., 2009; Y.-H. Chen et al., 2019), therefore, a deficiency in this estrogen-dependent mechanism has been thought to contribute to PD pathogenesis.

A 2024 study in primates showed that neural stem cells (NSCs) demonstrate protective effects through several estrogen-related pathways: they modulate estrogen balance, increase expression of PGC-1α (Peroxisome proliferator-activated receptor gamma co-activator 1 alpha), a master regulator of mitochondrial biogenesis, enhance production of fusion proteins like MFN2 (mitofusion 2) and OPA1 (Optic Atrophy 1) that maintain mitochondrial network integrity, and activate antioxidant defence systems that counteract the excessive reactive oxygen species associated with dopaminergic neuron degeneration in PD (He et al., 2024). The MitoPark mouse model, which develops PD-like symptoms due to mitochondrial impairment, shows clear sex differences - females exhibit less severe symptoms than males, while ovariectomized females display a disease progression similar to males (Y.-H. Chen et al., 2019). This evidence suggests that estrogen's actions on mitochondria contribute significantly to the observed lower susceptibility of females to PD.

Additionally, estrogen plays a role in regulating neuroinflammation in PD. In vitro studies in animals using microglial cell lines demonstrate that estrogen and GPER activation directly inhibit LPS induced inflammatory responses, including phagocytic activity, Nitric Oxide (NO) production, and inducible Nitric Oxide Synthase (iNOS) expression. G1, a GPER agonist, prevents LPS-induced microglial activation with effects completely abolished by the GPER antagonist G15 (Mendes-Oliveira et al., 2017; Morale et al., 2006; Zhao et al., 2016).

The timing of estrogen administration appears critical for neuroprotection. Time-course experiments in mouse models of PD reveal a strict temporal requirement; protection against neurotoxin damage requires pre-emptive hormone treatment. In female mice, administering estrogen prior to methamphetamine (at intervals of 24, 12, or as little as 0.5 hours beforehand) successfully prevents DA depletion in the striatum (Gajjar et al., 2003). However, this protection vanishes completely when estrogen is given after exposure, whether shortly afterward (0.25-2 hours post-toxin) or at longer

intervals (1 week later) (Gajjar et al., 2003; Gao and Dluzen, 2001). This sharp temporal cutoff demonstrates that estrogen's neuroprotective mechanisms activate preventive rather than restorative pathways in the nigrostriatal system, and it does so within physiological ranges as reviewed by Bourque et al., 2009.

1.2.1.1.3 Selective Estrogen Receptor Modulators as Therapeutic Targets in PD

Looking toward potential therapeutic applications for WwPD, selective estrogen receptor modulators (SERMs) represent a possible pharmacological approach for PD treatment without estrogen's feminising side effects (Goekoop et al., 2006). Tamoxifen, a widely used SERM originally developed for breast cancer treatment, exhibits complex effects in neurotoxin models of PD (Dean E. Dluzen et al., 2001; D. E. Dluzen et al., 2001; Gao and Dluzen, 2001; Yu et al., 2002). (Gao and Dluzen, 2001) Tamoxifen shows two opposing effects in neurotoxin models: it inhibits estrogen's protective benefits when given together in ovariectomized mice yet provides direct neuroprotection when used alone in intact animals (D. E. Dluzen et al., 2001; Gao and Dluzen, 2001; Yu et al., 2002). Its metabolite 4-hydroxytamoxifen further complicates this picture by boosting estrogen's protective capacity rather than interfering with it (Yu et al., 2002). Raloxifene, another clinically approved SERM used for osteoporosis treatment, shows variable but potentially beneficial effects against MPTP toxicity as it can act as estrogen antagonist in reproductive tissues while mimicking estrogen's beneficial effects in neural tissue (Callier et al., 2001; Grandbois et al., 2000; Ramirez et al., 2003).

1.2.1.2. Progesterone

1.2.1.2.1 Neuroprotective Evidence for Progesterone in PD

Recent work by Luchetti et al, in a human postmortem study, showed that the progesterone metabolite allopregnanolone exhibits a biphasic pattern, showing increased levels in the SN during early disease stages followed by significant reduction in advanced stages, suggesting an endogenous neuroprotective response that becomes dysregulated as the disease progresses (Luchetti et al., 2023). Supporting this connection to PD pathophysiology, both human and animal studies, progesterone metabolism is demonstrably impaired in PD, with reduced levels of the converting enzyme 5α-reductase in the SN and decreased concentrations of progesterone metabolites in plasma and cerebrospinal fluid (CSF), suggesting that dysregulation of progesterone metabolism may contribute to disease pathophysiology (Luchetti et al., 2023, 2010; Melcangi et al., 2012).
Unlike estrogen, which shows sex-specific neuroprotective effects in PD models, progesterone demonstrates robust neuroprotection in male mice against both MPTP and methamphetamine-induced neurotoxicity, although a substantial knowledge gap about possible sex variations in progesterone's therapeutic effects is evident, considering the dearth of thorough research in female models (Bourque et al., 2015; Dluzen et al., 1996a; Gao and Dluzen, 2001; Grandbois et al., 2000; Yu et al., 2002).

1.2.1.2.2 Mechanisms of Progesterone-Mediated Neuroprotection in PD

Progesterone exerts neuroprotective effects in PD animal models through multiple mechanisms, including preservation of dopamine transporter (DAT) integrity in the striatum, prevention of MPTP-

induced striatal DA depletion, and modulation of striatal neurotransmitter pathways, demonstrating effects beyond simple antioxidant activity (Callier et al., 2001; Casas et al., 2013; Grandbois et al., 2000). Progesterone's neuroprotective mechanisms also include modulation of BDNF, reduction of neuroinflammation through decreased glial fibrillary acidic protein (GFAP) expression in mice models, although this study only used male mice treated with MPTP (Litim et al., 2017).
The efficacy of progesterone appears to be highly dependent on administration parameters. In male mouse models of PD, both timing and dosage of progesterone administration are critical factors determining neuroprotective efficacy (Jarras et al., 2020; Litim et al., 2017). Litim et al did research in male mouse models of PD which showed that both timing and dosage of progesterone administration are critical factors determining neuroprotective efficacy, with treatment initiated within 24 hours after MPTP exposure proving effective at preserving striatal DA levels and transporters, whilst administration after 5 days fails to confer protection (Litim et al., 2017).

1.2.1.2.3 Translational Challenges and Therapeutic Potential of Progesterone in PD

Despite these promising preclinical findings, epidemiological data examining progesterone's relationship with PD risk is limited and inconclusive, with one exploratory analysis suggesting potentially increased risk associated with progestin-only hormone therapy in women, though small sample sizes and lack of specification of the specific progestins used prevent definitive conclusions (Gatto et al., 2014; Simon et al., 2009).
Interestingly, progesterone's protective effects extend beyond the central nervous system. It demonstrates neuroprotective effects in the myenteric plexus of the gut in male MPTP mouse models, preventing dopaminergic neuronal loss and reducing pro-inflammatory macrophage density, suggesting potential benefits for addressing gastrointestinal symptoms that often precede motor manifestations in PD (Jarras et al., 2020).
Given the complex and sometimes contradictory findings regarding progesterone's effects in PD models and frequent omission of female animal models, translational research remains challenging, though recent evidence supports consideration of progesterone as a potential disease-modifying therapy specifically in prodromal or early stages of PD rather than in established disease (Reviewed by Bourque et al., 2024; Jarras et al., 2020; Litim et al., 2017).

1.2.2. Menopause

Menopause appears to influence PD progression and vulnerability, with most women developing PD after menopause (Bovenzi et al., 2024b). The hormonal shifts during this transition may contribute to altered disease manifestation as reviewed by Smith and Dahodwala, 2014.

A woman's reproductive timeline significantly impacts PD vulnerability and progression. Extended exposure to endogenous estrogen appears protective, as women who had menopause at 50 years or later demonstrate reduced PD risk compared to those with early menopause (<40 years) (Yoo et al., 2020). Similarly, longer fertile lifespans correlate with less severe motor symptoms as measured by lower Hoehn & Yahr scores in postmenopausal WwPD and reduced PD risk (Bovenzi et al., 2024b; Yoo et al., 2020). This protective relationship is further supported by findings that early menopause may elevate PD risk, with Popat et al. (2005) noting significant reduction in risk for women with later final menstrual periods (Popat et al., 2005; Yoo et al., 2020).

The relationship between reproductive surgeries and PD risk provides additional evidence for hormonal influence. A case-control study of 72 WwPD and 72 age-matched controls has revealed that WwPD more frequently report prior hysterectomy with ovarian preservation compared to matched controls, suggesting that early reduction in endogenous estrogen, whether from pre-existing ovarian dysfunction that led to the surgery or from surgically-induced ovarian failure following the procedure, may increase PD risk (Benedetti et al., 2001). Benedetti et al. reported nearly three times higher rates of this surgical procedure among PwP. The connection potentially stems from disrupted ovarian function following hysterectomy, as research indicates women who undergo the procedure experience earlier ovarian failure despite ovary preservation, resulting in premature estrogen reduction (Benedetti et al., 2001; Siddle et al., 1987).

Postmenopausal WwPD show distinct hormonal alterations compared to age-matched controls, with significantly lower estrogen and higher testosterone levels (Bovenzi et al., 2024b). These hormonal differences directly influence disease manifestation and treatment requirements. Specifically, residual estrogen levels appear to modulate medication needs, as people with higher residual estrogen production experience less severe motor symptoms and require lower dopaminergic medication doses (Bovenzi et al., 2024b). This relationship between postmenopausal estrogen levels and treatment response highlights the persistent influence of sex hormones even after reproductive years have ended.

Beyond estrogen, other reproductive hormones may influence disease manifestation. Bovenzi et al showed that elevated follicle-stimulating hormone (FSH) levels in postmenopausal PD women were significantly associated with poorer cognitive performance, as measured by lower Mini-Mental State Examination (MMSE) scores and with CSF markers indicative of amyloid pathology (reduced Aβ42/Aβ40 ratios) and neuronal damage (increased total tau), suggesting FSH may contribute to cognitive deterioration through pathological protein accumulation (Bovenzi et al., 2024b).

Reproductive history also appears relevant to disease severity, as higher parity may be associated with increased motor dysfunction in postmenopausal WwPD, potentially due to reduced lifetime estrogen exposure (Bovenzi et al., 2024b; Windham et al., 2002).

1.2.2.1. Perimenopause

Perimenopause in WwPD is not well researched, but warrants careful consideration, as this represents a neurological transition state in which fluctuating estrogen and progesterone levels are associated with neurological changes that may increase PD-related neurodegeneration risk (Brinton et al., 2015; Genazzani et al., 2005). In a 2023 study, 8.8% of WwPD who responded to a questionnaire reported PD onset during perimenopause, whilst 6.1% were currently perimenopausal (Rao et al., 2023a). However, this literature search identified no original research articles examining the specific neurological impact of perimenopause in WwPD, a gap that has been highlighted as an unmet research need (Subramanian et al., 2022; Unda et al., 2022). Addressing this gap is essential for understanding PD risk mechanisms during this transition, informing optimal timing for HRT and developing more specific prevention and treatment strategies for WwPD (Saunders-Pullman et al., 2011; Song et al., 2019; Subramanian et al., 2022; Tsang et al., 2000).

1.2.2.2. HRT

Research on hormone replacement therapy (HRT) in postmenopausal WwPD shows complex and sometimes contradictory results (Yoo et al., 2020). While low-dose estrogen therapy demonstrated improvements in motor symptoms in some controlled trials (Tsang et al., 2000), larger meta-analyses found no consistent evidence supporting HRT's protective role against PD (Wang et al., 2014).

The inconsistency in research findings appears to stem from several key factors. The type of menopause significantly influences treatment response, with women experiencing natural versus surgical menopause showing different reactions to hormone therapies. One study found estrogen therapy was associated with a 2.6-fold increased risk in women with hysterectomy but suggested possible benefits in women with natural menopause (Popat et al., 2005). Similarly, hormone formulations demonstrate variable effects, with natural progesterone appearing to offer greater neuroprotective benefits than synthetic progestins like medroxyprogesterone acetate, which may reduce estrogen's protective effects (Kim et al., 2021; Yoo et al., 2020).

Methodological challenges further complicate the research landscape. The POETRY study (2011) identified that traditional research designs struggle to account for the complex interplay between hormonal changes and disease progression. This creates difficulty when trying to establish causality between hormone therapy and PD outcomes (POETRY, 2011).

Individual case reports provide interesting clinical insights despite the lack of conclusive population-level data. Case studies have indicated that discontinuing estrogen therapy can trigger or worsen motor fluctuations and "on-off" phenomena in some PwP, suggesting that estrogen withdrawal may acutely affect dopaminergic function in women with established PD (Sandyk, 1989).

The wide variation in study designs, participant characteristics, HRT formulations, and outcome measures makes drawing definitive conclusions challenging, highlighting the need for better-designed clinical trials specifically targeting postmenopausal WwPD (POETRY, 2011).

1.2.3. Pregnancy

As mentioned above, PD is uncommon before the age of 40 years. Therefore, PD is rare in women of fertile age, and its association with pregnancy is even rarer (Lamichhane et al., 2014). However, studying pregnancy is especially valuable because it presents a unique opportunity to observe dramatic physiological fluctuations in sex hormones, with estrogen and progesterone both progressively increasing throughout each trimester, reaching peak levels just before parturition, and then rapidly declining after birth (Tulchinsky et al., 1972).
Motor symptoms during pregnancy show variable responses, with some women reporting worsening, no change, or sometimes improvement of symptoms. Overall, studies indicate approximately half of women experience deterioration of motor symptoms during pregnancy (Golbe, 1987; Hagell et al., 1998; Seier and Hiller, 2017).
As noted earlier in this review, the current evidence indicates that estrogen provides neuroprotective effects and enhances dopaminergic function. However, during pregnancy, estriol becomes the predominant estrogen, and its specific impact on PD is not well characterised (Orzołek et al., 2022). This shift in dominant estrogen type has been proposed as one potential mechanism for the symptomatic changes that some WwPD experience during pregnancy (Seier and Hiller, 2017).
Beyond hormonal changes, medication management significantly influences symptom control during pregnancy. It should be noted that medication discontinuation often contributes to symptom worsening during pregnancy (reviewed by Bovenzi et al., 2024a and Young et al., 2020).
Additionally, changes in medication pharmacokinetics during pregnancy may affect symptom control (Loebstein et al., 1997).
Based on the available evidence, L-DOPA is considered the safest medication during pregnancy (Hagell et al., 1998; reviewed by Olivola et al., 2020; reviewed by Young et al., 2020).
On the contrary, amantadine is contraindicated during pregnancy due to teratogenic potential (Kranick et al., 2010).
The postpartum period can be associated with worsening of PD symptoms in women, requiring substantial increases in medication dosages, which some researchers attribute to the rapid decline in estrogen levels after delivery that were mentioned above (Reviewed by Bovenzi et al., 2024a; Shulman et al., 2000).

Looking at the above evidence from menstruation, pregnancy, and menopause, a pattern emerges where estrogen decline seems to correlate with worsened PD symptoms or higher disease risk. Still, it must be noted that the quality of research differs across these hormonal periods. For menstrual cycles, most studies are retrospective and lack objective measurements. Pregnancy studies, though useful for seeing dramatic hormonal changes, have very few cases to draw from. Menopause research has more data but faces the challenge of separating aging effects from hormonal influences.
These findings have shaped our research question, as we need to better understand how these hormonal changes affect symptoms and medication needs in WwPD. The wide variety in individual responses suggests complex biological interactions that need more advanced research methods if personalised treatment approaches for WwPD are to be achieved.

This study seeks to resolve several issues by facilitating real-time input of symptoms, medication usage, and menstrual dates and flow, thereby enabling prospective data collection. It will also gather demographic information on these women to analyse their demographic profiles and control for

specific effects. The study aims to address the knowledge gap by prospectively analysing longitudinal menstrual cycle and PD symptom data from WwPD, determining whether disparities in PD symptoms exist across various hormonal statuses, and examining the impact of hormonal therapy on these symptoms across different hormonal statuses.

1.3. Apps for PD

There is an increasing need for apps and technology for remote monitoring of PD symptoms and medications, with mobile applications emerging as promising tools to address the limitations of traditional clinical monitoring methods (Lee et al., 2022). Azadi et al. determined key features of PD-related applications, reporting large font and an easily accessible help section as critical aspects (Azadi et al., 2024). However, challenges include digital literacy among older adults and technical difficulties with app installation (Heart and Kalderon, 2013; Lee et al., 2024).

Apps for PD motor and non-motor symptom and medication tracking help with medication adherence and improve clinical visit quality (Lakshminarayana et al., 2017). Studies have supported electronic tracking compared to manual paper records by PwP, with electronic diaries showing superior compliance and accuracy, although two studies focused on motor symptoms only (Durmaz Celik et al., 2025; Lyons and Pahwa, 2007). Many existing PD apps integrate finger tapping tasks, cognitive function tests, wearable devices or online consultation (Lee et al., 2024, 2022; Lipsmeier et al., 2022; Shiraishi et al., 2023).

Such PD app platforms have been assessed to evaluate their clinical efficacy. Shiraishi et al. found that whilst PwP and carers were satisfied with online medical examinations and medication instruction, satisfaction with remote patient monitoring was only moderate. Integrating online PD care reduced burden for both PwP and carers (Shiraishi et al., 2023). Apps can benefit PwP by making them feel more involved with their care, helping with treatment compliance (Grosset and Grosset, 2005).
The MMM app used in this study, facilitates symptom selection and medication reporting to collect data for investigating patterns specific to WwPD related to hormonal status and menstrual cycles, with the aim of informing tailored care.

This study aims to utilise the My Moves Matter (MMM) app to gather data on symptoms, medications, and menstrual cycles, followed by gathering input on the user experience of the MMM app to evaluate the requirements of WwPD for an app that monitors both traditional PD information and facilitates period reporting (“My Moves Matter,” 2026a). The MMM app was developed, by Richelle Flanagan and Rene Reinbacher in 2022, to address a gap in mobile technology available to women WwPD. It is the first app to integrate menstrual cycle tracking with PD symptom and medication monitoring within a single platform, available free of charge on both iOS and Android (“My Moves Matter,” 2026b).

2. Methods.

2.1. Ethical Approval.

Initially, the ethical application was submitted through the University College Cork (UCC) Social Research Ethics Committee (SREC). The SREC redirected the study, and subsequently, the UCC Clinical Research Ethics Committee (CREC) approved the study (CREC Review Reference Number: ECM 4 (f) 05/12/2023 & ECM 3 (fff) 06/02/2024). This redirection occurred due to participants' involvement in monitoring their PD medication.

The submission documents to the CREC included the Patient Information Leaflet, which encompassed the necessary details for participants to provide informed consent, and the Agreement to Consent Form, that would be completed prior to taking the survey (Figure S6). This explicitly stated that their data would remain anonymous and be collected and stored in REDCap, a secure web application for online surveys and databases (https://projectredcap.org/about/, 2024). The Protocol and Surveys were also submitted for ethical approval.

Survey participants granted consent through initial questions preceding the information collection within the survey form. Participants responding with a “No” to any of the agreement-to-consent questions were automatically prevented from advancing to the next survey page containing the form questions.

The development of the Patient Information Leaflet and Consent Agreement Form adhered to the guidelines outlined in the CREC of the Cork Teaching Hospitals Application Manual, which provided a framework that was subsequently tailored for the PD study (https://www.ucc.ie/en/crec/, 2025).

An amendment to the follow-up survey questionnaire was desired, and therefore, CREC amendment of the study was attained. The CREC approved the study (CREC Review Reference Number: (ECM 4 (f) 05/12/2023 & ECM 3 (pp) 04/02/2025). This was due to the addition of mental health-related questions in the follow -up survey.

2.2. Participant Recruitment.

2.2.1. Recruitment.

All participants were recruited through the online survey. The Initial survey (ref survey) could be accessed by the link https://redcap.ucc.ie/surveys/?s=P3H8X484XD449AC7 or by scanning a QR code that led to the same site as the survey link, i.e., the consent form, unique code ID, and the initial survey. Participants were recruited via communication with PD, support groups and existing MMM app users.

2.2.2. Inclusion Criteria.

Participants whose data were analysed for the final dataset were those who were over 18 years old, had a PD diagnosis upon completion of the initial survey, and were assigned female biological sex at birth.
The 'stop' functionality on the REDCap website was triggered when a survey participant answered negatively to any of the agreement to consent form questions. Upon selecting “no” or “prefer not to say” in response to the question, "Have you ever been diagnosed with Parkinson's disease?" this stop functionality automatically terminated the survey. A pop-up message was designed to inform the participant that the survey was intended for women with a PD diagnosis. Checkboxes were provided, allowing the survey taker to express their preference regarding the continuation of the survey.

2.3. Survey Data Collection.

2.3.1. Survey Form Creation.

Two survey forms were made using the Online Designer feature in REDCap; “My Moves Matter UCC Parkinsons Study” (Initial survey) and “My Moves Matter Follow-Up Survey” (Follow-up survey).
Initial Survey

The Initial Survey comprised three formatted pages (Form 1S):
Page 1: Participant Information Leaflet and Consent Agreement Form:
Participants were provided with information and the consent agreement questions.
Page 2: Unique Identifier (ID) Code Generation:
Participants generated a Unique ID code (UCCXXXYYY), where the XXX represented the initial three digits of their phone number, and the YYY represented the last three characters of their postcode. This Unique ID served as a pseudonymised identifier for the MMM app and follow-up survey, allowing linkage of participant data.
Page 3: Questionnaire Page:
A copy of the initial survey, detailed in Figure S6, included demographic enquiries and questions regarding motor symptoms, NMS, PD medication, treatments and therapies. The MMM app information was matched with the data collected from this questionnaire using the Unique ID codes, thereby enabling the matching of the survey answers of participants to their MMM app tracking information. Progression to Pages 2 and 3 was dependent upon affirmative responses to the consent agreement questions; any "No" responses prevented advancement to these pages. The completion of the survey lead to a completion message which included a link and QR code to the MMM App to allow for participant progression over to the app for longitudinal data collection, this included directions on how

to log on to the MM app and the purpose of this step of the study.

Follow-Up Survey

The Follow-Up Survey comprised two formatted pages (Figure S7):

Page 1: Unique ID Code Input:

Participants were required to enter their Unique ID code (UCCXXXYYY). This code, established during the initial survey, was subsequently utilised upon the first login to the MMM app. It allowed the pseudoanonymisation and identification of participant data, ensuring consistency for researchers in identifying the same participant across the three databases: the initial survey, the MMM app, and the follow-up survey.

Page 3: Questionnaire Page:

A copy of the Follow-Up survey, detailed in Figure S7, included questions regarding motor symptoms, NMS, PD medication, treatments, therapies and mental health-related questions.

2.3.2. Survey Question Formatting.

Engaging with both surveys was voluntary, with the freedom to bypass specific questions if they so desired. No question was set to be 'required' for the completion of the form, excluding the essential step of confirming consent during the initial survey. The survey question structure took three data input formats:

- Select one box (single answer).
- Select multiple boxes.
- Provide text responses/free text entry.

Certain questions offered options such as "prefer not to say," "none of the above," or "other." If "other" was selected, participants were prompted to specify by typing a specific answer.

In certain instances, questions incorporated branching encoded queries, activating subsequent questions based on specific answer selections. This functionality was incorporated into the REDCap settings by modifying the branching logic to be triggered upon the selection of "all" within a group of answers or "any" in response to a particular question. This proved particularly significant for responses where participants were prompted to specify, in text, a certain symptom, medication, etc.

Field embedding represented another key feature in the construction of this survey using the REDCap online designer. This process involved the creation of a field capable of adopting any of the three formats specified earlier. This approach contributed to a more streamlined and user-friendly survey structure, as branching questions were not confined to appearing solely beneath the associated question. Instead, field elements could be repositioned on the survey page. This was accomplished by entering the designated variable name, intended for embedding in {} brackets (e.g., {pdmedication_01}), and incorporating it into the field label or descriptive field of another question within the same survey.

The PD medication table was made utilising the rich text editor feature of REDCap (Table 3). Each PD medication option was individually created as a field and subsequently integrated into the desired cell. Branching logic was employed in cases where certain medications could be taken in different doses, affording participants the flexibility to select their preferred medication dosage.

Table 3: PD Medications.

A

Ardatel	☐	Comtess	☐	Neupro	☐	Ropilynz	☐	Trihexyphenidryl/Benzhexol	☐
Adodespan PR	☐	Cogentin	☐	Nourianz	☐	Ropinirole	☐	Xadago	☐
Amantadine	☐	Dacepton	☐	Oprymea	☐	Rotigotine	☐	Zantiva	☐
Apogo	☐	Duodopa	☐	Ongentys	☐	Rytary	☐	Zelapar	☐
Apokyn	☐	Duopa	☐	Osmolex	☐	Sastravi	☐	Other	☐
Apomorphine	☐	Eldepryl	☐	Opicapone	☐	Selegiline	☐	None of the above	☐
Artane	☐	Gocovri	☐	Parlodel	☐	Slepark	☐		
Azilect	☐	Inbrija	☐	Parcopa	☐	Sinemet	☑		
Benztropine	☐	Ipinnia	☐	Pipexus	☐	Sinemet Plus			
Bromocriptine	☐	Kemadrin	☐	Pramipexole	☐	Spirico			
Caramet CR	☐	Kynmobi	☐	RalneaXL	☐	Stalevo			
Clonazepam	☐	Lecado	☐	Rasagiline	☐	Stanek			
Co-Beneldopa	☐	Madopar	☐	Requip	☐	Symmetrel	☐		
Co-Careldopa	☐	Mirapex	☐	Raponer	☐	Tasmar	☐		
Comtan	☐	Mirapexin	☐	Rivotril	☐	Tolcapone	☐		

Sinemet
Sinemet CR
Sinemet 12.5/50mg
Sinemet 10/100mg
Sinemet 25/100mg
Sinemet 25/250mg
Sinemet 50/200mg

B

PD Medication	Drop-down options
Amantadine	▪ Amantadine ER capsule ▪ Amantadine ER tablet
Apomorphine	▪ Apomorphine SL ▪ Apomorphine SQ
Madopar	▪ Madopar ▪ Madopar CR
Mirapex	▪ Mirapex ER ▪ Mirapexin ▪ Mirapex prolonged
Neupro	▪ Neupro ▪ Neupro patch
Parcopa	▪ Parcopa 10/100mg ▪ Parcopa 25/100mg ▪ Parcopa 25/200mg
Pramipexole	▪ Pramipexole ▪ Preamipexole ER
Requip	▪ Requip ▪ Requip XL
Ropinirole	▪ Ropinirole ▪ Ropinirole XL
Sinemet	▪ Sinemet ▪ Sinemet CR ▪ Sinemet 12.5/50mg

	▪ Sinemet 10/100mg ▪ Sinemet 25/100mg ▪ Sinemet 25/200mg ▪ Sinemet 50/200mg
Sinemet Plus	▪ Sinemet Plus ▪ Sinemet 12.5/50mg ▪ Sinemet 25/100mg
Stalevo	▪ Stalevo ▪ Stalevo 12.5/50/200mg ▪ Stalevo 18.75/75/200mg ▪ Stalevo 25/100/200mg ▪ Stalevo 31.25/125/200mg ▪ Stalevo 37.5/150/200mg ▪ Stalevo 50/200/200mg

Table 3: Parkinson's disease medication options available for selection in the initial survey. Part A displays the complete medication grid with checkbox selections. Part B shows drop-down options for medications with further subtypes of medication/dosage.

2.3.3 User Research.

The survey underwent testing for functionality; this was conducted by the research team involved in the project, comprising five individuals (n=1 with PD, n=4 controls), as well as by individuals external to the project for user research purposes, comprising four individuals (n=2 with PD, n=2 controls). Utilising the REDCap invitation sending feature, emails were dispatched containing a specific message clarifying that the email was not a recruitment communication. It emphasised that the study was in its design phase, purely for user research, and participation was entirely voluntary. Instructions on how to complete the survey were provided. Participants had the option to include comments about the survey's functionality at the end and were also given the opportunity to reach out to the research team via email with additional comments. This feedback proved instrumental in refining the survey, ensuring user functionality, and informing potential additions related to certain medications or PD symptoms.

2.4. Survey Data Cleaning and Coding.

A data coding sheet was prepared as a template for processing incoming survey data. Numerical values were assigned to categorical nominal/ordinal data to streamline data analysis; for example, 'Yes' was designated as 1 and 'No' as 2. Similarly, in the case of Menstruation Status,' 'Actively menstruating' was coded as 1, 'Post menstruation (no active period for 1 year)' as 2, 'Pregnant' as 3, 'Other' as 4, and 'Prefer not to say' as 5. This coding sheet was formulated in Excel, utilising a data dictionary retrieved from the final survey forms launched from the 'Project Home and Design' page on REDCap. A copy of the coding sheet is provided as supplementary material.

2.5. Data Analysis

Data preparation and analysis were conducted using R version 4.4.1 (R Core Team, 2024) within RStudio. Code was written with the assistance of an AI model for drafting, editing and debugging code (Claude.ai, 2025). All R scripts are provided as supplementary material.

2.5.1. Initial Survey

2.5.1.1. Data preparation and analysis (Script 03)

Survey data was exported from REDCap as a CSV file on the 3rd May 2025. The survey responses were filtered to include only completed surveys.

Variable labels were systematically applied to provide descriptive documentation for each survey field. Symptom variables were abbreviated for clarity, with original symptom descriptions converted to shortened forms (e.g., "Tremors or shaking, usually in the hands, arms, or legs" abbreviated to "Tremors") (Figure S5-S6). A standardised symptom ordering system was established to ensure consistent presentation across all analyses.

Categorical variables were converted to factors with predefined levels and labels (e.g., s01_age.factor with levels 1-13 for age ranges "18-29 years old" to "90 years old").

Unique participant IDs were created by combining the first three digits of participants' phone numbers with the last three characters of their postcodes. Both components were standardised by removing spaces, trimming whitespace, and converting postcode characters to uppercase, following the format "UCCXXXYYY". Test participant IDs were excluded, and duplicate submissions were removed, retaining the most recent completed response.

Sequential filtering was applied to create the final analytical dataset. From 607 completed initial

surveys, participants were excluded if they were test entries, had invalid unique IDs, were duplicate submissions (keeping the most recent), were male, answered ‘No’ to having a PD diagnosis, or had incomplete consent. This resulted in a final analytical dataset of 493 participants.
Participants who selected "Other" for the menstruation status question were manually reclassified. Text responses from the specification field were reviewed alongside participant age data by three researchers. Classifications categories were: Menstruating, Perimenopausal, and Postmenopausal. These manual classifications were integrated through RecordID matching, creating a definitive menstruation status table with four categories: Menstruating, Perimenopausal, Postmenopausal, and Other.
Two table types were generated for each survey variable. Individual-level data tables contained RecordID, Unique ID, numerical response codes, and descriptive text for each participant's response to that specific variable. Summary tables displayed frequency counts and percentages for each response category, excluding "Prefer not to say" responses. Multiple-choice questions, including symptoms, were restructured into binary format with separate columns for each option (1=selected, 0=not selected). Contraceptive usage was converted to overall Yes/No indicators. Medication category tables grouped individual drugs by therapeutic class, creating binary indicators for L-DOPA medications, DA agonists, MAO-B inhibitors, anticholinergics, COMT inhibitors, benzodiazepines, adenosine antagonists and NMDA antagonists. A standardised medication ordering system was established to ensure consistent presentation across all analyses.

2.5.1.2. Demographic characterisation by menstruation status (Script 04)

Demographic characteristics were analysed for participants with binary menstruation status (menstruating and postmenopausal only), excluding perimenopausal and other categories. Ten demographic variables were examined: age, gender, race, employment, marital status, age of diagnosis, symptom duration, family diagnosis, procedures, and hormonal contraceptive use. Participants who selected "Prefer not to say" responses were excluded from analysis. Question-specific sample sizes were calculated, and summary tables displayed frequency distributions and percentages for each demographic variable stratified by menstruation status.

2.5.1.3. Demographic count visualisation by menstruation status (Script 04.1)

Demographic characteristics were analysed by menstruation status, restricted to participants with binary classifications (menstruating and postmenopausal). Comparative bar charts displayed frequency counts for demographic variables and survey responses, with grouped bars showing counts for both menstruation status groups. Variables analysed included demographic characteristics (race, employment status, marital status, age distribution, age of diagnosis, symptom duration, family history of PD, and procedures/surgeries) and processed survey data (motor symptoms, secondary motor symptoms, non-motor symptoms, other medical conditions, current treatments, and other medications). "Prefer not to say" responses and missing data were excluded from all visualisations.

2.5.1.4. Symptom and medication comparison by menstruation status (Scripts 03.3 & 05)

Symptom and medication differences with menstruation status were analysed using processed tables (Script 03). Tables were filtered to include only engaged participants by retaining participants who reported at least one symptom or medication in each category excluding non-respondents. Menstruating and postmenopausal participants were compared from the menstruation status table (Script 03). For each symptom and medication variable, contingency tables were created with

symptom presence (coded as 1) or absence (coded as 0) cross-tabulated against menstruation status. Statistical test selection followed established criteria: Fisher's exact test was applied when any expected cell count was below five, with Pearson's chi-squared test used when all expected cell counts were 5 or greater. Statistical significance was set at $p<0.05$.
Each symptom and medication variable was tested independently as a separate comparison. Bonferroni correction for multiple comparisons was not applied, as the analysis was exploratory in nature, examining patterns across distinct symptom and medication domains rather than testing a single unified hypothesis.
Results were presented using diverging bar plots. Variables with p-values below 0.05 were identified as statistically significant and compiled into a summary table showing p-values and percentages for each menstruation group.

2.5.1.5. HRT usage in postmenopausal women (Script 03.2.1)

Symptom and medication tables were filtered to engaged participants only i.e. those that reported at least one symptom or medication category. Postmenopausal participants were filtered from the menstruation status table and matched with HRT usage data from the othermeds_table by UniqueID.

HRT users and non-HRT users were compared. For each symptom and medication variable, contingency tables were created, as per methods described in Section 2.5.1.4., and cross-tabulated against HRT status. Statistical analysis and bar plot creation were conducted as described in Section 2.5.1.4.

2.5.1.5.1. Age-stratified HRT analysis (03.2.1.1)

Postmenopausal participants from Script 03.2.1 were stratified by age categories and cross-tabulated with HRT status to identify viable age groups for analysis. Age groups with at least five participants in both HRT user and non-HRT user categories were included.

Three age groups met viability criteria: 50-54 years, 55-59 years and 60-64 years.
For each viable age group, participants were subset, and symptom/medication comparisons were conducted using contingency tables. Statistical analysis and bar plot creation were conducted as described in Section 2.5.1.4.

2.5.1.6. Contraceptive usage in menstruating women (Script 03.2.2)

Symptom and medication tables from Script 03 were filtered to engaged participants only, i.e. those that reported at least one symptom or medication category. Participants categorised as "Menstruating" were selected for contraceptive usage comparison.

Contraceptive status was determined from survey responses about current use of hormonal contraceptive methods (implants, intrauterine devices, injections, oral contraceptives, vaginal rings, and skin patches). Participants indicating use of any of these hormonal contraceptive methods were classified as "Contraceptive User" (Yes=1), while those indicating no current use were classified as "Non-Contraceptive User" (No=1).
Contraceptive users and non-contraceptive users were compared. For each symptom and medication variable, contingency tables were created, as per methods described in Section 2.5.1.4. and cross-tabulated against contraceptive status. Statistical analysis and bar plot creation were conducted as described in Section 2.5.1.4.

2.5.2. MMM App

2.5.2.1. App data processing and preparation (Script 1)

App data processing was conducted using R version 4.4.1 within RStudio. Following completion of the initial survey participants were invited to use the MMM mobile application for symptom and medication tracking. Participants accessed the app using their Unique IDs, enabling anonymous linkage between survey responses and app tracking data.
App data collection commenced on 29th January 2024 and ended on the 20th May 2025.

Unique IDs underwent standardisation using the same process described in Script 03: spaces were removed, characters converted to uppercase, and whitespace trimmed to ensure consistency across datasets. Test participant entries were systematically removed from both symptom and medication datasets.

Duplicate symptom entries occurring on the same date for individual participants were resolved by retaining the entry with the highest severity level. When severity levels were identical, the latest timestamp was retained.

The MMM app presented participants with 67 symptom options encompassing both motor and non-motor symptoms associated with PD. Symptoms were systematically classified as motor (n=13 symptoms) or non-motor (n=54 symptoms) based on established clinical categorisation. Original symptom descriptions were abbreviated using standardised nomenclature to facilitate data presentation and analysis (Table S7, S8).

Period tracking data was extracted from the symptom dataset and processed separately to identify menstrual cycle patterns. Period entries were clustered into menstrual phases using a gap-based algorithm: period logging entries separated by ≥6 days defined new menstrual cycles, whilst entries within 9 days were grouped within the same cycle (Bull et al., 2019; Cunningham et al., 2024).

Menstrual cycle assignment logic incorporated three categories: Before Menstrual Cycle (BMC, phase_id = -1) for symptom entries occurring before the first recorded period; Menstrual cycle (phase_id = 1, 2, 3…) numbered sequentially for each identified cycle; and After Menstrual Cycle (AMC) for entries occurring >25 days after the final recorded period to avoid inclusion of artifacts (Li et al., 2023).

Participants logged menstrual data were further categorised into three participant categories: multi-cycle participants with ≥2 menstrual cycles, single-cycle participants with 1 menstrual cycle, and no-period participants with no menstrual tracking data.

2.5.2.2. Mini data table creation (Script 2)

Mini data tables were created from the period tracking, symptom, and medication datasets generated in Script 1. These mini tables contained; period logging, symptom, scheduled medication and adhoc medication information. Where duplicate symptom entries existed for the same participant, date, and symptom type, only the entry with the highest severity level was retained.

Individual participant data tables were created through systematic merging of symptom, medication, and period tracking data using Unique IDs an example of which is shown in Table 4.

For participants with period tracking data, dates were assigned to menstrual cycles. The phase identifiers which represented menstrual cycles (see 2.5.2.1.) enabled temporal organisation of symptom and medication data relative to menstrual cycles.
Daily records were created with standardised column structures including dates, period information (flow levels and period entry indicators), symptom data (type, name, and severity), and medication information. Multiple data types recorded on the same date created separate rows, with period information appearing only on the first row per date and subsequent rows using "---" indicators to maintain data structure. Fields without applicable data for specific rows were coded as "No Entry" to maintain consistent table formatting.
Participants who did not log any periods had symptom and medication data merged chronologically without menstrual cycle phase assignment, resulting in simplified table structures containing dates, symptom details, and medication information (Table 4).
Scheduled medication information was processed separately and appeared once per participant in the first row of their individual table, formatted as pipe-delimited strings containing medication names, dosages, tablet quantities, and administration times. Ad-hoc medications created individual entries for each recorded dose, with duplicate prevention ensuring single entries per medication per date.
Quality control verification confirmed complete data preservation across all processing steps and no duplicate entries remained in final tables.
The resulting individual participant data tables provided structured tracking records enabling both participant-level analysis and aggregate comparisons. Table structures varied by participant type: period-tracking participants included phase_id columns and period flow data, whilst no-period participants contained streamlined symptom and medication tracking without menstrual cycle organisation.

Table 4: Example individual participant data table

research_id	phase_id	Dates	Flow	Symptom Type	Symptom	Severity	Adhoc med	Dosage	No. of Tablets	scheduled_meds_info
UCC0787HP	1	08/06/2024	1	Non-Motor	Constipation	1	No Entry	*NA*	*NA*	Rasagiline|1|1|8:00:00; Sinemet 25/100mg|1|1|6:30:00; Sinemet 25/100mg|25|1|12:30:00; Sinemet CR|1|1|21:30:00
UCC0787HP	1	08/06/2024	---	Non-Motor	Fatigue	1	No Entry	*NA*	*NA*	
UCC0787HP	1	09/06/2024	3	Non-Motor	Bloating	2	No Entry	*NA*	*NA*	
UCC0787HP	1	09/06/2024	---	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	1	09/06/2024	---	Non-Motor	Urinary urges	3	No Entry	*NA*	*NA*	
UCC0787HP	1	10/06/2024	3	Motor	Difficulty with typing	2	No Entry	*NA*	*NA*	
UCC0787HP	1	10/06/2024	---	Motor	Dystonia	1	No Entry	*NA*	*NA*	
UCC0787HP	1	10/06/2024	---	Non-Motor	Concentration issues	1	No Entry	*NA*	*NA*	
UCC0787HP	1	10/06/2024	---	Non-Motor	Dry eyes	2	No Entry	*NA*	*NA*	
UCC0787HP	1	10/06/2024	---	Non-Motor	Fatigue	3	No Entry	*NA*	*NA*	
UCC0787HP	1	10/06/2024	---	Non-Motor	Headache	1	No Entry	*NA*	*NA*	
UCC0787HP	1	11/06/2024	2	Motor	Difficulty with typing	1	No Entry	*NA*	*NA*	
UCC0787HP	1	11/06/2024	---	Non-Motor	Headache	2	No Entry	*NA*	*NA*	
UCC0787HP	1	11/06/2024	---	Non-Motor	Sweating	2	No Entry	*NA*	*NA*	
UCC0787HP	1	12/06/2024	1	Non-Motor	Bloating	1	No Entry	*NA*	*NA*	
UCC0787HP	1	13/06/2024	1	Motor	Dystonia	2	No Entry	*NA*	*NA*	
UCC0787HP	1	13/06/2024	---	Motor	Rigidity	1	No Entry	*NA*	*NA*	
UCC0787HP	1	13/06/2024	---	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	1	13/06/2024	---	Non-Motor	Sweating	3	No Entry	*NA*	*NA*	
UCC0787HP	1	14/06/2024	No Entry	Motor	Dystonia	1	No Entry	*NA*	*NA*	
UCC0787HP	1	14/06/2024	No Entry	Non-Motor	Sweating	3	No Entry	*NA*	*NA*	
UCC0787HP	1	15/06/2024	No Entry	Motor	Rigidity	1	No Entry	*NA*	*NA*	
UCC0787HP	1	15/06/2024	No Entry	Non-Motor	Dry eyes	1	No Entry	*NA*	*NA*	
UCC0787HP	1	15/06/2024	No Entry	Non-Motor	Fatigue	3	No Entry	*NA*	*NA*	

UCC0787HP	1	15/06/2024	No Entry	Non-Motor	Urinary urges	3	No Entry	*NA*	*NA*	
UCC0787HP	1	17/06/2024	No Entry	Non-Motor	Dry eyes	1	No Entry	*NA*	*NA*	
UCC0787HP	1	17/06/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	1	18/06/2024	No Entry	Non-Motor	Bloating	1	No Entry	*NA*	*NA*	
UCC0787HP	1	18/06/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	1	19/06/2024	No Entry	Non-Motor	Bloating	3	No Entry	*NA*	*NA*	
UCC0787HP	1	19/06/2024	No Entry	Non-Motor	Fatigue	3	No Entry	*NA*	*NA*	
UCC0787HP	1	20/06/2024	No Entry	Non-Motor	Bloating	2	No Entry	*NA*	*NA*	
UCC0787HP	1	20/06/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	1	21/06/2024	No Entry	Non-Motor	Bloating	2	No Entry	*NA*	*NA*	
UCC0787HP	1	21/06/2024	No Entry	Non-Motor	Fatigue	1	No Entry	*NA*	*NA*	
UCC0787HP	1	23/06/2024	No Entry	Non-Motor	Bloating	3	No Entry	*NA*	*NA*	
UCC0787HP	1	23/06/2024	No Entry	Non-Motor	Fatigue	3	No Entry	*NA*	*NA*	
UCC0787HP	1	24/06/2024	No Entry	Non-Motor	Bloating	3	No Entry	*NA*	*NA*	
UCC0787HP	1	24/06/2024	No Entry	Non-Motor	Daytime sleep iness	3	No Entry	*NA*	*NA*	
UCC0787HP	1	24/06/2024	No Entry	Non-Motor	Fatigue	3	No Entry	*NA*	*NA*	
UCC0787HP	1	25/06/2024	No Entry	Motor	Difficulty with typing	1	No Entry	*NA*	*NA*	
UCC0787HP	1	26/06/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	1	27/06/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	1	28/06/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	1	29/06/2024	No Entry	Non-Motor	Bloating	2	No Entry	*NA*	*NA*	
UCC0787HP	1	29/06/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	1	29/06/2024	No Entry	Non-Motor	Urinary urges	2	No Entry	*NA*	*NA*	
UCC0787HP	1	01/07/2024	No Entry	Non-Motor	Fatigue	1	No Entry	*NA*	*NA*	
UCC0787HP	1	02/07/2024	No Entry	Non-Motor	Bloating	2	No Entry	*NA*	*NA*	
UCC0787HP	1	02/07/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	1	03/07/2024	No Entry	Non-Motor	Bloating	2	No Entry	*NA*	*NA*	

UCC0787HP	1	03/07/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	1	04/07/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	1	05/07/2024	No Entry	Non-Motor	Fatigue	1	No Entry	*NA*	*NA*	
UCC0787HP	2	06/07/2024	1	Non-Motor	Constipation	1	No Entry	*NA*	*NA*	
UCC0787HP	2	06/07/2024	---	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	2	07/07/2024	2	Non-Motor	Dry eyes	2	No Entry	*NA*	*NA*	
UCC0787HP	2	07/07/2024	---	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	2	07/07/2024	---	Non-Motor	Restless legs	1	No Entry	*NA*	*NA*	
UCC0787HP	2	08/07/2024	2	Motor	Difficulty with typing	1	No Entry	*NA*	*NA*	
UCC0787HP	2	08/07/2024	---	Non-Motor	Brain fog	2	No Entry	*NA*	*NA*	
UCC0787HP	2	08/07/2024	---	Non-Motor	Fatigue	3	No Entry	*NA*	*NA*	
UCC0787HP	2	08/07/2024	---	Non-Motor	Headache	1	No Entry	*NA*	*NA*	
UCC0787HP	2	09/07/2024	2	No Entry	No Entry	No Entry	No Entry	*NA*	*NA*	
UCC0787HP	2	10/07/2024	2	No Entry	No Entry	No Entry	No Entry	*NA*	*NA*	
UCC0787HP	2	11/07/2024	2	No Entry	No Entry	No Entry	No Entry	*NA*	*NA*	
UCC0787HP	2	12/07/2024	1	Non-Motor	Fatigue	3	No Entry	*NA*	*NA*	
UCC0787HP	2	13/07/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	2	14/07/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	2	15/07/2024	No Entry	Non-Motor	Fatigue	3	No Entry	*NA*	*NA*	
UCC0787HP	2	16/07/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	2	17/07/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	2	18/07/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	2	19/07/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	
UCC0787HP	2	20/07/2024	No Entry	No Entry	No Entry	No Entry	Sinemet 25 /100mg	25	1	
UCC0787HP	2	21/07/2024	No Entry	Non-Motor	Fatigue	3	No Entry	*NA*	*NA*	
UCC0787HP	2	22/07/2024	No Entry	Non-Motor	Fatigue	2	No Entry	*NA*	*NA*	

Table 4: App tracking for one participant (Unique ID: UCC0787HP) across two tracking phases over 49 days. Phase ID: Numerical identifier for menstrual cycle phases. Flow values represent menstrual flow intensity (1=light, 2=moderate, 3=heavy). Symptom severity rated on 3-point scale (1=mild, 2=moderate, 3=severe). Scheduled medication information format: medication name|dose|frequency|time. Ad hoc medications, dosages, and tablet counts recorded when logged.

2.5.2.3. Participant flow chart statistics (Script 14.3)

Participant flow chart statistics were generated to quantify app engagement patterns by menstruation status. Menstruation status classifications were extracted from established menstruation status data. Logging criteria applied simple presence-based definitions without minimum thresholds. Period logging was defined as any entry in period data. Symptom logging required any entry in symptom data. Medication logging combined participants with valid entries in either adhoc or scheduled medication datasets, excluding missing or null values. No data quality filters or minimum logging requirements were applied, with only basic presence/absence of logging activity assessed. Overlap categories were calculated to identify participants logging multiple data types. Categories included participants logging all three data types (periods, symptoms, medications), symptoms and periods combined, and symptoms and medications combined.

2.5.2.4. Matching app IDs to initial survey IDs (Script 2.4)

App user Unique IDs were systematically matched to initial survey participant Unique IDs to establish a link between app usage data and survey responses.

This enabled verification of the menstruation status reported in the initial survey (i.e., actively menstruating, post-menstruation, pregnant, or other) and ensured that only participants who reported actively menstruating were included in menstruation-related analyses of app data.

Primary matching utilised exact string comparison between standardised IDs after applying the same standardisation process described in Script 03.

Where a direct Unique ID was not identified, alternative matching strategies addressed potential data entry errors through systematic approaches. These included numeric portion matching to identify IDs with identical digit sequences, character substitution detection for common transcription errors (particularly zero/letter O confusion), and partial string matching for formatting inconsistencies.

Unmatched app users underwent manual classification review using verification sheets containing potential matches and confidence ratings. Through this process, additional matches were identified and validated by researchers. Participants that could not be definitively matched to survey data remained unlinked, preventing erroneous data associations.

2.5.2.5. App symptom prevalence and severity (Script 14.1)

App symptom data was obtained from the dataset established in Script 2, extracting Unique IDs, symptom types (motor/non-motor), severity levels (0-3 scale), and dates. Valid symptom entries from participants were matched with menstruation status classification from Script 03, which ensure participants were study participants, not just app users.

Symptom prevalence was calculated as the percentage of participants within each

group reporting each symptom. Severity scores were calculated as mean severity levels per participant, then averaged across participants within each group.
Symptoms were analysed separately by type using the classification system from Script 03. Statistical comparisons used Pearson's chi-squared test. Statistical significance was set at $p<0.05$.
Sample size criteria required a minimum $n\geq5$ in either group for testing. Results tables displayed prevalence percentages, severity scores, and p-values with significance indicators.

2.5.2.6. Multi-cycle participant data export (Script 2.5)

Individual data tables were generated for all participants classified as multi-cycle in Script 2 and their complete app usage data was processed for export.
Quality control filtering was applied to exclude invalid menstrual cycle phase assignments. Rows with phase_id values were filtered to only include valid menstrual cycles.
Individual Excel sheets with each participant's filtered data were exported using their Unique ID as the sheet name.
The resulting Excel file contained complete tracking records, including dates, phase assignments, period information, symptom data, medication records, and scheduled medication regimens for each multi-cycle participant.
The exported participant data tables provided structured individual datasets enabling detailed menstrual cycle analyses and served as the foundation for subsequent participant-specific evaluations of app usage patterns and symptom tracking completeness.

2.5.2.7. Bleeding vs non-bleeding population analysis (Script 8)

Participants were classified into bleeding and non-bleeding groups using period tracking behaviour from multi-cycle and single-cycle participant datasets (Script 2). Participants were filtered to include only those with verified informed consent from the initial survey data. The bleeding group comprised participants who had logged menstrual flow data, whilst the non-bleeding group comprised participants who had logged symptoms but no menstrual data.
Eligibility required bleeding participants to have logged 3 or more consecutive period tracking and symptom tracking dates, whilst non-bleeding participants required only to have logged 3 or more consecutive symptom tracking dates. Average daily symptom severity was calculated for each participant by dividing the total sum of all symptom severity scores by the number of unique days on which symptoms were logged.
Statistical comparisons employed the Mann-Whitney U test and Welch's t-test for independent groups. Cohen's d was calculated using pooled standard deviation for effect size assessment. Boxplots with jittered points were created using ggplot2.

2.5.2.8. Motor vs non-motor between period tracking groups (Script 8.1)

Participants were selected using the approach described in Script 8, with additional eligibility requiring participants to have reported both symptom types.
Motor and non-motor severity were calculated separately. Calculations used total symptom severity divided by unique symptom logging days for each symptom type separately.
Statistical comparisons were performed separately for motor and non-motor symptoms using Mann-Whitney U test for independent groups.

2.5.2.9. Menstruating vs non-menstruating days analysis (Script 8.2)

Period-tracking participants from Script 8.1 were analysed to examine symptom severity differences within the same individuals on menstruating versus non-menstruating days. Each participant's symptom logging days were classified as follows: days with period entries were classified as "Menstruating" whilst symptom logging days without period entries were classified as "Non-Menstruating."
Motor and non-motor symptoms were analysed separately with different participant eligibility. Participants required motor symptoms on both day types for motor analysis and non-motor symptoms on both day types for non-motor analysis.
For each participant, average daily symptom severity was calculated by dividing total symptom severity by unique symptom logging days, calculated separately for menstruating and non-menstruating days for each symptom type independently.
Statistical analysis used Wilcoxon signed-rank tests to compare within-participant differences between day types performed separately for motor and non-motor symptoms. The paired approach compared each participant's menstruating day averages directly with their own non-menstruating day averages.

2.5.2.10. Post-menstrual symptom burden analysis (Script 9)

Using the menstruating participants identified from Script 2.5 process, post-menstrual symptom patterns were analysed for motor and non-motor symptoms separately. Analysis used the menstrual cycle dates established in Script 1 and 2. Menstruation periods were validated by filtering for periods containing at least three consecutive bleeding days and typical duration (3-7 days). Daily symptom burden was calculated as the sum of all symptom severity ratings per day, where individual symptoms were rated on a 0-3 scale (0 = absent, 1 = mild, 2 = moderate, 3 = severe), excluding "No Entry" records. Periods were included only if they contained at least six days of symptom data within the follow-up window.
Symptom burden was analysed for days during menstruation and twelve consecutive days following each menstruation period end. The relationship between days relative to menstruation end and daily symptom burden was assessed using Spearman rank correlation and linear regression modelling. Timelines were visualised using scatter

plots with LOESS smoothing curves and linear trend lines, with separate trend lines for motor and non-motor symptoms.

2.5.2.11. Selected participant timeline visualisation (Script 10)

The participants selected in Script 2.5 were assigned specific analysis date ranges. The research_ids_with_dates dataset was created, containing each participant's Unique ID paired with their designated analysis start and end dates.
Individual timeline plots were generated for each participant displaying symptom severity scores (0-3 scale) plotted against date, with motor and non-motor symptoms differentiated by colour coding and menstrual flow periods overlaid on the visualisation using different pink/red shades for flow levels 1-3. Plots included LOESS smoothing curves with severity constraints and adaptive date breaks based on tracking duration.

2.5.2.12. Menstrual phase symptom burden analysis (Script 11 & 11.1)

A 3-phase menstrual cycle analysis compared symptom burden across cycle phases using participants and timeline dates from Script 10. Symptom and period data was obtained from tables established in Script 2. Menstrual cycles were identified by filtering period entries to timeline dates, requiring ≥3 consecutive bleeding days per cycle. Phase boundaries were defined as: Premenstrual (5 days before cycle start), Menstrual (bleeding days), and Post-menstrual (cycle end to 5 days before next cycle), with boundaries truncated at timeline limits when necessary.

Initially, a 2-phase model (follicular and luteal phases) was attempted using cycle assignment logic from Script 1, but insufficient data for determining individual cycle lengths precluded accurate phase boundary determination. The 3-phase approach was adopted as it provided standardised analysis with conservative phase limits.

Symptom burden was calculated by collecting valid symptom entries within each phase period, excluding "No Entry" records. Daily symptom burden was calculated as the sum of all symptom severity scores reported on individual days. For each participant and phase, median daily burden was calculated across all days within that phase. Analysis was restricted to participants with data for all three phases and both symptom types. Boxplots visualised each participant's median daily symptom burden within each menstrual phase. Paired Wilcoxon signed-rank tests compared median burden scores between phases using complete case analysis, with six pairwise comparisons performed (three phase comparisons for each symptom type).

2.5.2.13. Symptom entry analysis across menstrual phases (Script 6)

Using the phase boundaries defined in Script 11, symptom entries were counted for each menstrual cycle phase. Symptom data was obtained from the datasets established in Script 2, excluding "No Entry" responses. Symptom entries were matched to each of the 3 phases outlined in Script 11. To account for varying phase durations, the number of symptoms per day was calculated for each phase period by

dividing total symptoms by phase duration in days. For each participant and phase, mean symptoms per day was calculated across all phase periods. Analysis was restricted to participants with complete data for all three phases. Group-level statistical analysis was performed using the Friedman test for repeated measures to test for differences across the three menstrual phases. Where the Friedman test showed significant differences ($p < 0.05$), post-hoc pairwise comparisons between phases were conducted using paired Wilcoxon signed-rank tests, with Bonferroni correction applied to control for multiple comparisons (three comparisons: Premenstrual vs Menstrual, Premenstrual vs Post-menstrual, Menstrual vs Post-menstrual; adjusted significance level $\alpha = 0.0167$). Where the Friedman test was non-significant, post-hoc comparisons were not performed.

2.5.2.14. Individual symptom boxplot analysis across menstrual phases (Script 4.2)

Individual symptom analysis across menstrual cycle phases was conducted using the participants established in Script 11.

Symptom representation criteria required ≥3 participants who reported each symptom in ≥1 menstrual cycle phase. For manually specified symptoms, participant-level average severity scores were calculated within each phase (premenstrual, menstrual, post-menstrual) from individual symptom entries rated on a 0-3 severity scale.

. Statistical analysis employed linear mixed effects models with participant as a random intercept. Overall phase effects were assessed using F-tests. Where the F-test showed significant phase effects ($p < 0.05$), post-hoc pairwise comparisons between phases were conducted with Bonferroni correction for three comparisons (Premenstrual vs Menstrual, Premenstrual vs Post-menstrual, Menstrual vs Post-menstrual; adjusted significance threshold $p < 0.0167$).

Boxplot visualizations were created using ggplot2, displaying individual participant data points overlaid on boxplots grouped by menstrual phase.

2.5.2.15. Medication categorisation (Script 2.6)

Medication categorisation was applied to the mini medication datasets created in Script 2. Processing was conducted separately for the three participant groups outlined in Script 1.

Medication data underwent standardisation and selective deduplication prior to categorisation. Scheduled medication entries were deduplicated to retain one entry per unique combination of participant, medication name, and dosage, eliminating multiple identical regimen entries. Adhoc medication entries were preserved without deduplication to maintain complete usage records.

Drug classification utilised an Excel table containing medications organised within two overarching systems: PD medications (L-DOPA, DA Agonist, MAO-B inhibitor, Anticholinergic, COMT Inhibitor) and non-PD medications (Contraceptives, Estrogen-

only, Combined Estrogen/Progesterone, Progesterone-only, Antidepressants, Anxiolytics). A drug_classification_lookup table was created from the reference file and applied using left_join() functions for exact text matching between participant-reported medication names and standardised drug classifications.
Three final medication tables were generated combining scheduled and adhoc entries: multi_cycle_medication_table, single_cycle_medication_table, and no_period_medication_table.

2.5.2.16. L-DOPA dose comparison between menstruating and postmenopausal participants (Script 15)

L-DOPA medication users were identified from mini scheduled medication dataset created in Script 2 to compare daily dosing requirements between menstruating and postmenopausal participants. Only L-DOPA components were extracted from combination medications: Sinemet 25/100mg used the 100mg L-DOPA component, excluding the 25mg carbidopa component; Stalevo 25/100/200mg used the 100mg L-DOPA component, excluding carbidopa and entacapone components. Extraction used a lookup table approach for parseable medication names, with logged dosage values retained for unparseable medications.
Daily L-DOPA doses were calculated by summing L-DOPA components across all scheduled medications per participant. Participants were matched with menstruation status from new_menstruationstatus_table (Script 03) using Unique IDs.
The analysis compared menstruating and postmenopausal participants only.
A Mann-Whitney U test was used. Effect size was calculated using Cohen's d. Extreme outliers exceeding clinical dosing ranges were flagged but retained in statistical calculations. Statistical significance was set at $p<0.05$. Boxplots displayed were created using ggplot2.

2.5.2.17. Follow-Up cohort selection (Script 03)

An engagement criterion was established to identify participants suitable for follow-up survey invitations. Participants meeting eligibility requirements demonstrated ≥3 distinct days of logging activity in any of the following categories: Period tracking (≥3 distinct days of menstrual logging); Symptom tracking (≥3 distinct days of symptom recording); Medication tracking (≥3 distinct days of valid medication adherence logging, combining scheduled medications taken on time/early/late and ad-hoc medications taken as needed).

2.5.3. Follow- Up Survey

2.5.3.1. Follow-up survey data preparation and analysis (Script 103 &102.0.)

Data preparation followed the same approach described in Script 03 with the following differences. Follow-up survey data was exported from REDCap as CSV file on 26th June 2025.

Unique IDs used variables p2 and a2 instead of p and a, following the same standardisation process. Follow-up participants were validated against the combined cohort of initial survey participants (Script 03), app users (Script 2.4), and invited follow-up participants with non-study participants removed to create the follow-up dataset. Additional variables specific to the follow-up survey included mental health status (s02_mental_health), mental health effects (s02_mental_health_effect), and support mechanisms (s02_supports). Manual classification of menstruation status was not required as follow-up participants retained established classifications. Variable labelling and table generation processes were identical to Script 03 with the prefix 'fu_' applied to all table names.

2.5.3.2. Follow-up survey symptom and medication comparison by menstruation status (Script 105)

Approach was identical to Script 05. Follow-up data from Script 103 included 23 participants with clear menstruation status. Contingency tables were created using follow-up symptom and medication data with fu_ prefixes (fu_motorsymptoms_table, fu_nonmotorsymptoms_table, fu_med_category_table). Additional analysis included follow-up-specific variables such as mental health measures (fu_mental_health_table, fu_mental_health_effect_table) and support mechanisms (fu_supports_table).

2.5.3.3. App experience response extraction (Script 103.2)

App experience responses were extracted from follow-up survey participants. From the follow-up dataset, app-invited participants were identified.

Three categories of app experience data was extracted using specific REDCap variables: life events ("Have you experienced any significant life events or changes since you completed our initial survey?"), app tracking experience ("How did you find your experience tracking on the app overall?"), and app usability survey (Table 5).

Three separate tables were created with all responses preserved as raw participant input. Results were exported to Excel format. The responses to the app tracking experience question were manually categorised into positive, negative, and mixed experiences following data export.

2.5.3.4. App usability visualisations (Script 103.3)

To examine to app usability survey responses, two complementary visualisation to were created. The 18-question usability survey responses were measured on a 7-point Likert scale (1 = Strongly Disagree, 7 = Strongly Agree). Individual question analysis created box plots displaying response distributions for each of the 18 usability questions.

Summary statistics including mean, median, and mode, were calculated for each question using direct frequency tables.
Thematic analysis grouped the 18 questions into five categories: Core Usability (5 questions), User Experience & Satisfaction (4 questions), Practical Integration (3 questions), Future Intent (1 question), and Health & Clinical Value (5 questions).

Table 5. App Usability Survey Questions Thematic Categorisation

Theme	**Questions**
Core Usability	The app was easy to use
	The app was easy to learn
	Navigation was consistent throughout the app
	The interface provided all necessary functions
	It was easy to recover from mistakes
User Experience & Satisfaction	I liked the app interface
	Information was well organised
	The app provided acknowledgement of my progress
	Overall, I was satisfied with the app
Practical Integration	I felt comfortable using the app in social settings
	The app fitted well into my schedule
	The app worked well even with poor internet connection
Future Intent	I would use this app again
Health & Clinical Value	The app was useful for my health and wellbeing
	The app improved my access to healthcare
	The app helped me manage my health
	The app had all the functions I expected
	The app provided acceptable healthcare services

Table 5: Survey questions and their corresponding thematic categories used in the usability assessment of the MyMovesMatter app. Survey questions were grouped thematically for analysis as described in Section 2.5.3.4.

3. Results

3.1. Participant Flow Chart Statistics

A total of 493 participants completed the initial survey. Of these, 125 were categorised as menstruating and 283 as postmenopausal. The remaining 85 participants did not fit either category due to being perimenopausal, classified as 'other', or not providing menstruation status information.

48 menstruating participants (38.4% retention) and 83 postmenopausal participants (29.3% retention) used the app. Among menstruating participants, n=36 logged periods, 45 logged symptoms, and n=35 logged medications. Among postmenopausal participants, n=72 logged symptoms and n=64 logged medications.

27 menstruating participants logged all three data types, compared to n=53 postmenopausal participants who logged both symptoms and medications. 17 participants met sufficient information criteria, as established in Section 2.5.2.6.

Figure 4: Participant Flow Chart from Initial Survey to App Engagement

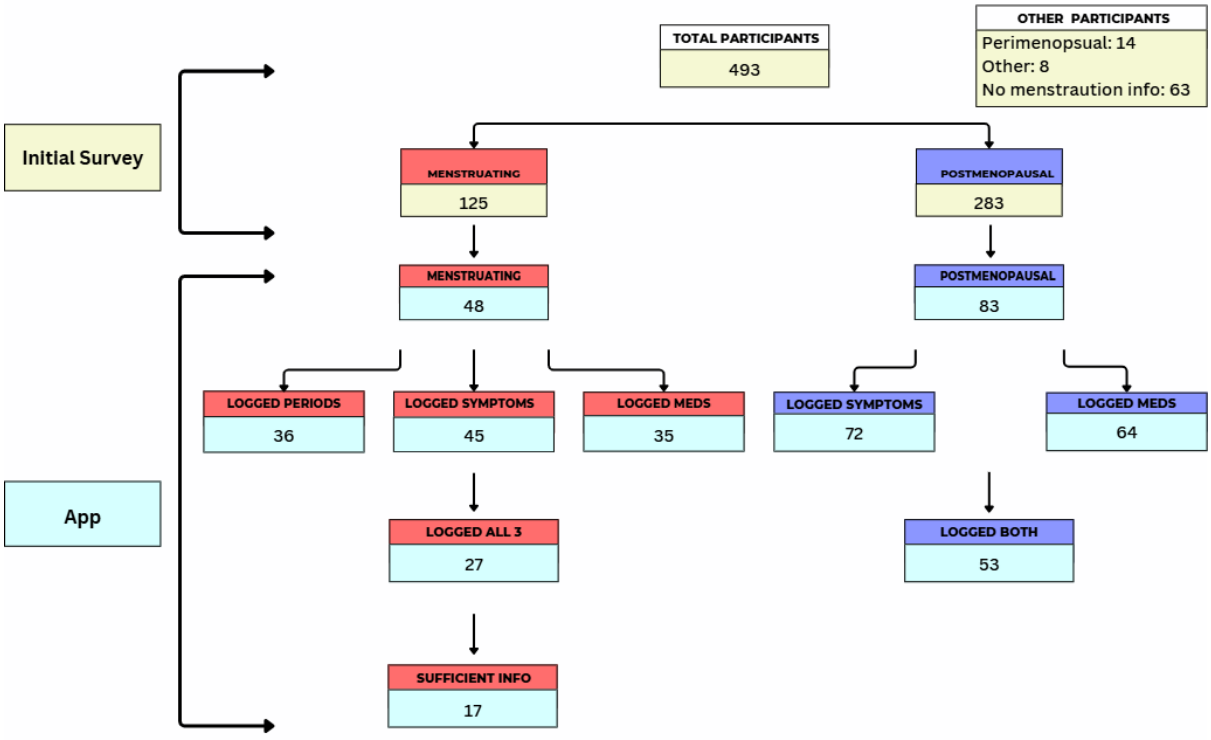


Figure 4: Participant flow from initial survey (n = 493) through app engagement and data logging patterns. Menstruating participants = red boxes, postmenopausal participants = blue boxes. Initial survey usage numbers are shown in yellow. App usage numbers are shown in teal. Seventeen participants met 'sufficient logging' criteria, as established in Section 2.5.2.6.

3.2 Survey Data

3.2.1. Demographics by Menstruation Status

Demographic characteristics were analysed for 408 participants with binary menstruation status: 125 menstruating and 283 postmenopausal participants (Table 6). Question-specific sample sizes varied due to individual response patterns, with complete demographic data available for most participants in both groups.

Age distribution differed between groups, with menstruating participants concentrated in the 40-49 years category (65.8%) compared to postmenopausal participants who were distributed across older age groups, with the highest proportions in 55-59 years (26.4%) and 60-64 years (26.1%). Employment status that showed menstruating participants had higher rates of full-time employment (42.6% vs 15.3%) whilst postmenopausal participants had higher retirement rates (43.1% vs 2.5%).

Age of diagnosis patterns showed that menstruating participants were predominantly diagnosed in their 30s (33.6%) and 40s (55.7%), compared to postmenopausal participants, who had higher proportions diagnosed in their 50s (23.4% aged 50-54 years, 18.3% aged 55-59 years) and 60s (16.9% aged 60-64 years, 10.6% aged 65-69 years). Contraceptive use was higher in menstruating participants (29.3% vs 8.0%).

Both groups were predominantly white (92.5% menstruating, 97.1% postmenopausal) and married (65.0% menstruating, 70.5% postmenopausal). Family history of PD was reported by 24.0% of menstruating participants and 19.4% of postmenopausal participants.

Table 6: Demographics by Menstruation Status

Category	Option	Menstruating (%)	Postmenopausal (%)
Age (M: N=123, PM: N=276)	**30 - 39 years**	12.2	0.36
	40 - 49 years	65.85	2.9
	50 - 54 years	18.7	14.13
	55 - 59 years	3.25	26.45
	60 - 64 years	0	26.09
	65 - 69 years	0	14.49
	70 - 74 years	0	10.14
	>75 years	0	5.43
Race (M: N=120, PM: N=275)	**White**	92.5	97.09
Employment (M: N=122, PM: N=274)	**Employed (full-time)**	42.62	15.33
	Employed (part-time)	20.49	11.31
	Retired	2.46	43.07
Marital Status (M: N=123, PM: N=271)	**Single**	12.2	7.38
	In a relationship	9.76	4.43
	Married	65.04	70.48
	Divorced	6.5	12.92
Age of Diagnosis (M: N=122, PM: N=273)	**18 - 29 years**	4.1	0.37
	30 - 39 years	33.61	3.3
	40 - 49 years	55.74	21.25
	50 - 54 years	5.74	23.44
	55 - 59 years	0.82	18.32
	60 - 64 years	0	16.85
	>65 years	0	16.49
PD Symptom Duration (M: N=113, PM: N=256)	**0 - 3 years**	27.43	21.88
	4 - 6 years	23.89	27.34
	7 - 9 years	23.01	16.02
	10 - 12 years	10.62	12.89
	13 - 15 years	6.19	8.98
	>15 years	8.85	12.89
PD Family Diagnosis (M: N=121, PM: N=273)	**Yes**	23.97	19.41
PD-related Procedure (M: N=118, PM: N=260)	**Yes**	15.25	14.62
Hormonal Contraceptive Use (M: N=123, PM: N=276)	**Yes**	29.27	7.97

Table 6: Demographic characteristics of menstruating (n=125) and postmenopausal (n=283) women showing percentage distribution within each menstruation status. Question-specific sample sizes: Age (M: N=123, PM: N=276); Race (M: N=120, PM: N=275); Employment (M: N=122, PM: N=274); Marital Status (M: N=123, PM: N=271); Age of Diagnosis (M: N=122, PM: N=273); Symptom Duration (M: N=113, PM: N=256); Family Diagnosis (M: N=121, PM: N=273); Procedures (M: N=118, PM: N=260); Hormonal Contraceptive Use (M: N=123, PM: N=276).
Abbreviations and survey question descriptions :M = Menstruating, PM = Postmenopausal. Age = "What is your age?"; Race = "What is your race/ethnicity?"; Employment = "What is your employment status?"; Marital Status = "What is your marital status?"; Age of Diagnosis = "What age were you when you were diagnosed with Parkinson's disease?"; PD Symptom Duration = "How long have you been experiencing Parkinson's symptoms?"; PD Family Diagnosis = "Do you have any family members who also have Parkinson's disease?"; PD-related Procedure = "Have you ever had any surgeries or procedures related to your Parkinson's disease?"; Hormonal Contraceptive Use = "Do you take an oral Hormonal Contraceptive?"

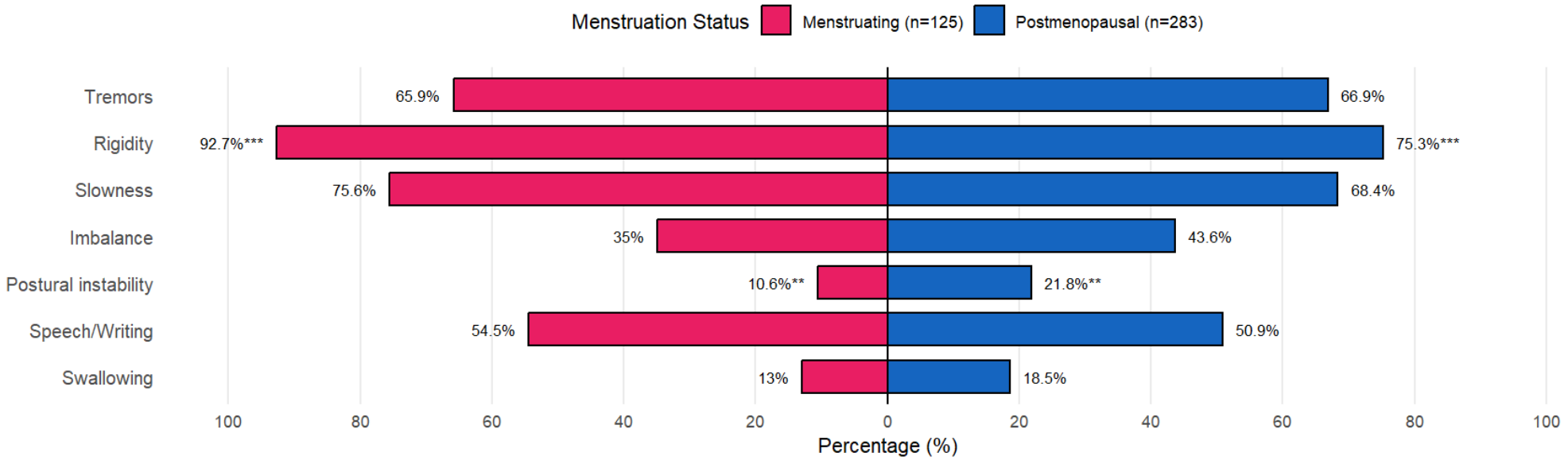
3.2.2. Symptom and PD Medication Prevalence by Menstruation Status

Motor and non-motor symptom prevalence was compared between menstruating (n=125) and postmenopausal (n=283) women who provided symptom and medication data. Motor symptoms analysis included seven symptoms, with significant differences observed for rigidity, postural instability and dyskinesia (Figure 5A) (Figure B) (Table 7). Rigidity was reported by 92.7% of menstruating women compared to 75.3% of postmenopausal women (p=0.0000484; Odds Ratio(OR)=4.15, 95% CI: 1.97–9.82). Dyskinesia was reported by 50.4% of menstruating women compared to 36.2% of postmenopausal women (p=0.00776; OR=1.79, 95% CI: 1.14–2.81). Postural instability was reported by 10.6% of menstruating women compared to 21.8% of postmenopausal women (p=0.00737; OR=2.36, 95% CI: 1.21–4.89) (Table 7).
Non-motor symptoms analysis encompassed 17 symptoms, with five showing significant differences between groups (Figure 5C). Depression and anxiety was reported by 60.7% of menstruating women versus 48.5% of postmenopausal women (p=0.0259; OR=1.63, 95% CI: 1.04–2.59). Hyposmia was less prevalent in menstruating women (34.4%) compared to postmenopausal women (50.4%) (p=0.00333; OR=1.93, 95% CI: 1.22–3.10). Constipation was reported by 46.7% of menstruating women versus 57.8% of postmenopausal women (p=0.0419; OR=1.56, 95% CI: 0.99–2.46). Pain and discomfort was reported by 59% of menstruating women versus 45.9% of postmenopausal women (p=0.0164; OR=1.69, 95% CI: 1.08–2.68). Mood swings and emotional changes showed the largest difference, with 48.4% of menstruating women reporting this symptom compared to 28.9% of postmenopausal women (p=0.000181; OR=2.30, 95% CI: 1.45–3.67).
Medication category analysis examined eight therapeutic classes, with no statistically

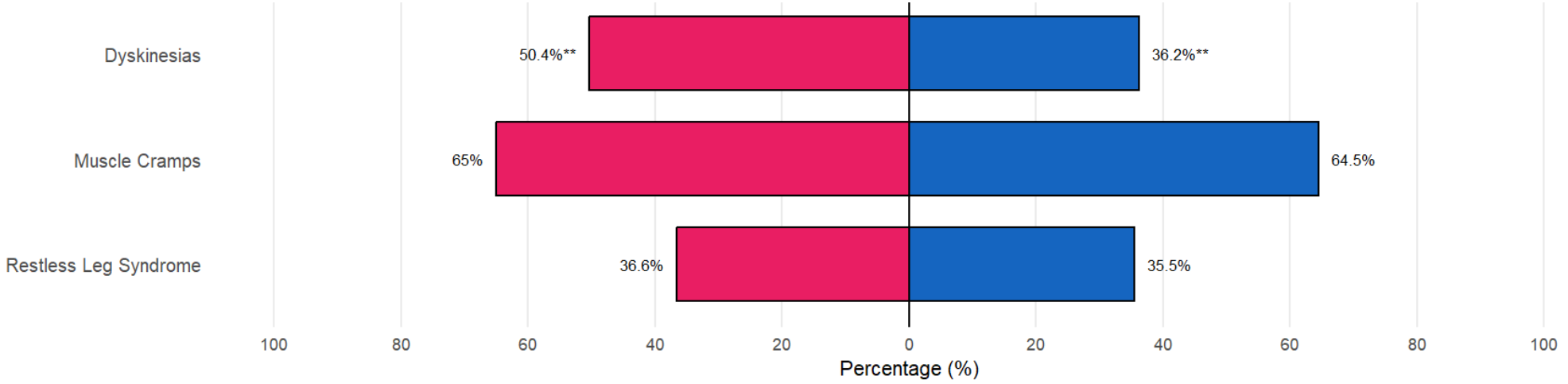
significant differences observed between menstruating and postmenopausal women (Figure 5D). L-DOPA usage was similar between groups (80.7% menstruating vs 86.4% postmenopausal), as were DA agonist usage rates (39.4% vs 38.3%) and other medication categories.
Statistically significant results are summarised in Table 7.

Figure 5A-D: Symptoms and PD Medication by Menstruation Status

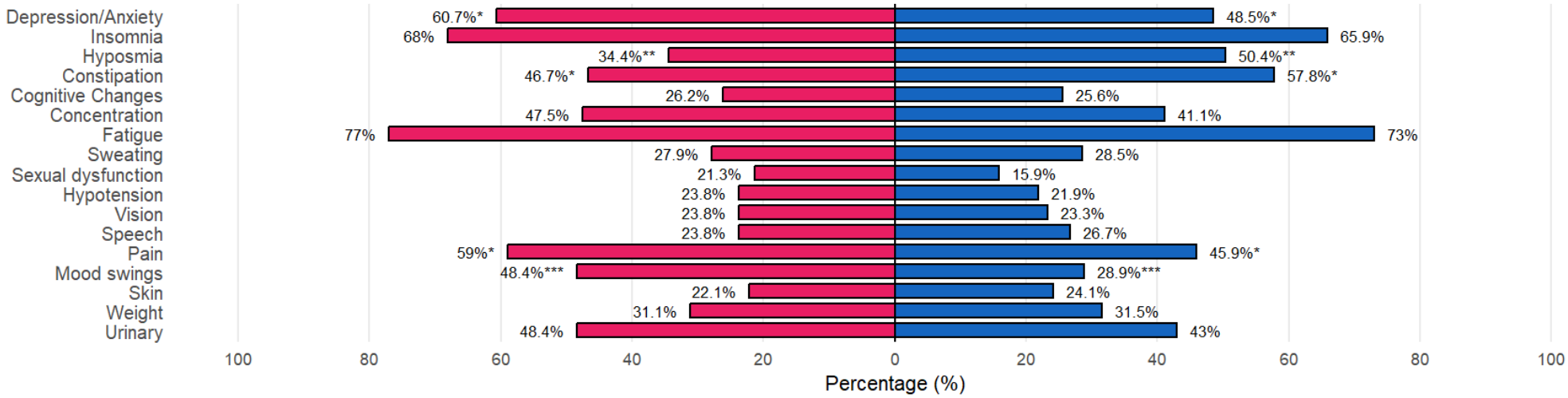
A



B



C



D

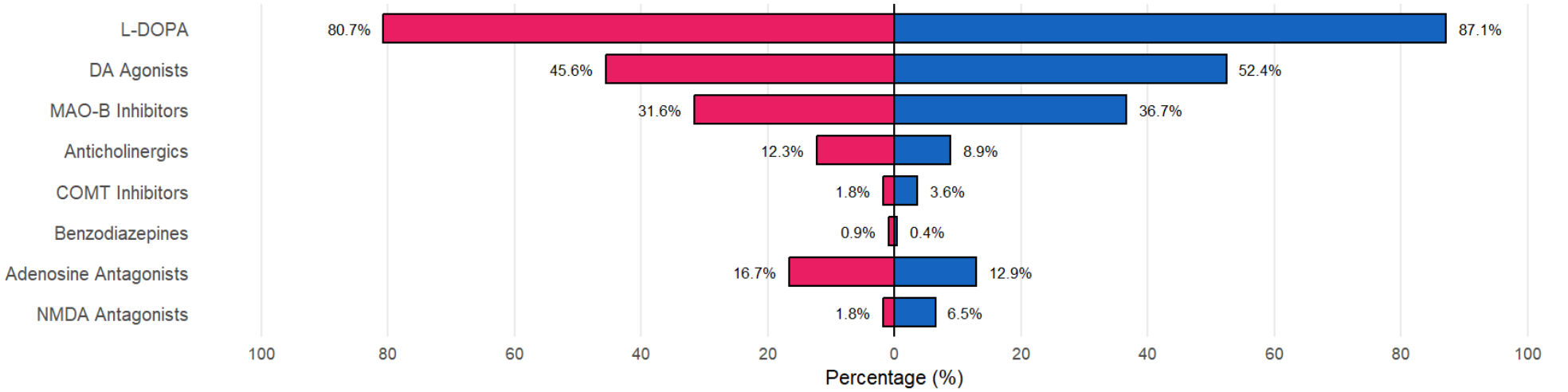


*Figure 5: (A)Motor symptom prevalence. (B) Secondary motor symptoms. (C) Non-motor symptom prevalence. (D) PD Medication category usage in menstruating (n=125, pink) and postmenopausal (n=283, blue) women. Statistical significance was determined using chi-squared tests, with Fisher's exact tests applied when expected cell counts <5. * p<0.05, ** p<0.01, *** p<0.001.*

Symptom Abbreviations:

Motor Symptoms: Tremors = Tremors or shaking, usually in the hands, arms, or legs; Rigidity = Stiffness or rigidity of the muscles; Slowness = Slowness of movement (bradykinesia); Imbalance = Difficulty with balance and coordination; Postural instability = Postural instability which can cause falls; Speech/Writing = Changes in speech and writing, such as slurred speech or small handwriting; Swallowing = Difficulty with swallowing.
Non-Motor Symptoms: Depression/Anxiety = Depression and anxiety; Insomnia = Sleep disturbances, including insomnia; Hyposmia = Loss of sense of smell; Constipation = Constipation; Cognitive Changes = Cognitive changes, including memory problems; Concentration = Difficulty concentrating; Fatigue = Fatigue and lack of energy; Sweating = Excessive sweating; Sexual dysfunction = Sexual dysfunction; Hypotension = Orthostatic hypotension; Vision = Visual disturbances, including blurred vision; Speech = Speech and swallowing problems; Pain = Pain and discomfort; Mood swings = Mood swings and emotional changes; Skin = Skin problems, including seborrheic dermatitis; Weight = Weight loss or gain; Urinary = Urinary Problems.
Medication Categories: L-DOPA medications = Madopar, Sinemet, Sinemet Plus, Co-Careldopa, Co-Beneldopa, Stalevo, Duodopa, Duopa, Adodespan PR, Caramet CR, Inbrija, Lecado, Parcopa, Rytary; DA (Dopamine) Agonists = Mirapex, Mirapexin, Pramipexole, Requip, Ropinirole, Neupro, Rotigotine, Apomorphine, Apokyn, Bromocriptine, Parlodel, Adartel, Apogo, Dacepton, Ipinnia, Kynmobi, Oprymea, Pipexus, RalneaXL, Raponer, Ropilynz, Spirico, Zantiva; MAOB-Inhibitors = Azilect, Rasagiline, Eldepryl, Selegiline, Zelapar, Xadago. Anticholinergics: Artane, Trihexyphenidryl/Benzhexol, Benztropine, Cogentin, Kemadrin; COMT Inhibitors = Comtan, Comtess, Tasmar, Tolcapone, Opicapone, Ongentys, Sastravi, Stanek; Benzodiazepines = Clonazepam, Rivotril; Adenosine_Antagonist = Nourianz; NMDA antagonists = Amantadine, Gocovri, Osmolex, Symmetrel.

Table 7: Significant Differences between Menstruation Status and Clinical Variables

Category	Variable	P-Value	Menstruating %	Postmenopausal %
Motor Symptoms	Rigidity	0.0000484	92.7%	75.3%
	Postural instability	0.00737	10.6%	21.8%
Non-Motor Symptoms	Depression/Anxiety	0.0259	60.7%	48.5%
	Hyposmia	0.00333	34.4%	50.4%
	Constipation	0.0419	46.7%	57.8%
	Pain	0.0164	59.0%	45.9%
	Mood swings	0.000182	48.4%	28.9%

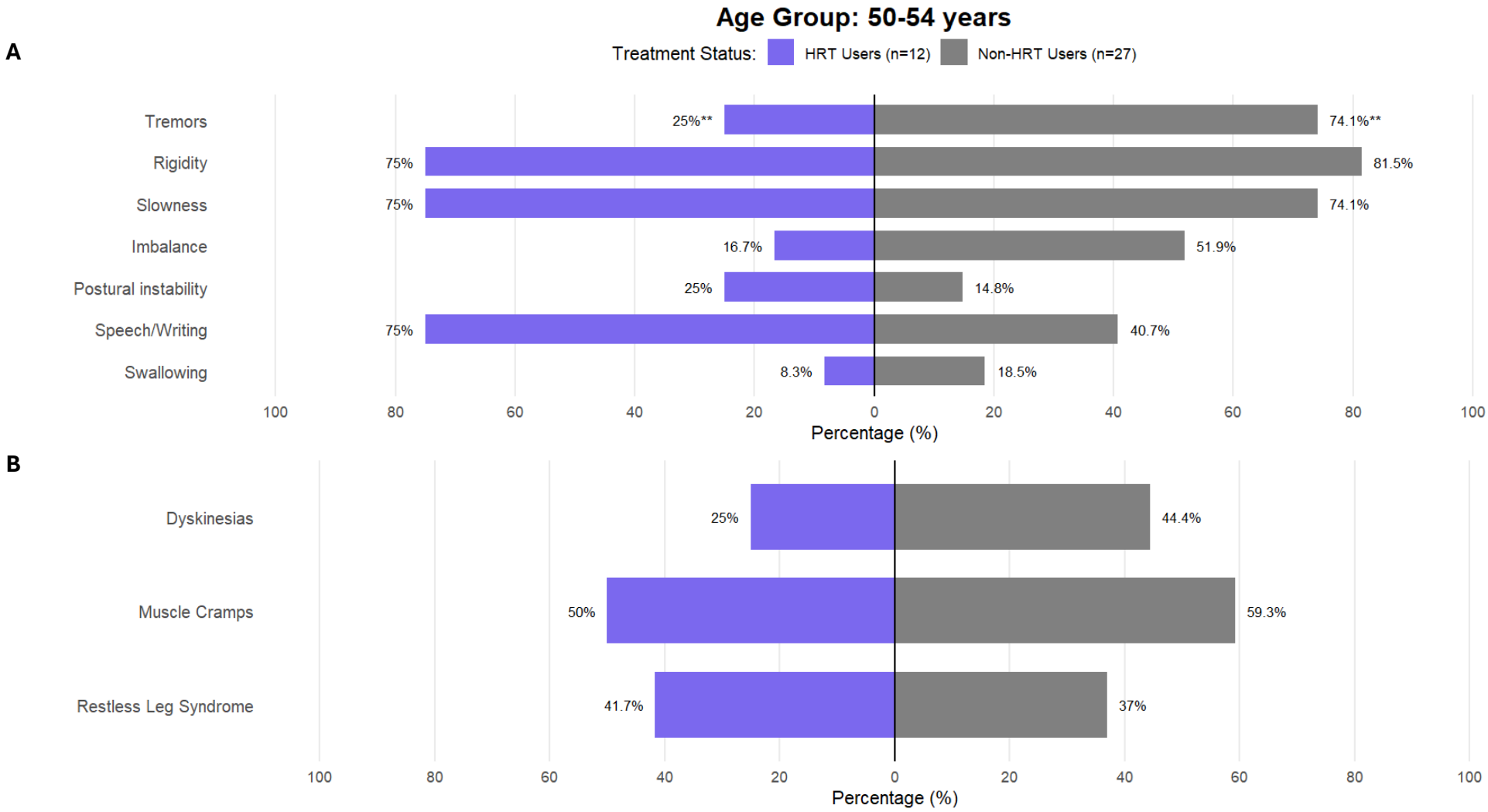
Table 7: Statistically significant differences ($p<0.05$) between menstruation status and initial survey symptoms reported, in menstruating (n=125) and postmenopausal (n=283) women. Statistical significance was determined using chi-squared tests without continuity correction. Bonferroni correction was not applied due to the exploratory nature of the analysis.
Symptom Abbreviations:
Motor Symptoms: Rigidity = Stiffness or rigidity of the muscles; Postural instability = Postural instability which can cause falls.
Non-Motor Symptoms: Depression/Anxiety = Depression and anxiety; Hyposmia: Loss of sense of smell; Constipation: Constipation; Pain: Pain and discomfort; Mood swings: Mood swings and emotional changes.

3.2.3. Age-Stratified HRT Analysis in Postmenopausal Women

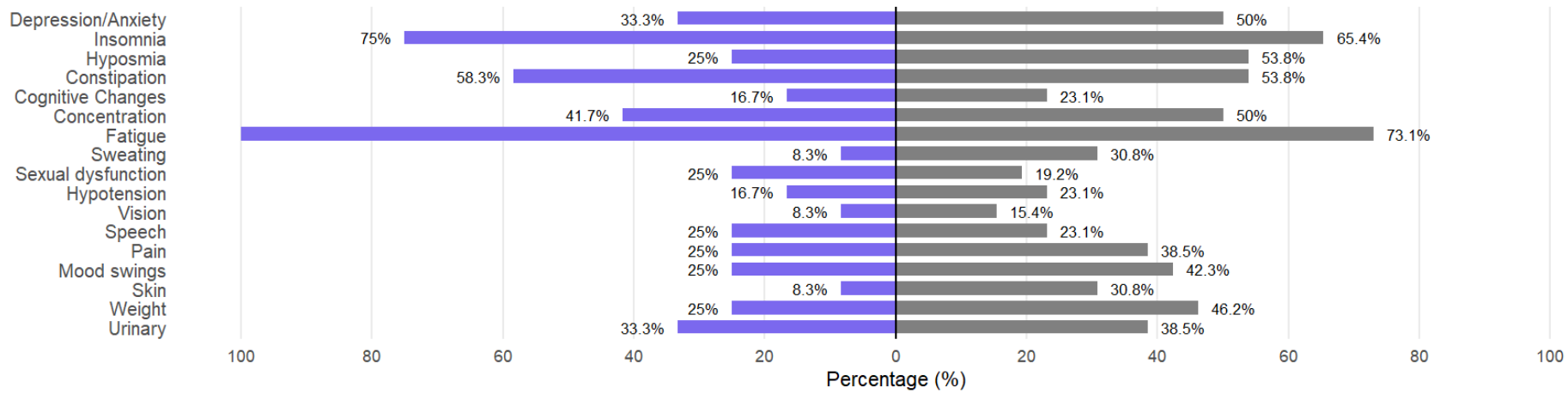
Age-stratified analysis was conducted to control for potential age confounding effects in HRT comparisons among postmenopausal women who provided symptom and medication data. Overall HRT analysis results are presented in supplementary material (Figure S3). Three age groups were viable for analysis: 50-54 years (n=12 HRT users, n=27 non-HRT users), 55-59 years (n=12 HRT users, n=61 non-HRT users), and 60-64 years (n=11 HRT users, n=60 non-HRT users).

Figure 6A-I present the prevalence of motor symptoms and non-motor symptoms reported by women, stratified by HRT status within each of the three age groups. Motor and non-motor symptom prevalence was compared between HRT users and non-HRT users within each age group (Figure 6A-F). Most symptoms demonstrated no significant differences between HRT users and non-HRT users across all age groups. However, one significant difference was observed. In the 50-54 years age group, Tremors or shaking, usually in the hands, arms, or legs were significantly less prevalent amongst HRT users compared to non-HRT users (25.0% vs 74.1%, p=0.006; OR=8.02, 95% CI: 1.48–59.80).

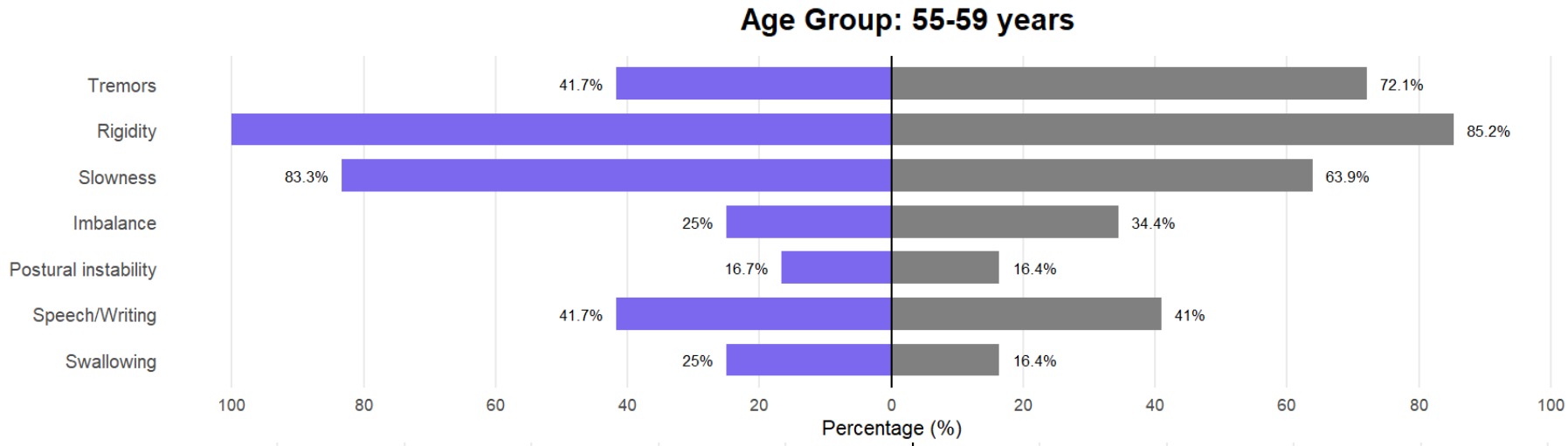
Figure 6A-F: Motor and non-motor symptoms by HRT status split by age groups



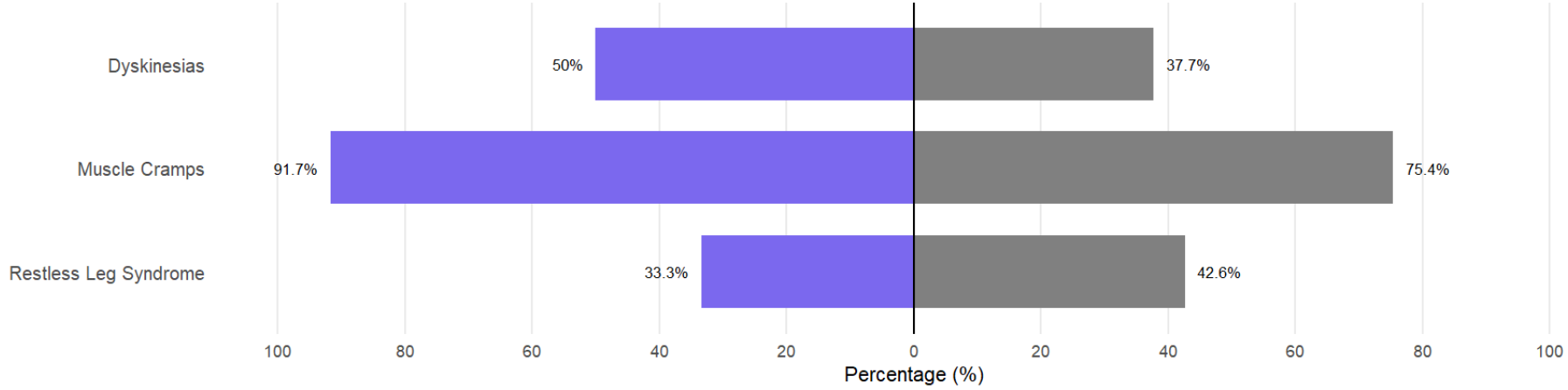
C


Depression/Anxiety
Insomnia
Hyposmia
Constipation
Cognitive Changes
Concentration
Fatigue
Sweating
Sexual dysfunction
Hypotension
Vision
Speech
Pain
Mood swings
Skin
Weight
Urinary
33.3%
50%
75%
65.4%
25%
53.8%
58.3%
53.8%
16.7%
23.1%
41.7%
50%
73.1%
8.3%
30.8%
25%
19.2%
16.7%
23.1%
8.3%
15.4%
25%
23.1%
25%
38.5%
25%
42.3%
8.3%
30.8%
25%
46.2%
33.3%
38.5%
100
80
60
40
20
0
20
40
60
80
100
Percentage (%)

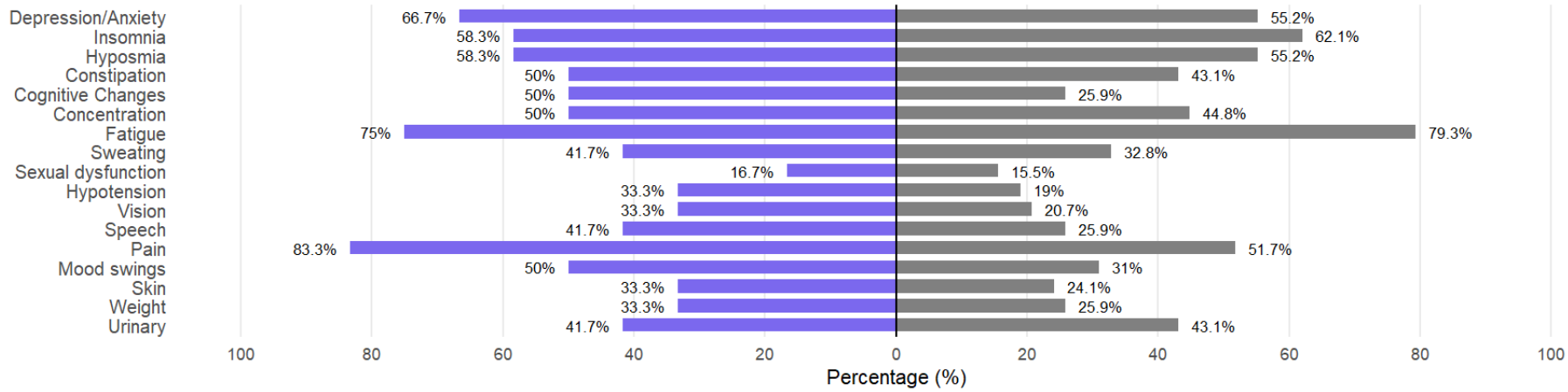
D


Age Group: 55-59 years
Tremors
Rigidity
Slowness
Imbalance
Postural instability
Speech/Writing
Swallowing
41.7%
72.1%
83.3%
85.2%
25%
63.9%
16.7%
34.4%
41.7%
16.4%
25%
41%
16.4%
100
80
60
40
20
0
20
40
60
80
100
Percentage (%)

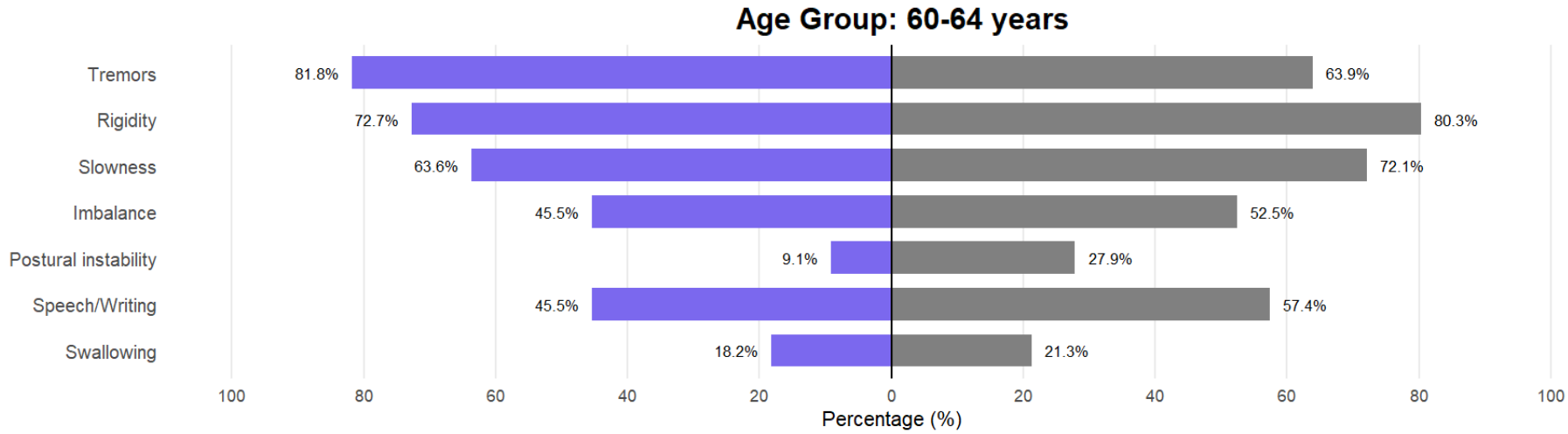
E


Dyskinesias
Muscle Cramps
Restless Leg Syndrome
50%
37.7%
91.7%
75.4%
33.3%
42.6%
100
80
60
40
20
0
20
40
60
80
100
Percentage (%)

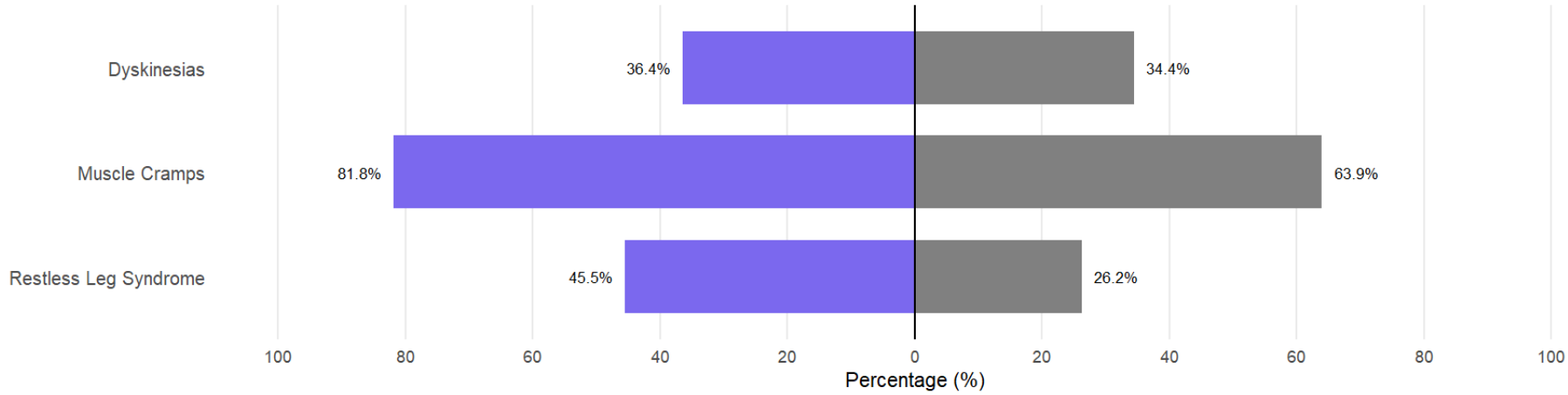
F


Depression/Anxiety
Insomnia
Hyposmia
Constipation
Cognitive Changes
Concentration
Fatigue
Sweating
Sexual dysfunction
Hypotension
Vision
Speech
Pain
Mood swings
Skin
Weight
Urinary
66.7%
58.3%
58.3%
50%
50%
50%
75%
41.7%
16.7%
33.3%
33.3%
41.7%
83.3%
50%
33.3%
33.3%
41.7%
55.2%
62.1%
55.2%
43.1%
25.9%
44.8%
79.3%
32.8%
15.5%
19%
20.7%
25.9%
51.7%
31%
24.1%
25.9%
43.1%
100
80
60
40
20
0
20
40
60
80
100
Percentage (%)

G


Age Group: 60-64 years
Tremors
Rigidity
Slowness
Imbalance
Postural instability
Speech/Writing
Swallowing
81.8%
63.9%
72.7%
80.3%
63.6%
72.1%
45.5%
52.5%
9.1%
27.9%
45.5%
57.4%
18.2%
21.3%
100
80
60
40
20
0
20
40
60
80
100
Percentage (%)

H


Dyskinesias
Muscle Cramps
Restless Leg Syndrome
36.4%
34.4%
81.8%
63.9%
45.5%
26.2%
100
80
60
40
20
0
20
40
60
80
100
Percentage (%)

I

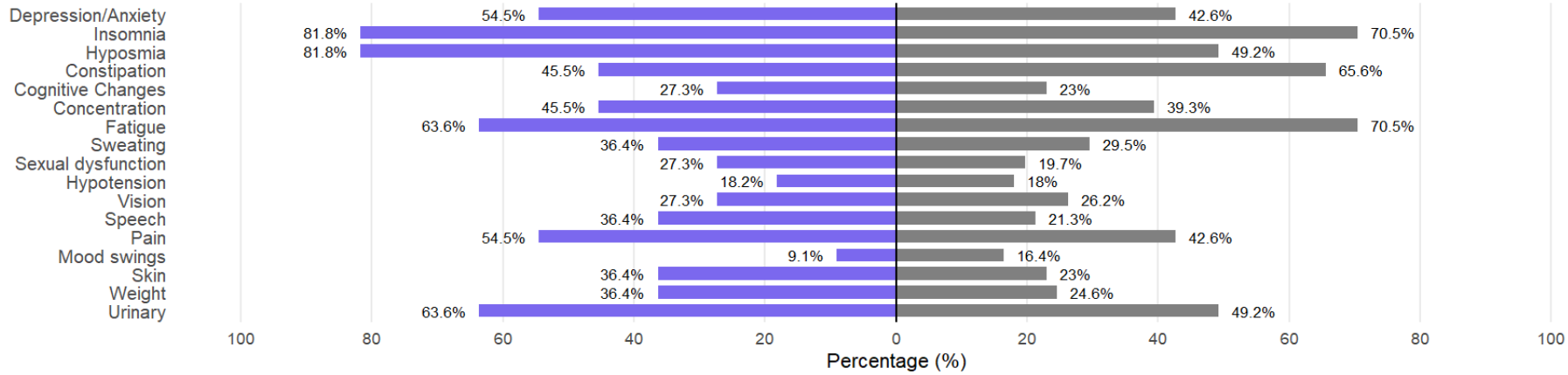


*Figure 6A-F: Symptom prevalence comparing HRT users (purple bars) and non-HRT users (grey bars) stratified by age groups. Motor symptoms are shown in panels A, D, and G; secondary motor symptoms in panels B, E and H; non-motor symptoms in panels C, F, and I. Age groups: 50-54 years (A-C), 55-59 years (D-F), and 60-64 years (G-I). Sample sizes: 50-54 years (n=12 HRT users, n=27 non-HRT users); 55-59 years (n=12 HRT users, n=61 non-HRT users); 60-64 years (n=11 HRT users, n=60 non-HRT users). Statistical significance determined using chi-squared tests, with Fisher's exact tests applied when expected cell counts <5. * p<0.05, ** p<0.01, *** p<0.00.*

Symptom Abbreviations:

Motor Symptoms: Tremors = Tremors or shaking, usually in the hands, arms, or legs; Rigidity = Stiffness or rigidity of the muscles; Slowness = Slowness of movement (bradykinesia); Imbalance = Difficulty with balance and coordination; Postural instability = Postural instability which can cause falls; Speech/Writing = Changes in speech and writing, such as slurred speech or small handwriting; Swallowing = Difficulty with swallowing.

Non-Motor Symptoms: Depression/Anxiety = Depression and anxiety; Insomnia = Sleep disturbances, including insomnia; Hyposmia = Loss of sense of smell; Constipation = Constipation; Cognitive Changes = Cognitive changes, including memory problems; Concentration = Difficulty concentrating; Fatigue = Fatigue and lack of energy; Sweating = Excessive sweating; Sexual dysfunction = Sexual dysfunction; Hypotension = Orthostatic hypotension; Vision = Visual disturbances, including blurred vision; Speech = Speech and swallowing problems; Pain = Pain and discomfort; Mood swings = Mood swings and emotional changes; Skin = Skin problems, including seborrheic dermatitis; Weight = Weight loss or gain; Urinary = Urinary Problems.

3.2.4. Contraceptive Usage Analysis in Menstruating Women

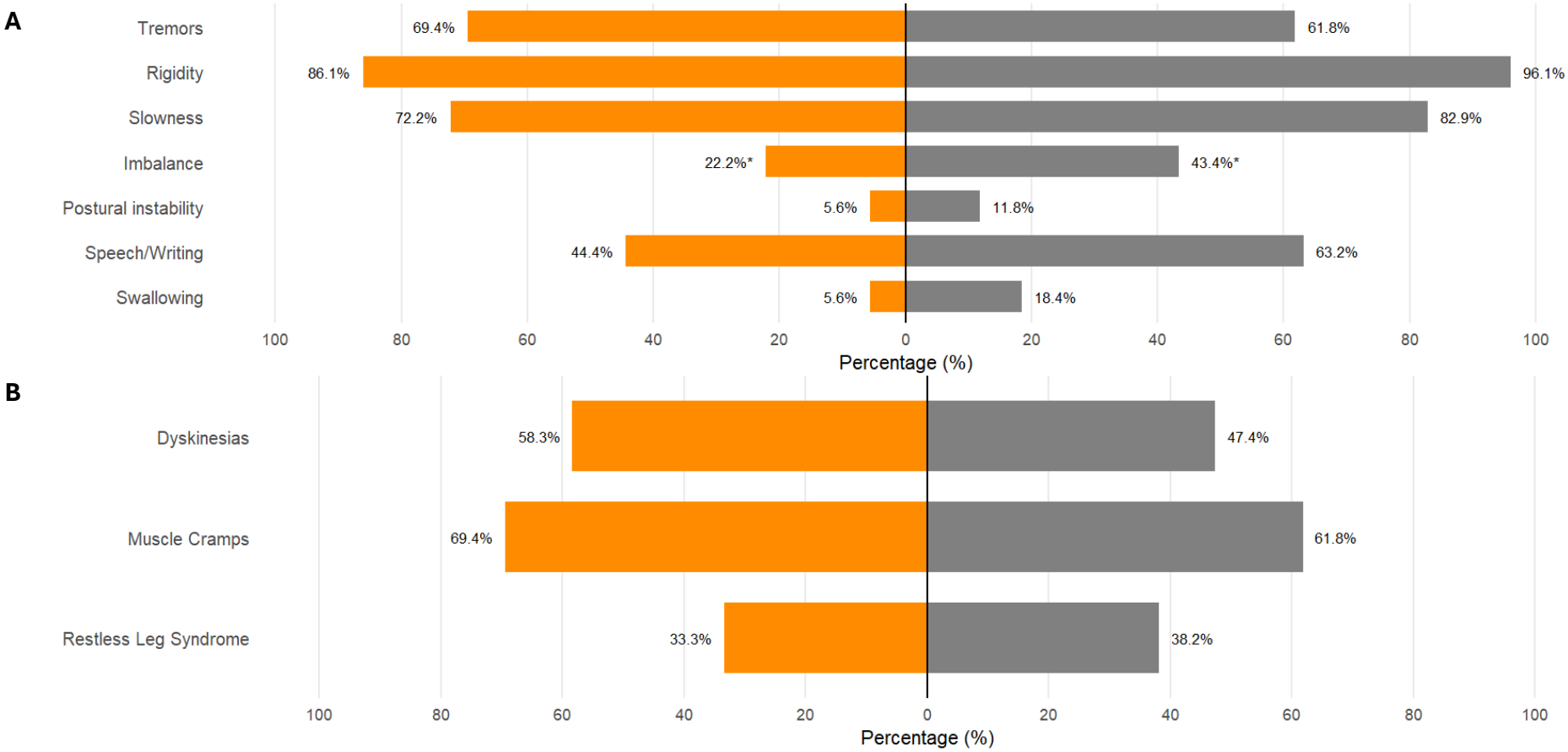
Analysis of contraceptive usage was conducted in menstruating women who provided symptom and medication data with complete contraceptive data (n=112 of 125 menstruating women, 89.6%). The analysis included 36 contraceptive users and 76 non-contraceptive users.

Demographic characteristics (age, race, employment status, marital status, PD symptom duration, and age at PD diagnosis) did not differ significantly between contraceptive users and non-contraceptive users (all $p > 0.05$).

Figure 7A shows prevalence of motor symptoms by contraceptive status in menstruating women. One motor symptom showed a statistically significant difference: Difficulty with balance and coordination; this symptom was reported by 22.2% of contraceptive users compared to 43.4% of non-contraceptive users ($p = 0.049$; OR=2.66, 95% CI: 1.02–7.67). The prevalence of the remaining motor symptoms showed no significant differences between groups.

Figure 7B shows prevalence of non-motor symptoms in contraceptive users and non-users. One non-motor symptom demonstrated a statistically significant difference: skin symptoms, which were reported by 8.6% of contraceptive users compared to 27.6% of non-contraceptive users ($p = 0.044$; OR=4.03, 95% CI: 1.08–22.74). The prevalence of all other non-motor symptoms showed no significant differences between contraceptive status groups.

Figure 7: Symptom Prevalence by Hormonal Contraceptive Status in Menstruating Women



C

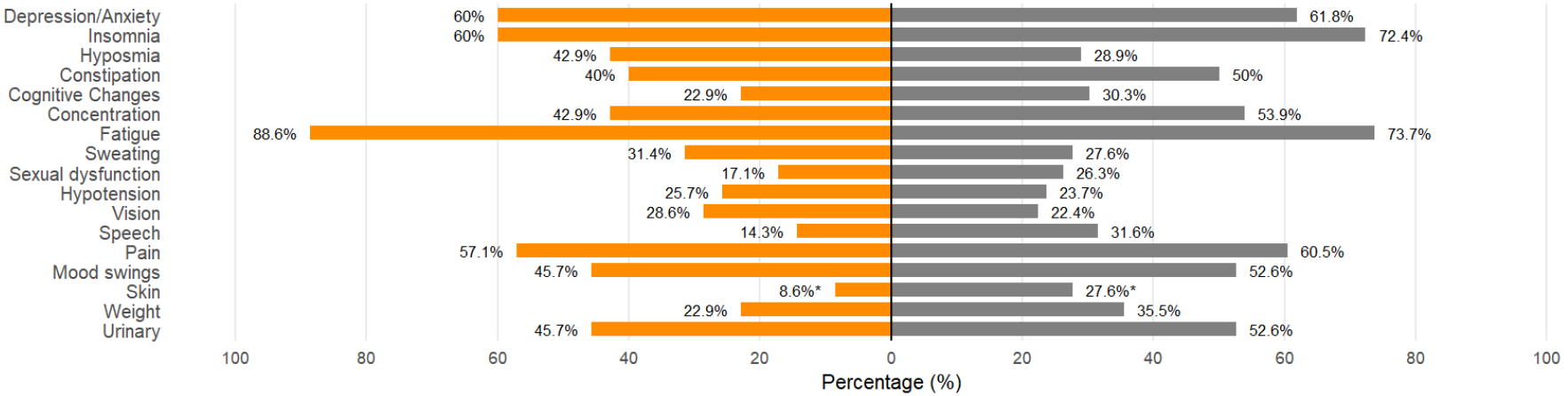


Figure 7: Bar charts showing *percentage of women reporting each of (A) Motor symptom prevalence (B) Secondary motor symptoms and (C) Non-motor symptom prevalence, in hormonal contraceptive users (orange bars, left) and non-users (grey bars, right). Hormonal contraceptive methods assessed included implants, intrauterine devices, injections, oral contraceptives, vaginal rings, and skin patches. Sample sizes: hormonal contraceptive users (n=36), non-users (n=76) Statistical significance determined using chi-squared tests, with Fisher's exact tests applied when expected cell counts <5. * p<0.05, ** p<0.01, *** p<0.00.*

Symptom Abbreviations:

Motor Symptoms: Tremors = Tremors or shaking, usually in the hands, arms, or legs; Rigidity = Stiffness or rigidity of the muscles; Slowness = Slowness of movement (bradykinesia); Imbalance = Difficulty with balance and coordination; Postural instability = Postural instability which can cause falls; Speech/Writing = Changes in speech and writing, such as slurred speech or small handwriting; Swallowing = Difficulty with swallowing.

Non-Motor Symptoms: Depression/Anxiety = Depression and anxiety; Insomnia = Sleep disturbances, including insomnia; Hyposmia = Loss of sense of smell; Constipation = Constipation; Cognitive Changes = Cognitive changes, including memory problems; Concentration = Difficulty concentrating; Fatigue = Fatigue and lack of energy; Sweating = Excessive sweating; Sexual dysfunction = Sexual dysfunction; Hypotension = Orthostatic hypotension; Vision = Visual disturbances, including blurred vision; Speech = Speech and swallowing problems; Pain = Pain and discomfort; Mood swings = Mood swings and emotional changes; Skin = Skin problems, including seborrheic dermatitis; Weight = Weight loss or gain; Urinary = Urinary Problems.

3.3 App Data

3.3.1 App Symptom and Medication Analysis

3.3.1.1. L-DOPA Dose Comparison Between Menstruating and Postmenopausal Women

The analysis included 65 women taking L-DOPA medications based on their scheduled medication entries in the app (22 menstruating, 43 postmenopausal). Women were excluded if they lacked menstruation status data from the initial survey or were classified as perimenopausal as outlined in Section 2.5.1.1.

The Mann-Whitney U test revealed no statistically significant difference in daily L-DOPA doses between menstruating and postmenopausal women (U = 423, p = 0.492). The effect size was 0.106, indicating a small effect.

The median daily L-DOPA dose was 866 mg (562-1814 mg) for menstruating women compared to 1050 mg (602-1948 mg) for postmenopausal women (Figure 8).

Figure 8: Daily L-DOPA Dose Comparison Between Menstruating and Postmenopausal Women

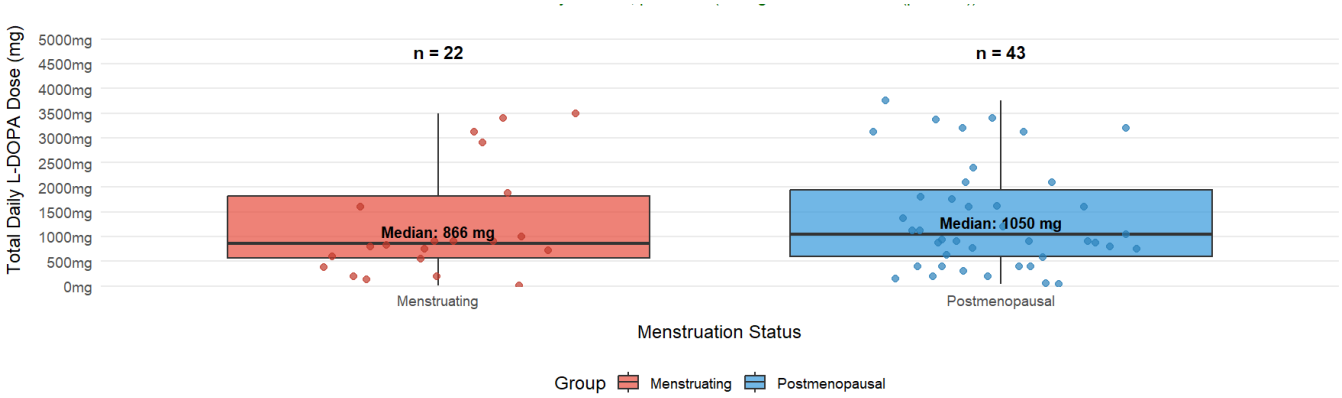


Figure 8: Menstruating n= 22 (red) and postmenopausal n= 43 (blue) women. Statistical comparisons used Mann-Whitney U test. Note: Two extreme outliers (>4495 mg) not shown for clarity but were included in statistical analysis.

L-DOPA PD Medication Options on MyMovesMatter App: Caramet CR, Co- Beneldopa, Co-Careldopa, Dopicar 25/250mg, Duodopa, Duopa, Inbrija, Lecado, Madopar, Modopar, Parcopa, Prolopa , Rytary, Sastravi, Sinemet, Stalevo.

3.3.1.2. App Symptom Reporting by Menstruation Status

App tracking data was analysed for differences between menstruation status and symptom logging patterns among 117 app participants: n=45 menstruating and n=72 postmenopausal participants. Eight significant differences were identified ($p<0.05$) (Table 8).

Motor symptoms showed one significant association. Dyskinesia was logged by 31.1%

(14/45) of menstruating women compared to 13.9% (10/72) of postmenopausal women (p=0.045; OR=2.77, 95% CI: 1.01–7.86).
Non-motor symptoms showed seven significant differences. Irritability was logged by 37.8% (17/45) of menstruating women versus 12.5% (9/72) of postmenopausal women (p=0.003; OR=4.19, 95% CI: 1.55–12.09). Crying spells was logged by 31.1% (14/45) of menstruating women compared to 11.1% (8/72) of postmenopausal women (p=0.014; OR=3.57, 95% CI: 1.25–10.95). Excessive daytime sleepiness (EDS) was logged by 31.1% (14/45) of menstruating women versus 12.5% (9/72) of postmenopausal women (p=0.026; OR=3.13, 95% CI: 1.12–9.18). Hot flushes were logged by 28.9% (13/45) of menstruating women compared to 8.3% (6/72) of postmenopausal women (p=0.007; OR=4.41, 95% CI: 1.41–15.52). Pelvic cramps were logged by 24.4% (11/45) of menstruating women versus 2.8% (2/72) of postmenopausal women (p=0.001; OR=11.09, 95% CI: 2.24–108.51). Mood swings were logged by 15.6% (7/45) of menstruating women compared to 4.2% (3/72) of postmenopausal women (p=0.043; OR=4.18, 95% CI: 0.89–26.51). Increased appetite was logged by 15.6% (7/45) of menstruating women versus 2.8% (2/72) of postmenopausal women (p=0.026; OR=6.34, 95% CI: 1.13–65.62).

Table 8: Motor and non-motor symptoms by menstruation status among app participants.

Symptom Type	Symptom	Menstruating (n=45)	Postmenopausal (n=72)	P-value
Motor	**Dyskinesia**	14	10	0.045*
Non-Motor	**Irritable**	17	9	0.003**
	Crying Spells	14	8	0.014*
	Exs Daytime Sleep	14	9	0.026*
	Hot Flushes	13	6	0.007**
	Pelvic Cramps	11	2	0.001***
	Mood Swings	7	3	0.043*
	↑Appetite	7	2	0.026*

*Table 8: Numbers represent women who logged each symptom at least once during their app usage period. Menstruation status determined from initial survey responses. Statistical significance determined using chi-squared tests, with Fisher's exact tests applied when expected cell counts <5. * p<0.05, ** p<0.01, *** p<0.001.*
Symptom Abbreviations:
Motor Symptoms: Dyskinesia: Jerky movements (Dyskinesia).
Non-Motor Symptoms: Crying spells: Crying spells; Exs Daytime sleepiness: Excessive

daytime sleepiness; Irritability: Feeling irritable; Hot flushes: Hot flushes; ↑Appetite: Increased appetite; Mood swings: Mood swings; Pelvic cramps: Pelvic cramps.

3.3.1.3. Motor vs Non-Motor Symptom Burden by Menstruation Tracking Status

Average daily symptom burden from app tracking data was compared between women who tracked periods (n=27) and women who did not track periods (n=37) (Figure 9). Results are presented as mean ± standard error of the mean (SEM). Motor symptom burden averaged 3.07 ± 0.326 for women who tracked periods compared to 2.70 ± 0.236 for women who did not track periods (p=0.363). Non-motor symptom burden averaged 3.61 ± 0.364 for women who tracked periods compared to 4.24 ± 0.532 for women who did not track periods (p=0.332). No statistically significant differences were observed between groups.

Figure 9: Motor and Non-Motor Symptom Burden by Menstruation Tracking Status

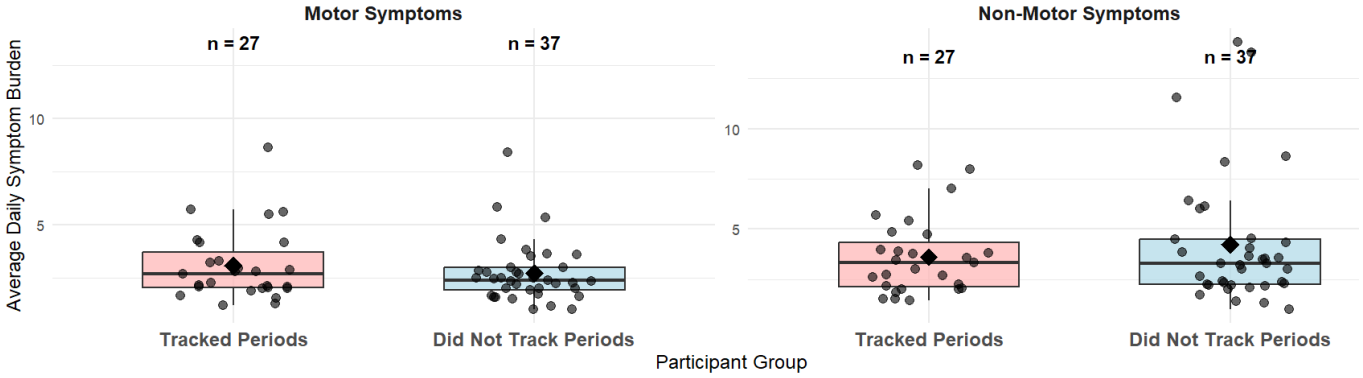


Figure 9: Average daily symptom burden in women who tracked periods (n=27, pink) and women who did not track periods (n=37, blue) for motor and non-motor symptoms. Daily symptom burden was calculated as the sum of all symptom severity ratings per day, where individual symptoms were rated on a 0-3 scale (0 = absent, 1 = mild, 2 = moderate, 3 = severe). Observed daily burden scores ranged from 1.22–8.63 (motor) and 1.44–8.17 (non-motor) in women who tracked periods, and 1.00–8.40 (motor) and 1.00–14.32 (non-motor) in women who did not track periods. Statistical comparisons using Mann-Whitney U test.

3.3.1.4. Motor vs Non-Motor Symptoms: Menstruating vs Non-Menstruating Days

A within-participant comparison of motor and non-motor symptom severity was conducted between menstruating and non-menstruating days using women who tracked periods (Figure 10). Motor symptom analysis included n=21 women with symptom data for both day types. Non-motor symptom analysis included n=22 women with symptom data for both day types. Motor symptom severity averaged 2.76 ± 0.35 on

menstruating days compared to 2.93 ± 0.39 on non-menstruating days (p=0.398). Non-motor symptom severity averaged 3.51 ± 0.35 on menstruating days compared to 3.31 ± 0.40 on non-menstruating days (p=0.549). No statistically significant differences were observed between menstruating and non-menstruating days for either motor or non-motor symptoms.

Figure 10: Motor vs Non-Motor Symptoms: Menstruating vs Non-Menstruating Days

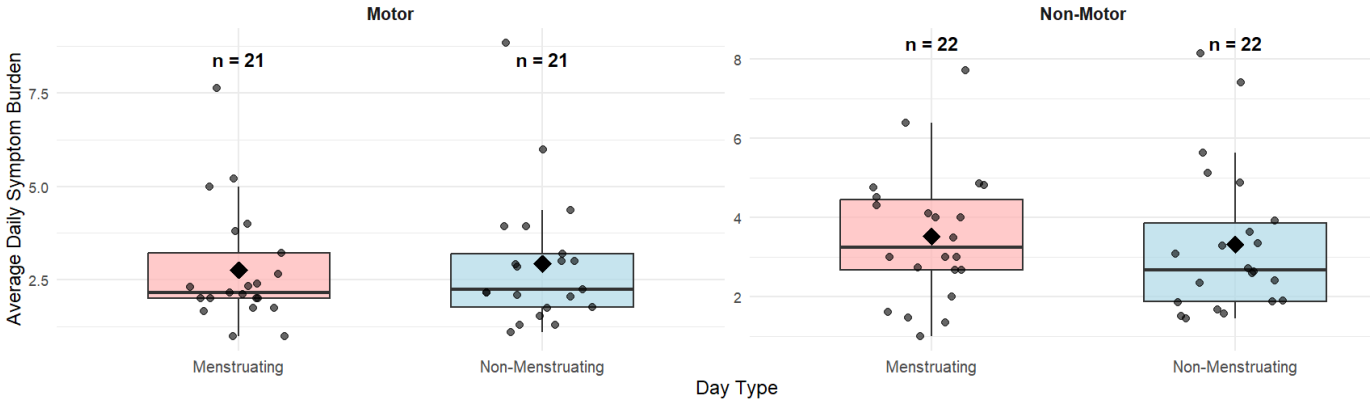


Figure 10: Paired comparison of average daily symptom severity, in women who tracked periods on the MyMovesMatter app, between menstruating and non-menstruating days for motor symptoms (n=21) and non-motor symptoms (n=22). Daily symptom burden was calculated as the sum of all symptom severity ratings per day, where individual symptoms were rated on a 0-3 scale (0 = absent, 1 = mild, 2 = moderate, 3 = severe). Observed daily burden scores ranged from 1.00–7.64 (motor) and 1.00–7.72 (non-motor) during menstruating days, and 1.10–8.85 (motor) and 1.43–8.14 (non-motor) during non-menstruating days. Statistical comparisons using Wilcoxon signed-rank test.

3.3.1.5. Post-Menstrual Symptom Burden Analysis

Post-menstrual symptom patterns were analysed in 8 women across 15 menstruation periods (Figure 11). Motor symptom burden averaged 3.52 ± 0.42 during menstruation and 4.18 ± 0.40 post-menstruation. Non-motor symptom burden averaged 4.12 ± 0.53 during menstruation and 3.51 ± 0.25 post-menstruation.

Spearman correlation analysis revealed no significant association between days since menstruation end and daily symptom burden (rho=0.013, p=0.878). Linear regression modelling showed no significant temporal trend in symptom burden across the analysis period (slope=0.106, R^2=0.005, p=0.413). Visual analysis demonstrated relatively stable symptom patterns across both menstrual and post-menstrual periods for motor and non-motor symptoms.

Figure 11: Motor and Non-Motor Symptom Burden During and After Menstruation

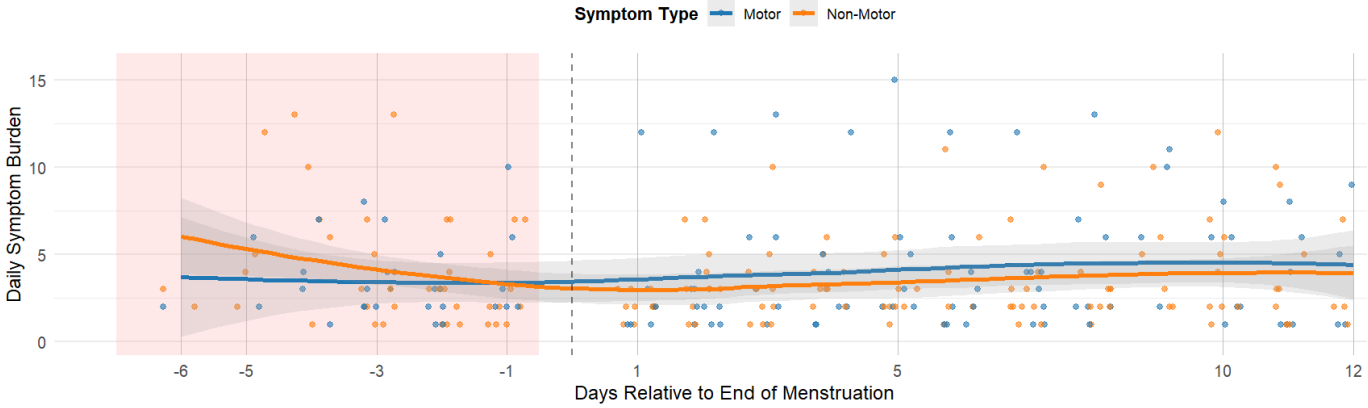


Figure 11: Motor (blue) and non-motor (orange) symptom burden across menstrual and post-menstrual periods in n=8 women. Pink background indicates menstruation period (days -6 to -1), dashed line marks end of menstruation. Daily symptom burden was calculated as the sum of all symptom severity ratings per day, where individual symptoms were rated on a 0-3 scale (0 = absent, 1 = mild, 2 = moderate, 3 = severe). Observed daily burden scores ranged from 0–19 for motor symptoms and 0–19 for non-motor symptoms. Statistical analysis using Spearman correlation and linear regression. Trend lines with 95% confidence intervals.

3.3.1.6. Menstrual phase symptom burden analysis

Motor symptom burden scores, as described in Section 2.5.2.12. ranged from 1.0 to 15.0, whilst non-motor scores ranged from 1.0 to 28.0. Analysis was restricted to 8 women who had data for all three menstrual phases and both symptom types (Figure 12).

For motor symptoms, median symptom burden scores were 2.25 ± 1.1 (IQR: 2.0-5.0) for Premenstrual, 2.25 ± 1.06 (IQR: 2.0-4.5) for Menstrual, and 2.0 ± 0.84 (IQR: 1.75-3.25) for Post-menstrual phases. For non-motor symptoms, median scores were 4.0 ± 1.13 (IQR: 2.0-4.88) for Premenstrual, 3.5 ±0.92 (IQR: 2.75-5.0) for Menstrual, and 2.0 ± 0.41 (IQR: 2.0-2.25) for Post-menstrual phases.

Paired Wilcoxon signed-rank tests revealed no significant differences between menstrual phases for either symptom type. For motor symptoms: Premenstrual vs Menstrual (p=0.773), Premenstrual vs Post-menstrual (p=0.099), and Menstrual vs Post-menstrual (p=0.410). For non-motor symptoms: Premenstrual vs Menstrual (p=1.000), Premenstrual vs Post-menstrual (p=0.114), and Menstrual vs Post-menstrual (p=0.058).

Figure 12: Symptom burden across menstrual cycle phases

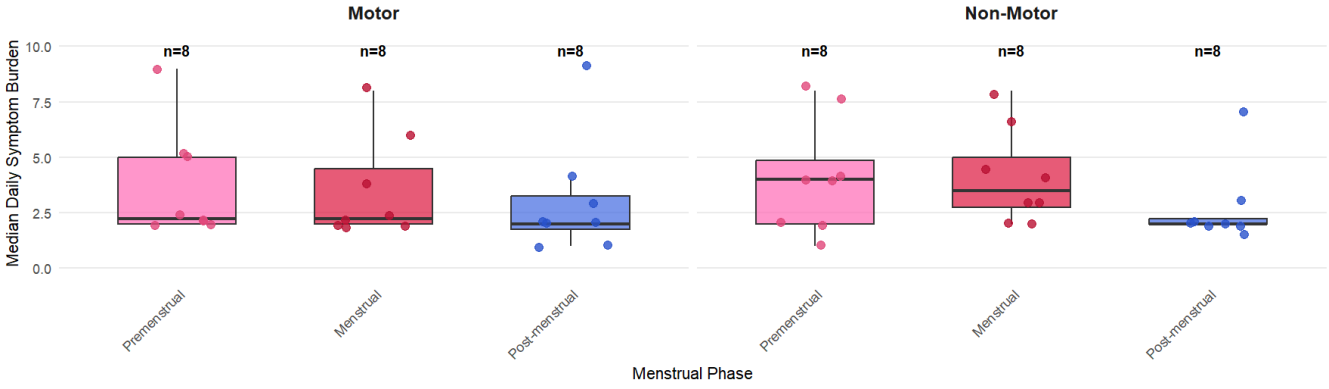


Figure 12: Average daily symptom burden for motor and non-motor symptoms across menstrual cycle phases. Each point represents one participant's median daily burden for that phase. Box plots show median, quartiles, and range. Premenstrual phase (pink), Menstrual phase (red), Post-menstrual phase (blue). Daily symptom burden was calculated as the sum of all symptom severity ratings per day, where individual symptoms were rated on a 0-3 scale (0 = absent, 1 = mild, 2 = moderate, 3 = severe). Observed daily burden scores ranged from 2.00–9.00 (motor) and 1.00–8.00 (non-motor) in the premenstrual phase, 2.00–8.00 (motor) and 2.00–8.00 (non-motor) in the menstrual phase, and 1.00–9.00 (motor) and 1.50–7.00 (non-motor) in the post-menstrual phase. Statistical comparisons using paired Wilcoxon signed-rank tests. No significant differences found between phases (all p > 0.05). n=8 women.

3.3.1.7. Dyskinesia severity analysis

Individual symptom analysis was conducted for dyskinesia across menstrual cycle phases. 7 out of 16 menstruating participants met the inclusion criteria outlined in Section 2.5.2.14. Linear mixed effects analysis revealed no significant difference in dyskinesia severity across phases (F=1.59, p=0.294) (Figure 13).

Figure 13: Dyskinesia severity across menstrual cycle phases

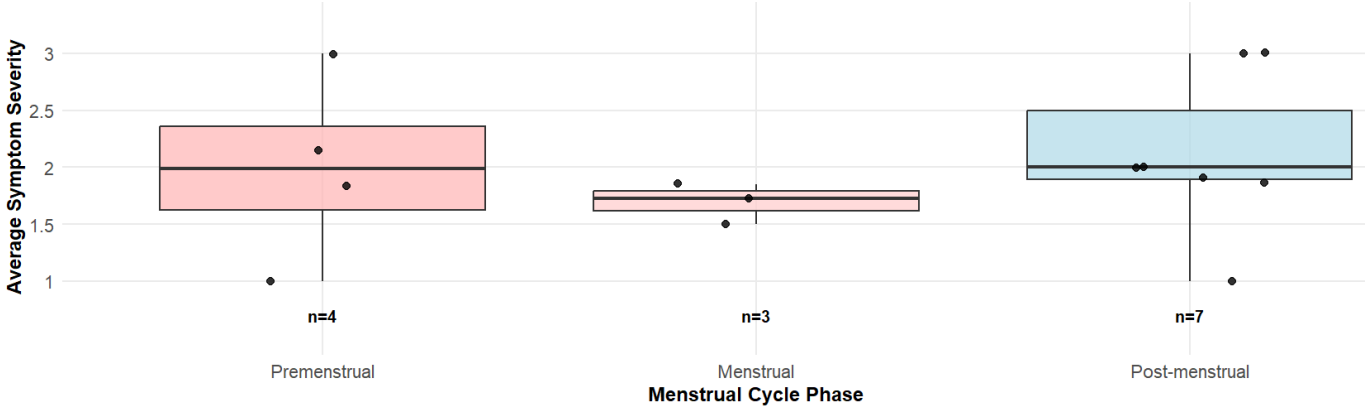


Figure 13: Average dyskinesia severity per participant per phase. Premenstrual phase (pink), menstrual phase (red), post-menstrual phase (blue). Black dots represent individual participants. Sample sizes indicated below each phase. Severity scale: 0-3. Statistical analysis employed linear mixed effects models with participant as a random intercept to assess overall phase effects (F = 1.59, p = 0.294). Post-hoc pairwise comparisons with Bonferroni correction were not performed due to non-significant F-test.

3.3.2. App Usage and Engagement

3.3.2.1. Symptom entry analysis across menstrual phases

Daily symptom reporting frequencies were calculated across menstrual phases n=15 women (Figure 14). The average number of symptoms logged per day was 2.10 ± 0.47 for menstrual phases, 1.91 ± 0.59 for premenstrual phases, and 1.36 ± 0.39 for post-menstrual phases. A Friedman test for repeated measures found no significant differences in daily symptom reporting frequency across phases (p = 0.127).

Figure 14: Average no. of symptoms per day by phase

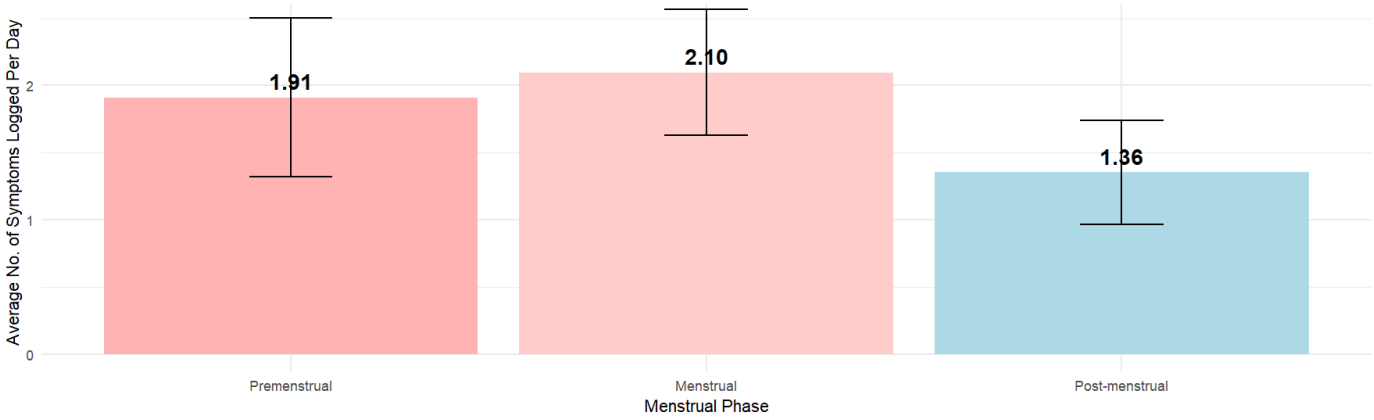


Figure 14: Daily symptom reporting frequency across menstrual cycle phases. Premenstrual phases shown in pink, menstrual phases in light pink, post-menstrual phases in blue. Error bars represent standard error. Values displayed above bars indicate mean symptoms per day. Statistical comparisons using Friedman test (χ^2 = 4.133, df = 2, p = 0.127). n=15 women. The Friedman test did not reach significance (p = 0.127), therefore post-hoc pairwise comparisons with Bonferroni correction were not conducted.

3.3.2.2. App Usability Survey Results

From 148 total app users, 105 participants met at least one eligibility criterion outlined in Section 2.5.2.17., whilst 43 participants had insufficient logging activity for follow-up inclusion. Of 38 app users who completed the follow-up survey, n=27 women fully completed all 18 usability questions, n=8 women provided partial responses, and 3 women provided no usability response.

Individual question analysis revealed "Easy To Learn" as the highest rated aspect (mean 5.63 ± 0.28), whilst "Improved Healthcare Access" received the lowest rating (mean 3.24 ± 0.24).

Thematic analysis grouped questions into five categories using the approach described in Section 2.5.3.4. Core Usability achieved the highest thematic mean (5.35 ±0.26), followed by User Experience & Satisfaction (5.06 ± 0.28), Future Intent (4.94 ± 0.32), Practical Integration (4.65 ± 0.26), and Health & Clinical Value (4.31 ± 0.23) (Figure 15).

Figure 15: App Usability Survey Thematic Summary Analysis

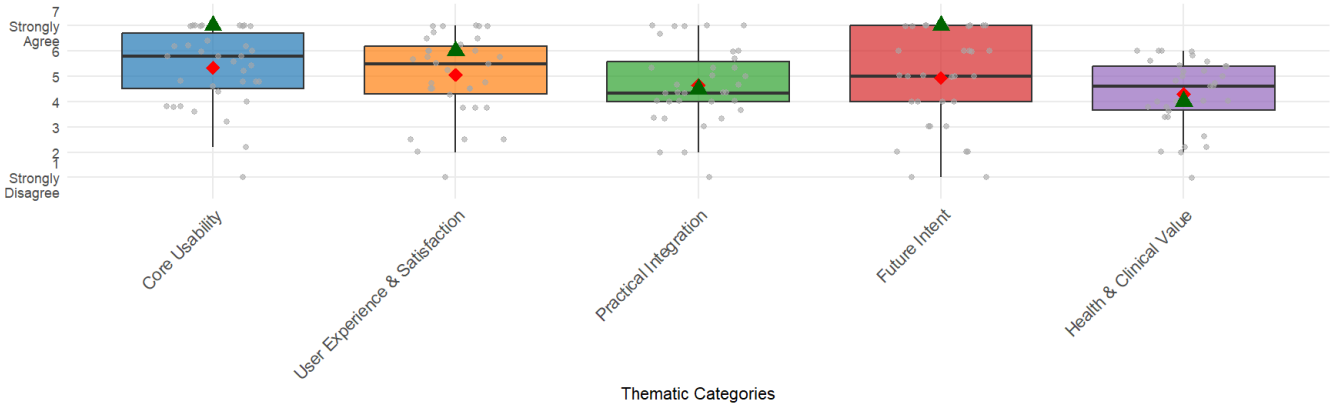


Figure 15: App usability survey responses grouped by thematic categories. Red diamonds = mean thematic score, green triangles = most common thematic score, grey dots = individual participant averages. Scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Somewhat Disagree, 4 = Neither Agree nor Disagree, 5 = Somewhat Agree, 6 = Agree, 7 = Strongly Agree. n=35 app users.

3.3.2.3. App Tracking Experience

Of the 38 app users who completed the follow-up survey, n=30 women responded to the question "How did you find your experience tracking on the app overall?" (78.9%). Of the 30 responders, n=16 women reported positive experiences (53.3%), n=11 women reported negative experiences (36.7%), and n=3 participants reported mixed experiences (10.0%).

Examples of positive responses included "Very useful" and "Mostly very positive (helped me track and understand my main symptoms in relation to, e.g., diet or HRT dose)".

Negative responses included "too much to do", "overwhelming", and "tedious".

Mixed responses included "Afraid I didn't manage to get into the habit of using it" and "ok - only track meds and rem disorder."

4. Discussion

4.1. Study Population

The distributions of the age of the study participants, and the age at which they first experienced PD symptoms, were as expected. Those participants who were menstruating were younger than those who were not menstruating, and since these has presented with young-onset PD (before the age of 50 years), by definition had an earlier disease onset. Employment patterns of the study participants also followed expected trends, with menstruating women predominantly in full-time employment, whilst most postmenopausal women were retired (Table 6). This finding is consistent with 2016 data from Ireland showing that 95.0% of women aged 65+ were not in employment (Nolan and Barrett, 2019) . It is important to consider the employment status of participants when interpreting the results, as employment status can affect participants' daily symptom experience, and furthermore, experiences of symptoms may interfere with work responsibilities. Employment status influences how PwP perceive and report their symptoms, Murphy et al. found that workplace-related stigma concerns significantly affected PwP's symptom experiences and help-seeking behaviours (Murphy et al., 2013). Given the substantial employment differences between menstruating and postmenopausal women in this study, work-related factors may contribute to the observed differences in symptom prevalence between these groups.

The marital status of WwPD is important for contextualising their symptoms and their effect on the quality of life of participants, considering the interpersonal, emotional and physical strains that PD places on relationships as Fleming et al. found that while sometimes relationships can strengthen following diagnosis, others break down due to increased dependence on partners, and sexual difficulties caused by physical symptoms (Fleming et al., 2004).

It is important to consider the predominantly white cohort composition in our study when interpreting the findings, as PD presentation can vary across racial populations (Di Luca et al., 2023; Van Den Eeden et al., 2003). In our study, the percentage of both groups that reported having a family history of PD was 23.97% and 19.4% for menstruating and postmenopausal women respectively. This is in line with previous studies, which reported that ~15% of people with PD have a family history of PD (Tambasco et al., 2016), thus giving confidence that our study cohort represents a normal population of people with PD. The slightly higher percentage of family history among menstruating women aligns with expectations, as familial-linked cases typically present with younger onset ages (Bonifati et al., 1995; Papapetropoulos et al., 2007).

4.2. Differences between menstruating and postmenopausal participants

An attempt was made to further subdivide menstruating versus post-menopausal women into age brackets to control for age, equivalent to the methods described for the analysis of HRT users vs non-users (Section 2.5.1.5.1.). However, the overlap in age brackets between the two groups was too small to allow robust statistical analysis.

4.2.1. Medications

The PD medication category usage patterns observed (Figure 5D) confirm that this population represents normal contemporary prescribing practices. L-DOPA usage was high in both groups, reflecting current guidelines which establish this drug as first-line treatment (Pringsheim et al., 2021). The similar PD medication patterns in both menstruating and postmenopausal women is noteworthy given that current treatment guidelines emphasise age-based prescribing considerations (Pringsheim et al., 2021). DA agonists are recommended for younger PwP who face an elevated risk of dyskinesia (Pringsheim et al., 2021), yet the menstruating cohort, despite being predominantly younger, showed nearly identical DA agonist usage patterns to the postmenopausal group (Figure 5D). The clinical implications of these prescribing patterns in relation to dyskinesias are discussed further below.

The absence of statistically significant differences in L-DOPA dosing between menstruating and postmenopausal women (Figure 8) suggests that current prescribing practices may not account for hormonal status when determining dosages. It is likely that the lower median dose in menstruating women reflects the menstruating cohort's earlier disease stage (Table 6). The finding that these women experience higher dyskinesia rates despite receiving lower doses warrants consideration (Table 8). Given the observed cyclical symptom fluctuations in menstruating WwPD (Table 1), this finding suggests that static dosing approaches may be inadequate for addressing the unique challenges posed by hormonal fluctuations in younger WwPD.

4.2.2. Symptoms

Rigidity

The higher prevalence of 'stiffness or rigidity of the muscles 'or rigidity, one of the cardinal symptoms of PD, in menstruating women in this cohort is an interesting and novel finding. While familial and genetic forms of PD typically present at younger ages, comparative research shows no significant differences in rigidity rates between familial and sporadic PwP at disease onset (Papapetropoulos et al., 2007; Vibha et al., 2010). Therefore, even if some women in the menstruating cohort had familial forms of PD, established literature suggests that this would not account for the observed higher

rigidity prevalence, indicating that other mechanisms, such as hormonal factors, may contribute to this novel finding.

Postural Instability

The higher prevalence of postural instability, which can cause falls, in postmenopausal women compared to menstruating women is consistent with expected age-related disease progression, as postmenopausal women in this cohort were older and therefore at a more advanced stage of PD. This reflects the literature that postural instability onset is primarily determined by age of onset, which was older in the postmenopausal cohort, rather than by disease duration (Table 6) (Becker et al., 2022). It should be noted that, as mentioned in the introduction, higher fall risk has been documented in WwPD compared to men (Baba et al., 2005; Parashos et al., 2018). Additionally, literature shows that postural instability presents therapeutic challenges, with L-DOPA sometimes being unhelpful for this symptom (Curtze et al., 2015). However, the potential for hormonal influences cannot be entirely dismissed, as a 1994 case study documented deterioration in postural reflexes the week before menstruation in one woman with PD (Session et al., 1994) (Table 1). Additionally, Castrioto et al., found that 55% of WwPD had worsening of 'balance and clumsiness' in the week before menses compared to the week after (Castrioto et al., 2010) (Table 1). However, it is difficult to attribute this different in prevalence to hormonal status and the observed difference between menstruating and postmenopausal groups more likely reflects the established age-dependent emergence of postural instability.

Hyposmia

The higher prevalence of hyposmia and constipation in postmenopausal women aligns with the established progression of PD, where these prodromal NMS symptoms typically emerge years before motor symptom onset and increase with disease progression (Fullard et al., 2017; Ross et al., 2008; Savica et al., 2009).

The prevalence of loss of sense of smell or hyposmia amongst postmenopausal women could be clinically significant as, beyond its role as a prodromal marker, research demonstrates that olfactory impairment is associated with reduced QoL in PwP (Lin et al., 2022). Furthermore, hyposmia is linked to not only greater severity of motor and cognitive deterioration and increased reliance on PD medication, but also accelerated rate of progression of these outcomes (Fullard et al., 2016; He et al., 2020). Hyposmia has also been associated with unintentional weight loss in PwP (Sharma and Turton, 2012).

Treatment of hyposmia in PD remains challenging, with limited and conflicting evidence regarding potential pharmacological interventions. Although Haehner et al. reported better odour discrimination abilities in early-stage PwP receiving rasagiline (Haehner et al., 2015), a 2022 systematic review found that MAO-B inhibitors do not offer reliable or clinically significant benefits for olfactory dysfunction (Tsuboi et al., 2022).

Constipation

Notably, constipation was prevalent across both groups, affecting 57.8% of postmenopausal women and 46.7% of menstruating women, which aligns with established literature demonstrating that WwPD experience substantial constipation (Martinez-Martin et al., 2012; Picillo et al., 2021), with longitudinal evidence showing consistently higher constipation severity at disease onset and throughout early disease progression in WwPD than men (Kurlawala et al., 2021). The prevalence of constipation in the menstruating cohort is noteworthy, as early constipation in PD has been associated with greater severity of motor and cognitive symptoms, potentially indicative of a more severe disease phenotype (De Pablo-Fernandez et al., 2017; Jones et al., 2020). This is further supported by Picillo et al. (2021), who confirmed that prodromal constipation is sex-specifically associated with cognitive dysfunction and apathy in WwPD (Picillo et al., 2021).

A 2025 study suggests that constipation in PwP may be indicative of a peripheral "body-first" PD subtype rather than disease stage (Camacho et al., 2025). Camacho et al. found that after confounder adjustment, constipation remained stable over four years and was most strongly predicted by prodromal Rapid Eye Movement (REM) Sleep Behaviour Disorder history, a marker of this subtype. The higher prevalence of constipation in postmenopausal women may therefore reflect a greater proportion of this subtype within this group, rather than more advanced disease.

The lower prevalence of constipation in the menstruating cohort may be partly attributable to menstrual cycle-related gastrointestinal effects independent of PD pathology. A rise in prostaglandins, which reduce bowel absorption, and concurrent decline in progesterone at the onset of menstruation have been associated with increased rates of diarrhoea in healthy menstruating women, potentially reducing prevalence of constipation in the menstruating cohort (Bernstein et al., 2014; Dajani et al., 2003; Judkins et al., 2020).

Regardless of its aetiology, the high constipation rates observed may have important therapeutic implications, as delayed gastric emptying impairs L-DOPA absorption and bioavailability (Camacho et al., 2025; Müller et al., 2006; Pfeiffer et al., 2020). Given that women already demonstrate altered gastrointestinal transit times (Datz et al., 1987; Jung et al., 2009) and different pharmacokinetic responses to L-DOPA (Martinelli et al., 2003) compared to men, the elevated constipation rates in postmenopausal women may necessitate consideration, regarding PD treatment tailoring for this population. Furthermore, similarly to hyposmia, prodromal constipation in WwPD has been linked to cognitive issues and increased reliance on L-DOPA.(Picillo et al., 2021). This evidence collectively suggests that postmenopausal WwPD with constipation represent a clinically complex subgroup warranting targeted PD medication regimens and lifestyle intervention (Camacho et al., 2025).

PMS-related

Menstruating women consistently demonstrated higher prevalence of mood-related and physiological symptoms across both survey and app data, with all six examined symptoms (mood swings, irritability, crying spells, pelvic cramps, increased appetite, and pain) showing the same directional pattern of increased prevalence compared to postmenopausal women (Table 7-8). The higher prevalence of these established PMS-related symptoms in menstruating women validates the reliability of the PD-specific motor findings.

The elevated pain prevalence observed in the menstruating cohort may be linked to their higher rates of depression and anxiety, as PwP experiencing pain have been shown to exhibit significantly higher depression scores compared to those without pain (Valkovic et al., 2015). For these menstruating women these mood-related symptoms occur in the context of complex hormonal fluctuations during the menstrual cycle (Figure 3). As mentioned previously, it is difficult to differentiate between PMS symptoms and PD-specific NMS symptoms that are made worse by PD and hormonal interference (Kompoliti et al., 2000; Valkovic et al., 2015). Martignoni et al. found that WwPD retrospectively reported significantly higher rates of PMS during their reproductive years compared to healthy controls (Martignoni et al., 2003).

The smaller app cohort sample size compared to the larger survey cohort should be noted. Comparisons with healthy populations would be required to determine whether these were simply PMS-related symptom prevalence rates, whilst increased app engagement would have facilitated symptom timing relative to menstrual cycles.

Hot Flushes

The significantly higher prevalence of hot flushes in menstruating WwPD ($p=0.007$) is a novel finding (Table 8). However, it may reflect a transition to perimenopause, given that the menstruating cohort was predominantly aged 40-49 years (65.85%) (Table 6). Given that postmenopausal women commonly experience hot flushes for years following menopause, the higher rate observed in menstruating WwPD is particularly noteworthy (Freeman et al., 2014). Previous retrospective questionnaire-based research found that WwPD reported more hot flushes than healthy controls (Martignoni et al., 2003).

PD has been shown to interfere with thermoregulation through dopaminergic deficits in hypothalamic temperature control centres, and estrogen generally promotes lower body temperatures and progesterone promotes higher temperatures (Reviewed by Charkoudian and Stachenfeld, 2016; LeDoux, 2005). Therefore, this combination of PD-related thermoregulatory impairment and cyclical nature of sex hormones (Figure 3) in menstruating WwPD may exacerbate an already struggling thermoregulation processes in PwP and manifest as hot flushes, as found in our study (Table 8).

The low engagement with the MMM app unfortunately limited investigation of symptom timing relative to menstrual cycles, which would have provided valuable insights into this hypothesis.

Excessive daytime sleepiness

The higher prevalence of EDS in the menstruating cohort is noteworthy (Figure 5C) given that 42.6% of menstruating women work full-time and may need to drive to work, and EDS has been identified as a critical factor for traffic safety and accident risk in PwP (Meindorfner et al., 2005). EDS would not have been expected to be higher in the menstruating cohort, who were both younger and had been experiencing symptoms for fewer years, as EDS has been associated with PD progression (Table 6) (Gjerstad et al., 2006; Tholfsen et al., 2015). Sleep disturbances and fatigue were highly prevalent in both menstruating and postmenopausal women (Figure 5C), suggesting there may be nighttime sleep problems in this study cohort. However, if sleep disturbances and fatigue are problematic in both groups, it is not clear why EDS is significantly higher only in menstruating women. Menstruating women may experience iron deficiency due to monthly blood loss, which can contribute to RLS and therefore poor sleep quality, potentially leading to EDS (Allen et al., 2013; Fernandez-Jimenez et al., 2020; Harvey et al., 2005; Liu et al., 2022; Mao et al., 2018). However, there was no statistically significant difference in sleep disturbance reporting between groups (Figure 5C). The higher prevalence of dyskinesia in menstruating women may be related to sleep problems, as poor sleep quality has been associated with motor complications in PD (Mao et al., 2018).

Dyskinesia

The app tracking data revealed that menstruating women were more likely to report dyskinesia. This findings aligns with previous literature, which reported that people diagnosed with YOPD (diagnosed before age 50), have in increased risk of dyskinesia development (Kostic et al., 1991; Kumar et al., 2005). Additional risk factors for dyskinesia include low body weight and female sex (Hassin-Baer et al., 2011; Hely et al., 1995). In our study, whilst all participants were biologically female, anthropometric questions were not asked in the initial survey, precluding analysis of weight-related dyskinesia risk in this study cohort.

DA agonist usage was very similar between menstruating and postmenopausal participants, indicating that observed dyskinesia differences cannot be attributed to differential prescribing patterns of these medications, which have been shown to delay dyskinesia onset due their longer half-life and continuous DA stimulation compared to L-DOPA 's pulsatile effects (Pringsheim et al., 2021).

As discussed in the introduction, estrogen modulates DA and fluctuates cyclically in menstruating women (Figure 3), creating periods of varying DA availability that could influence dyskinesia expression differently than in the stable low-estrogen state of

postmenopausal women.
An interconnection between dyskinesia and sleep disturbances has been previously reported established, with poor nighttime sleep quality found to significantly increase dyskinesia risk in PwP (Mao et al., 2018; Tang et al., 2022). This suggests that the elevated EDS observed in menstruating women may be linked to their motor complications, creating a bidirectional relationship where sleep disturbances could both contribute to and result from dyskinesia.
Safinamide has shown some promise in managing dyskinesia with one clinical trial demonstrating increased "on" time without dyskinesia when used as adjunctive therapy to L-DOPA (Schapira et al., 2017).
This finding prompted examination of whether menstrual phases were influencing the onset of dyskinesia in those women who are menstruating (Figure 13); the findings are covered later in this discussion.

Depression/Anxiety

'Depression and anxiety' were reported by significantly more menstruating women (Figure 5C). Depression has been identified as a risk factor for developing PD and depression and anxiety are prodromal symptoms of the disease. (Leentjens et al., 2003; Shiba et al., 2000). The clinical significance of these findings is underscored by evidence that depression and anxiety substantially reduce QoL in PwP, regardless of their motor symptom burden, highlighting the importance of addressing these mental health symptoms (Storch et al., 2013). Furthermore, the higher depression and anxiety prevalence in menstruating women indicates worse PD severity in this cohort, as depression is associated with increased motor impairment and poorer disease prognosis in PwP (Bega et al., 2015; Post et al., 2011; Rao et al., 2023b).
This finding contrasts with previous literature suggesting that depression and anxiety symptoms become more severe with increasing age in WwPD, and that women are at increased risk during perimenopause and menopause (Bromberger et al., 2011; Maas et al., 2024). The survey questionnaire presented 'depression and anxiety' as a combined measure. As stated in the limitations, it would have been useful to separate these symptoms; however, the validity of this finding is not undermined, as both conditions frequently co-occur in PD (Chang et al., 2020). The survey did not ask about antidepressant or anti-anxiety medication usage.
Given that the majority of menstruating women in this cohort were employed (63.1%) and married (65.0%) (Table 6), depression and anxiety symptoms may significantly impact their work performance, daily functioning and concerns about losing employment may add to psychological burden (Fleming et al., 2004; Mullin et al., 2018; Schrag and Banks, 2006). Marital status may particularly contribute to the prevalence of these NMS, as WwPD experience changing relationships with partners, altered self-perception, and relationship stress that can lead to marital breakdown (Fleming et al., 2004). It is of value to note that lower levels of BDNF have been identified as a

biomarker of depression in WwPD, which may aid in providing these women with early support in addressing the prevalence of anxiety and depression in this group (Huang et al., 2021).
A 2019 review outlined how treatment of depression and anxiety in PD remains problematic due to variable drug responses and psychiatric side effects (Reviewed by Carrarini et al., 2019). Potential interventions for menstruating women include enhanced social support networks, structured exercise programmes, and consideration of MAOB inhibitors to address their elevated depression and anxiety rates (Abraham et al., 2019; Dauwan et al., 2021; Tsuboi et al., 2022).

4.3. Hormonal Medication

4.3.1. HRT

The potential difference in symptoms reported by participants was explored by comparing HRT users to non-HRT users across the entire population (Figure S3). Subsequently, age-stratified analysis was conducted to control for confounding age effects, given the established relationship between PD severity and disease progression with advancing age (Figure 6).
In the 50-54 years age group, tremors were less prevalent in HRT users compared to non-HRT users. This findings align with evidence that higher residual estrogen levels correlate with reduced dopaminergic medication requirements and milder motor presentations (Bovenzi et al., 2024b). Additionally, Saunders-Pullman et al. found that HRT users not yet requiring L-DOPA exhibited lower UPDRS scores, though most participants in this study population are already established on L-DOPA therapy (Figure 5D) (Saunders-Pullman et al., 1999).
The relationship between estrogen deficiency and symptom exacerbation is further supported by observations of a WwPD reporting increased tremor severity and reduced L-DOPA efficacy during menstruation, when estrogen levels reach their nadir (Table 1) (Quinn and Marsden, 1986).
It should be noted that there is no information available on how long the HRT users in this study had been taking HRT, nor on the type of HRT formulation that they use and these observational findings require cautious interpretation given the potential for unmeasured confounders.
Nevertheless, this finding raises the question of whether a clinical trial examining HRT use in postmenopausal women in this age range should be conducted, and whether HRT could be considered as an option for treating tremor-dominant PD.

4.3.2. Contraceptives

Imbalance

Analysis of motor symptom prevalence between contraceptive users and non-users revealed that difficulty with balance and coordination was significantly less reported by

contraceptive users compared to non-contraceptive users (Figure 7A). This finding contrasts with research in healthy women, where contraceptive users demonstrated reduced postural stability compared to those not taking contraceptives (Mokosáková et al., 2014).
The relationship between estrogen fluctuations and balance impairment in WwPD is further supported by case studies documenting worsening postural reflexes premenstrually (Table 1). Session et al. reported that one woman experienced worsening of postural reflexes premenstrually (Session et al., 1994). Castrioto et al. showed worsening of balance and clumsiness issues reported by more than half of WwPD during the week before and during their periods (Castrioto et al., 2010).
Stable hormone levels from contraceptive use may maintain consistent PD medication effectiveness, as a study reported that two sisters with PD experienced reduced OFF periods with continuous oral contraceptive therapy (Sprenger et al., 2014).
Although the remaining motor symptoms examined in our study showed no significant differences between groups, this imbalance finding suggests that stable estrogen levels may improve motor control in PD.

Skin

Analysis of NMS prevalence between contraceptive users and non-users revealed that skin problems, including dry skin and rashes, was significantly less reported by contraceptive users compared to non-contraceptive users (Figure 7B).
Skin problems are more common in men, and increase with age in PwP (Tomic et al., 2022; Tucker and Masood, 2025). Other skin conditions that have been reported to be more prevalent in PwP include malignant melanoma and bullous pemphigoid (Chen et al., 2011; Huang et al., 2015). Additionally, iatrogenic problems from PD medications or treatments may be the cause of skin problems in PwP (reviewed by Niemann et al., 2021). Several specific dermatological conditions are overrepresented in PwP compared to the general population. Seborrhoeic dermatitis (SD) is one of the most common skin problems affecting PwP and is related to motor symptom severity (Fischer et al., 2001; Wilkowski et al., 2025).
Hyperhidrosis is another commonly reported NMS in PwP, though there was no correlation between 'excessive sweating' and contraceptive use in this cohort (Figure 7B) (Schestatsky et al., 2006).
There is no information on whether the contraceptive users in our study are taking estrogen-only, combined or progesterone contraceptives. However, it has been established that estrogens reduce sebum production (Pye et al., 1977), which can ameliorate SD (Huber et al., 2000; Katz et al., 2000). Therefore, the reduction in sebum production mediated by estrogen exposure may explain the decreased prevalence of skin problems observed in the contraceptive user cohort, accounting for the significant difference between users and non-users in our study. Consequently, contraceptives

could be considered as a therapeutic option for menstruating WwPD who suffer from SD related to their condition.

4.4. Menstrual Cycle Tracking Findings

Menstruating vs non-menstruating

Given the low app engagement rates and to include perimenopausal women who had not been discussed previously, symptom burden was compared between women who tracked periods and those who did not track periods (Figure 9). The results showed no significant difference in average daily burden of motor symptoms or NMS between the two groups, suggesting that period tracking behaviour, as a proxy for menstrual status, is not associated with symptom burden. Both groups demonstrated higher NMS burden compared to motor symptoms, which is clinically relevant given that NMS are one of the biggest factors that predict QoL in PwP (Prakash et al., 2016; Santos García et al., 2019).

A within-participant comparison assessed symptom burden on menstruating versus non-menstruating days among women who tracked periods (Figure 10). Motor symptom burden showed little difference between menstruating and non-menstruating days however, NMS burden approached significance with higher burden occurring on menstruating days (Figure 10). As discussed with PMS-related symptom prevalence in menstruating versus postmenopausal women, this result appears to be due to non-PD-specific symptoms and PMS-related symptoms. Given the prevalence of depression and anxiety in the menstruating cohort (Figure 5C) and the known impact of PD NMS on QoL, this trend and higher burden may warrant clinical intervention in these menstruating WwPD.

Menstrual Cycle Findings

Motor and non-motor symptoms across menstrual cycle

Given the limited literature on WwPD and menstrual cycle effects on PD symptoms, our study examined daily symptom burden across the menstrual cycle (Figure 11). No statistically significant correlation emerged between cycle days and symptom burden; however, consistent with the lack of difference between menstruating and non-menstruating days. NMS showed a spike at menstruation onset followed by a decline during and after menstruation. Motor symptoms showed minimal variation across the cycle, contrasting with previous literature that reported perimenstrual worsening of motor symptoms. It should be noted that the analysis was limited to 8 women who provided sufficient data for assessment of trends in their cycles.

These 8 women underwent further analysis divided into 3 phases: premenstrual, menstrual, and postmenstrual (Figure 12). No significant differences were found between the phases; however, NMS showed a pattern of premenstrual peaking,

remaining elevated during menstruation, then declining postmenstrually. Motor symptoms showed their nadir postmenstrually. The estrogen hypothesis presented in the introduction (Benedetti et al., 2001; Bovenzi et al., 2024a, 2024b; Cereda et al., 2013; Dluzen et al., 1996a, 1996b; Gatto et al., 2014; Gomez-Mancilla and Bédard, 1992; Hirohata et al., 2009; Murray et al., 2003; Yoo et al., 2020) would predict a more pronounced version of this pattern.

App Usage Timing

No statistically significant difference emerged in symptom logging frequency across menstrual phases, although logging increased around menstruation periods (Figure 14). This suggests two possibilities: menstruation serves as a biological reminder to log symptoms, or women genuinely experience more symptoms during menstruation.

Dyskinesias across the menstrual cycle

Dyskinesia was the only motor symptom that differed significantly between menstruating and postmenopausal groups in the app tracking data, prompting investigation of menstrual phase effects on dyskinesia occurrence and severity (Figure 13). Of the 16 participants with analysable menstrual cycle data, n=7 reported dyskinesia across their cycle. This prevalence is consistent with rates reported in the literature for PwP with similar disease duration (Holloway et al., 2004; Quinn et al., 1987).

Analysis of daily symptom reporting frequencies across menstrual cycle phases controlled for potential reporting differences between cycle phases and found no statistically significant variation in symptom reporting across phases. The higher dyskinesia reporting in the postmenstrual phase therefore represents genuine occurrence rather than reporting bias. This pattern is especially significant considering that menstruation phase showed the highest overall symptom reporting (Figure 14). Dyskinesia severity did not differ significantly across phases; therefore, menstrual phases appear to influence occurrence likelihood rather than actual severity in menstruating WwPD. A review by Bentall and Parr-Brownlie hypothesised that higher estrogen states allow for greater DA control and therefore reduced LID risk. This studies results contradict this, as the postmenstrual phase showed higher dyskinesia reporting despite presumably elevated estrogen levels (Figure 3) suggesting the increase in estrogen might cause fluctuations in DA availability that increase LIDs.

The limited statistical power of the study, and low tracking levels by participants may have prevented detection of menstrual cycle trends in the app data. The inclusion of medication timings in relation to the onset of dyskinesia would have been an ideal extension of this investigation. However, while the app had functionality to allow this data collection, engagement was too low to conduct this analysis. Due to these limitations, the findings should be interpreted with caution and no definitive conclusion about menstrual cycle influence on PD symptoms can be made. A more sophisticated

MMM app incorporating engaging design features and time-sensitive notifications could enhance data collection for similar future menstrual cycle investigations.

4.5. App Usability and Tracking Experience

4.5.1. MMM App Survey Analysis

The MMM app usability survey responses explained user experience and provided context for the low participant engagement.

Core usability averaged 5.35 on the 7-point scale, indicating that most women somewhat agreed with the usability statements, as defined in the methods (Table 5). Therefore, the MMM app's ease of use, functions, and navigation were adequate but could be improved. Future versions of the MMM app for commercial, research, or personal use by WwPD should address these aspects to enhance engagement.

User experience and satisfaction scored 5.06, demonstrating positive feedback on organisation, interface, and overall experience, with room for improvement.

Practical integration scored 4.65, indicating some positive feedback regarding comfort of use, social settings, scheduling, and internet connectivity, but requiring significant improvement.

Most women reported that they would use the MMM app again. Given the growing interest in this area of PD, this is good feedback, as the app could be used in another study, and some participants could be recruited again with a more sophisticated app design that addresses the areas identified for improvement in this study.

The health and clinical value category received the lowest average score; therefore, users found the MMM app's daily healthcare impact relatively neutral (Table 5). Given that the MMM app is intended as an adjunct for self-management of PD, practical improvements should prioritise interface function and ease of use to boost overall health and clinical value.

The feedback offers important lessons for app design and improvements, with users providing generally positive responses overall. This feedback is especially notable considering the low engagement of the menstruating participants who are a younger cohort (Table 6) and therefore, although their explicit digital literacy levels were not asked, it can be assumed they didn’t have the barrier to entry of lack of digital literacy that some of the older WwPD in the postmenopausal cohort may have had (Heart and Kalderon, 2013). Less than 1 in 7 users access apps more than a day after downloading them, making user retention a known problem for all mobile apps (Chen, 2025).

When comparing the MMM app to existing applications for symptom and period tracking, the Bearable app permits tracking of chronic illness symptoms alongside menstrual cycle monitoring; it is not, however, specific to PD, though it is notably user-friendly and features a well-designed interface (Bearable, 2026). The StrivePD app offers more sophisticated symptom tracking and medication logging for PD, and

provides insights derived from passive data that influence PD symptoms by integrating with a user's Apple Watch, but does not have a period tracking option ("StrivePD," 2026). The APDA Symptom Tracker app is similarly specific to PD and enables symptom and medication logging, yet is less sophisticated than StrivePD and likewise lacks period tracking capabilities ("APDA Symptom Tracker App," 2026). The MMM app is, therefore, the only currently available app that integrates symptom, medication, and period tracking within a single platform.

4.5.1.1. Future Directions for the MMM App

Future versions of the MMM app for commercial or research use by WwPD should address the retention challenges evident in this study, where only 48 menstruating (38.4%) and 83 postmenopausal (29.3%) women engaged with the app from their respective cohorts (Figure 4).

As noted in the introduction, the MMM app was freely available, a factor that may have contributed to low user engagement ("My Moves Matter," 2026a). In a feasibility study of a similarly free lifestyle app, investigators observed a substantial decline in participation, attributing this partly to limited user investment. The authors suggested that prompt, personalised feedback and gamification strategies may improve retention (Reviewed by Amagai et al., 2022; McConnell et al., 2017).

Interactive symptom visualisation, medication reminders incorporating positive reinforcement for adherence, and real-time correlation analysis between motor symptoms and tracked variables (mood, movement, menstrual cycle phase) could be potential updates to enhance the application's perceived health and clinical value (Lakshminarayana et al., 2017; Lee et al., 2024).

Future iterations could incorporate more intuitive interfaces and data-driven insights akin to those offered by Bearable and StrivePD. Sleep tracking could also be incorporated as a passive measure via smartwatch linkage, providing the kind of personalised trend insights already offered by Bearable and StrivePD, and giving a more objective and wider view of the factors influencing symptom occurrence, severity, and medication effectiveness in WwPD than the standalone sleep-related symptom tracking currently available in the MMM app (Figure S6) (Bearable, 2026; "StrivePD," 2026).

Future studies using the MMM app, should also consider incorporating food logging, as estrogen-rich foods, protein intake and caffeine consumption could also be useful to capture in WwPD, given their known interactions with levodopa efficacy and estrogen-mediated pathways that could play a confounding role in observing hormonal effects on PD symptoms and medication effectiveness in WwPD (Ascherio et al., 2004, 2003; Mischley et al., 2017; Virmani et al., 2016).

4.5.2. MMM Tracking Thematic Analysis

While most women (53.3%) using the MMM app gave positive feedback about their experience, the negative or mixed responses from others warrant consideration. Three themes were identified based on the response about tracking experience: utility, overwhelm and adherence.

Most experiences were positive, but some highlighted areas needing improvement. Many users described the MMM app as helpful: *"Very useful"*, *"Very helpful"*. One woman found the app *"so so useful"* but reported that logging symptoms was difficult: *"have to search a long and often repetitive symptoms list"*.

“I found the app so so useful and thanks for the opportunity to try it!”
“Mostly very positive (helped me track and understand my main symptoms in relation to e.g. diet or HRT dose”

Participants feeling overwhelmed emerged as a theme, with users describing the app as *"a little overwhelming"* and noting that *"tracking so many symptoms takes longer than you'd think!"*. Other responses reflected this overwhelm, with users finding the app *"tiring"* and reporting *"too much to do,"* leading one woman to stop using it: *"could not continue"*. This theme of being overwhelmed helps explain the low engagement levels found in the app data that led to limitations in analysis and interpretation. WwPD must balance many aspects of daily living with PD; therefore, PD apps should be designed as smoothly as possible to facilitate integration into their routines.

“a little overwhelming - tracking so many symptoms takes longer than you'd think!”
“too stressful, could not continue”

Finally, adherence presented challenges, as reflected in responses like *"Afraid I didn't manage to get into the habit of using it"* and *“time consuming”*. This reported difficulty with adherence likely reflects accumulated app design issues that hinder integration for WwPD. These struggles led users to abandon the app entirely, with responses like *"tedious and stopped completing it"* and *"Easy to do but forgot some days, then stopped altogether."*

"Keeping up with the clock with medications is so difficult”
“sadly I ran out of enthusiasm and stopped filling in the tracking data”
“the app doesn't make it easy to record multiple (more than 3) symptoms as you have to search a long and often repetitive symptoms list”

This feedback informs app improvements and identifies what PwP require from mobile apps for managing symptoms and medications. Therefore, if the MMM app is to continue to be used, its user experience should be improved.

4.6. Limitations of the study

When the results from this study, it is essential to recognise its limitations. Firstly, there were no males as controls, nor were there people who do not have PD as controls. The lack of the latter control cohort is particularly notable when discussing the menstruating group, as some of the symptoms that were categorised as NMS in this study may be the same, or similar to, symptoms that are related to PMS. A control group of people without PD would have allowed us to distinguish between PD-related NMS and PMS-related symptoms, and to investigate whether some PMS symptoms may be exacerbated by having PD. Martignoni et al. reported that overall characteristics of premenstrual syndrome were similar between WwPD and healthy controls (Martignoni et al., 2002). However, this was a retrospective interview-based study and prospective investigation may be warranted in future studies.

All data points were self-reported by participants without clinician verification. This is a limitation because retrospective self-reporting is susceptible to recall bias, which can result in participants providing erroneous responses that depend on their ability to recall past events and if not given proper consideration, can either underestimate or overestimate the true effect or association (Kip et al., 2001; Neugebauer and Ng, 1990). Although symptom questionnaires in the initial survey focused on the past month to minimise recall errors, the potential for bias remains. Despite these limitations, self-reporting is valuable as it captures information and experiences that only WwPD can provide and helps avoids observer bias (Black, 2013; Mahtani et al., 2018; Subramanian et al., 2022).

The design of the questionnaire for the initial survey could have been improved in several aspects. The questionnaire was designed to ask participants to select age ranges rather than providing exact ages to protect identifiable data points (Figure S6); the latter more precise information on age would have facilitated logistic regression and enabled more accurate age profiles to be generated, whilst controlling for age as a covariate. Additionally, 'perimenopause' was not an explicit option for participants to select for their menstrual status; this resulted in many perimenopausal women selecting 'other', which then required manual reclassification by the researcher. Having 'perimenopause' as an explicit option could have resulted in more accurate categorisation and enabled specific perimenopausal cohort analysis. Furthermore, 'depression and anxiety' as symptoms were presented as a single combined option in the initial survey, but to improve precision were separated into two distinct symptoms in the follow-up survey.

Additional, more detailed data collection in the initial survey would have enhanced the analysis, including age of HRT initiation, specific HRT formulation, route of

administration and duration of therapy, as these variables have been shown to significantly influence risk outcomes and findings in menopause and PD research (Kim et al., 2021; Liu et al., 2014). Similarly, information about contraceptive type would have been informative for evaluating the effects of different hormonal combinations.

The MMM app lacked an ability to send notifications to users to prompt symptom input, potentially contributing to poor engagement. Consequently, most participants did not use the app for the full duration of the study, evidenced by only 35.42% of menstruating app users tracking for sufficient duration to provide analysable symptom and menstrual cycle data, and only 36.19% of eligible participants completing the follow-up survey. A more interactive app with notification capabilities may have enabled more comprehensive symptom analysis throughout cycles and more accurate period reporting. The low engagement level made it challenging to determine whether users forgot to log symptoms or were genuinely asymptomatic, as well as making it difficult to infer details of participants' regular cycles. Consequently, aggregate menstrual phases could only be deduced from n=17 participants who had provided distinct cycle information. Therefore, all findings should be interpreted with consideration of the fact that the initial survey had more robust numbers and engagement than the data which was generated from participants engaging with the app.

Finally, statistical significance testing was applied to observational survey data, which presents methodological challenges as inferential statistics are typically employed in controlled hypothesis-testing scenarios. This approach was used identify prominent differences between groups, but the results should be interpreted as descriptive rather than inferential. This approach is more methodologically sound for the app tracking data, where participants served as their own controls and minimal variation was observed across menstrual cycles. The confounding variables between these populations preclude causal inferences from the observed differences between menstruating and post-menopausal groups.

4.7. Conclusion and Future Directions

This study demonstrates clear hormonal status-related differences in PD symptom presentation between menstruating and postmenopausal women, with menstruating women exhibiting distinct profiles of motor and non-motor symptoms characterised by higher rates of rigidity, dyskinesias, and mood-related symptoms. These findings were consistent across both cross-sectional survey data and longitudinal app tracking, providing convergent evidence for hormonal influences on symptom expression.

The lack of medication differentiation contrasts sharply with the significant symptom differences between menstruating and postmenopausal women. This raises a question of whether women with different symptom profiles should receive tailored medication regimens. Current PD treatment protocols inadequately integrate sex differences, and hormonal status may represent the next consideration for personalising medication requirements in WwPD. HRT findings suggest potential therapeutic benefits for specific age groups (tremor reduction in 50- to 54-year-olds). Contraceptive users reported significantly fewer skin problems, suggesting potential benefits for managing seborrhoeic dermatitis in menstruating WwPD. These findings also raise the question of whether hormonal treatments could be used as an adjunct to the treatment of WwPD. Menstruating women exhibited higher dyskinesia rates, despite receiving lower L-DOPA doses, which may warrant consideration in clinical assessment and dosing protocols. However, study limitations, particularly low app engagement rates and methodological constraints, prevented definitive treatment recommendations at this stage.

To properly investigate these therapeutic possibilities, subsequent investigations will require several methodological improvements. This study was in a predominately white cohort, and broader racial demographics would be feasible in a similar study design due to the remote nature of data collection, so further studies might launch the MMM app in a wider range of countries. Given the findings related to HRT and contraceptives, new studies should address the specific types of HRT and contraceptives that WwPD are taking. Information on the precise age of participants, rather than age ranges, should be collected, in order to control properly for age-related effects. Anthropometric data would also be informative to collect in future studies, to further inform findings on the development of LIDs, which has been reported previously (Martinez-Ramirez et al., 2014). This would enable examination of how specific hormonal formulations may influence PD symptoms and medication effectiveness. Additionally, such research should explicitly include perimenopausal status, as this is a particularly interesting and unexplored aspect of PD in women. From a practical standpoint, subsequent work collecting survey-type demographic data should integrate the survey into the study-related app to prevent loss of participant engagement across multiple platforms and enhance accessibility.

Considering the differences found between menstruating and postmenopausal women in this study, extended longitudinal studies spanning several years would be valuable, to following WwPD who have a normal menstrual cycle as they transition through their perimenopausal and postmenopausal phases. This approach could reveal how their symptom profile and medication needs may evolve over time, and to assess disease progression not just in relation to ageing but also to hormonal transitions.

Realising these future research goals will require significant technical improvements to data collection methods. Limited engagement with the MMM app prevented robust assessment of menstrual cycle-related fluctuations; therefore, enhanced versions might allow for more comprehensive data collection and better characterisation of symptom patterns in menstruating women. Future iterations of the MMM app, or other PD-related apps, should incorporate the usability and tracking experience responses presented in this paper and might include a more interactive interface, real-time machine learning insight into symptom correlation to medication intake and period logging, as well as time-sensitive notifications for reminders to improve engagement and statistical power.

In conclusion, this study identifies consistent symptom differences between menstruating and postmenopausal WwPD across survey and app data, suggesting that hormonal status may influence symptom presentation. While limitations prevent definitive treatment conclusions, the findings highlight the importance of considering hormonal factors in PD research and provide preliminary evidence for future personalised approaches in WwPD. Overall, the results of this study provide valuable information on the experiences of PD in women and raise opportunities for future studies on symptoms and medication usage in women with PD, involving apps or other technologies.

5. Bibliography

A randomized pilot trial of estrogen replacement therapy in post-menopausal women with Parkinson's disease, 2011. . Parkinsonism Relat. Disord. 17, 757–760. https://doi.org/10.1016/j.parkreldis.2011.07.007

Abraham, D.S., Gruber-Baldini, A.L., Magder, L.S., McArdle, P.F., Tom, S.E., Barr, E., Schrader, K., Shulman, L.M., 2019. Sex Differences in Parkinson's Disease Presentation and Progression. Parkinsonism Relat. Disord. 69, 48–54. https://doi.org/10.1016/j.parkreldis.2019.10.019

Al Sweidi, S., Sánchez, M.G., Bourque, M., Morissette, M., Dluzen, D., Di Paolo, T., 2012. Oestrogen Receptors and Signalling Pathways: Implications for Neuroprotective Effects of Sex Steroids in Parkinson's Disease. J. Neuroendocrinol. 24, 48–61. https://doi.org/10.1111/j.1365-2826.2011.02193.x

Allen, R.P., Auerbach, S., Bahrain, H., Auerbach, M., Earley, C.J., 2013. The prevalence and impact of restless legs syndrome on patients with iron deficiency anemia. Am. J. Hematol. 88, 261–264. https://doi.org/10.1002/ajh.23397

Alonso, A., García Rodríguez, L.A., Logroscino, G., Hernán, M.A., 2009. Use of antidepressants and the risk of Parkinson's disease: a prospective study. J. Neurol. Neurosurg. Psychiatry 80, 671–674. https://doi.org/10.1136/jnnp.2008.152983

Amagai, S., Pila, S., Kaat, A.J., Nowinski, C.J., Gershon, R.C., 2022. Challenges in Participant Engagement and Retention Using Mobile Health Apps: Literature Review. J. Med. Internet Res. 24, e35120. https://doi.org/10.2196/35120

APDA Symptom Tracker App, 2026. . Am. Park. Dis. Assoc. URL https://www.apdaparkinson.org/apda-symptom-tracker/ (accessed 3.7.26).

Ascherio, A., Chen, H., Schwarzschild, M.A., Zhang, S.M., Colditz, G.A., Speizer, F.E., 2003. Caffeine, postmenopausal estrogen, and risk of Parkinson's disease. Neurology 60, 790–795. https://doi.org/10.1212/01.wnl.0000046523.05125.87

Ascherio, A., Weisskopf, M.G., O'Reilly, E.J., McCullough, M.L., Calle, E.E., Rodriguez, C., Thun, M.J., 2004. Coffee consumption, gender, and Parkinson's disease mortality in the Cancer Prevention Study II cohort: The modifying effects of estrogen. Am. J. Epidemiol. 160, 977–984. https://doi.org/10.1093/aje/kwh312

Azadi, H., Akbarzadeh-Totonchi, M.-R., Jahani, Y., Gohari, S.H., Khajouei, R., 2024. Evaluation of mobile applications related to patients with Parkinson's disease based on their essential features and capabilities. BMC Med. Inform. Decis. Mak. 24, 407. https://doi.org/10.1186/s12911-024-02804-9

Ba, F., Poonja, S., Torabi, P., Yen, K., Daud Shah, B., Noyes, E., Dimapilis, G., Corrick, S., Sankar, T., Rajput, A., Monchi, O., Vitali, A.M., Salami, B., Shetty, A., Miyasaki, J.M., 2025. Ethnic and Gender Disparities in Access to Deep Brain Stimulation Surgery for Parkinson's Disease. Mov. Disord. Clin. Pract. n/a. https://doi.org/10.1002/mdc3.70312

Baba, Y., Putzke, J.D., Whaley, N.R., Wszolek, Z.K., Uitti, R.J., 2005. Gender and the Parkinson'sdisease phenotype. J. Neurol. 252, 1201–1205. https://doi.org/10.1007/s00415-005-0835-7

Baik, K., Chung, S.J., Yoo, H.S., Lee, Y.H., Jung, J.H., Sohn, Y.H., Lee, P.H., 2020. Sex-dependent association of urate on the patterns of striatal dopamine depletion in

Parkinson's disease. Eur. J. Neurol. 27, 773–778. https://doi.org/10.1111/ene.14152
Bakeberg, M.C., Jefferson, A., Riley, M., Byrnes, M., Ghosh, S., Mastaglia, F.L., Horne, M.K., McGregor, S., Stell, R., Kenna, J., Walters, S., Hince, D., Anderton, R.S., 2019. Elevated Serum Homocysteine Levels Have Differential Gender-Specific Associations with Motor and Cognitive States in Parkinson's Disease. Park. Dis. 2019, 3124295. https://doi.org/10.1155/2019/3124295
Bates, N., Chin, M., Becker, T. (Eds.), 2022. Measuring Sex, Gender Identity, and Sexual Orientation. National Academies Press, Washington, D.C. https://doi.org/10.17226/26424
Beach, T.G., Sue, L.I., Walker, D.G., Lue, L.F., Connor, D.J., Caviness, J.N., Sabbagh, M.N., Adler, C.H., 2007. Marked microglial reaction in normal aging human substantia nigra: correlation with extraneuronal neuromelanin pigment deposits. Acta Neuropathol. (Berl.) 114, 419–424. https://doi.org/10.1007/s00401-007-0250-5
Bearable, 2026. Bearable Symptom Tracker App | Track Pain, Mood & Medication. URL https://bearable.app/ (accessed 3.7.26).
Becker, D., Maric, A., Schreiner, S.J., Büchele, F., Baumann, C.R., Waldvogel, D., 2022. Onset of Postural Instability in Parkinson's Disease Depends on Age rather than Disease Duration. Park. Dis. 2022, 6233835. https://doi.org/10.1155/2022/6233835
Beery, A.K., 2018. Inclusion of females does not increase variability in rodent research studies. Curr. Opin. Behav. Sci. 23, 143–149. https://doi.org/10.1016/j.cobeha.2018.06.016
Bega, D., Luo, S., Fernandez, H., Chou, K., Aminoff, M., Parashos, S., Walker, H., Russell, D.S., Christine, C.W., Dhall, R., Singer, C., Bodis-Wollner, I., Hamill, R., Truong, D., Mari, Z., Glazmann, S., Huang, M., Houston, E., Simuni, T., 2015. Impact of Depression on Progression of Impairment and Disability in Early Parkinson's Disease. Mov. Disord. Clin. Pract. 2, 371–378. https://doi.org/10.1002/mdc3.12205
Benedetti, M.D., Maraganore, D.M., Bower, J.H., McDonnell, S.K., Peterson, B.J., Ahlskog, J.E., Schaid, D.J., Rocca, W.A., 2001. Hysterectomy, menopause, and estrogen use preceding Parkinson's disease: An exploratory case-control study. Mov. Disord. 16, 830–837. https://doi.org/10.1002/mds.1170
Bereczki, D., 2010. The description of all four cardinal signs of Parkinson's disease in a Hungarian medical text published in 1690. Parkinsonism Relat. Disord. 16, 290–293. https://doi.org/10.1016/j.parkreldis.2009.11.006
Bernstein, M.T., Graff, L.A., Avery, L., Palatnick, C., Parnerowski, K., Targownik, L.E., 2014. Gastrointestinal symptoms before and during menses in healthy women. BMC Womens Health 14, 14. https://doi.org/10.1186/1472-6874-14-14
Beyer, C., Pilgrim, C., Reisert, I., 1991. Dopamine content and metabolism in mesencephalic and diencephalic cell cultures: sex differences and effects of sex steroids. J. Neurosci. 11, 1325–1333. https://doi.org/10.1523/JNEUROSCI.11-05-01325.1991
Black, N., 2013. Patient reported outcome measures could help transform healthcare. https://doi.org/10.1136/bmj.f167

Boccalini, C., Carli, G., Pilotto, A., Padovani, A., Perani, D., 2022. Gender differences in dopaminergic system dysfunction in de novo Parkinson's disease clinical subtypes. Neurobiol. Dis. 167, 105668. https://doi.org/10.1016/j.nbd.2022.105668

Bonifati, V., Fabrizio, E., Vanacore, N., De Mari, M., Meco, G., 1995. Familial Parkinson's disease: a clinical genetic analysis. Can. J. Neurol. Sci. J. Can. Sci. Neurol. 22, 272–279. https://doi.org/10.1017/s0317167100039469

Boudíková, B., Szumlanski, C., Maidak, B., Weinshilboum, R., 1990. Human liver catechol-O-methyltransferase pharmacogenetics. Clin. Pharmacol. Ther. 48, 381–389. https://doi.org/10.1038/clpt.1990.166

Bourque, M., Dluzen, D.E., Di Paolo, T., 2009. Neuroprotective actions of sex steroids in Parkinson's disease. Front. Neuroendocrinol., Steroids and Neuroprotection 30, 142–157. https://doi.org/10.1016/j.yfrne.2009.04.014

Bourque, M., Morissette, M., Al Sweidi, S., Caruso, D., Melcangi, R.C., Di Paolo, T., 2015. Neuroprotective Effect of Progesterone in MPTP-Treated Male Mice. Neuroendocrinology 103, 300–314. https://doi.org/10.1159/000438789

Bourque, M., Morissette, M., Di Paolo, T., 2024. Neuroactive steroids and Parkinson's disease: Review of human and animal studies. Neurosci. Biobehav. Rev. 156, 105479. https://doi.org/10.1016/j.neubiorev.2023.105479

Bovenzi, R., Conti, M., Degoli, G.R., Cerroni, R., Artusi, C.A., Pierantozzi, M., Stefani, A., Mercuri, N.B., Schirinzi, T., 2024a. Pregnancy, fertile life factors, and associated clinical course in PRKN early-onset Parkinson's disease. Neurol. Sci. 45, 591–599. https://doi.org/10.1007/s10072-023-07029-8

Bovenzi, R., Schirinzi, T., Conti, M., Sancesario, G.M., Zenuni, H., Simonetta, C., Bissacco, J., Mascioli, D., Pieri, M., Cerroni, R., Stefani, A., Mercuri, N.B., Pierantozzi, M., 2024b. A biological characterization of patients with postmenopausal Parkinson's disease. J. Neurol. 271, 3610–3615. https://doi.org/10.1007/s00415-024-12258-8

Brakedal, B., Toker, L., Haugarvoll, K., Tzoulis, C., 2022. A nationwide study of the incidence, prevalence and mortality of Parkinson's disease in the Norwegian population. Npj Park. Dis. 8, 1–8. https://doi.org/10.1038/s41531-022-00280-4

Brinton, R.D., Yao, J., Yin, F., Mack, W.J., Cadenas, E., 2015. Perimenopause as a neurological transition state. Nat. Rev. Endocrinol. 11, 393–405. https://doi.org/10.1038/nrendo.2015.82

Bromberger, J.T., Kravitz, H.M., Chang, Y.-F., Cyranowski, J.M., Brown, C., Matthews, K.A., 2011. Major depression during and after the menopausal transition: Study of Women's Health Across the Nation (SWAN). Psychol. Med. 41, 1879–1888. https://doi.org/10.1017/S003329171100016X

Brooks, W.J., Jarvis, M.F., Wagner, G.C., 1989. Influence of sex, age and strain on MPTP-induced neurotoxicity. Res. Commun. Subst. Abuse 10, 181–184.

Bull, J.R., Rowland, S.P., Scherwitzl, E.B., Scherwitzl, R., Danielsson, K.G., Harper, J., 2019. Real-world menstrual cycle characteristics of more than 600,000 menstrual cycles. Npj Digit. Med. 2, 1–8. https://doi.org/10.1038/s41746-019-0152-7

Callier, S., Morissette, M., Grandbois, M., Pélaprat, D., Di Paolo, T., 2001. Neuroprotective properties of 17beta-estradiol, progesterone, and raloxifene in

MPTP C57Bl/6 mice. Synap. N. Y. N 41, 131–138. https://doi.org/10.1002/syn.1067
Camacho, M., Greenland, J.C., Daruwalla, C., Scott, K.M., Patel, B., Apostolopoulos, D., Ribeiro, J., O'Reilly, M., Hu, M.T., Williams-Gray, C.H., 2025. The profile of gastrointestinal dysfunction in prodromal to late-stage Parkinson's disease. NPJ Park. Dis. 11, 123. https://doi.org/10.1038/s41531-025-00900-9
Cantuti-Castelvetri, I., Keller-McGandy, C., Bouzou, B., Asteris, G., Clark, T.W., Frosch, M.P., Standaert, D.G., 2007. Effects of Gender on Nigral Gene Expression and Parkinson Disease. Neurobiol. Dis. 26, 606–614. https://doi.org/10.1016/j.nbd.2007.02.009
Carrarini, C., Russo, M., Dono, F., Di Pietro, M., Rispoli, M.G., Di Stefano, V., Ferri, L., Barbone, F., Vitale, M., Thomas, A., Sensi, S.L., Onofrj, M., Bonanni, L., 2019. A Stage-Based Approach to Therapy in Parkinson's Disease. Biomolecules 9, 388. https://doi.org/10.3390/biom9080388
Casas, S., Giuliani, F., Cremaschi, F., Yunes, R., Cabrera, R., 2013. Neuromodulatory effect of progesterone on the dopaminergic, glutamatergic, and GABAergic activities in a male rat model of Parkinson's disease. Neurol. Res. 35, 719–725. https://doi.org/10.1179/1743132812Y.0000000142
Castrioto, A., Hulliger, S., Poon, Y.-Y., Lang, A.E., Moro, E., 2010. A Survey on the Impact of the Menstrual Cycle on Movement Disorders Severity. Can. J. Neurol. Sci. J. Can. Sci. Neurol. 37, 478–481. https://doi.org/10.1017/S0317167100010490
Cereda, E., Barichella, M., Cassani, E., Caccialanza, R., Pezzoli, G., 2013. Reproductive factors and clinical features of Parkinson's disease. Parkinsonism Relat. Disord. 19, 1094–1099. https://doi.org/10.1016/j.parkreldis.2013.07.020
Chang, Y.-P., Lee, M.-S., Wu, D.-W., Tsai, J.-H., Ho, P.-S., Lin, C.-H.R., Chuang, H.-Y., 2020. Risk factors for depression in patients with Parkinson's disease: A nationwide nested case-control study. PLOS ONE 15, e0236443. https://doi.org/10.1371/journal.pone.0236443
Charkoudian, N., Stachenfeld, N., 2016. Sex hormone effects on autonomic mechanisms of thermoregulation in humans. Auton. Neurosci. Basic Clin. 196, 75–80. https://doi.org/10.1016/j.autneu.2015.11.004
Chaturvedi, R.K., Beal, M.F., 2008. PPAR: a therapeutic target in Parkinson's disease. J. Neurochem. 106, 506–518. https://doi.org/10.1111/j.1471-4159.2008.05388.x
Chen, A., 2025. Mobile retention benchmarks for 2014 vs 2013 show a 50% drop in D1 retention (Guest post) at andrewchen. URL https://andrewchen.com/mobile-retention-benchmarks-for-2014-vs-2013-show-a-50-drop-in-d1-retention-guest-post/ (accessed 8.27.25).
Chen, J.-Q., Cammarata, P.R., Baines, C.P., Yager, J.D., 2009. Regulation of Mitochondrial Respiratory Chain Biogenesis by Estrogens/Estrogen Receptors and Physiological, Pathological and Pharmacological Implications. Biochim. Biophys. Acta 1793, 1540–1570. https://doi.org/10.1016/j.bbamcr.2009.06.001
Chen, T.-Y., Lin, C.-L., Wang, L.-F., Tsai, K.-L., Lin, J.-Y., Hsu, C., 2019. Targeting GPER1 to suppress autophagy as a male-specific therapeutic strategy for iron-induced striatal injury. Sci. Rep. 9, 6661. https://doi.org/10.1038/s41598-019-43244-0
Chen, Y. j., Wu, C. y., Lin, M. w., Chen, T. j., Liao, K. k., Chen, Y. c., Hwang, C. y., Chu, S. y., Chen, C. c., Lee, D. d., Chang, Y. t., Wang, W. j., Liu, H. n., 2011. Comorbidity profiles among patients with bullous pemphigoid: a nationwide population-

based study. Br. J. Dermatol. 165, 593–599. https://doi.org/10.1111/j.1365-2133.2011.10386.x
Chen, Y.-H., Wang, V., Huang, E.Y.-K., Chou, Y.-C., Kuo, T.-T., Olson, L., Hoffer, B.J., 2019. Delayed Dopamine Dysfunction and Motor Deficits in Female Parkinson Model Mice. Int. J. Mol. Sci. 20, 6251. https://doi.org/10.3390/ijms20246251
Cilia, R., Siri, C., Rusconi, D., Allegra, R., Ghiglietti, A., Sacilotto, G., Zini, M., Zecchinelli, A.L., Asselta, R., Duga, S., Paganoni, A.M., Pezzoli, G., Seia, M., Goldwurm, S., 2014. LRRK2 mutations in Parkinson's disease: Confirmation of a gender effect in the Italian population. Parkinsonism Relat. Disord. 20, 911–914. https://doi.org/10.1016/j.parkreldis.2014.04.016
Classen, J., Koschel, J., Oehlwein, C., Seppi, K., Urban, P., Winkler, C., Wüllner, U., Storch, A., 2017. Nonmotor fluctuations: phenotypes, pathophysiology, management, and open issues. J. Neural Transm. 124, 1029–1036. https://doi.org/10.1007/s00702-017-1757-0
Clayton, J.A., 2016. Studying both sexes: a guiding principle for biomedicine. FASEB J. 30, 519–524. https://doi.org/10.1096/fj.15-279554
Clayton, J.A., Collins, F.S., 2014. Policy: NIH to balance sex in cell and animal studies. Nature 509, 282–283. https://doi.org/10.1038/509282a
Colombo, D., Abbruzzese, G., Antonini, A., Barone, P., Bellia, G., Franconi, F., Simoni, L., Attar, M., Zagni, E., Haggiag, S., Stocchi, F., 2015. The "Gender Factor" in Wearing-Off among Patients with Parkinson's Disease: A Post Hoc Analysis of DEEP Study. Sci. World J. 2015, 787451. https://doi.org/10.1155/2015/787451
Conti, V., Izzo, V., Russillo, M.C., Picillo, M., Amboni, M., Scaglione, C.L.M., Nicoletti, A., Cani, I., Cicero, C.E., De Bellis, E., Charlier, B., Giudice, V., Somma, G., Corbi, G., Barone, P., Filippelli, A., Pellecchia, M.T., 2022. Gender Differences in Levodopa Pharmacokinetics in Levodopa-Naïve Patients With Parkinson's Disease. Front. Med. 9. https://doi.org/10.3389/fmed.2022.909936
Corwin, C., Nikolopoulou, A., Pan, A.L., Nunez-Santos, M., Vallabhajosula, S., Serrano, P., Babich, J., Figueiredo-Pereira, M.E., 2018. Prostaglandin D2/J2 signaling pathway in a rat model of neuroinflammation displaying progressive parkinsonian-like pathology: potential novel therapeutic targets. J. Neuroinflammation 15, 272. https://doi.org/10.1186/s12974-018-1305-3
Costa, G., Pinna, A., Porceddu, P.F., Casu, M.A., Di Maio, A., Napolitano, F., Usiello, A., Morelli, M., 2018. Rhes Counteracts Dopamine Neuron Degeneration and Neuroinflammation Depending on Gender and Age. Front. Aging Neurosci. 10, 163. https://doi.org/10.3389/fnagi.2018.00163
Costa, G., Porceddu, P.F., Serra, M., Casu, M.A., Schiano, V., Napolitano, F., Pinna, A., Usiello, A., Morelli, M., 2019. Lack of Rhes Increases MDMA-Induced Neuroinflammation and Dopamine Neuron Degeneration: Role of Gender and Age. Int. J. Mol. Sci. 20, 1556. https://doi.org/10.3390/ijms20071556
Cullity, E.R., Madsen, H.B., Perry, C.J., Kim, J.H., 2019. Postnatal developmental trajectory of dopamine receptor 1 and 2 expression in cortical and striatal brain regions. J. Comp. Neurol. 527, 1039–1055. https://doi.org/10.1002/cne.24574
Cunningham, A.C., Pal, L., Wickham, A.P., Prentice, C., Goddard, F.G.B., Klepchukova, A., Zhaunova, L., 2024. Chronicling menstrual cycle patterns across the reproductive lifespan with real-world data. Sci. Rep. 14, 10172. https://doi.org/10.1038/s41598-024-60373-3

Curtze, C., Nutt, J.G., Carlson-Kuhta, P., Mancini, M., Horak, F.B., 2015. Levodopa is a Double-Edged Sword for Balance and Gait in People with Parkinson's Disease. Mov. Disord. Off. J. Mov. Disord. Soc. 30, 1361–1370. https://doi.org/10.1002/mds.26269

Dajani, E.Z., Shahwan, T.G., Dajani, N.E., 2003. Prostaglandins and brain-gut axis. J. Physiol. Pharmacol. Off. J. Pol. Physiol. Soc. 54 Suppl 4, 155–164.

Datz, F.L., Christian, P.E., Moore, J., 1987. Gender-Related Differences in Gastric Emptying. J. Nucl. Med. 28, 1204–1207.

Dauwan, M., Begemann, M.J.H., Slot, M.I.E., Lee, E.H.M., Scheltens, P., Sommer, I.E.C., 2021. Physical exercise improves quality of life, depressive symptoms, and cognition across chronic brain disorders: a transdiagnostic systematic review and meta-analysis of randomized controlled trials. J. Neurol. 268, 1222–1246. https://doi.org/10.1007/s00415-019-09493-9

De Pablo-Fernández, E., Lees, A.J., Holton, J.L., Warner, T.T., 2019. Prognosis and Neuropathologic Correlation of Clinical Subtypes of Parkinson Disease. JAMA Neurol. 76, 470–479. https://doi.org/10.1001/jamaneurol.2018.4377

De Pablo-Fernandez, E., Tur, C., Revesz, T., Lees, A.J., Holton, J.L., Warner, T.T., 2017. Association of Autonomic Dysfunction With Disease Progression and Survival in Parkinson Disease. JAMA Neurol. 74, 970–976. https://doi.org/10.1001/jamaneurol.2017.1125

Del Prete, E., Schmitt, E., Meoni, S., Fraix, V., Castrioto, A., Pelissier, P., Ceravolo, R., Moro, E., 2022. Do neuropsychiatric fluctuations temporally match motor fluctuations in Parkinson's disease? Neurol. Sci. 43, 3641–3647. https://doi.org/10.1007/s10072-021-05833-8

Dexter, D.T., Jenner, P., 2013. Parkinson disease: from pathology to molecular disease mechanisms. Free Radic. Biol. Med., Neurodegeneration 62, 132–144. https://doi.org/10.1016/j.freeradbiomed.2013.01.018

Di Luca, D.G., Luo, S., Liu, H., Cohn, M., Davis, T.L., Ramirez-Zamora, A., Rafferty, M., Dahodwala, N., Naito, A., Neault, M., Beck, J., Marras, C., 2023. Racial and Ethnic Differences in Health-Related Quality of Life for Individuals With Parkinson Disease Across Centers of Excellence. Neurology 100, e2170–e2181. https://doi.org/10.1212/WNL.0000000000207247

Dinkelbach, L., Möller, B., Witt, K., Schnitzler, A., Südmeyer, M., 2017. How to improve patient education on deep brain stimulation in Parkinson's disease: the CARE Monitor study. BMC Neurol. 17, 36. https://doi.org/10.1186/s12883-017-0820-7

Dluzen, D., 1997. Estrogen decreases corpus striatal neurotoxicity in response to 6-hydroxydopamine. Brain Res. 767, 340–344. https://doi.org/10.1016/S0006-8993(97)00630-6

Dluzen, Dean E., Mcdermott, J.L., Anderson, L.I., 2001. Tamoxifen eliminates estrogen's neuroprotective effect upon MPTP-induced neurotoxicity of the nigrostriatal dopaminergic system. Neurotox. Res. 3, 291–300. https://doi.org/10.1007/BF03033268

Dluzen, D. E., McDermott, J.L., Anderson, L.I., 2001. Tamoxifen Diminishes Methamphetamine-Induced Striatal Dopamine Depletion in Intact Female and Male Mice. J. Neuroendocrinol. 13, 618–624. https://doi.org/10.1046/j.1365-2826.2001.00675.x

Dluzen, D.E., McDermott, J.L., Bourque, M., Di Paolo, T., Darvesh, A.S., Buletko, A.B., Laping, N.J., 2011. Markers Associated with Sex Differences in Methamphetamine-Induced Striatal Dopamine Neurotoxicity. Curr. Neuropharmacol. 9, 40–44. https://doi.org/10.2174/157015911795017399

Dluzen, D.E., McDermott, J.L., Liu, B., 1996a. Estrogen as a neuroprotectant against MPTP-Induced neurotoxicity in C57/B1 mice. Neurotoxicol. Teratol. 18, 603–606. https://doi.org/10.1016/0892-0362(96)00086-4

Dluzen, D.E., McDermott, J.L., Liu, B., 1996b. Estrogen Alters MPTP-Induced Neurotoxicity in Female Mice: Effects on Striatal Dopamine Concentrations and Release. J. Neurochem. 66, 658–666. https://doi.org/10.1046/j.1471-4159.1996.66020658.x

Donzuso, G., Cicero, C.E., Vinciguerra, E., Sergi, R., Luca, A., Mostile, G., Terravecchia, C., Zappia, M., Nicoletti, A., 2023. Gender differences in non-motor fluctuations in Parkinson's disease. J. Neural Transm. 130, 1249–1257. https://doi.org/10.1007/s00702-023-02679-6

dos Santos, E.U.D., Sampaio, T.F., Tenório dos Santos, A.D., Bezerra Leite, F.C., da Silva, R.C., Crovella, S., Asano, A.G.C., Asano, N.M.J., de Souza, P.R.E., 2019. The influence of SLC6A3 and DRD2 polymorphisms on levodopa-therapy in patients with sporadic Parkinson's disease. J. Pharm. Pharmacol. 71, 206–212. https://doi.org/10.1111/jphp.13031

Durmaz Celik, N., Yaman Kula, A., Murat, N., Kuzu Kumcu, M., Topal, A., Özkan, S., 2025. Evaluation of compliance and accuracy in Parkinson's disease motor symptom tracking: a comparative study of digital and traditional paper diaries using a smartphone application (MyParkinson's). Front. Neurol. 16. https://doi.org/10.3389/fneur.2025.1522721

Ekue, Boulanger, Morissette, M., Di Paolo, T, 2002. Lack of Effect of Testosterone and Dihydrotestosterone Compared to 17β-Oestradiol in 1-Methyl-4-Phenyl-1,2,3,6, Tetrahydropyridine-Mice - Ekue - 2002 - Journal of Neuroendocrinology - Wiley Online Library [WWW Document]. URL https://onlinelibrary.wiley.com/doi/full/10.1046/j.1365-2826.2002.00833.x?saml_referrer (accessed 6.29.25).

Factor, S.A., 1993. Acetazolamide therapy of menstrual-related fluctuations in parkinson's disease. Mov. Disord. 8, 240–241. https://doi.org/10.1002/mds.870080231

Feng, Z.-H., Wang, T.-G., Li, D.-D., Fung, P., Wilson, B.C., Liu, B., Ali, S.F., Langenbach, R., Hong, J.-S., 2002. Cyclooxygenase-2-deficient mice are resistant to 1-methyl-4-phenyl1, 2, 3, 6-tetrahydropyridine-induced damage of dopaminergic neurons in the substantia nigra. Neurosci. Lett. 329, 354–358. https://doi.org/10.1016/S0304-3940(02)00704-8

Fernandez-Jimenez, M.C., Moreno, G., Wright, I., Shih, P.-C., Vaquero, M.P., Remacha, A.F., 2020. Iron Deficiency in Menstruating Adult Women: Much More than Anemia. Womens Health Rep. 1, 26–35. https://doi.org/10.1089/whr.2019.0011

Ferreira, J.J., Katzenschlager, R., Bloem, B.R., Bonuccelli, U., Burn, D., Deuschl, G., Dietrichs, E., Fabbrini, G., Friedman, A., Kanovsky, P., Kostic, V., Nieuwboer, A., Odin, P., Poewe, W., Rascol, O., Sampaio, C., Schüpbach, M., Tolosa, E., Trenkwalder, C., Schapira, A., Berardelli, A., Oertel, W.H., 2013. Summary of the recommendations of the EFNS/MDS-ES review on therapeutic management of

Parkinson's disease. Eur. J. Neurol. 20, 5–15. https://doi.org/10.1111/j.1468-1331.2012.03866.x
Fischer, M., Gemende, I., Marsch, W.Ch., Fischer, P.A., 2001. Skin function and skin disorders in Parkinson's disease. J. Neural Transm. 108, 205–213. https://doi.org/10.1007/s007020170088
Fleming, V., Tolson, D., Schartau, E., 2004. Changing perceptions of womanhood: living with Parkinson's Disease. Int. J. Nurs. Stud. 41, 515–524. https://doi.org/10.1016/j.ijnurstu.2003.12.004
Freeman, E.W., Sammel, M.D., Sanders, R.J., 2014. Risk of long-term hot flashes after natural menopause: evidence from the Penn Ovarian Aging Study cohort. Menopause 21, 924. https://doi.org/10.1097/GME.0000000000000196
Fullard, M.E., Dafoe, A., Shelton, E., Kern, D.S., Matlock, D.D., Morris, M.A., 2024. How Women and Men with Parkinson's Disease Approach Decision-Making for Deep Brain Stimulation Surgery. Mov. Disord. Clin. Pract. 12, 453–463. https://doi.org/10.1002/mdc3.14284
Fullard, M.E., Morley, J.F., Duda, J.E., 2017. Olfactory Dysfunction as an Early Biomarker in Parkinson's Disease. Neurosci. Bull. 33, 515–525. https://doi.org/10.1007/s12264-017-0170-x
Fullard, M.E., Tran, B., Xie, S.X., Toledo, J.B., Scordia, C., Linder, C., Purri, R., Weintraub, D., Duda, J.E., Chahine, L.M., Morley, J.F., 2016. Olfactory impairment predicts cognitive decline in early Parkinson's disease. Parkinsonism Relat. Disord. 25, 45–51. https://doi.org/10.1016/j.parkreldis.2016.02.013
Gajjar, T.M., Anderson, L.I., Dluzen, D.E., 2003. Acute effects of estrogen upon methamphetamine induced neurotoxicity of the nigrostriatal dopaminergic system. J. Neural Transm. 110, 1215–1224. https://doi.org/10.1007/s00702-003-0045-3
Gao, L., Yang, Y., Cai, L., Xiong, Y., 2022. Gender Differences in Pain Subtypes among Patients with Parkinson's Disease. J. Integr. Neurosci. 21, 120. https://doi.org/10.31083/j.jin2104120
Gao, X., Dluzen, D.E., 2001. Tamoxifen abolishes estrogen's neuroprotective effect upon methamphetamine neurotoxicity of the nigrostriatal dopaminergic system. Neuroscience 103, 385–394. https://doi.org/10.1016/S0306-4522(01)00014-8
Gatto, N.M., Deapen, D., Stoyanoff, S., Pinder, R., Narayan, S., Bordelon, Y., Ritz, B., 2014. Lifetime exposure to estrogens and Parkinson's disease in California teachers. Parkinsonism Relat. Disord. 20, 1149–1156. https://doi.org/10.1016/j.parkreldis.2014.08.003
Genazzani, A.R., Bernardi, F., Pluchino, N., Begliuomini, S., Lenzi, E., Casarosa, E., Luisi, M., 2005. Endocrinology of menopausal transition and its brain implications. CNS Spectr. 10, 449–457. https://doi.org/10.1017/s1092852900023142
Gillies, G.E., Murray, H.E., Dexter, D., McArthur, S., 2004. Sex dimorphisms in the neuroprotective effects of estrogen in an animal model of Parkinson's disease. Pharmacol. Biochem. Behav. 78, 513–522. https://doi.org/10.1016/j.pbb.2004.04.022

Gjerstad, M.D., Alves, G., Wentzel-Larsen, T., Aarsland, D., Larsen, J.P., 2006. Excessive daytime sleepiness in Parkinson disease. Neurology 67, 853–858. https://doi.org/10.1212/01.wnl.0000233980.25978.9d

Goekoop, R., Barkhof, F., Duschek, E.J.J., Netelenbos, C., Knol, D.L., Scheltens, P., Rombouts, S.A.R.B., 2006. Raloxifene treatment enhances brain activation during recognition of familiar items: a pharmacological fMRI study in healthy elderly males. Neuropsychopharmacol. Off. Publ. Am. Coll. Neuropsychopharmacol. 31, 1508–1518. https://doi.org/10.1038/sj.npp.1300956

Golbe, L.I., 1987. Parkinson's disease and pregnancy. Neurology 37, 1245–1245. https://doi.org/10.1212/WNL.37.7.1245

Gomez-Mancilla, B., Bédard, P.J., 1992. Effect of estrogen and progesterone on l-DOPA induced dyskinesia in MPTP-treated monkeys. Neurosci. Lett. 135, 129–132. https://doi.org/10.1016/0304-3940(92)90152-W

Grandbois, M., Morissette, M., Callier, S., Di Paolo, T., 2000. Ovarian steroids and raloxifene prevent MPTP-induced dopamine depletion in mice. NeuroReport 11, 343.

Grosset, K.A., Grosset, D.G., 2005. Patient-perceived involvement and satisfaction in Parkinson's disease: Effect on therapy decisions and quality of life. Mov. Disord. 20, 616–619. https://doi.org/10.1002/mds.20393

Guimarães, I., Póvoa, A.M., 2020. Primary Dysmenorrhea: Assessment and Treatment. RBGO Gynecol. Obstet. 42, 501–507. https://doi.org/10.1055/s-0040-1712131

Haaxma, C.A., Bloem, B.R., Borm, G.F., Oyen, W.J.G., Leenders, K.L., Eshuis, S., Booij, J., Dluzen, D.E., Horstink, M.W.I.M., 2007. Gender differences in Parkinson's disease. J. Neurol. Neurosurg. Psychiatry 78, 819–824. https://doi.org/10.1136/jnnp.2006.103788

Haehner, A., Habersack, A., Wienecke, M., Storch, A., Reichmann, H., Hummel, T., 2015. Early Parkinson's disease patients on rasagiline present with better odor discrimination. J. Neural Transm. 122, 1541–1546. https://doi.org/10.1007/s00702-015-1433-1

Hagell, P., Odin, P., Vinge, E., 1998. Pregnancy in Parkinson's disease: A review of the literature and a case report. Mov. Disord. 13, 34–38. https://doi.org/10.1002/mds.870130110

Hamberg, K., Hariz, G.-M., 2014. The decision-making process leading to deep brain stimulation in men and women with parkinson's disease – an interview study. BMC Neurol. 14, 89. https://doi.org/10.1186/1471-2377-14-89

Hanamsagar, R., Alter, M.D., Block, C.S., Sullivan, H., Bolton, J.L., Bilbo, S.D., 2017. Generation of a microglial developmental index in mice and in humans reveals a sex difference in maturation and immune reactivity. Glia 65, 1504–1520. https://doi.org/10.1002/glia.23176

Hariz, G.-M., Limousin, P., Zrinzo, L., Tripoliti, E., Aviles-Olmos, I., Jahanshahi, M., Hamberg, K., Foltynie, T., 2013. Gender differences in quality of life following subthalamic stimulation for Parkinson's disease. Acta Neurol. Scand. 128, 281–285. https://doi.org/10.1111/ane.12127

Harvey, L.J., Armah, C.N., Dainty, J.R., Foxall, R.J., Lewis, D.J., Langford, N.J., Fairweather-Tait, S.J., 2005. Impact of menstrual blood loss and diet on iron

deficiency among women in the UK. Br. J. Nutr. 94, 557–564. https://doi.org/10.1079/BJN20051493
Hassin-Baer, S., Molchadski, I., Cohen, O.S., Nitzan, Z., Efrati, L., Tunkel, O., Kozlova, E., Korczyn, A.D., 2011. Gender effect on time to levodopa-induced dyskinesias. J. Neurol. 258, 2048–2053. https://doi.org/10.1007/s00415-011-6067-0
He, R., Zhao, Y., He, Y., Zhou, Y., Yang, J., Zhou, Xiaoting, Zhu, L., Zhou, Xun, Liu, Z., Xu, Q., Sun, Q., Tan, J., Yan, X., Tang, B., Guo, J., 2020. Olfactory Dysfunction Predicts Disease Progression in Parkinson's Disease: A Longitudinal Study. Front. Neurosci. 14, 569777. https://doi.org/10.3389/fnins.2020.569777
He, Y., Li, R., Yu, Y., Huang, C., Xu, Z., Wang, T., Chen, M., Huang, H., Qi, Z., 2024. Human neural stem cells promote mitochondrial genesis to alleviate neuronal damage in MPTP-induced cynomolgus monkey models. Neurochem. Int. 175, 105700. https://doi.org/10.1016/j.neuint.2024.105700
Heart, T., Kalderon, E., 2013. Older adults: Are they ready to adopt health-related ICT? Int. J. Med. Inf. 82, e209–e231. https://doi.org/10.1016/j.ijmedinf.2011.03.002
Hely, M.A., Morris, J.G.L., Reid, W.G.J., O'Sullivan, D. j., Williamson, P.M., Broe, G.A., Adena, M.A., 1995. Age at onset: the major determinant of outcome in Parkinson's disease. Acta Neurol. Scand. 92, 455–463. https://doi.org/10.1111/j.1600-0404.1995.tb00480.x
Hirohata, M., Ono, K., Morinaga, A., Ikeda, T., Yamada, M., 2009. Anti-aggregation and fibril-destabilizing effects of sex hormones on α-synuclein fibrils *in vitro*. Exp. Neurol. 217, 434–439. https://doi.org/10.1016/j.expneurol.2009.03.003
Ho, D.H., Yi, S., Seo, H., Son, I., Seol, W., 2014. Increased DJ-1 in Urine Exosome of Korean Males with Parkinson's Disease. BioMed Res. Int. 2014, 704678. https://doi.org/10.1155/2014/704678
Holloway, R., Shoulson, I., Kieburtz, K., 2004. Pramipexole vs Levodopa as Initial Treatment for Parkinson Disease: A 4-Year Randomized Controlled Trial. Arch. Neurol. 61, 1044–1053. https://doi.org/10.1001/archneur.61.7.1044
Hu, L., Huang, Y.-J., Wei, Y.-D., Li, T., Ke, W., Chen, G.-H., Dong, M.-X., 2024. Plasma metabolomics profiles indicate sex differences of lipid metabolism in patients with Parkinson's disease. Sci. Rep. 14, 31262. https://doi.org/10.1038/s41598-024-82674-3
Huang, P., Yang, X.-D., Chen, S.-D., Xiao, Q., 2015. The association between Parkinson's disease and melanoma: a systematic review and meta-analysis. Transl. Neurodegener. 4, 21. https://doi.org/10.1186/s40035-015-0044-y
Huang, Y., Huang, C., Zhang, Q., Wu, W., Sun, J., 2021. Serum BDNF discriminates Parkinson's disease patients with depression from without depression and reflect motor severity and gender differences. J. Neurol. 268, 1411–1418. https://doi.org/10.1007/s00415-020-10299-3
Huber, J., Foidart, J.M., Wuttke, W., Merki-Feld, G.S., The, H.S., Gerlinger, C., Schellschrmdt, I., Heithecker, R., 2000. Efficacy and tolerability of a monophasic oral contraceptive containing ethinylestradiol and drospirenone. Eur. J. Contracept. Reprod. Health Care 5, 25–34. https://doi.org/10.1080/13625180008500375
Ikeda-Matsuo, Y., Miyata, H., Mizoguchi, T., Ohama, E., Naito, Y., Uematsu, S., Akira, S., Sasaki, Y., Tanabe, M., 2019. Microsomal prostaglandin E synthase-1 is a critical

factor in dopaminergic neurodegeneration in Parkinson's disease. Neurobiol. Dis. 124, 81–92. https://doi.org/10.1016/j.nbd.2018.11.004
Iwaki, H., Blauwendraat, C., Leonard, H.L., Makarious, M.B., Kim, J.J., Liu, G., Maple-Grødem, J., Corvol, J.-C., Pihlstrøm, L., van Nimwegen, M., Smolensky, L., Amondikar, N., Hutten, S.J., Frasier, M., Nguyen, K.-D.H., Rick, J., Eberly, S., Faghri, F., Auinger, P., Scott, K.M., Wijeyekoon, R., Van Deerlin, V.M., Hernandez, D.G., Gibbs, R.J., Day-Williams, A.G., Brice, A., Alves, G., Noyce, A.J., Tysnes, O.-B., Evans, J.R., Breen, D.P., Estrada, K., Wegel, C.E., Danjou, F., Simon, D.K., Andreassen, O.A., Ravina, B., Toft, M., Heutink, P., Bloem, B.R., Weintraub, D., Barker, R.A., Williams-Gray, C.H., van de Warrenburg, B.P., Van Hilten, J.J., Scherzer, C.R., Singleton, A.B., Nalls, M.A., 2021a. Differences in the Presentation and Progression of Parkinson's Disease by Sex. Mov. Disord. Off. J. Mov. Disord. Soc. 36, 106–117. https://doi.org/10.1002/mds.28312
Iwaki, H., Blauwendraat, C., Leonard, H.L., Makarious, M.B., Kim, J.J., Liu, G., Maple-Grødem, J., Corvol, J.-C., Pihlstrøm, L., van Nimwegen, M., Smolensky, L., Amondikar, N., Hutten, S.J., Frasier, M., Nguyen, K.-D.H., Rick, J., Eberly, S., Faghri, F., Auinger, P., Scott, K.M., Wijeyekoon, R., Van Deerlin, V.M., Hernandez, D.G., Gibbs, R.J., Day-Williams, A.G., Brice, A., Alves, G., Noyce, A.J., Tysnes, O.-B., Evans, J.R., Breen, D.P., Estrada, K., Wegel, C.E., Danjou, F., Simon, D.K., Andreassen, O.A., Ravina, B., Toft, M., Heutink, P., Bloem, B.R., Weintraub, D., Barker, R.A., Williams-Gray, C.H., van de Warrenburg, B.P., Van Hilten, J.J., Scherzer, C.R., Singleton, A.B., Nalls, M.A., 2021b. Differences in the Presentation and Progression of Parkinson's Disease by Sex. Mov. Disord. 36, 106–117. https://doi.org/10.1002/mds.28312
Jaber, S.M., Bordt, E.A., Bhatt, N.M., Lewis, D.M., Gerecht, S., Fiskum, G., Polster, B.M., 2018. Sex differences in the mitochondrial bioenergetics of astrocytes but not microglia at a physiologically relevant brain oxygen tension. Neurochem. Int. 117, 82–90. https://doi.org/10.1016/j.neuint.2017.09.003
Jankovic, J., 2008. Parkinson's disease: clinical features and diagnosis. J. Neurol. Neurosurg. Psychiatry 79, 368–376. https://doi.org/10.1136/jnnp.2007.131045
Jarras, H., Bourque, M., Poirier, A.-A., Morissette, M., Coulombe, K., Di Paolo, T., Soulet, D., 2020. Neuroprotection and immunomodulation of progesterone in the gut of a mouse model of Parkinson's disease. J. Neuroendocrinol. 32, e12782. https://doi.org/10.1111/jne.12782
Jesús, S., Pérez, I., Cáceres-Redondo, M.T., Carrillo, F., Carballo, M., Gómez-Garre, P., Mir, P., 2013. Low serum uric acid concentration in Parkinson's disease in southern Spain. Eur. J. Neurol. 20, 208–210. https://doi.org/10.1111/j.1468-1331.2012.03745.x
Jones, J.D., Rahmani, E., Garcia, E., Jacobs, J.P., 2020. Gastrointestinal symptoms are predictive of trajectories of cognitive functioning in de novo Parkinson's disease. Parkinsonism Relat. Disord. 72, 7–12. https://doi.org/10.1016/j.parkreldis.2020.01.009
Jost, S.T., Strobel, L., Rizos, A., Loehrer, P.A., Ashkan, K., Evans, J., Rosenkranz, F., Barbe, M.T., Fink, G.R., Franklin, J., Sauerbier, A., Nimsky, C., Sattari, A., Ray Chaudhuri, K., Antonini, A., Timmermann, L., Martinez-Martin, P., Silverdale, M., Kalbe, E., Visser-Vandewalle, V., Dafsari, H.S., 2022. Gender gap in deep brain

stimulation for Parkinson's disease. Npj Park. Dis. 8, 47. https://doi.org/10.1038/s41531-022-00305-y
Judkins, T.C., Dennis-Wall, J.C., Sims, S.M., Colee, J., Langkamp-Henken, B., 2020. Stool frequency and form and gastrointestinal symptoms differ by day of the menstrual cycle in healthy adult women taking oral contraceptives: a prospective observational study. BMC Womens Health 20, 136. https://doi.org/10.1186/s12905-020-01000-x
Jung, H., Choung, R.S., Locke, G.R., Schleck, C.D., Zinsmeister, A.R., Szarka, L.A., Mullan, B., Talley, N.J., 2009. The incidence, prevalence and outcomes of patients with gastroparesis in Olmsted County, Minnesota from 1996 to 2006. Gastroenterology 136, 1225–1233. https://doi.org/10.1053/j.gastro.2008.12.047
Kang, K.W., Choi, S.-M., Kim, B.C., 2022. Gender differences in motor and non-motor symptoms in early Parkinson disease. Medicine (Baltimore) 101, e28643. https://doi.org/10.1097/MD.0000000000028643
Katz, H.I., Kempers, S., Akin, M.D., Dunlap, F., Whiting, D., Norbart, T.C.J., 2000. Effect of a desogestrel-containing oral contraceptive on the skin. Eur. J. Contracept. Reprod. Health Care 5, 248–255. https://doi.org/10.1080/13625180008500411
Kim, Y.J., Soto, M., Branigan, G.L., Rodgers, K., Brinton, R.D., 2021. Association between menopausal hormone therapy and risk of neurodegenerative diseases: Implications for precision hormone therapy. Alzheimers Dement. N. Y. N 7, e12174. https://doi.org/10.1002/trc2.12174
Kip, K.E., Cohen, F., Cole, S.R., Wilhelmus, K.R., Patrick, D.L., Blair, R.C., Beck, R.W., 2001. Recall bias in a prospective cohort study of acute time-varying exposures: Example from the herpetic eye disease study. J. Clin. Epidemiol. 54, 482–487. https://doi.org/10.1016/S0895-4356(00)00310-3
Koharudin, L.M.I., Liu, H., Di Maio, R., Kodali, R.B., Graham, S.H., Gronenborn, A.M., 2010. Cyclopentenone prostaglandin-induced unfolding and aggregation of the Parkinson disease-associated UCH-L1. Proc. Natl. Acad. Sci. 107, 6835–6840. https://doi.org/10.1073/pnas.1002295107
Kompoliti, K., Adler, C.H., Raman, R., Pincus, J.H., Leibowitz, M.T., Ferry, J.J., Blasucci, L., Caviness, J.N., Leurgans, S., Chase, W.M., Yones, L.C., Tan, E., Carvey, P., Goetz, C.G., 2002. Gender and pramipexole effects on levodopa pharmacokinetics and pharmacodynamics. Neurology 58, 1418–1422. https://doi.org/10.1212/WNL.58.9.1418
Kompoliti, K., Comella, C.L., Jaglin, J.A., Leurgans, S., Raman, R., Goetz, C.G., 2000. Menstrual-related changes in motoric function in women with Parkinson's disease. Neurology 55, 1572–1575. https://doi.org/10.1212/WNL.55.10.1572
Kostic, V., Przedborski, S., Flaster, E., Sternic, N., 1991. Early development of levodopa-induced dyskinesias and response fluctuations in young-onset Parkinson's disease. Neurology 41, 202–202. https://doi.org/10.1212/WNL.41.2_Part_1.202
Kovács, M., Makkos, A., Aschermann, Z., Janszky, J., Komoly, S., Weintraut, R., Karádi, K., Kovács, N., 2016. Impact of Sex on the Nonmotor Symptoms and the Health-Related Quality of Life in Parkinson's Disease. Park. Dis. 2016, 7951840. https://doi.org/10.1155/2016/7951840
Kranick, S.M., Mowry, E.M., Colcher, A., Horn, S., Golbe, L.I., 2010. Movement disorders and pregnancy: a review of the literature. Mov. Disord. Off. J. Mov. Disord. Soc. 25, 665–671. https://doi.org/10.1002/mds.23071

Kumagai, T., Nagayama, H., Ota, T., Nishiyama, Y., Mishina, M., Ueda, M., 2014. Sex Differences in the Pharmacokinetics of Levodopa in... : Clinical Neuropharmacology.

Kumar, N., Van Gerpen, J.A., Bower, J.H., Ahlskog, J.E., 2005. Levodopa-dyskinesia incidence by age of Parkinson's disease onset. Mov. Disord. 20, 342–344. https://doi.org/10.1002/mds.20360

Küppers, E., Ivanova, T., Karolczak, M., Beyer, C., 2000. Estrogen: a multifunctional messenger to nigrostriatal dopaminergic neurons. J. Neurocytol. 29, 375–385. https://doi.org/10.1023/a:1007165307652

Kurlawala, Z., Shadowen, P.H., McMillan, J.D., Beverly, L.J., Friedland, R.P., 2021. Progression of Nonmotor Symptoms in Parkinson's Disease by Sex and Motor Laterality. Park. Dis. 2021, 8898887. https://doi.org/10.1155/2021/8898887

Lakshminarayana, R., Wang, D., Burn, D., Chaudhuri, K.R., Galtrey, C., Guzman, N.V., Hellman, B., Ben James, Pal, S., Stamford, J., Steiger, M., Stott, R.W., Teo, J., Barker, R.A., Wang, E., Bloem, B.R., van der Eijk, M., Rochester, L., Williams, A., 2017. Using a smartphone-based self-management platform to support medication adherence and clinical consultation in Parkinson's disease. Npj Park. Dis. 3, 2. https://doi.org/10.1038/s41531-016-0003-z

Lamichhane, D., Narayanan, N.S., Gonzalez-Alegre, P., 2014. Two cases of pregnancy in Parkinson's disease. Parkinsonism Relat. Disord. 20, 239–240. https://doi.org/10.1016/j.parkreldis.2013.10.013

Lau, B., Meier, N., Serra, G., Czernecki, V., Schuepbach, M., Navarro, S., Cornu, P., Grabli, D., Agid, Y., Vidailhet, M., Karachi, C., Welter, M.-L., 2019. Axial symptoms predict mortality in patients with Parkinson disease and subthalamic stimulation. Neurology 92. https://doi.org/10.1212/WNL.0000000000007562

LeDoux, M.S., 2005. Thermoregulatory Dysfunction, in: Pfeiffer, R.F., Bodis-Wollner, I. (Eds.), Parkinson's Disease and Nonmotor Dysfunction. Humana Press, Totowa, NJ, pp. 159–172. https://doi.org/10.1385/1-59259-859-5:159

Lee, J., Suh, Y., Kim, E., Yoo, S., Kim, Y., 2024. A Mobile App for Comprehensive Symptom Management in People With Parkinson's Disease: A Pilot Usability Study. Comput. Inform. Nurs. CIN 42. https://doi.org/10.1097/CIN.0000000000001089

Lee, J., Yeom, I., Chung, M.L., Kim, Y., Yoo, S., Kim, E., 2022. Use of Mobile Apps for Self-care in People With Parkinson Disease: Systematic Review. JMIR MHealth UHealth 10, e33944. https://doi.org/10.2196/33944

Lee, Y., Park, Y.-H., Lee, J.J., Sohn, Y.H., Lee, J.-M., Lee, P.H., 2018. Gender-specific effect of uric acid on resting-state functional networks in de novo Parkinson's disease. Parkinsonism Relat. Disord. 52, 49–54. https://doi.org/10.1016/j.parkreldis.2018.03.023

Leentjens, A.F.G., Van den Akker, M., Metsemakers, J.F.M., Lousberg, R., Verhey, F.R.J., 2003. Higher incidence of depression preceding the onset of Parkinson's disease: a register study. Mov. Disord. Off. J. Mov. Disord. Soc. 18, 414–418. https://doi.org/10.1002/mds.10387

Lejri, I., Grimm, A., Eckert, A., 2018. Mitochondria, Estrogen and Female Brain Aging. Front. Aging Neurosci. 10. https://doi.org/10.3389/fnagi.2018.00124

Li, H., Gibson, E.A., Jukic, A.M.Z., Baird, D.D., Wilcox, A.J., Curry, C.L., Fischer-Colbrie, T., Onnela, J.-P., Williams, M.A., Hauser, R., Coull, B.A., Mahalingaiah, S., 2023.

Menstrual cycle length variation by demographic characteristics from the Apple Women's Health Study. Npj Digit. Med. 6, 1–11. https://doi.org/10.1038/s41746-023-00848-1

Lin, C.-Y., Chang, T.-Y., Chang, M.-H., 2022. Dysosmia Is a Predictor of Motor Function and Quality of Life in Patients with Parkinson's Disease. J. Pers. Med. 12, 754. https://doi.org/10.3390/jpm12050754

Lipsmeier, F., Taylor, K.I., Postuma, R.B., Volkova-Volkmar, E., Kilchenmann, T., Mollenhauer, B., Bamdadian, A., Popp, W.L., Cheng, W.-Y., Zhang, Y.-P., Wolf, D., Schjodt-Eriksen, J., Boulay, A., Svoboda, H., Zago, W., Pagano, G., Lindemann, M., 2022. Reliability and validity of the Roche PD Mobile Application for remote monitoring of early Parkinson's disease. Sci. Rep. 12, 12081. https://doi.org/10.1038/s41598-022-15874-4

Litim, N., Morissette, M., Di Paolo, T., 2017. Effects of progesterone administered after MPTP on dopaminergic neurons of male mice. Neuropharmacology 117, 209–218. https://doi.org/10.1016/j.neuropharm.2017.02.007

Littleton-Kearney, M.T., Ostrowski, N.L., Cox, D.A., Rossberg, M.I., Hurn, P.D., 2002. Selective Estrogen Receptor Modulators: Tissue Actions and Potential for CNS Protection. CNS Drug Rev. 8, 309–330. https://doi.org/10.1111/j.1527-3458.2002.tb00230.x

Liu, H., Li, J., Wang, X., Huang, J., Wang, T., Lin, Z., Xiong, N., 2022. Excessive Daytime Sleepiness in Parkinson's Disease. Nat. Sci. Sleep 14, 1589–1609. https://doi.org/10.2147/NSS.S375098

Liu, R., Baird, D., Park, Y., Freedman, N.D., Huang, X., Hollenbeck, A., Blair, A., Chen, H., 2014. Female reproductive factors, menopausal hormone use, and Parkinson's disease. Mov. Disord. Off. J. Mov. Disord. Soc. 29, 889–896. https://doi.org/10.1002/mds.25771

Loebstein, R., Lalkin, A., Koren, G., 1997. Pharmacokinetic Changes During Pregnancy and Their Clinical Relevance. Clin. Pharmacokinet. 33, 328–343. https://doi.org/10.2165/00003088-199733050-00002

Luchetti, S., Bossers, K., Frajese, G.V., Swaab, D.F., 2010. Neurosteroid Biosynthetic Pathway Changes in Substantia Nigra and Caudate Nucleus in Parkinson's Disease. Brain Pathol. 20, 945–951. https://doi.org/10.1111/j.1750-3639.2010.00396.x

Luchetti, S., Liere, P., Pianos, A., Verwer, R.W.H., Sluiter, A., Huitinga, I., Schumacher, M., Swaab, D.F., Mason, M.R.J., 2023. Disease stage-dependent changes in brain levels and neuroprotective effects of neuroactive steroids in Parkinson's disease. Neurobiol. Dis. 183, 106169. https://doi.org/10.1016/j.nbd.2023.106169

Lyons, K.E., Pahwa, R., 2007. Electronic motor function diary for patients with Parkinson's disease: A feasibility study. Parkinsonism Relat. Disord. 13, 304–307. https://doi.org/10.1016/j.parkreldis.2006.06.008

Maas, B.R., Göttgens, I., Tijsse Klasen, H.P.S., Kapelle, W.M., Radder, D.L.M., Bloem, B.R., Post, B., de Vries, N.M., Darweesh, S.K.L., 2024. Age and gender differences in non-motor symptoms in people with Parkinson's disease. Front. Neurol. 15, 1339716. https://doi.org/10.3389/fneur.2024.1339716

Mahtani, K., Spencer, E.A., Brassey, J., Heneghan, C., 2018. Catalogue of bias: observer bias. BMJ Evid.-Based Med. 23, 23–24. https://doi.org/10.1136/ebmed-2017-110884

Mao, C., Yang, Y., Chen, Ju-ping, Wang, F., Chen, Jing, Zhang, J., Zhang, H., Zhuang, S., Xiong, Y., Gu, C., Yuan, W., Huang, J., Fay, A., Zhong, C., Liu, C., 2018. Poor nighttime sleep is positively associated with dyskinesia in Parkinson's disease patients. Parkinsonism Relat. Disord. 48, 68–73. https://doi.org/10.1016/j.parkreldis.2017.12.022

Mariani, S., Ventriglia, M., Simonelli, I., Bucossi, S., Siotto, M., Donno, S., Vernieri, F., Squitti, R., 2016. Association between sex, systemic iron variation and probability of Parkinson's disease. Int. J. Neurosci. 126, 354–360. https://doi.org/10.3109/00207454.2015.1020113

Marin, R., Casañas, V., Pérez, J.A., Fabelo, N., Fernandez, C.E., Diaz, M., 2013. Oestrogens as modulators of neuronal signalosomes and brain lipid homeostasis related to protection against neurodegeneration. J. Neuroendocrinol. 25, 1104–1115. https://doi.org/10.1111/jne.12068

Marin, R., Diaz, M., 2018. Estrogen Interactions With Lipid Rafts Related to Neuroprotection. Impact of Brain Ageing and Menopause. Front. Neurosci. 12, 128. https://doi.org/10.3389/fnins.2018.00128

Marrinan, S., Emmanuel, A.V., Burn, D.J., 2014. Delayed gastric emptying in Parkinson's disease. Mov. Disord. Off. J. Mov. Disord. Soc. 29, 23–32. https://doi.org/10.1002/mds.25708

Martignoni, E., Nappi, R.E., Citterio, A., Calandrella, D., Corengia, E., Fignon, A., Zangaglia, R., Riboldazzi, G., Pacchetti, C., Nappi, G., 2002. Parkinson's disease and reproductive life events. Neurol. Sci. 23, s85–s86. https://doi.org/10.1007/s100720200082

Martignoni, E., Nappi, R.E., Citterio, A., Calandrella, D., Zangaglia, R., Mancini, F., Corengia, E., Riboldazzi, G., Polatti, F., Nappi, G., 2003. Reproductive life milestones in women with Parkinson's disease. Funct. Neurol.

Martinelli, P., Contin, M., Scaglione, C., Riva, R., Albani, F., Baruzzi, A., 2003. Levodopa pharmacokinetics anddyskinesias: are there sex-related differences? Neurol. Sci. 24, 192–193. https://doi.org/10.1007/s10072-003-0125-z

Martinez-Martin, P., Falup Pecurariu, C., Odin, P., van Hilten, J.J., Antonini, A., Rojo-Abuin, J.M., Borges, V., Trenkwalder, C., Aarsland, D., Brooks, D.J., Ray Chaudhuri, K., 2012. Gender-related differences in the burden of non-motor symptoms in Parkinson's disease. J. Neurol. 259, 1639–1647. https://doi.org/10.1007/s00415-011-6392-3

Martinez-Ramirez, D., Giugni, J., Vedam-Mai, V., Shukla, A.W., Malaty, I.A., McFarland, N.R., Rodriguez, R.L., Foote, K.D., Okun, M.S., 2014. The "Brittle Response" to Parkinson's Disease Medications: Characterization and Response to Deep Brain Stimulation. PLoS ONE 9, e94856. https://doi.org/10.1371/journal.pone.0094856

Martin-Jiménez, C., Gaitán-Vaca, D.M., Areiza, N., Echeverria, V., Ashraf, G.M., González, J., Sahebkar, A., Garcia-Segura, L.M., Barreto, G.E., 2018. Astrocytes Mediate Protective Actions of Estrogenic Compounds after Traumatic Brain Injury. Neuroendocrinology 108, 142–160. https://doi.org/10.1159/000495078

Mattammal, M.B., Strong, R., Lakshmi, V.M., Chung, H.D., Stephenson, A.H., 1995. Prostaglandin H Synthetase-Mediated Metabolism of Dopamine: Implication for Parkinson's Disease. J. Neurochem. 64, 1645–1654. https://doi.org/10.1046/j.1471-4159.1995.64041645.x

McConnell, M.V., Shcherbina, A., Pavlovic, A., Homburger, J.R., Goldfeder, R.L., Waggot, D., Cho, M.K., Rosenberger, M.E., Haskell, W.L., Myers, J., Champagne, M.A., Mignot, E., Landray, M., Tarassenko, L., Harrington, R.A., Yeung, A.C., Ashley, E.A., 2017. Feasibility of Obtaining Measures of Lifestyle From a Smartphone App: The MyHeart Counts Cardiovascular Health Study. JAMA Cardiol. 2, 67. https://doi.org/10.1001/jamacardio.2016.4395

McFarland, N.R., Burdett, T., Desjardins, C., Frosch, M.P., Schwarzschild, M.A., 2013. Postmortem Brain Levels of Urate and Precursors in Parkinson disease and Related Disorders. Neurodegener. Dis. 12, 10.1159/000346370. https://doi.org/10.1159/000346370

Mehanna, R., Jankovic, J., 2019. Young-onset Parkinson's disease: Its unique features and their impact on quality of life. Parkinsonism Relat. Disord. 65, 39–48. https://doi.org/10.1016/j.parkreldis.2019.06.001

Meindorfner, C., Körner, Y., Möller, J.C., Stiasny-Kolster, K., Oertel, W.H., Krüger, H.-P., 2005. Driving in Parkinson's disease: Mobility, accidents, and sudden onset of sleep at the wheel. Mov. Disord. 20, 832–842. https://doi.org/10.1002/mds.20412

Melcangi, R.C., Caruso, D., Levandis, G., Abbiati, F., Armentero, M.-T., Blandini, F., 2012. Modifications of Neuroactive Steroid Levels in an Experimental Model of Nigrostriatal Degeneration: Potential Relevance to the Pathophysiology of Parkinson's Disease. J. Mol. Neurosci. 46, 177–183. https://doi.org/10.1007/s12031-011-9570-y

Mendes-Oliveira, J., Lopes Campos, F., Videira, R.A., Baltazar, G., 2017. GPER activation is effective in protecting against inflammation-induced nigral dopaminergic loss and motor function impairment. Brain. Behav. Immun. 64, 296–307. https://doi.org/10.1016/j.bbi.2017.04.016

Miller, D.B., Ali, S.F., O'callaghan, J.P., Laws, S.C., 1998. The Impact of Gender and Estrogen on Striatal Dopaminergic Neurotoxicity. Ann. N. Y. Acad. Sci. 844, 153–165. https://doi.org/10.1111/j.1749-6632.1998.tb08230.x

Mischley, L.K., Lau, R.C., Bennett, R.D., 2017. Role of Diet and Nutritional Supplements in Parkinson's Disease Progression. Oxid. Med. Cell. Longev. 2017, 6405278. https://doi.org/10.1155/2017/6405278

Moisan, F., Kab, S., Mohamed, F., Canonico, M., Le Guern, M., Quintin, C., Carcaillon, L., Nicolau, J., Duport, N., Singh-Manoux, A., Boussac-Zarebska, M., Elbaz, A., 2016. Parkinson disease male-to-female ratios increase with age: French nationwide study and meta-analysis. J. Neurol. Neurosurg. Psychiatry 87, 952–957. https://doi.org/10.1136/jnnp-2015-312283

Mokosáková, M., Krsková, L., Hlavacka, F., Zeman, M., 2014. Diurnal changes of the postural control in young women without and with hormonal contraceptive treatment. Neuro Endocrinol. Lett. 35, 230–235.

Morale, M.C., Serra, P.A., L'Episcopo, F., Tirolo, C., Caniglia, S., Testa, N., Gennuso, F., Giaquinta, G., Rocchitta, G., Desole, M.S., Miele, E., Marchetti, B., 2006. Estrogen, neuroinflammation and neuroprotection in Parkinson's disease: Glia

dictates resistance versus vulnerability to neurodegeneration. Neuroscience 138, 869–878. https://doi.org/10.1016/j.neuroscience.2005.07.060
Morissette, M., Di Paolo, T., 1993. Sex and Estrous Cycle Variations of Rat Striatal Dopamine Uptake Sites. Neuroendocrinology 58, 16–22. https://doi.org/10.1159/000126507
Mulcachy, Dr.T., 2025. Femtech in Ireland: The Case for Prioritising Women's Health Research and Innovation. Health Innov. Hub Irel.
Müller, T., Erdmann, C., Bremen, D., Schmidt, W.E., Muhlack, S., Woitalla, D., Goetze, O., 2006. Impact of Gastric Emptying on Levodopa Pharmacokinetics in Parkinson Disease Patients. Clin. Neuropharmacol. 29, 61.
Mullin, R.L., Chaudhuri, K.R., Andrews, T.C., Martin, A., Gay, S., White, C.M., 2018. A study investigating the experience of working for people with Parkinson's and the factors that influence workplace success. Disabil. Rehabil. 40, 2032–2039. https://doi.org/10.1080/09638288.2017.1323237
Murphy, R., Tubridy, N., Kevelighan, H., O'Riordan, S., 2013. Parkinson's disease: how is employment affected? Ir. J. Med. Sci. 182, 415–419. https://doi.org/10.1007/s11845-013-0902-5
Murray, H.E., Pillai, A.V., McArthur, S.R., Razvi, N., Datla, K.P., Dexter, D.T., Gillies, G.E., 2003. Dose- and sex-dependent effects of the neurotoxin 6-hydroxydopamine on the nigrostriatal dopaminergic pathway of adult rats: differential actions of estrogen in males and females. Neuroscience 116, 213–222. https://doi.org/10.1016/S0306-4522(02)00578-X
My Moves Matter [WWW Document], 2026a. . My Moves Matter. URL https://www.mymovesmatter.com/get-the-app (accessed 1.31.26).
My Moves Matter [WWW Document], 2026b. . Park. UK Tech Guide. URL https://techguide.parkinsons.org.uk/catalogue/my-moves-matter (accessed 1.31.26).
Nabizadeh, F., Seyedmirzaei, H., Rafiei, N., Maryam Vafaei, S., Shekouh, D., Mehrtabar, E., Mirzaaghazadeh, E., Mirzaasgari, Z., 2024. Global prevalence and incidence of Young Onset Parkinson's disease: A systematic review and meta-analysis. J. Clin. Neurosci. 125, 59–67. https://doi.org/10.1016/j.jocn.2024.05.015
Nebe, A., Ebersbach, G., 2009. Pain intensity on and off levodopa in patients with Parkinson's disease. Mov. Disord. 24, 1233–1237. https://doi.org/10.1002/mds.22546
Neugebauer, R., Ng, S., 1990. Differential recall as a source of bias in epidemiologic research. J. Clin. Epidemiol. 43, 1337–1341. https://doi.org/10.1016/0895-4356(90)90100-4
Niemann, N., Billnitzer, A., Jankovic, J., 2021. Parkinson's disease and skin. Parkinsonism Relat. Disord. 82, 61–76. https://doi.org/10.1016/j.parkreldis.2020.11.017
Nolan, A., Barrett, A., 2019. Working Beyond age 65 in Ireland. J. Popul. Ageing 12, 299–326. https://doi.org/10.1007/s12062-019-09249-3
Nyholm, D., Karlsson, E., Lundberg, M., Askmark, H., 2010. Large differences in levodopa dose requirement in Parkinson's disease: men use higher doses than women. Eur. J. Neurol. 17, 260–266. https://doi.org/10.1111/j.1468-1331.2009.02866.x

Olivola, S., Xodo, S., Olivola, E., Cecchini, F., Londero, A.P., Driul, L., 2020. Parkinson's Disease in Pregnancy: A Case Report and Review of the Literature. Front. Neurol. 10.

Orr-Urtreger, A., Shifrin, C., Rozovski, U., Rosner, S., Bercovich, D., Gurevich, T., Yagev-More, H., Bar-Shira, A., Giladi, N., 2007. The LRRK2 G2019S mutation in Ashkenazi Jews with Parkinson disease. Neurology 69, 1595–1602. https://doi.org/10.1212/01.wnl.0000277637.33328.d8

Orzołek, I., Sobieraj, J., Domagała-Kulawik, J., 2022. Estrogens, Cancer and Immunity. Cancers 14, 2265. https://doi.org/10.3390/cancers14092265

Papapetropoulos, S., Adi, N., Ellul, J., Argyriou, A.A., Chroni, E., 2007. A Prospective Study of Familial versus Sporadic Parkinson's Disease. Neurodegener. Dis. 4, 424–427. https://doi.org/10.1159/000107702

Parashos, S.A., Bloem, B.R., Browner, N.M., Giladi, N., Gurevich, T., Hausdorff, J.M., He, Y., Lyons, K.E., Mari, Z., Morgan, J.C., Post, B., Schmidt, P.N., Wielinski, C.L., Wu, S.S., 2018. What predicts falls in Parkinson disease? Neurol. Clin. Pract. 8, 214–222. https://doi.org/10.1212/CPJ.0000000000000461

PD Avengers Championing 'Women and PD' [WWW Document], 2021. . PD Aveng. URL https://www.pdavengers.com/blog/women-and-pd (accessed 11.15.23).

Pedersen, A.L., Nelson, L.H., Saldanha, C.J., 2016. Centrally Synthesized Estradiol Is a Potent Anti-Inflammatory in the Injured Zebra Finch Brain. Endocrinology 157, 2041–2051. https://doi.org/10.1210/en.2015-1991

Pellecchia, M.T., Picillo, M., Russillo, M.C., Andreozzi, V., Oliveros, C., Cattaneo, C., 2023. The effects of safinamide according to gender in Chinese parkinsonian patients. Sci. Rep. 13, 20632. https://doi.org/10.1038/s41598-023-48067-8

Pellecchia, M.T., Picillo, M., Russillo, M.C., De Pandis, M.F., Bonizzoni, E., Marjanovic, I., Cattaneo, C., 2021. Efficacy of Safinamide and Gender Differences During Routine Clinical Practice. Front. Neurol. 12. https://doi.org/10.3389/fneur.2021.756304

Pfeiffer, R.F., Isaacson, S.H., Pahwa, R., 2020. Clinical implications of gastric complications on levodopa treatment in Parkinson's disease. Parkinsonism Relat. Disord. 76, 63–71. https://doi.org/10.1016/j.parkreldis.2020.05.001

Picillo, M., Palladino, R., Erro, R., Alfano, R., Colosimo, C., Marconi, R., Antonini, A., Barone, P., 2021. The PRIAMO study: age- and sex-related relationship between prodromal constipation and disease phenotype in early Parkinson's disease. J. Neurol. 268, 448–454. https://doi.org/10.1007/s00415-020-10156-3

Picillo, M., Palladino, R., Moccia, M., Erro, R., Amboni, M., Vitale, C., Barone, P., Pellecchia, M.T., 2016. Gender and non motor fluctuations in Parkinson's disease: A prospective study. Parkinsonism Relat. Disord. 27, 89–92. https://doi.org/10.1016/j.parkreldis.2016.04.001

Pierre, S.-R., Lemmens, M.A., Figueiredo-Pereira, M.E., 2009. Subchronic infusion of the product of inflammation prostaglandin J2 models sporadic Parkinson's disease in mice. J. Neuroinflammation 6, 18. https://doi.org/10.1186/1742-2094-6-18

Popat, R.A., Van Den Eeden, S.K., Tanner, C.M., McGuire, V., Bernstein, A.L., Bloch, D.A., Leimpeter, A., Nelson, L.M., 2005. Effect of reproductive factors and postmenopausal hormone use on the risk of Parkinson disease. Neurology 65, 383–390. https://doi.org/10.1212/01.wnl.0000171344.87802.94

Post, B., Muslimovic, D., van Geloven, N., Speelman, J.D., Schmand, B., de Haan, R.J., Group, on behalf of the C., 2011. Progression and prognostic factors of motor impairment, disability and quality of life in newly diagnosed Parkinson's disease. Mov. Disord. 26, 449–456. https://doi.org/10.1002/mds.23467

Powell, A.M., Chan, W.Y., Alvin, P., Litt, I.F., 1985. Menstrual-PGF2 alpha, PGE2 and TXA2 in normal and dysmenorrheic women and their temporal relationship to dysmenorrhea. Prostaglandins 29, 273–290. https://doi.org/10.1016/0090-6980(85)90208-4

Prakash, K.M., Nadkarni, N.V., Lye, W.-K., Yong, M.-H., Tan, E.-K., 2016. The impact of non-motor symptoms on the quality of life of Parkinson's disease patients: a longitudinal study. Eur. J. Neurol. 23, 854–860. https://doi.org/10.1111/ene.12950

Pringsheim, T., Day, G.S., Smith, D.B., Rae-Grant, A., Licking, N., Armstrong, M.J., de Bie, R.M.A., Roze, E., Miyasaki, J.M., Hauser, R.A., Espay, A.J., Martello, J.P., Gurwell, J.A., Billinghurst, L., Sullivan, K., Fitts, M.S., Cothros, N., Hall, D.A., Rafferty, M., Hagerbrant, L., Hastings, T., O'Brien, M.D., Silsbee, H., Gronseth, G., Lang, A.E., 2021. Dopaminergic Therapy for Motor Symptoms in Early Parkinson Disease Practice Guideline Summary. Neurology 97, 942–957. https://doi.org/10.1212/WNL.0000000000012868

Pye, R.J., Meyrick, G., Pye, M.J., Burton, J.L., 1977. Effect of oral contraceptives on sebum excretion rate. Br. Med. J. 2, 1581–1582. https://doi.org/10.1136/bmj.2.6102.1581

Quinn, N., Critchley, P., Marsden, C.D., 1987. Young onset Parkinson's disease. Mov. Disord. 2, 73–91. https://doi.org/10.1002/mds.870020201

Quinn, N.P., Marsden, C.D., 1986. Menstrual-related fluctuations in Parkinson's disease. Mov. Disord. 1, 85–87. https://doi.org/10.1002/mds.870010112

Ramirez, A.D., Liu, X., Menniti, F.S., 2003. Repeated Estradiol Treatment Prevents MPTP-Induced Dopamine Depletion in Male Mice. Neuroendocrinology 77, 223–231. https://doi.org/10.1159/000070277

Rao, S.C., Li, Y., Lapin, B., Pattipati, S., Ghosh Galvelis, K., Naito, A., Gutierrez, N., Leal, T.P., Salim, A., Salles, P.A., De Leon, M., Mata, I.F., 2023a. Association of women-specific health factors in the severity of Parkinson's disease. Npj Park. Dis. 9, 86. https://doi.org/10.1038/s41531-023-00524-x

Rao, S.C., Li, Y., Lapin, B., Pattipati, S., Ghosh Galvelis, K., Naito, A., Gutierrez, N., Leal, T.P., Salim, A., Salles, P.A., De Leon, M., Mata, I.F., 2023b. Association of women-specific health factors in the severity of Parkinson's disease. Npj Park. Dis. 9, 1–16. https://doi.org/10.1038/s41531-023-00524-x

Rivest, R., Falardeau, P., Di Paolo, T., 1995. Brain dopamine transporter: gender differences and effect of chronic haloperidol. Brain Res. 692, 269–272. https://doi.org/10.1016/0006-8993(95)00611-S

Roediger, J., Artusi, C.A., Romagnolo, A., Boyne, P., Zibetti, M., Lopiano, L., Espay, A.J., Fasano, A., Merola, A., 2019. Effect of subthalamic deep brain stimulation on posture in Parkinson's disease: A blind computerized analysis. Parkinsonism Relat. Disord. 62, 122–127. https://doi.org/10.1016/j.parkreldis.2019.01.003

Ross, G.W., Petrovitch, H., Abbott, R.D., Tanner, C.M., Popper, J., Masaki, K., Launer, L., White, L.R., 2008. Association of olfactory dysfunction with risk for future

Parkinson's disease. Ann. Neurol. 63, 167–173. https://doi.org/10.1002/ana.21291
Rugbjerg, K., Christensen, J., Tjønneland, A., Olsen, J.H., 2013. Exposure to estrogen and women's risk for Parkinson's disease: A prospective cohort study in Denmark. Parkinsonism Relat. Disord. 19, 457–460. https://doi.org/10.1016/j.parkreldis.2013.01.008
Sampaio, T.F., dos Santos, E.U.D., de Lima, G.D.C., dos Anjos, R.S.G., da Silva, R.C., Asano, A.G.C., Asano, N.M.J., Crovella, S., de Souza, P.R.E., 2018. MAO-B and COMT Genetic Variations Associated With Levodopa Treatment Response in Patients With Parkinson's Disease. J. Clin. Pharmacol. 58, 920–926. https://doi.org/10.1002/jcph.1096
Sandyk, R., 1989. Estrogens and the Pathophysiology of Parkinson's Disease. Int. J. Neurosci. 45, 119–122. https://doi.org/10.3109/00207458908986223
Santos García, D., de Deus Fonticoba, T., Suárez Castro, E., Borrué, C., Mata, M., Solano Vila, B., Cots Foraster, A., Álvarez Sauco, M., Rodríguez Pérez, A.B., Vela, L., Macías, Y., Escalante, S., Esteve, P., Reverté Villarroya, S., Cubo, E., Casas, E., Arnaiz, S., Carrillo Padilla, F., Pueyo Morlans, M., Mir, P., Martinez-Martin, P., Adarmes, A.D., Almeria, M., Alonso Losada, G., Alonso Cánovas, A., Alonso-Frech, F., Arribas, S., Ascunde Vidondo, A., Aquilar, M., Ávila, M.A., Bernardo Lambrich, N., Bejr-Kasem, H., Blázquez Estrada, M., Botí, M., Cabello González, C., Cabo López, I., Caballol, N., Cámara Lorenzo, A., Carrillo, F., Catalán, M.J., Clavero, P., Cortina Fernández, A., Crespo Cuevas, A., de Fábregues-Boixar, O., Díez-Fairen, M., Erro, E., Estelrich Peyret, E., Fernández Guillán, N., Gámez, P., Gallego, M., García Caldentey, J., García Campos, C., García Moreno, J.M., Gastón, I., Gómez Garre, M.P., González Aloy, J., González-Aramburu, I., González Ardura, J., González García, B., González Palmás, M.J., González Toledo, G.R., Golpe Díaz, A., Grau Solá, M., Guardia, G., Hernández-Vara, J., Horta Barba, A., Infante, J., Jesús, S., Kulisevsky, J., Kurtis, M., Labandeira, C., Labrador, M.A., Lacruz, F., Lage Castro, M., Legarda, I., López Ariztegui, N., López Díaz, L.M., López Manzanares, L., López Seoane, B., Martí Andres, G., Martí, M.J., Martínez-Castrillo, J.C., McAfee, D., Meitín, M.T., Menéndez González, M., Méndez del Barrio, C., Miranda Santiago, J., Morales Casado, M.I., Moreno Diéguez, A., Nogueira, V., Novo Amado, A., Novo Ponte, S., Ordás, C., Pagonabarraga, J., Pareés, I., Pascual-Sedano, B., Pastor, P., Pérez Fuertes, A., Pérez Noguera, R., Planellas, L., Pol Fuster, J., Prats, M.A., Prieto Jurczynska, C., Puente, V., Redondo Rafales, N., Rodríguez Méndez, L., Roldán, F., Ruíz De Arcos, M., Ruíz Martínez, J., Sánchez Alonso, P., Sánchez-Carpintero, M., Sánchez Díez, G., Sánchez Rodríguez, A., Santacruz, P., Segundo Rodríguez, J.C., Seijo, M., Serarols, A., Sierra Peña, M., Tartari, J.P., Vargas, L., Vázquez Gómez, R., Villanueva, C., Vives, B., Villar, M.D., 2019. Non-motor symptoms burden, mood, and gait problems are the most significant factors contributing to a poor quality of life in non-demented Parkinson's disease patients: Results from the COPPADIS Study Cohort. Parkinsonism Relat. Disord. 66, 151–157. https://doi.org/10.1016/j.parkreldis.2019.07.031
Santos-Galindo, M., Acaz-Fonseca, E., Bellini, M.J., Garcia-Segura, L.M., 2011. Sex differences in the inflammatory response of primary astrocytes to lipopolysaccharide. Biol. Sex Differ. 2, 7. https://doi.org/10.1186/2042-6410-2-7

Santos-García, D., Laguna, A., Hernández-Vara, J., de Deus Fonticoba, T., Cores Bartolomé, C., Feal Painceiras, M.J., Íñiguez-Alvarado, M.C., García Díaz, I., Jesús, S., Boungiorno, M.T., Planellas, L., Cosgaya, M., García Caldentey, J., Caballol, N., Legarda, I., Cabo, I., López Manzanares, L., González Aramburu, I., Ávila Rivera, M.A., Gómez Mayordomo, V., Nogueira, V., Puente, V., Dotor García-Soto, J., Borrué, C., Solano Vila, B., Álvarez Sauco, M., Vela, L., Escalante, S., Cubo, E., Carrillo Padilla, F., Castrillo, J.C.M., Sánchez Alonso, P., Alonso Losada, M.G., López Ariztegui, N., Gastón, I., Kulisevsky, J., Menéndez González, M., Seijo, M., Rúiz Martínez, J., Valero, C., Kurtis, M., Ardura, J.G., Alonso Redondo, R., Ordás, C., López Díaz, L.M., McAfee, D., Martinez-Martin, P., Mir, P., 2023. Sex Differences in Motor and Non-Motor Symptoms among Spanish Patients with Parkinson's Disease. J. Clin. Med. 12, 1329. https://doi.org/10.3390/jcm12041329

Saunders-Pullman, R., Gordon-Elliott, J., Parides, M., Fahn, S., Saunders, H.R., Bressman, S., 1999. The effect of estrogen replacement on early Parkinson's disease. Neurology 52, 1417–1417. https://doi.org/10.1212/WNL.52.7.1417

Saunders-Pullman, R., Wang, C., Stanley, K., Bressman, S.B., 2011. Diagnosis and Referral Delay in Women With Parkinson's Disease. Gend. Med. 8, 209–217. https://doi.org/10.1016/j.genm.2011.05.002

Savica, R., Carlin, J.M., Grossardt, B.R., Bower, J.H., Ahlskog, J.E., Maraganore, D.M., Bharucha, A.E., Rocca, W.A., 2009. Medical records documentation of constipation preceding Parkinson disease. Neurology 73, 1752–1758. https://doi.org/10.1212/WNL.0b013e3181c34af5

Schapira, A.H.V., Fox, S.H., Hauser, R.A., Jankovic, J., Jost, W.H., Kenney, C., Kulisevsky, J., Pahwa, R., Poewe, W., Anand, R., 2017. Assessment of Safety and Efficacy of Safinamide as a Levodopa Adjunct in Patients With Parkinson Disease and Motor Fluctuations: A Randomized Clinical Trial. JAMA Neurol. 74, 216–224. https://doi.org/10.1001/jamaneurol.2016.4467

Schestatsky, P., Valls-Solé, J., Ehlers, J.A., Rieder, C.R.M., Gomes, I., 2006. Hyperhidrosis in Parkinson's disease. Mov. Disord. 21, 1744–1748. https://doi.org/10.1002/mds.21006

Schrag, A., Banks, P., 2006. Time of loss of employment in Parkinson's disease. Mov. Disord. 21, 1839–1843. https://doi.org/10.1002/mds.21030

Schwarzschild, M.A., Marek, K., Eberly, S., Oakes, D., Shoulson, I., Jennings, D., Seibyl, J., Ascherio, A., 2011. Serum urate and probability of dopaminergic deficit in early 'Parkinson disease.' Mov. Disord. Off. J. Mov. Disord. Soc. 26, 1864–1868. https://doi.org/10.1002/mds.23741

Seier, M., Hiller, A., 2017. Parkinson's disease and pregnancy: An updated review. Parkinsonism Relat. Disord. 40, 11–17. https://doi.org/10.1016/j.parkreldis.2017.05.007

Session, D.R., Pearlstone, M.M., Jewelewicz, R., Kelly, A.C., 1994. Estrogens and Parkinson's disease. Med. Hypotheses 42, 280–282. https://doi.org/10.1016/0306-9877(94)90133-3

Seyfried, T.N., Choi, H., Chevalier, A., Hogan, D., Akgoc, Z., Schneider, J.S., 2018. Sex-Related Abnormalities in Substantia Nigra Lipids in Parkinson's Disease. ASN NEURO 10, 1759091418781889. https://doi.org/10.1177/1759091418781889

Sharma, J.C., Bachmann, C.G., Linazasoro, G., 2010. Classifying risk factors for dyskinesia in Parkinson's disease. Parkinsonism Relat. Disord. 16, 490–497. https://doi.org/10.1016/j.parkreldis.2010.06.003
Sharma, J.C., Turton, J., 2012. Olfaction, dyskinesia and profile of weight change in Parkinson's disease: Identifying neurodegenerative phenotypes. Parkinsonism Relat. Disord. 18, 964–970. https://doi.org/10.1016/j.parkreldis.2012.05.004
Shiba, M., Bower, J.H., Maraganore, D.M., McDonnell, S.K., Peterson, B.J., Ahlskog, J.E., Schaid, D.J., Rocca, W.A., 2000. Anxiety disorders and depressive disorders preceding Parkinson's disease: A case-control study. Mov. Disord. 15, 669–677. https://doi.org/10.1002/1531-8257(200007)15:4%253C669::AID-MDS1011%253E3.0.CO;2-5
Shiraishi, M., Kamo, T., Kumazawa, R., Sasaki, N., Isahaya, K., Akiyama, H., Furusawa, Y., Onodera, R., Fernandez, J., Otsuka, M., Yamano, Y., 2023. A multicenter, prospective, observational study to assess the satisfaction of an integrated digital platform of online medical care and remote patient monitoring in Parkinson's disease. Neurol. Clin. Neurosci. 11, 152–163. https://doi.org/10.1111/ncn3.12709
Shpiner, D.S., Luca, D.G.D., Cajigas, I., Diaz, J.S., Margolesky, J., Moore, H., Levin, B.E., Singer, C., Jagid, J., Luca, C.C., 2019. Gender Disparities in Deep Brain Stimulation for Parkinson's Disease. Neuromodulation 22, 484–488. https://doi.org/10.1111/ner.12973
Shu, L., Zhang, Y., Pan, H., Xu, Q., Guo, J., Tang, B., Sun, Q., 2018. Clinical Heterogeneity Among LRRK2 Variants in Parkinson's Disease: A Meta-Analysis. Front. Aging Neurosci. 10, 283. https://doi.org/10.3389/fnagi.2018.00283
Shulman, L.M., Minagar, A., Weiner, W.J., 2000. The effect of pregnancy in Parkinson's disease. Mov. Disord. 15, 132–135. https://doi.org/10.1002/1531-8257(200001)15:1%253C132::AID-MDS1020%253E3.0.CO;2-C
Siani, F., Greco, R., Levandis, G., Ghezzi, C., Daviddi, F., Demartini, C., Vegeto, E., Fuzzati-Armentero, M.-T., Blandini, F., 2017. Influence of Estrogen Modulation on Glia Activation in a Murine Model of Parkinson's Disease. Front. Neurosci. 11, 306. https://doi.org/10.3389/fnins.2017.00306
Siddle, N., Sarrel, P., Whitehead, M., 1987. The effect of hysterectomy on the age at ovarian failure: identification of a subgroup of women with premature loss of ovarian function and literature review*. Fertil. Steril. 47, 94–100. https://doi.org/10.1016/S0015-0282(16)49942-5
Simon, K.C., Chen, H., Gao, X., Schwarzschild, M.A., Ascherio, A., 2009. Reproductive factors, exogenous estrogen use and risk of Parkinson's disease. Mov. Disord. Off. J. Mov. Disord. Soc. 24, 1359–1365. https://doi.org/10.1002/mds.22619
Simunovic, F., Yi, M., Wang, Y., Stephens, R., Sonntag, K.C., 2010. Evidence for Gender-Specific Transcriptional Profiles of Nigral Dopamine Neurons in Parkinson Disease. PLoS ONE 5, e8856. https://doi.org/10.1371/journal.pone.0008856
Smith, K.M., Dahodwala, N., 2014. Sex differences in Parkinson's disease and other movement disorders. Exp. Neurol., Special Issue: The importance of sex in the etiology, presentation, and treatment of neurological disorders 259, 44–56. https://doi.org/10.1016/j.expneurol.2014.03.010
Song, Z., Yang, H., Peckham, E.M., Becker, J.B., 2019. Estradiol-Induced Potentiation of Dopamine Release in Dorsal Striatum Following Amphetamine Administration

Requires Estradiol Receptors and mGlu5. eNeuro 6, ENEURO.0446-18.2019. https://doi.org/10.1523/ENEURO.0446-18.2019
Sprenger, F.S., Seppi, K., Wolf, E., Poewe, W., 2014. Perimenstrual Fluctuations in Two Siblings With Early-Onset Parkinson's Disease. Mov. Disord. Clin. Pract. 1, 125–127. https://doi.org/10.1002/mdc3.12036
Storch, A., Schneider, C.B., Wolz, M., Stürwald, Y., Nebe, A., Odin, P., Mahler, A., Fuchs, G., Jost, W.H., Chaudhuri, K.R., Koch, R., Reichmann, H., Ebersbach, G., 2013. Nonmotor fluctuations in Parkinson disease. Neurology 80, 800–809. https://doi.org/10.1212/WNL.0b013e318285c0ed
StrivePD [WWW Document], 2026. URL https://www.strive.group/ (accessed 3.7.26).
Subramanian, I., Mathur, S., Oosterbaan, A., Flanagan, R., Keener, A.M., Moro, E., 2022. Unmet Needs of Women Living with Parkinson's Disease: Gaps and Controversies. Mov. Disord. Off. J. Mov. Disord. Soc. 37, 444–455. https://doi.org/10.1002/mds.28921
Szewczyk-Krolikowski, K., Tomlinson, P., Nithi, K., Wade-Martins, R., Talbot, K., Ben-Shlomo, Y., Hu, M.T.M., 2014. The influence of age and gender on motor and non-motor features of early Parkinson's disease: Initial findings from the Oxford Parkinson Disease Center (OPDC) discovery cohort. Parkinsonism Relat. Disord. 20, 99–105. https://doi.org/10.1016/j.parkreldis.2013.09.025
Tambasco, N., Nigro, P., Romoli, M., Prontera, P., Simoni, S., Calabresi, P., 2016. A53T in a parkinsonian family: a clinical update of the SNCA phenotypes. J. Neural Transm. 123, 1301–1307. https://doi.org/10.1007/s00702-016-1578-6
Tang, X., Yang, J., Zhu, Y., Gong, H., Sun, H., Chen, F., Guan, Q., Yu, L., Wang, W., Zhang, Z., Li, L., Ma, G., Wang, X., 2022. High PSQI score is associated with the development of dyskinesia in Parkinson's disease. Npj Park. Dis. 8, 124. https://doi.org/10.1038/s41531-022-00391-y
Teismann, P., 2012. COX-2 in the neurodegenerative process of Parkinson's disease. Biofactors Oxf. Engl. 38, 395–397. https://doi.org/10.1002/biof.1035
Tholfsen, L.K., Larsen, J.P., Schulz, J., Tysnes, O.-B., Gjerstad, M.D., 2015. Development of excessive daytime sleepiness in early Parkinson disease. Neurology 85, 162–168. https://doi.org/10.1212/WNL.0000000000001737
Tolson, D., Fleming, V., Schartau, E., 2002. Coping with menstruation: understanding the needs of women with Parkinson's disease. J. Adv. Nurs. 40, 513–521. https://doi.org/10.1046/j.1365-2648.2002.02408.x
Tomic, S., Kuric, I., Kuric, T.G., Popovic, Z., Kragujevic, J., Zubonja, T.M., Rajkovaca, I., Matosa, S., 2022. Seborrheic Dermatitis Is Related to Motor Symptoms in Parkinson's Disease. J. Clin. Neurol. Seoul Korea 18, 628–634. https://doi.org/10.3988/jcn.2022.18.6.628
Tsang, K.-L., Ho, S.-L., Lo, S.-K., 2000. Estrogen improves motor disability in parkinsonian postmenopausal women with motor fluctuations. Neurology 54, 2292–2298. https://doi.org/10.1212/WNL.54.12.2292
Tsuboi, T., Satake, Y., Hiraga, K., Yokoi, K., Hattori, M., Suzuki, M., Hara, K., Ramirez-Zamora, A., Okun, M.S., Katsuno, M., 2022. Effects of MAO-B inhibitors on non-motor symptoms and quality of life in Parkinson's disease: A systematic review. NPJ Park. Dis. 8, 75. https://doi.org/10.1038/s41531-022-00339-2
Tucker, D., Masood, S., 2025. Seborrheic Dermatitis, in: StatPearls. StatPearls Publishing, Treasure Island (FL).

Tulchinsky, D., Hobel, C.J., Yeager, E., Marshall, J.R., 1972. Plasma estrone, estradiol, estriol, progesterone, and 17-hydroxyprogesterone in human pregnancy: I. Normal pregnancy. Am. J. Obstet. Gynecol. 112, 1095–1100. https://doi.org/10.1016/0002-9378(72)90185-8

Tzeng, S.-F., Hsiao, H.-Y., Mak, O.-T., 2005. Prostaglandins and cyclooxygenases in glial cells during brain inflammation. Curr. Drug Targets Inflamm. Allergy 4, 335–340. https://doi.org/10.2174/1568010054022051

Unda, S.R., Marciano, S., Milner, T.A., Marongiu, R., 2022. State-of-the-art review of the clinical research on menopause and hormone replacement therapy association with Parkinson's disease: What meta-analysis studies cannot tell us. Front. Aging Neurosci. 14. https://doi.org/10.3389/fnagi.2022.971007

Valkovic, P., Minar, M., Singliarova, H., Harsany, J., Hanakova, M., Martinkova, J., Benetin, J., 2015. Pain in Parkinson´s Disease: A Cross-Sectional Study of Its Prevalence, Types, and Relationship to Depression and Quality of Life. PLoS ONE 10, e0136541. https://doi.org/10.1371/journal.pone.0136541

Van Den Eeden, S.K., Tanner, C.M., Bernstein, A.L., Fross, R.D., Leimpeter, A., Bloch, D.A., Nelson, L.M., 2003. Incidence of Parkinson's Disease: Variation by Age, Gender, and Race/Ethnicity. Am. J. Epidemiol. 157, 1015–1022. https://doi.org/10.1093/aje/kwg068

Vibha, D., Sureshbabu, S., Shukla, G., Goyal, V., Srivastava, A.K., Singh, S., Behari, M., 2010. Differences between familial and sporadic Parkinson's disease. Parkinsonism Relat. Disord. 16, 486–487. https://doi.org/10.1016/j.parkreldis.2010.04.012

Villa, A., Gelosa, P., Castiglioni, L., Cimino, M., Rizzi, N., Pepe, G., Lolli, F., Marcello, E., Sironi, L., Vegeto, E., Maggi, A., 2018. Sex-Specific Features of Microglia from Adult Mice. Cell Rep. 23, 3501–3511. https://doi.org/10.1016/j.celrep.2018.05.048

Vinke, R.S., Georgiev, D., Selvaraj, A.K., Rahimi, T., Bloem, B.R., Bartels, R.H.M.A., Esselink, R.A.J., 2022. Gender Distribution in Deep Brain Stimulation for Parkinson's Disease: The Effect of Awake versus Asleep Surgery. J. Park. Dis. 12, 1965–1968. https://doi.org/10.3233/JPD-223315

Virmani, T., Tazan, S., Mazzoni, P., Ford, B., Greene, P.E., 2016. Motor fluctuations due to interaction between dietary protein and levodopa in Parkinson's disease. J. Clin. Mov. Disord. 3, 8. https://doi.org/10.1186/s40734-016-0036-9

Wang, L.-F., Yokoyama, K.K., Chen, T.-Y., Hsiao, H.-W., Chiang, P.-C., Hsieh, Y.-C., Lo, S., Hsu, C., 2015. Male-Specific Alleviation of Iron-Induced Striatal Injury by Inhibition of Autophagy. PLoS ONE 10, e0131224. https://doi.org/10.1371/journal.pone.0131224

Wang, P., Li, J., Qiu, S., Wen, H., Du, J., 2014. Hormone replacement therapy and Parkinson's disease risk in women: a meta-analysis of 14 observational studies. Neuropsychiatr. Dis. Treat. 11, 59–66. https://doi.org/10.2147/NDT.S69918

Warren Olanow, C., Kieburtz, K., Rascol, O., Poewe, W., Schapira, A.H., Emre, M., Nissinen, H., Leinonen, M., Stocchi, F., Investigators, S.R. in D.E. in P.D. (STRIDE-P., 2013. Factors predictive of the development of Levodopa-induced dyskinesia and wearing-off in Parkinson's disease. Mov. Disord. 28, 1064–1071. https://doi.org/10.1002/mds.25364

Weaver, F.M., Follett, K., Stern, M., Hur, K., Harris, C., Marks, W.J., Rothlind, J., Sagher, O., Reda, D., Moy, C.S., Pahwa, R., Burchiel, K., Hogarth, P., Lai, E.C., Duda, J.E., Holloway, K., Samii, A., Horn, S., Bronstein, J., Stoner, G., Heemskerk, J., Huang, G.D., 2009. Bilateral Deep Brain Stimulation vs Best Medical Therapy for Patients With Advanced Parkinson Disease. JAMA J. Am. Med. Assoc. 301, 63. https://doi.org/10.1001/jama.2008.929

Wickremaratchi, M.M., Ben-Shlomo, Y., Morris, H.R., 2009. The effect of onset age on the clinical features of Parkinson's disease. Eur. J. Neurol. 16, 450–456. https://doi.org/10.1111/j.1468-1331.2008.02514.x

Wilkowski, C.M., Alhajj, M., Kilbane, C.W., Kumar, Y., Bordeaux, J.S., Carroll, B.T., 2025. The association of seborrheic dermatitis and Parkinson's disease: A cross-sectional study. J. Am. Acad. Dermatol. 92, 157–159. https://doi.org/10.1016/j.jaad.2024.08.074

Willis, A.W., Roberts, E., Beck, J.C., Fiske, B., Ross, W., Savica, R., Van Den Eeden, S.K., Tanner, C.M., Marras, C., 2022. Incidence of Parkinson disease in North America. Npj Park. Dis. 8, 1–7. https://doi.org/10.1038/s41531-022-00410-y

Willis, A.W., Schootman, M., Kung, N., Wang, X.-Y., Perlmutter, J.S., Racette, B.A., 2014. Disparities in deep brain stimulation surgery among insured elders with Parkinson disease. Neurology 82, 163–171. https://doi.org/10.1212/WNL.0000000000000017

Windham, G.C., Elkin, E., Fenster, L., Waller, K., Anderson, M., Mitchell, P.R., Lasley, B., Swan, S.H., 2002. Ovarian Hormones in Premenopausal Women: Variation by Demographic, Reproductive and Menstrual Cycle Characteristics. Epidemiology 13, 675.

Yoo, J.E., Shin, D.W., Jang, W., Han, K., Kim, D., Won, H.-S., Park, H.S., 2020. Female reproductive factors and the risk of Parkinson's disease: a nat ionwide cohort study. Eur. J. Epidemiol. 35, 871–878. https://doi.org/10.1007/s10654-020-00672-x

Young, C., Phillips, R., Ebenezer, L., Zutt, R., Peall, K.J., 2020. Management of Parkinson's Disease During Pregnancy: Literature Review and Multidisciplinary Input. Mov. Disord. Clin. Pract. 7, 419–430. https://doi.org/10.1002/mdc3.12925

Yu, L., Kuo, Y., Cherng, C.G., Chen, H.-H., Hsu, C.-H., 2002. Ovarian Hormones Do Not Attenuate Methamphetamine-Induced Dopaminergic Neurotoxicity in Mice Gonadectomized at 4 Weeks Postpartum. Neuroendocrinology 75, 282–287. https://doi.org/10.1159/000057337

Yu, L., Liao, P.-C., 2000. Sexual differences and estrous cycle in methamphetamine-induced dopamine and serotonin depletions in the striatum of mice. J. Neural Transm. 107, 419–427. https://doi.org/10.1007/s007020070084

Zappia, M., Annesi, G., Nicoletti, G., Arabia, G., Annesi, F., Messina, D., Pugliese, P., Spadafora, P., Tarantino, P., Carrideo, S., Civitelli, D., De Marco, E.V., Cirò-Candiano, I.C., Gambardella, A., Quattrone, A., 2005. Sex Differences in Clinical and Genetic Determinants of Levodopa Peak-Dose Dyskinesias in Parkinson Disease: An Exploratory Study. Arch. Neurol. 62, 601–605. https://doi.org/10.1001/archneur.62.4.601

Zhao, T., Ding, Q., Hu, J., He, S., Shi, F., Ma, L., 2016. GPER expressed on microglia mediates the anti-inflammatory effect of estradiol in ischemic stroke. Brain Behav. 6, e00449. https://doi.org/10.1002/brb3.449

Zirra, A., Rao, S.C., Bestwick, J., Rajalingam, R., Marras, C., Blauwendraat, C., Mata, I.F., Noyce, A.J., 2022. Gender Differences in the Prevalence of Parkinson's Disease. Mov. Disord. Clin. Pract. 10, 86–93. https://doi.org/10.1002/mdc3.13584

6. Acknowledgements

I would like to express my heartfelt gratitude to my supervisors, Prof. Aideen Sullivan, Dr. Lucy Collins Stack, and Dr. Aonghus Lavelle, for their unwavering guidance and encouragement throughout the course of this study. Their expertise, support, and mentorship have been invaluable. I also want to extend my thanks to Richelle Flanagan and René Reinbacher, co-founders of the My Moves Matter app, for their generous support during the project.

7. Supplementary Material

Figure S1: Initial Survey Completion Message

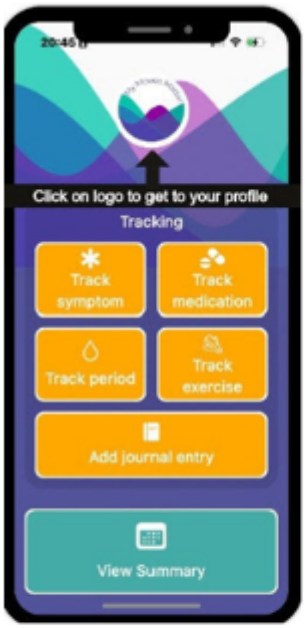
Thank you for taking the survey

For the next important part to continue with the study, we ask you to download the My Moves Matter app.

1. **Please go to your mobile and download the My Moves Matter app by clicking the link**: https://www.mymovesmatter.com/get-the-app or you can scan the barcode below:

Download My Moves Matter

2. Register on the app. You do this by clicking on the My Moves Matter logo on the front page of the app



3. After you have entered your registration details, **enter your UCC study code on the app** in the box that says Clinical trial ID which is at the bottom of the registration page on the app



This is a really important step as this will link your anonymous tracking data to your surveys to analyse the data to produce the study results.

4. Once you download the app, then add your medications.
5. Then start tracking from the day you register on the app
    a. Track your medication intake.
    b. Track your PD symptoms.
    c. If you still menstruate, even if erratic, simply track all the days of your period or any menstrual bleeding during each of the 4 months.

We need you to track everyday for 4 months.

The reason we ask for 4 months is so there is enough data to see if there are indeed patterns where different hormonal stages might impact the effectiveness of medication and so impact Parkinson's symptoms.

After you complete the study, you can keep using the app for free and you will be able to see all your own tracked data from the 4 months.

Thank you again for taking part in this study, your participation will help us understand the impact of the different hormonal stages, whether pre-menopausal, pregnant, peri/menopausal or post-menopausal, on PD symptoms.

Please remember your Unique Study Code for input to the My Moves Matter App

UCCXXXYYY

Figure S1: Completion message displayed to participants upon finishing the initial survey.

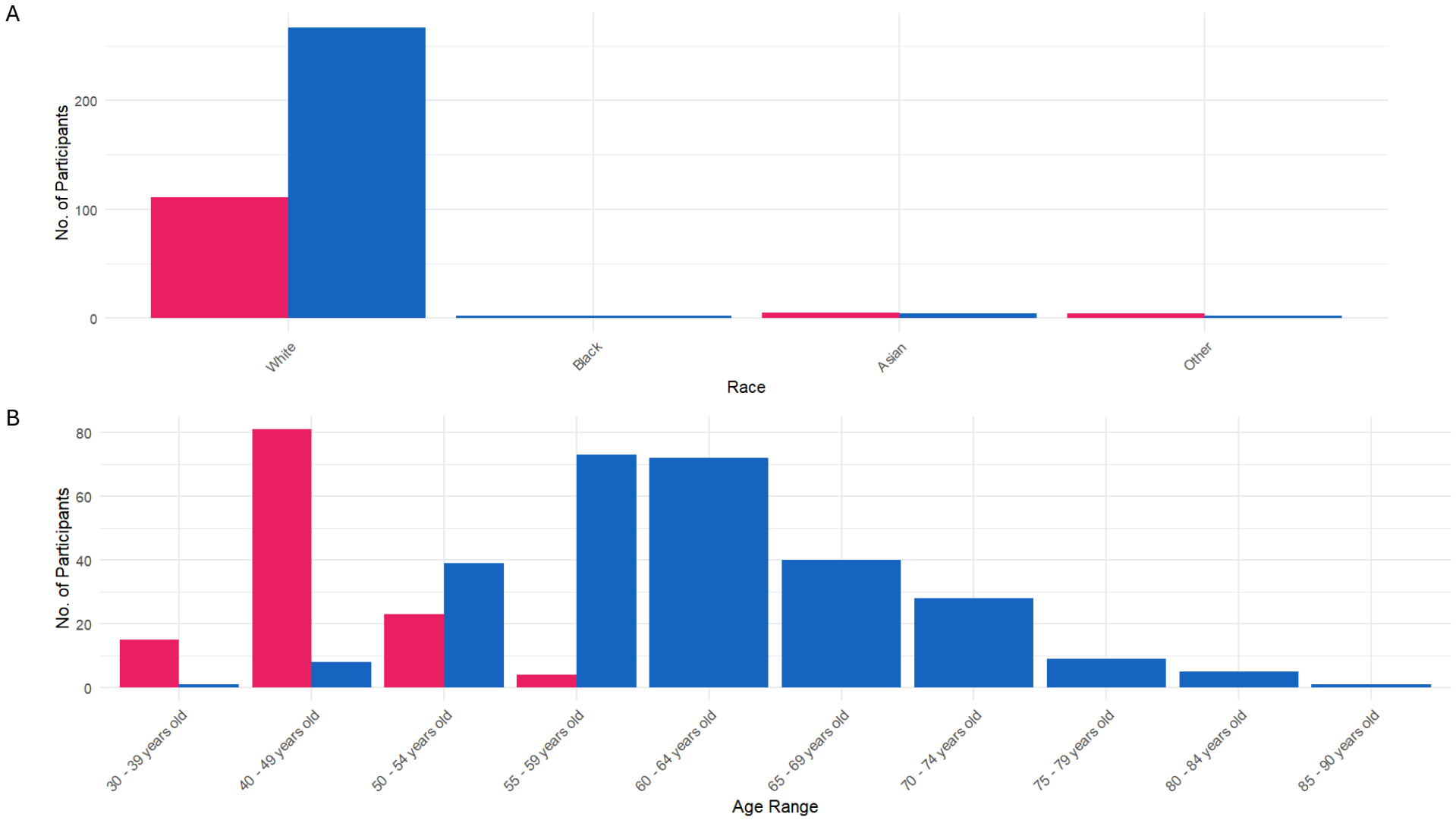
Table S1: Extended Demographics by Menstruation Status

Category	Option	Menstruating	Postmenopausal
Age (M: N=123, PM: N=276)	**30 - 39 years**	12.2	0.36
	40 - 49 years	65.85	2.9
	50 - 54 years	18.7	14.13
	55 - 59 years	3.25	26.45
	60 - 64 years	0	26.09
	65 - 69 years	0	14.49
	70 - 74 years	0	10.14
	75 - 79 years	0	3.26
	80 - 84 years	0	1.81
	85 - 90 years	0	0.36
Gender (M: N=123, PM: N=276)	**Female (biological from birth)**	100	100
Race (M: N=120, PM: N=275)	**White**	92.5	97.09
	Black	0	0.73
	Asian	4.17	1.45
Employment (M: N=122, PM: N=274)	**Employed (full-time)**	42.62	15.33
	Employed (part-time)	20.49	11.31
	Self-employed	9.84	6.57
	Unemployed (actively seeking employment)	1.64	1.09
	Unemployed (not actively seeking employment)	7.38	6.93
	Student (full-time)	1.64	0
	Homemaker	8.2	4.01
	Retired	2.46	43.07
	Disabled	3.28	10.58
Marital Status (M: N=123, PM: N=271)	**Single**	12.2	7.38
	In a relationship	9.76	4.43
	Married	65.04	70.48
	Widowed	0.81	2.21
	Divorced	6.5	12.92
	Separated	4.07	1.48
	Domestic partnership	0.81	1.11
	Civil union	0.81	0
Age of Diagnosis (M: N=122, PM: N=273)	**18 - 29 years**	4.1	0.37
	30 - 39 years	33.61	3.3
	40 - 49 years	55.74	21.25
	50 - 54 years	5.74	23.44

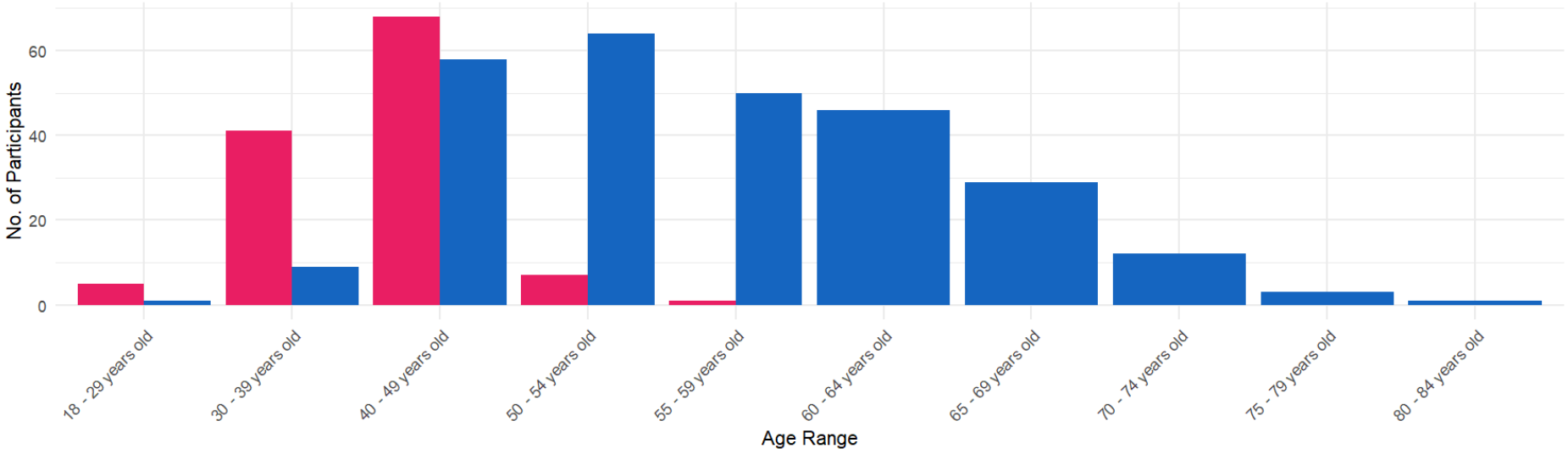
	55 - 59 years	0.82	18.32
	60 - 64 years	0	16.85
	65 - 69 years	0	10.62
	70 - 74 years	0	4.4
	75 - 79 years	0	1.1
	80 - 84 years	0	0.37
Symptom Duration (M: N=113, PM: N=256)	**0 - 3 years**	27.43	21.88
	4 - 6 years	23.89	27.34
	7 - 9 years	23.01	16.02
	10 - 12 years	10.62	12.89
	13 - 15 years	6.19	8.98
	>15 years	8.85	12.89
Family Diagnosis (M: N=121, PM: N=273)	**Yes**	23.97	19.41
	No	68.6	74.73
Procedures (M: N=118, PM: N=260)	**Yes**	15.25	14.62
	No	84.75	85.38
Hormonal Contraceptive Use (M: N=123, PM: N=276)	**No**	70.73	92.03
	Yes	29.27	7.97

Table S1: Extended table showing demographic characteristics of menstruating (n=125) and postmenopausal (n=283) women showing percentage distribution within each menstruation status. Question-specific sample sizes: Age (M: N=123, PM: N=276); Gender (M: N=123, PM: N=276); Race (M: N=120, PM: N=275); Employment (M: N=122, PM: N=274); Marital Status (M: N=123, PM: N=271); Age of Diagnosis (M: N=122, PM: N=273); Symptom Duration (M: N=113, PM: N=256); Family Diagnosis (M: N=121, PM: N=273); Procedures (M: N=118, PM: N=260); Hormonal Contraceptive Use (M: N=123, PM: N=276). M = Menstruating, PM = Postmenopausal.

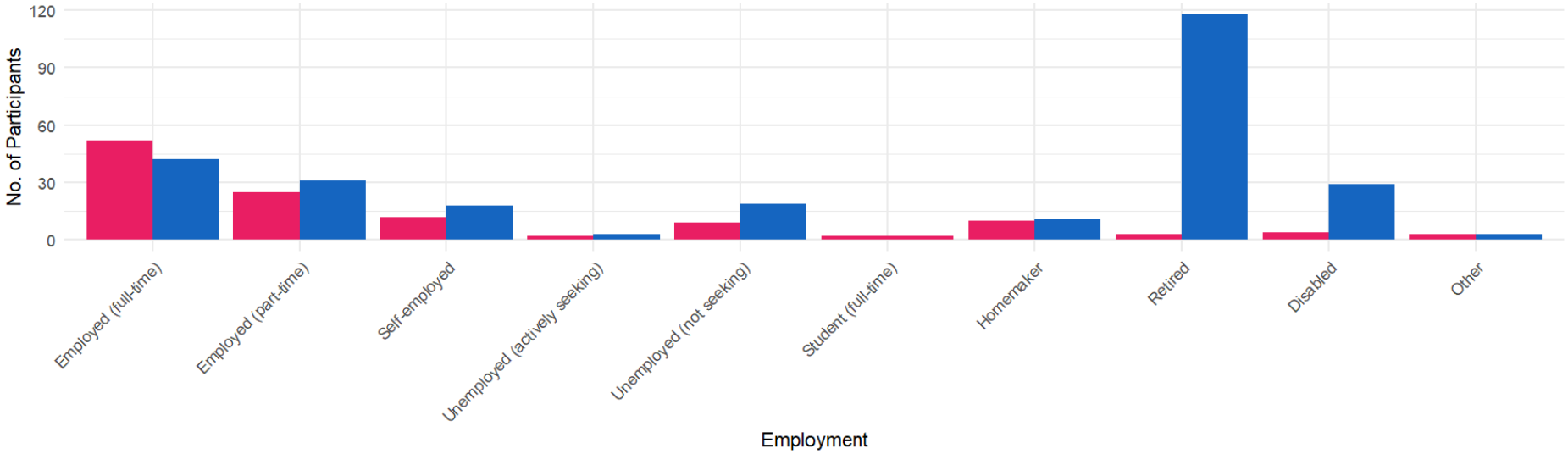
Figure S2: Demographic Counts by Menstruation Status



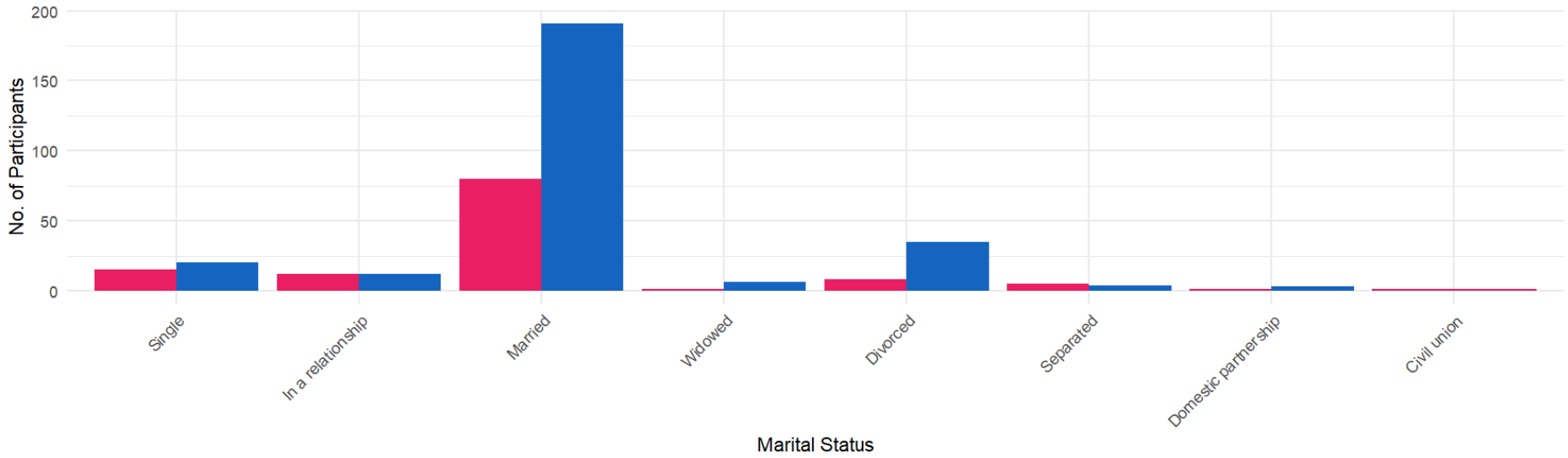
C


No. of Participants
0
20
40
60
18 - 29 years old
30 - 39 years old
40 - 49 years old
50 - 54 years old
55 - 59 years old
60 - 64 years old
65 - 69 years old
70 - 74 years old
75 - 79 years old
80 - 84 years old
Age Range

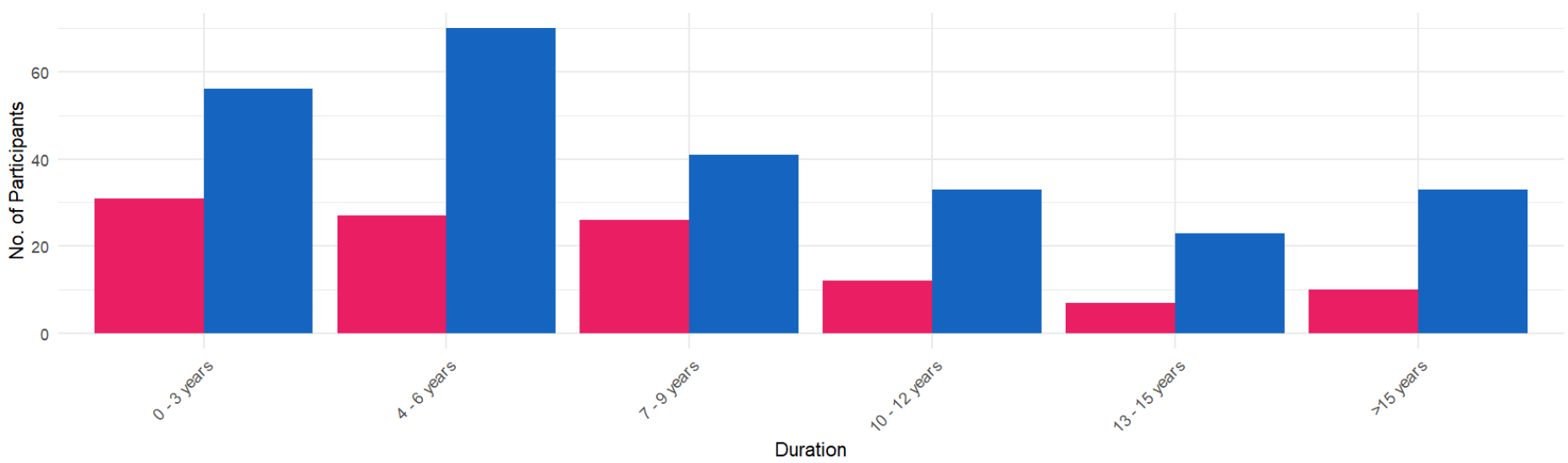
D


No. of Participants
0
30
60
90
120
Employed (full-time)
Employed (part-time)
Self-employed
Unemployed (actively seeking)
Unemployed (not seeking)
Student (full-time)
Homemaker
Retired
Disabled
Other
Employment

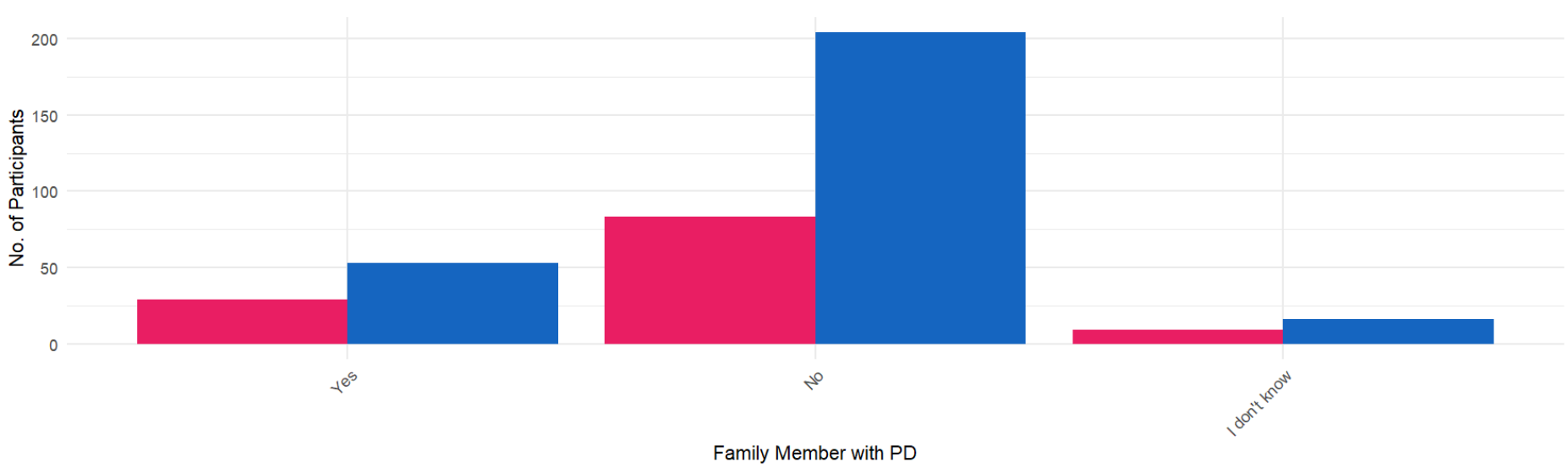
E


200
150
100
50
0
No. of Participants
Single
In a relationship
Married
Widowed
Divorced
Separated
Domestic partnership
Civil union
Marital Status

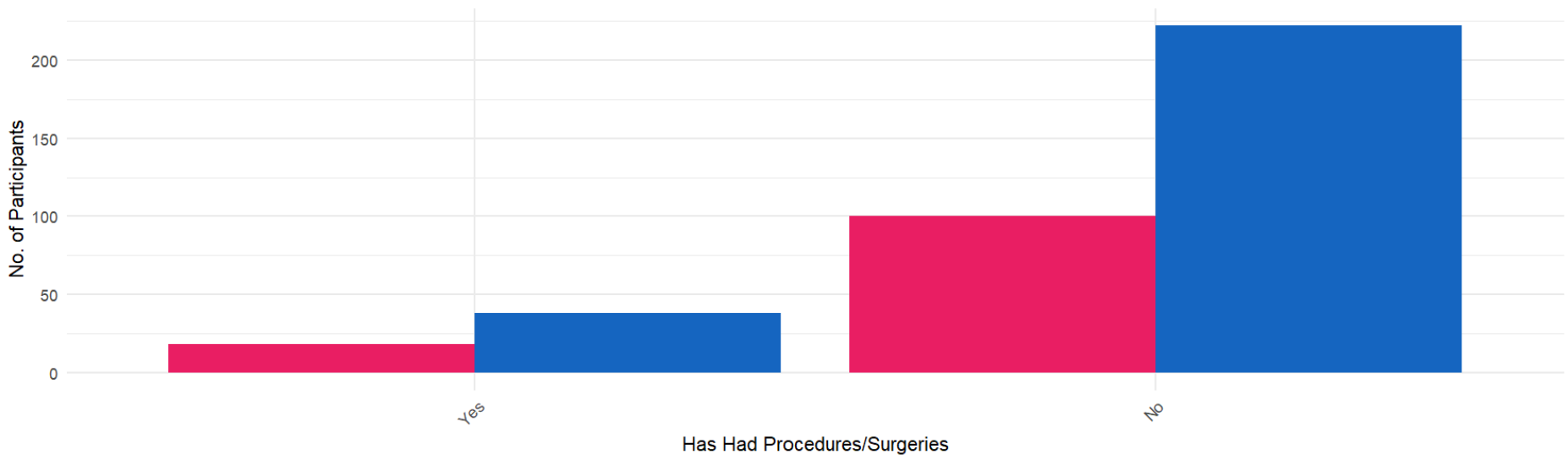
F


60
40
20
0
No. of Participants
0 - 3 years
4 - 6 years
7 - 9 years
10 - 12 years
13 - 15 years
>15 years
Duration

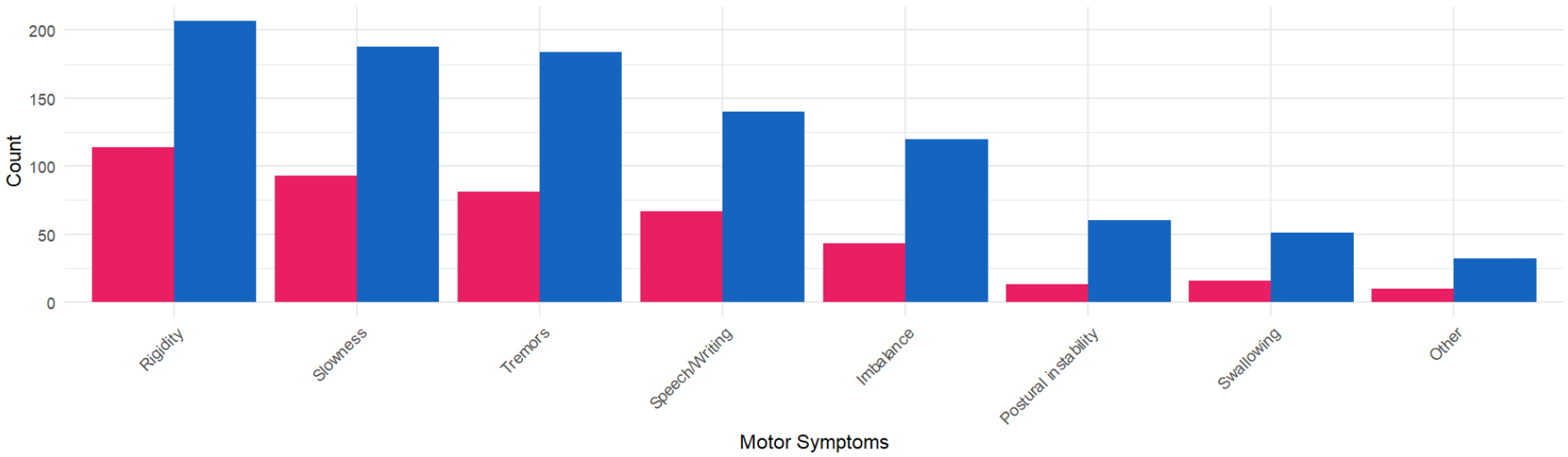
G


200
150
100
50
0
No. of Participants
Yes
No
I don't know
Family Member with PD

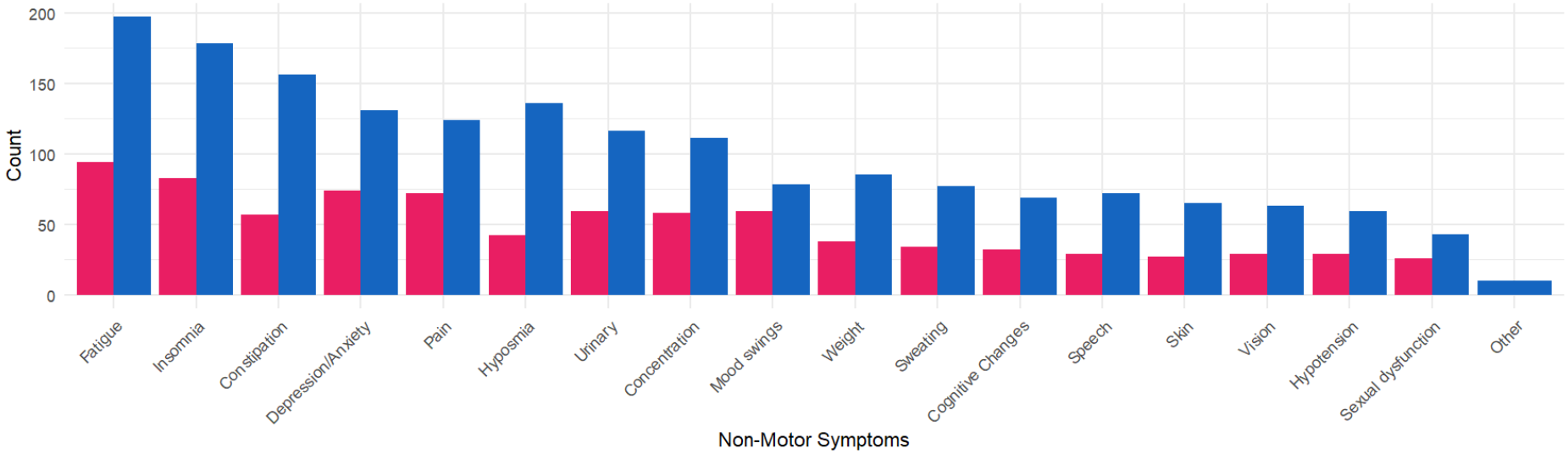
H


200
150
100
50
0
No. of Participants
Yes
No
Has Had Procedures/Surgeries

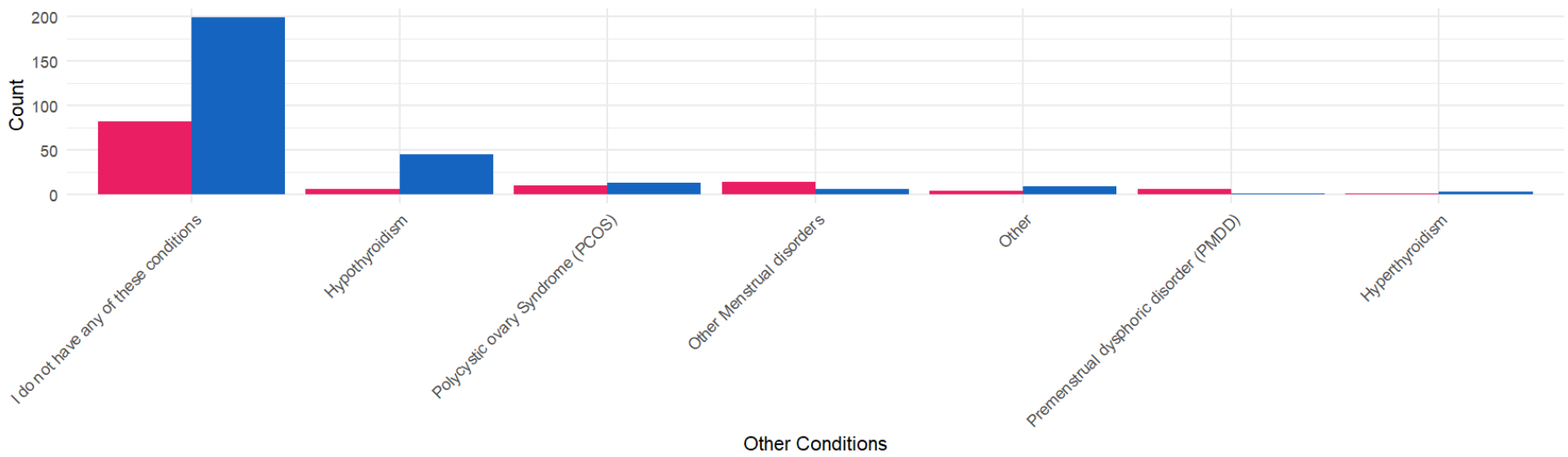
I


Count
200
150
100
50
0
Rigidity
Slowness
Tremors
Speech/Writing
Imbalance
Postural instability
Swallowing
Other
Motor Symptoms

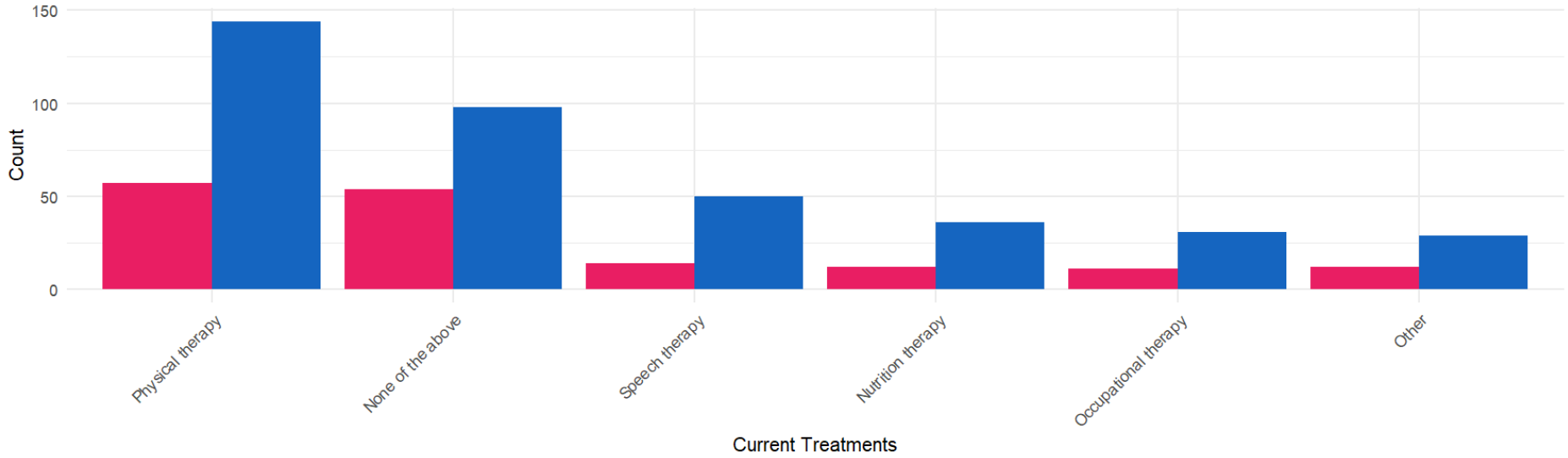
J


Count
200
150
100
50
0
Fatigue
Insomnia
Constipation
Depression/Anxiety
Pain
Hyposmia
Urinary
Concentration
Mood swings
Weight
Sweating
Cognitive Changes
Speech
Skin
Vision
Hypotension
Sexual dysfunction
Other
Non-Motor Symptoms

K


Count
200
150
100
50
0
I do not have any of these conditions
Hypothyroidism
Polycystic ovary Syndrome (PCOS)
Other Menstrual disorders
Other
Premenstrual dysphoric disorder (PMDD)
Hyperthyroidism
Other Conditions

L


Count
150
100
50
0
Physical therapy
None of the above
Speech therapy
Nutrition therapy
Occupational therapy
Other
Current Treatments

M

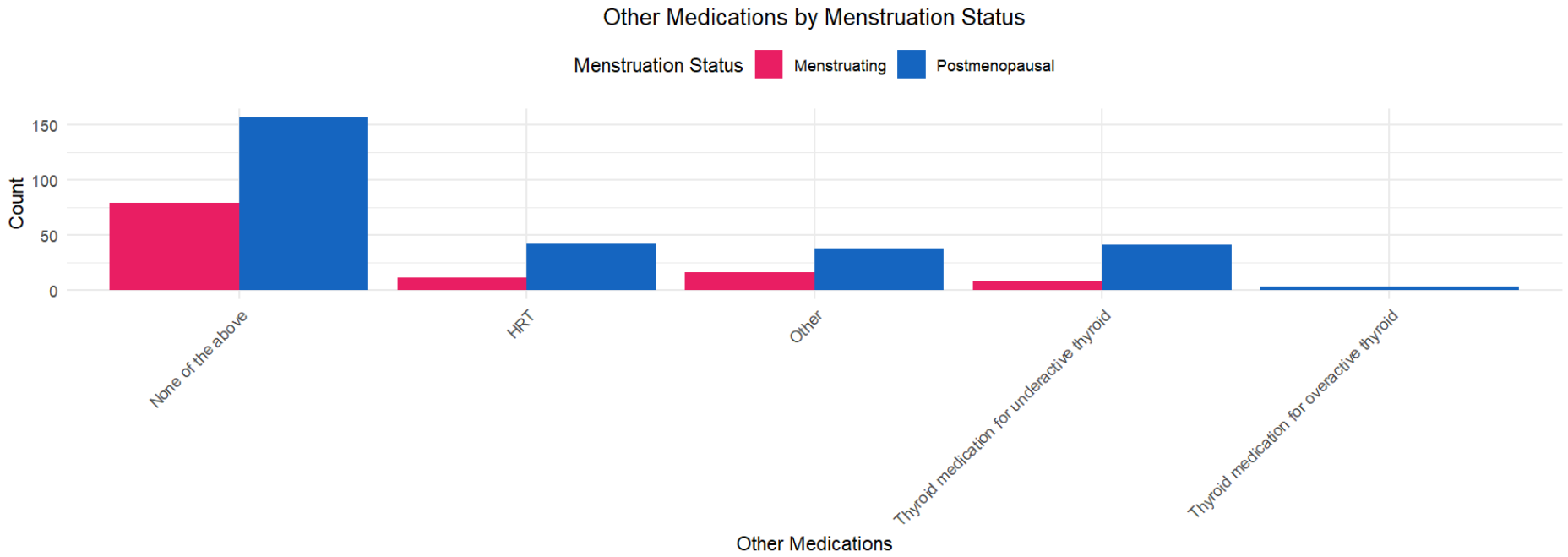
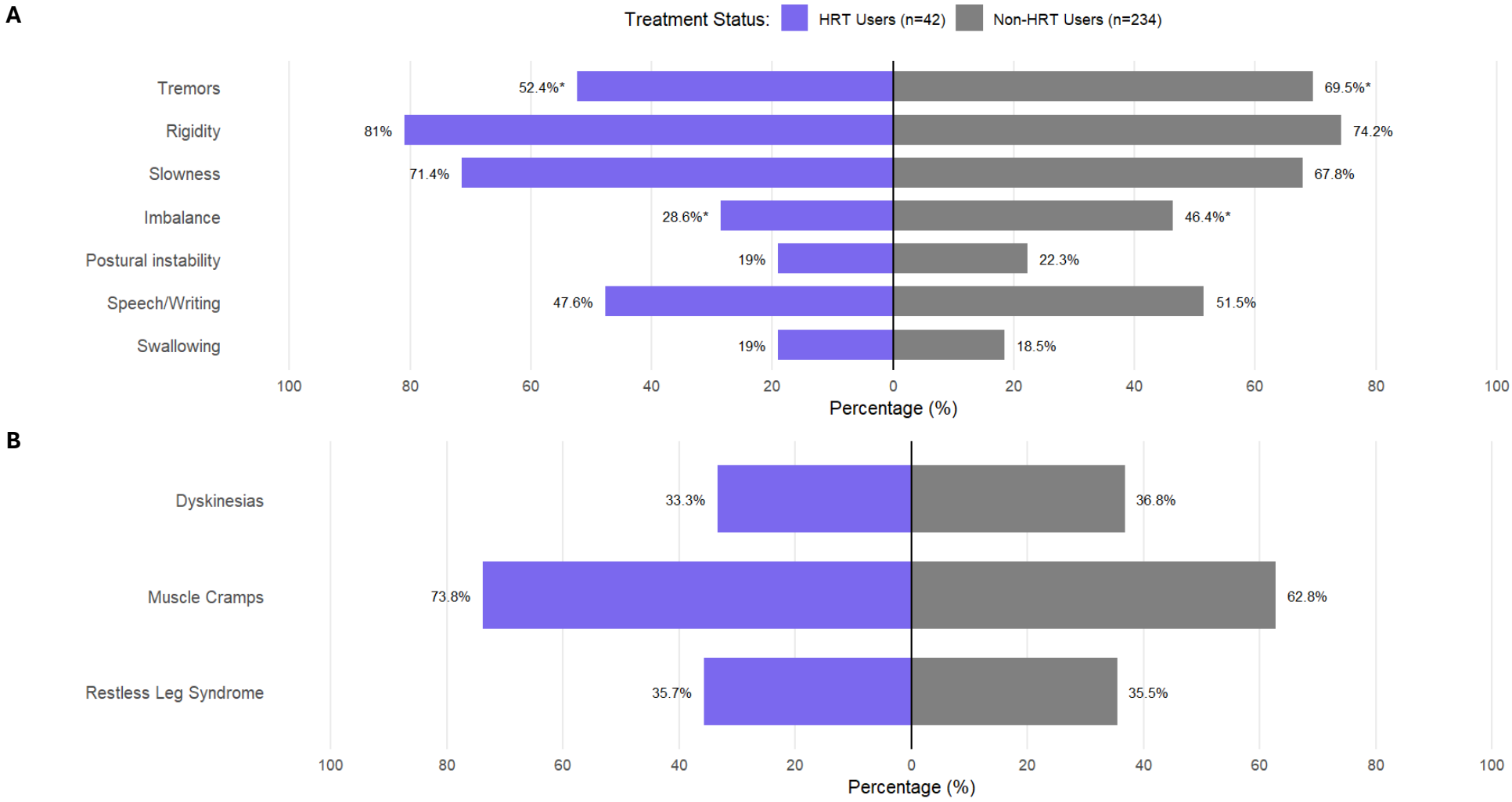


Figure S2(A-M): Demographic characteristics counts compared between menstruating (n=125, pink) and postmenopausal (n=283, blue) women.

Figure S3: HRT Analysis of Symptom in Users vs Non-Users



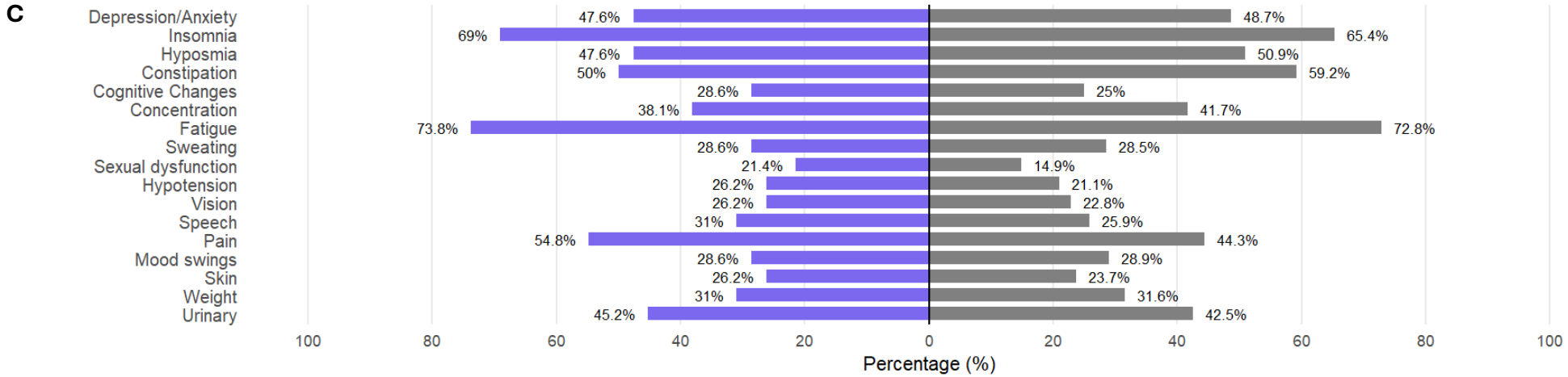


Figure S3: Comparison of (A)Motor symptom prevalence, (B) Secondary motor symptoms and (C) Non-motor symptom prevalence between n=42 HRT users (purple bars) and n=234 non-HRT users (grey bars).

Symptom Abbreviations:

Motor Symptoms: Tremors = Tremors or shaking, usually in the hands, arms, or legs; Rigidity = Stiffness or rigidity of the muscles; Slowness = Slowness of movement (bradykinesia); Imbalance = Difficulty with balance and coordination; Postural instability = Postural instability which can cause falls; Speech/Writing = Changes in speech and writing, such as slurred speech or small handwriting; Swallowing = Difficulty with swallowing.

Non-Motor Symptoms: Depression/Anxiety = Depression and anxiety; Insomnia = Sleep disturbances, including insomnia; Hyposmia = Loss of sense of smell; Constipation = Constipation; Cognitive Changes = Cognitive changes, including memory problems; Concentration = Difficulty concentrating; Fatigue = Fatigue and lack of energy; Sweating = Excessive sweating; Sexual dysfunction = Sexual dysfunction; Hypotension = Orthostatic hypotension; Vision = Visual disturbances, including blurred vision; Speech = Speech and swallowing problems; Pain = Pain and discomfort; Mood swings = Mood swings and emotional changes; Skin = Skin problems, including seborrheic dermatitis; Weight = Weight loss or gain; Urinary = Urinary Problems.

Table S2: MMM App Motor Symptoms: Menstruating vs Postmenopausal Comparison

	Sample Size		Mean Severity Score		Statistical Test
Symptom	Menstruating (n=45)	Postmenopausal (n=72)	Menstruating Severity	Postmenopausal Severity	P-value
Tremor	26	39	1.66	1.31	0.848
Dystonia	16	26	1.93	2.13	1
Rigidity	22	25	1.98	1.83	0.185
Bradykinesia	20	22	1.78	1.57	0.185
Difficulty with typing	10	16	1.91	1.64	1
Dyskinesia	14	10	1.91	1.59	0.045*
Micrographia	7	11	1.72	2.17	1
Balance loss	6	10	1.56	1.58	1
Clumsiness	9	9	1.52	1.18	0.406
Swallowing issues	8	7	1.12	1.72	0.325
Facial masking	2	5	1.25	2.07	0.706

*Table S2: Motor symptom prevalence and mean severity scores comparing menstruating (n=45) and postmenopausal (n=72) women using MyMovesMatter app tracking data. Sample sizes represent women who logged each symptom at least once during their app usage period. Mean severity scores calculated on 0-3 scale for participants who reported each symptom. Statistical significance determined using chi-squared tests, with Fisher's exact tests applied when expected cell counts <5. * p<0.05. Motor symptom abbreviations:*
Tremor = Tremor; Dystonia = Muscle cramps (Dystonia); Rigidity = Rigidity (stiff muscles); Bradykinesia = Slow movement (Bradykinesia); Difficulty with typing = Difficulty typing; Dyskinesia = Jerky movements (Dyskinesia); Micrographia = Small handwriting (Micrographia); Balance loss = Difficulty with balance and coordination; Clumsiness = Clumsiness; Swallowing issues = Swallow issues; Facial masking = Facial masking.

Table S3: MMM App Non-Motor Symptoms: Menstruating vs Postmenopausal Comparison

	Sample Size		Mean Severity Score		Statistical Test
Symptom	Menstruating (n=45)	Postmenopausal (n=72)	Menstruating Severity	Postmenopausal Severity	P-value
Constipated	10	28	1.44	1.66	0.095
Muscle/Joint Pains	16	24	1.75	1.66	0.963
Fatigue	22	22	2.19	1.77	0.073
Insomnia	17	20	1.84	1.68	0.354
Anxious	16	20	1.59	1.94	0.496
Dizzy	18	17	1.43	1.58	0.094
Irritable	17	9	1.66	2.25	0.003**
Bloated	16	15	1.75	1.91	0.124
Backache	11	15	1.98	1.72	0.819
Apathy	8	15	1.62	1.63	0.869
Crying Spells	14	8	1.79	1.67	0.014*
Exs Daytime Sleep	14	9	1.79	1.77	0.026*
Sudden Urge to Urinate	12	14	1.77	1.76	0.493
Hot Flushes	13	6	1.58	2.19	0.007**
Nausea	13	10	1.48	1.31	0.081
Brain Fog	12	13	2	1.66	0.382
Headache	11	13	1.66	1.38	0.55
Body Pain	9	13	2.11	1.91	0.985
SOB	6	13	1.81	1.39	0.677
Depressed	12	12	1.7	1.97	0.286
Pelvic Cramps	11	2	1.59	2.33	0.001***
Diff Concentrating	5	10	1.5	1.75	0.878
Fast Heartbeat	5	10	1.89	1.67	0.878
Hypophonia	2	10	1.75	1.37	0.126

Cravings	9	7	1.75	2.08	0.194
Diff Finding Words	9	9	1.28	1.69	0.406
RLS	8	9	1.7	1.61	0.604
Low BP	7	9	1.9	2.06	0.848
Exs Sweating	4	9	2.12	1.95	0.762
Low Libido	2	9	1.25	2.25	0.2
Vaginal Dryness	1	9	1	1.7	0.086
Dry Eyes	8	7	1.47	2.17	0.325
Anhedonia	6	8	1.96	2	0.946
Night Sweats	6	8	1.62	1.84	0.946
Pins & Needles	6	8	1.4	1.41	0.946
Memory Problems	5	8	1.31	1.17	1
Tinnitus	3	8	1.6	1.35	0.527
Mood Swings	7	3	1.92	1.8	0.043*
Sialorrhea	7	7	1.68	1.79	0.514
↑Appetite	7	2	1.83	1.33	0.026*
Diarrhea	5	7	1.62	1.88	1
Blurry Vision	6	5	1.24	1.75	0.331
Low Noise/Light Tolerance	5	5	1.75	1.8	0.505
Compulsivity	4	5	2.12	2.17	0.732
Diff MultiTasking	4	5	1.43	1.69	0.732

*Table S3: Non-motor symptom prevalence and mean severity scores comparing menstruating (n=45) and postmenopausal (n=72) women using MyMovesMatter app tracking data. Sample sizes represent women who logged each symptom at least once during their app usage period. Mean severity scores calculated on 0-3 scale for participants who reported each symptom. Statistical significance determined using chi-squared tests, with Fisher's exact tests applied when expected cell counts <5. * p<0.05, ** p<0.01, *** p<0.001.*

Non-motor symptom abbreviations:

Constipated = Constipated; Muscle/Joint Pains = Muscle and joint pains; Fatigue =

Fatigue; Insomnia = Difficulty in sleeping (Insomnia); Anxious = Feeling anxious, tense or nervous; Dizzy = Feeling dizzy or faint; Irritable = Feeling irritable; Bloated = Bloated; Backache = Backache; Apathy = Lack of interest (Apathy); Crying Spells = Crying spells; Exs Daytime Sleep = Excessive daytime sleepiness; Sudden Urge to Urinate = Sudden urge to urinate; Hot Flushes = Hot flushes; Nausea = Nausea; Brain Fog = Brain fog (Slowed thinking); Headache = Headache; Body Pain = Pain in your body; SOB = Shortness of breath; Depressed = Feeling depressed; Pelvic Cramps = Pelvic cramps; Diff Concentrating = Difficulty concentrating; Fast Heartbeat = Heart beating quickly or strongly; Hypophonia = Low voice (Hypophonia); Cravings = Food cravings; Diff Finding Words = Difficulty finding words; RLS = Restless leg syndrome; Low BP = Low blood pressure; Exs Sweating = Excessive sweating; Low Libido = Less interest in sex (Low Libido); Vaginal Dryness = Vaginal dryness; Dry Eyes = Dry eyes; Anhedonia = Lack of joy (Anhedonia); Night Sweats = Night Sweats; Pins & Needles = Pins and needles in any part of the body; Memory Problems = Memory problems; Tinnitus = Tinnitus (ringing or buzzing in the ear); Mood Swings = Mood swings; Sialorrhea = Excessive salivation (Sialorrhea); ↑Appetite = Increased appetite; Diarrhea = Diarrhoea; Blurry Vision = Blurry Vision; Low Noise/Light Tolerance = Lower tolerance for noise or light; Compulsivity = Compulsive and impulsive behaviour; Diff MultiTasking = Difficulty multi-tasking.

Table S4: MMM App Medication Categories: Menstruating vs Postmenopausal Comparison

	Sample Size		Statistical Test
Medication Category	**Menstruating (n=35)**	**Postmenopausal (n=64)**	**P-value**
L-DOPA	29	55	0.908
DA Agonist	19	27	0.346
MAO-B inhibitor	10	24	0.501
Estrogen-only	1	6	0.416
COMT Inhibitor	5	4	0.272

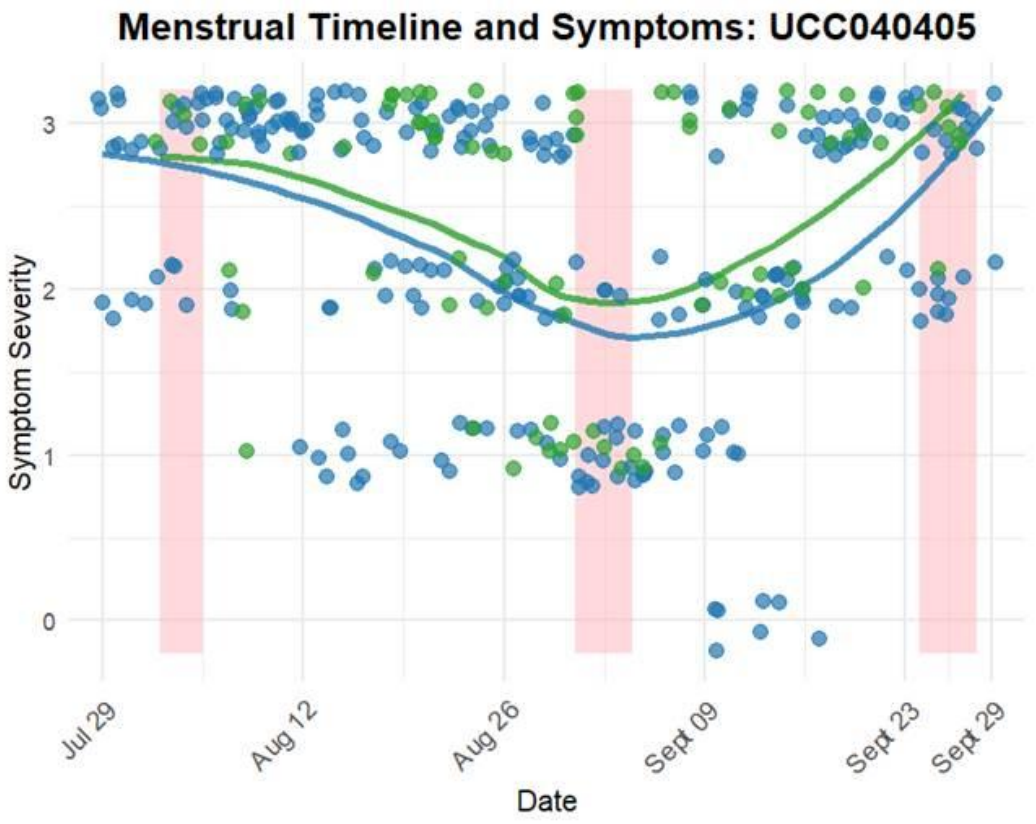
Table S4: PD Medication Options on MyMovesMatter App: Medication category prevalence rates comparing menstruating (n=35) and postmenopausal (n=64) women using MyMovesMatter app tracking data. Sample sizes represent women who logged each medication category at least once during their app usage period. Statistical significance determined using chi-squared tests, with Fisher's exact tests applied when expected cell counts <5.

L-DOPA: Caramet CR, Co- Beneldopa, Co-Careldopa, Dopicar 25/250mg, Duodopa,

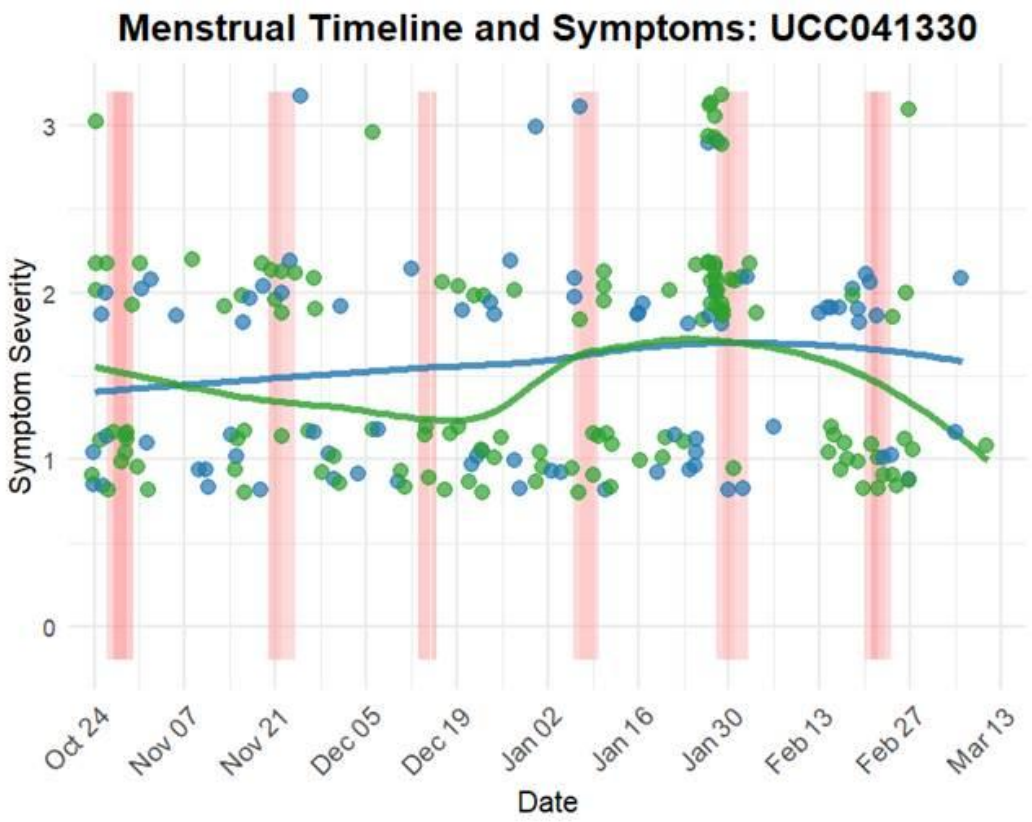
Duopa, Inbrija, Lecado, Madopar, Modopar, Parcopa, Prolopa , Rytary, Sastravi, Sinemet, Stalevo. DA Agonists: Adartel, Apo-go, Apokinon (apomorphine) stylo, Apokyn, Apomorphine, Bromocriptine, Dacepton, Gocovri, Ipinnia, Kynmobi, Mantadix (amantadine) 100 mg, Mirapex, Neupro, Oprymea, Osmolex, Parlodel, Pipexus, Pramipexole, RalneaXL, Raponer, Requip, Ropilynz, Ropinirole, Rotigotine, Sifrol, Symmetrel, Trivastal. MAO-B Inhibitor: Azilect, Deprenyl (selegiline), Eldepryl, Rasagiline, Safinamide, Selegiline, Xadago, Zelapar. COMT Inhibitors: Comtan, Comtess, Entacapone, Ongentys, Opicapone, Stanek, Tasmar (tolcapone), Tolcapone. Estrogen-only HRT: Divigel, Elleste-Solo, Estradot, Estreva, Estrofem, Estrogen only gel HRT, Estrogen only pill HRT, Evorel, Fematab, FemSeven, Imvaggis, Lenzetto, Oestrogel, Oromone, Ovestin, Physiogine, Premarin, Progynova, Provames, Sandrena, Vagifem, Zumenon.

Figure S4(A-Q): Individual Participant Menstrual Cycle and Symptom Timelines

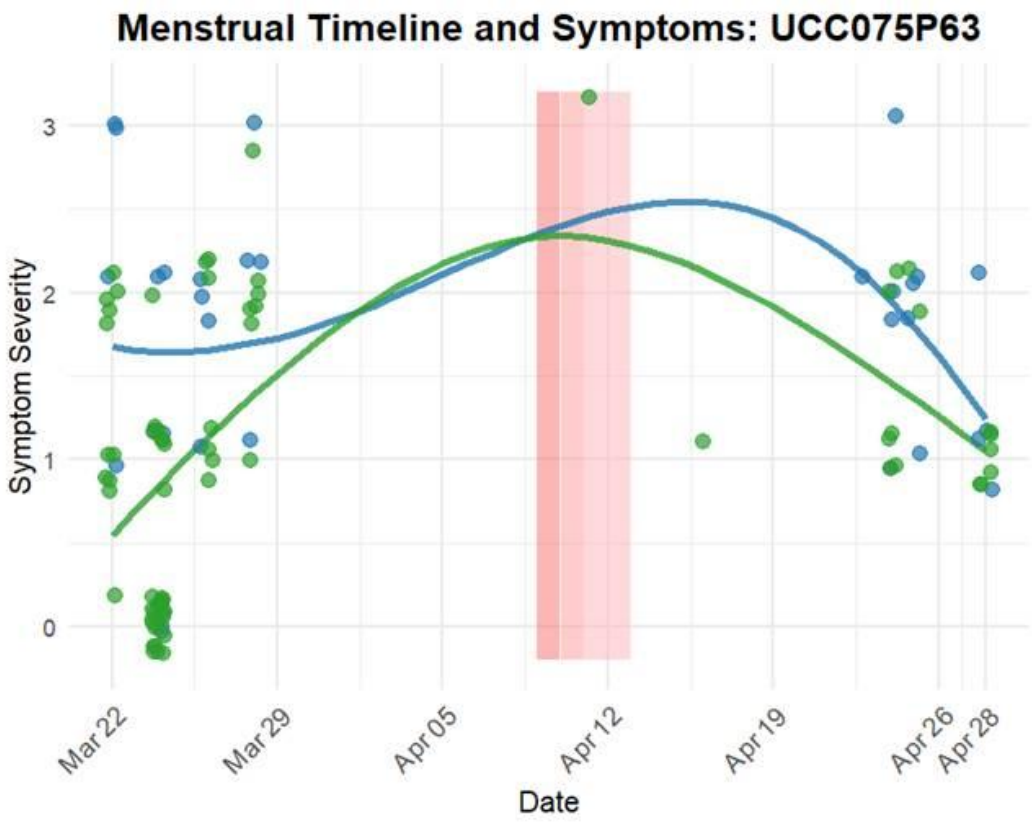
A



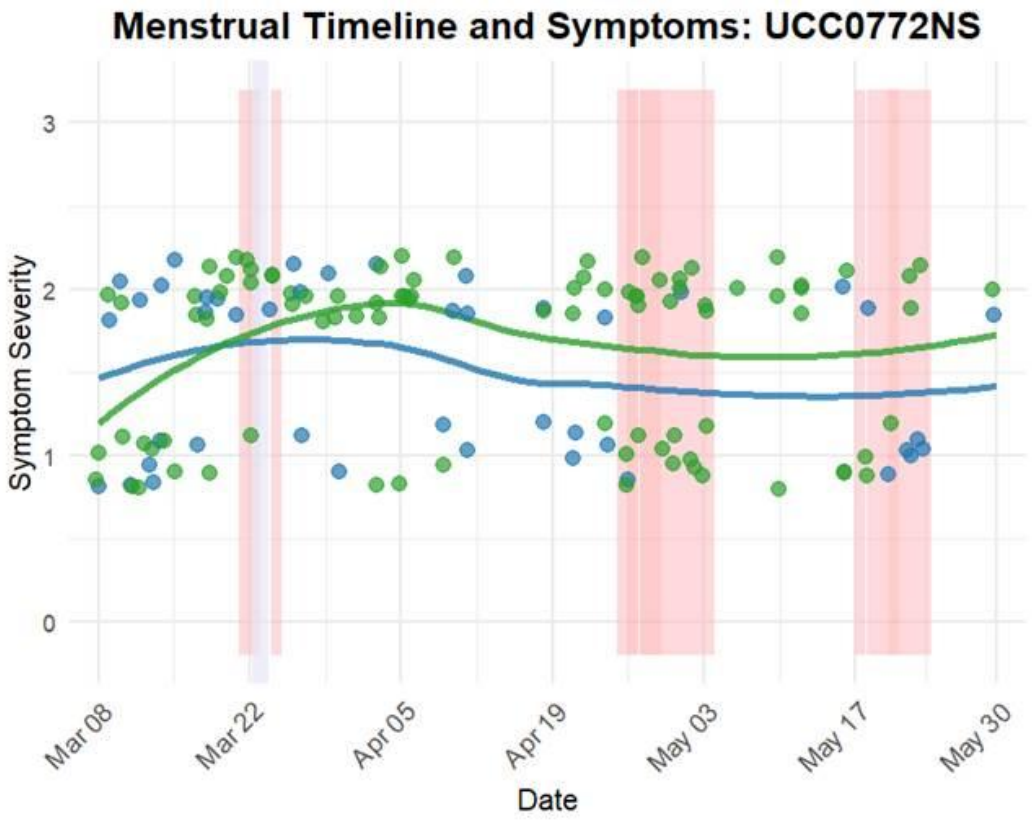
B


Menstrual Timeline and Symptoms: UCC041330
Symptom Severity
0
1
2
3
Oct 24
Nov 07
Nov 21
Dec 05
Dec 19
Jan 02
Jan 16
Jan 30
Feb 13
Feb 27
Mar 13
Date

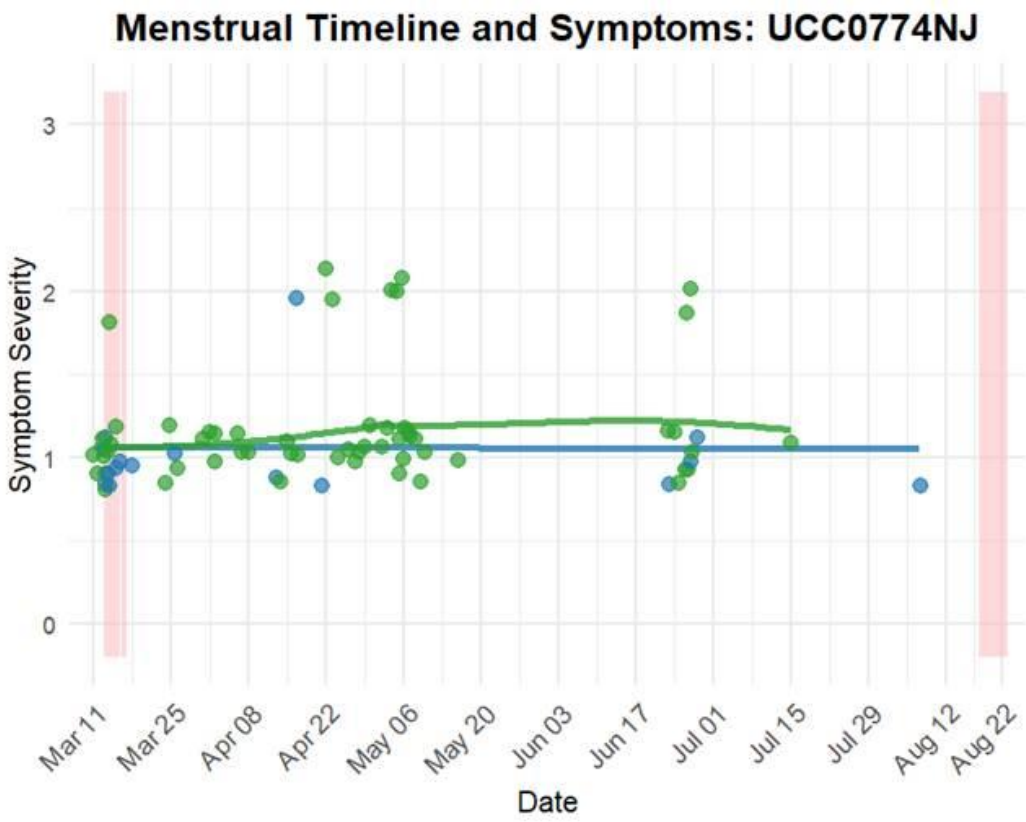
C


Menstrual Timeline and Symptoms: UCC075P63
Symptom Severity
0
1
2
3
Mar 22
Mar 29
Apr 05
Apr 12
Apr 19
Apr 26
Apr 28
Date

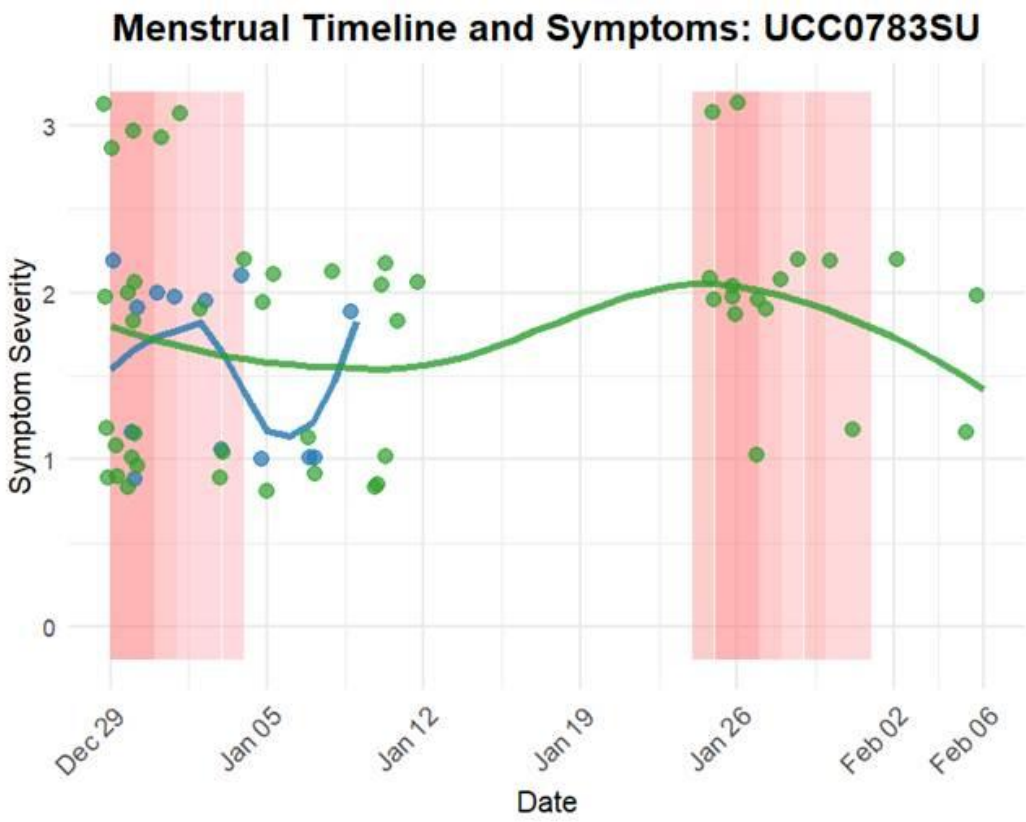
D


Menstrual Timeline and Symptoms: UCC0772NS
Symptom Severity
3
2
1
0
Mar 08
Mar 22
Apr 05
Apr 19
May 03
May 17
May 30
Date

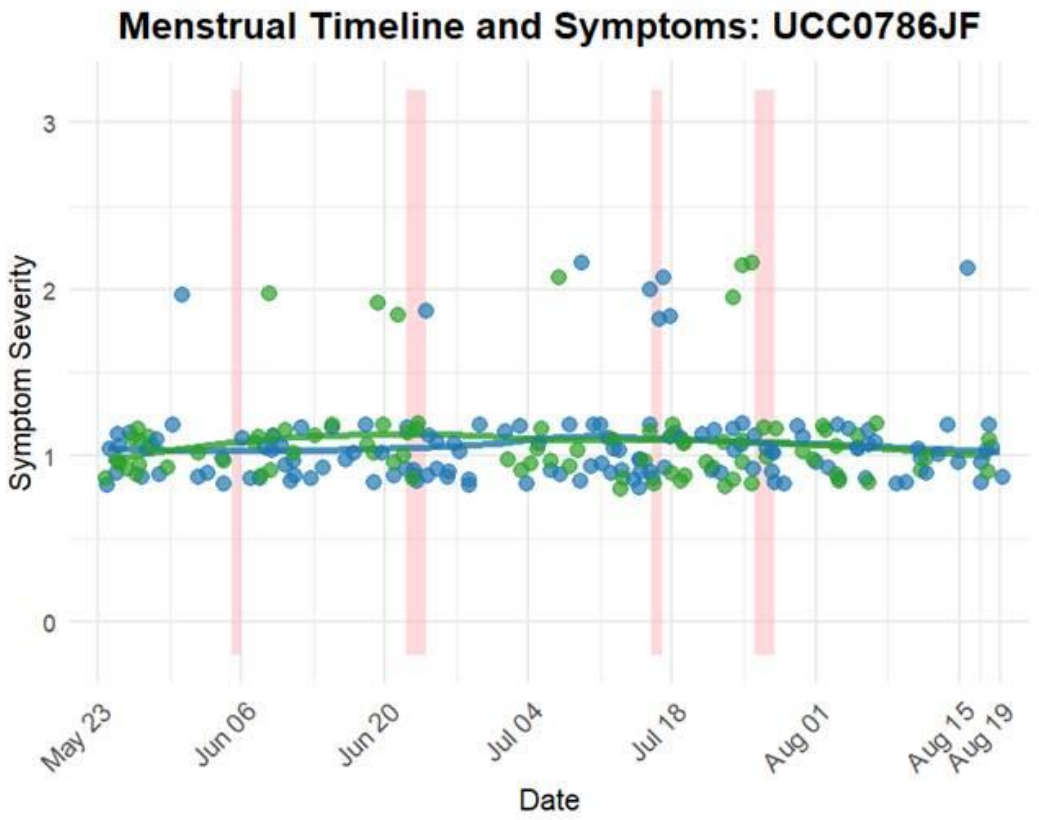
E


Menstrual Timeline and Symptoms: UCC0774NJ
Symptom Severity
0
1
2
3
Mar 11
Mar 25
Apr 08
Apr 22
May 06
May 20
Jun 03
Jun 17
Jul 01
Jul 15
Jul 29
Aug 12
Aug 22
Date

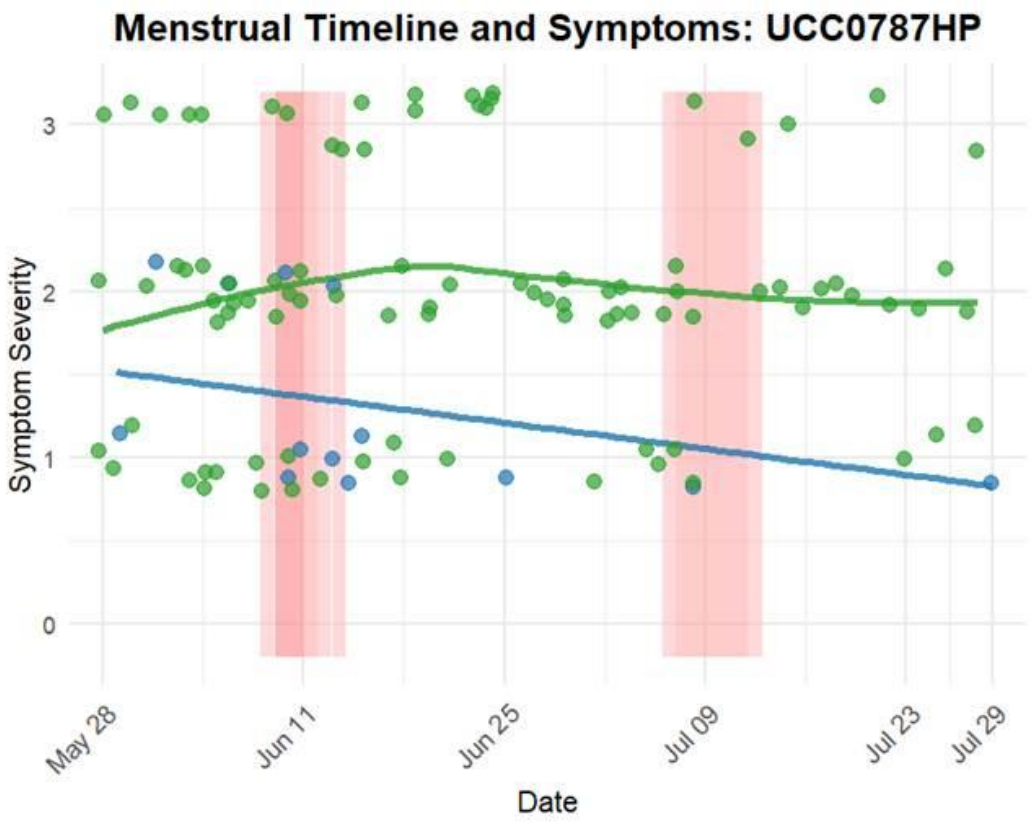
F


Menstrual Timeline and Symptoms: UCC0783SU
Symptom Severity
0
1
2
3
Dec 29
Jan 05
Jan 12
Jan 19
Jan 26
Feb 02
Feb 06
Date

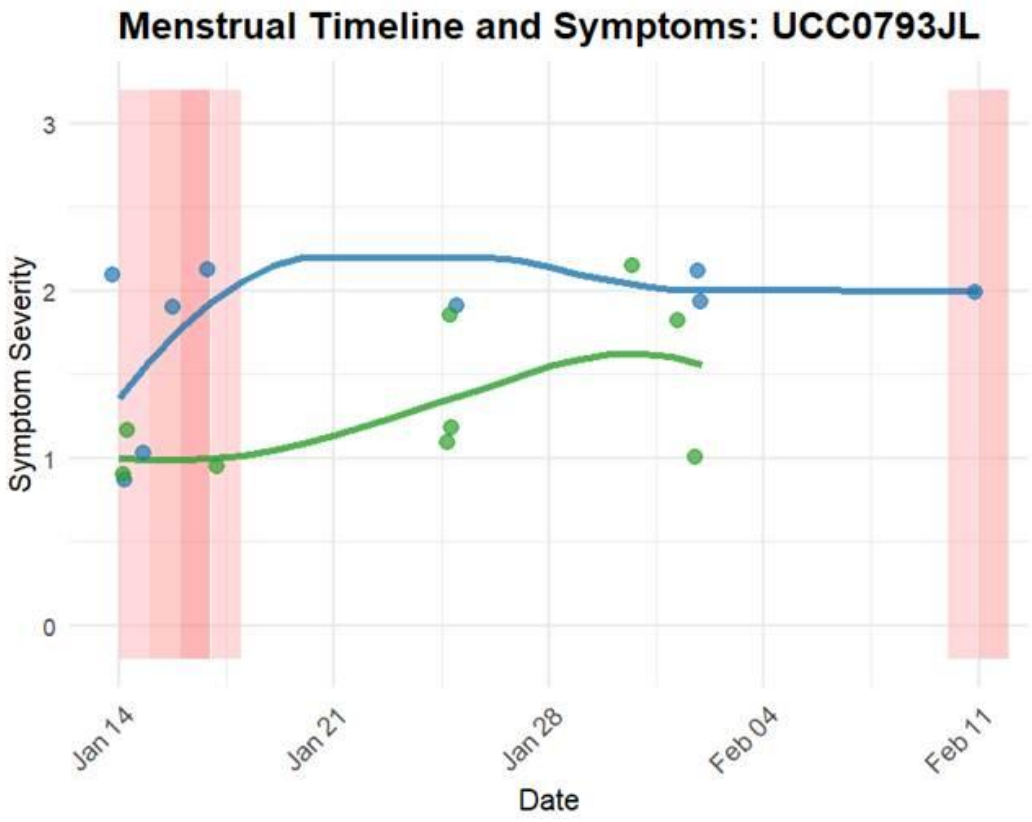
G


Menstrual Timeline and Symptoms: UCC0786JF
Symptom Severity
3
2
1
0
May 23
Jun 06
Jun 20
Jul 04
Jul 18
Aug 01
Aug 15
Aug 19
Date

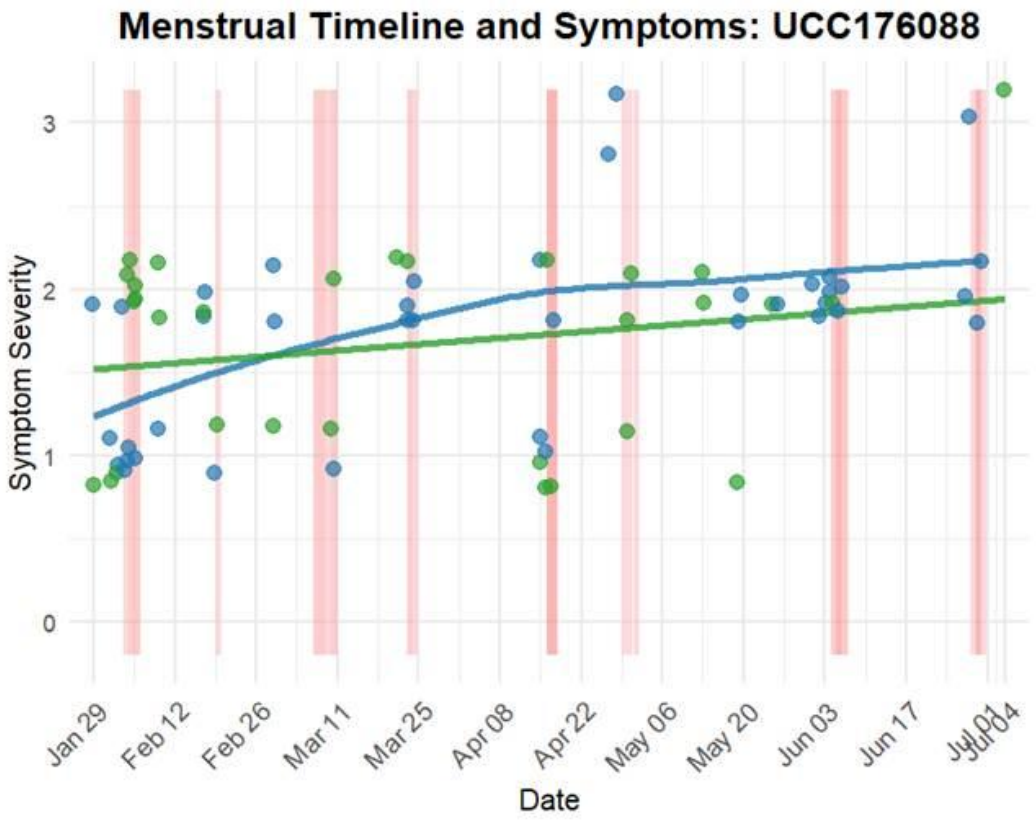
H


Menstrual Timeline and Symptoms: UCC0787HP
Symptom Severity
0
1
2
3
May 28
Jun 11
Jun 25
Jul 09
Jul 23
Jul 29
Date

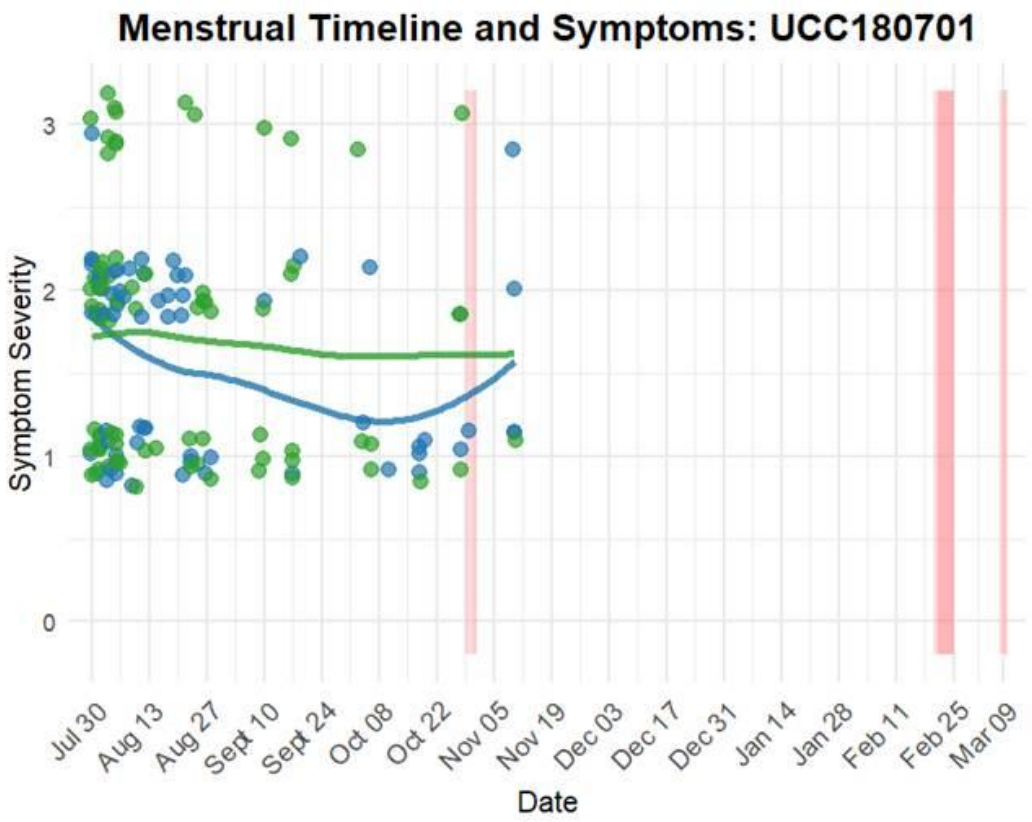
I


Menstrual Timeline and Symptoms: UCC0793JL
Symptom Severity
0
1
2
3
Jan 14
Jan 21
Jan 28
Feb 04
Feb 11
Date

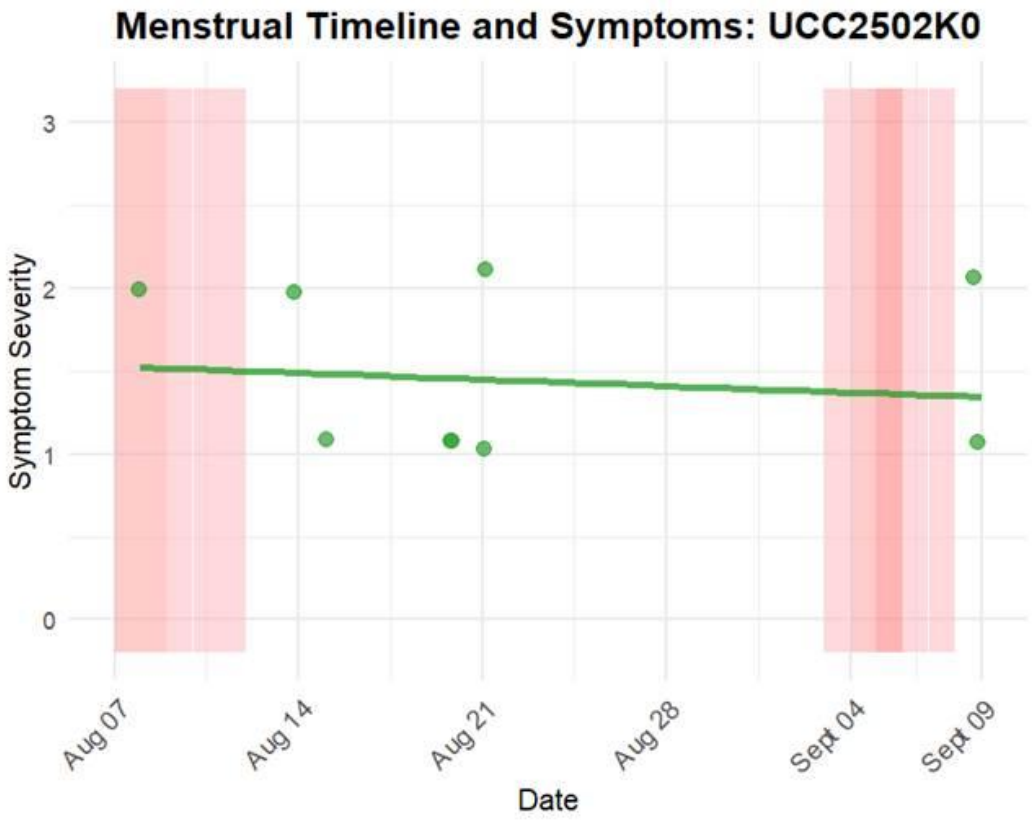
J


Menstrual Timeline and Symptoms: UCC176088
Symptom Severity
3
2
1
0
Jan 29
Feb 12
Feb 26
Mar 11
Mar 25
Apr 08
Apr 22
May 06
May 20
Jun 03
Jun 17
Jul 01
Jul 04
Date

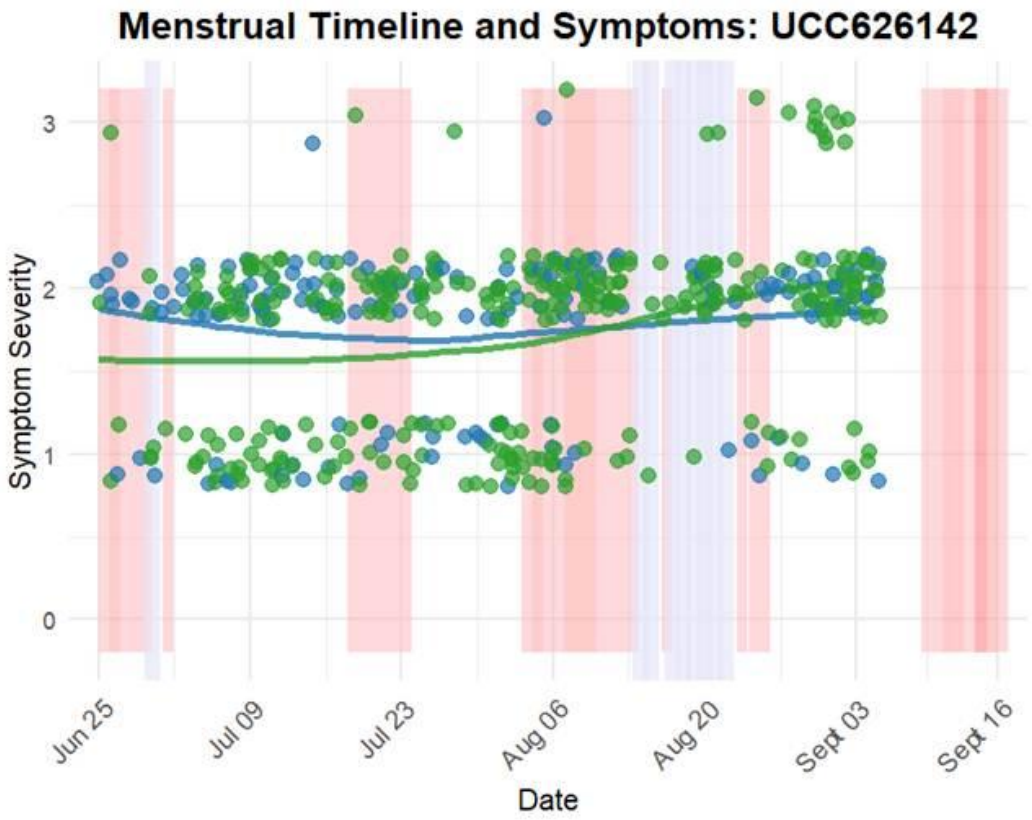
K


Menstrual Timeline and Symptoms: UCC180701
Symptom Severity
0
1
2
3
Jul 30
Aug 13
Aug 27
Sept 10
Sept 24
Oct 08
Oct 22
Nov 05
Nov 19
Dec 03
Dec 17
Dec 31
Jan 14
Jan 28
Feb 11
Feb 25
Mar 09
Date

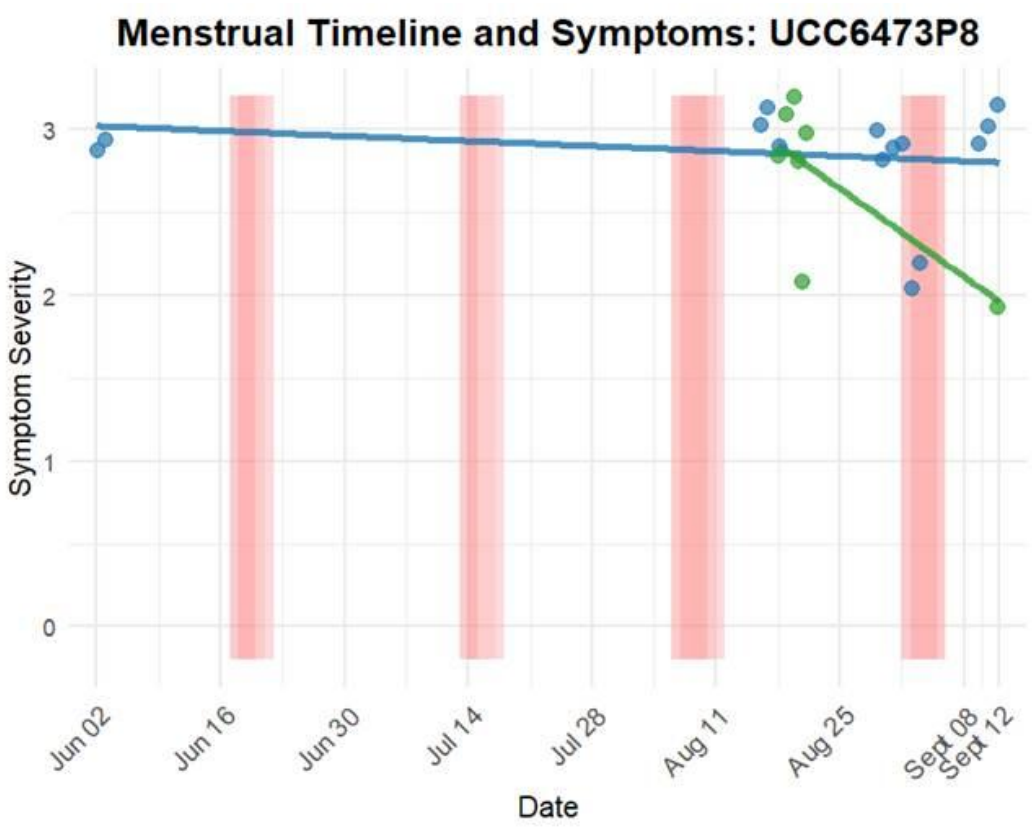
L


Menstrual Timeline and Symptoms: UCC2502K0
Symptom Severity
0
1
2
3
Aug 07
Aug 14
Aug 21
Aug 28
Sept 04
Sept 09
Date

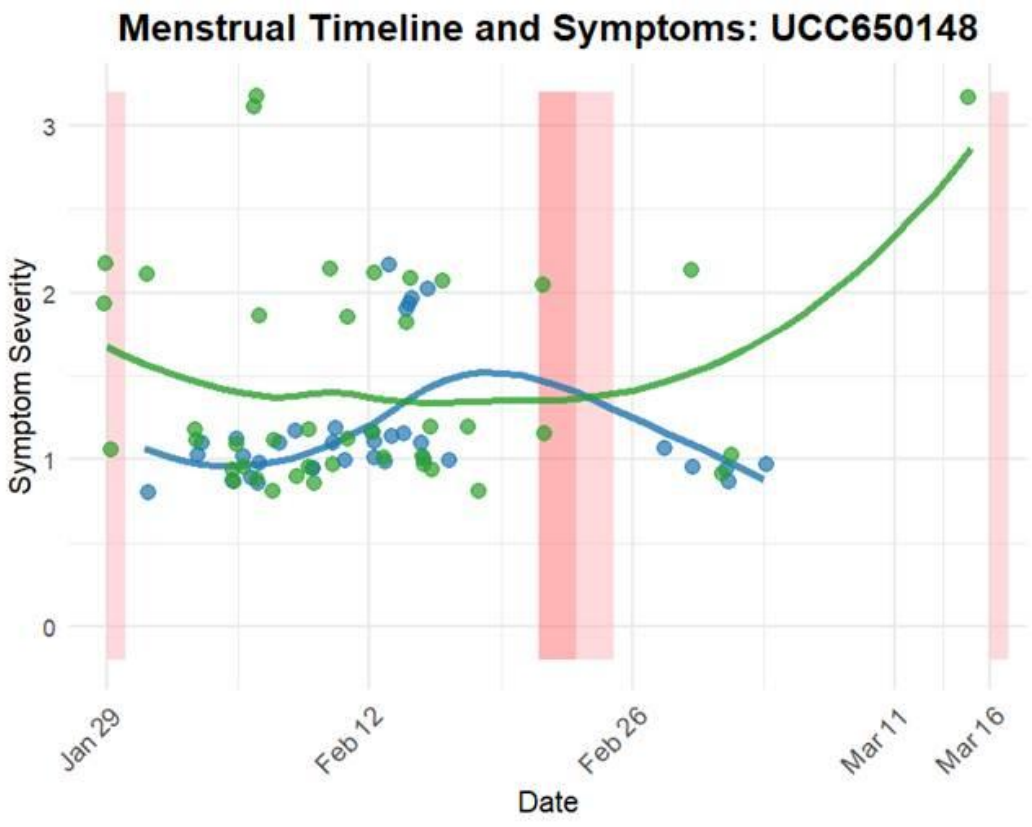
M


Menstrual Timeline and Symptoms: UCC626142
Symptom Severity
3
2
1
0
Jun 25
Jul 09
Jul 23
Aug 06
Aug 20
Sept 03
Sept 16
Date

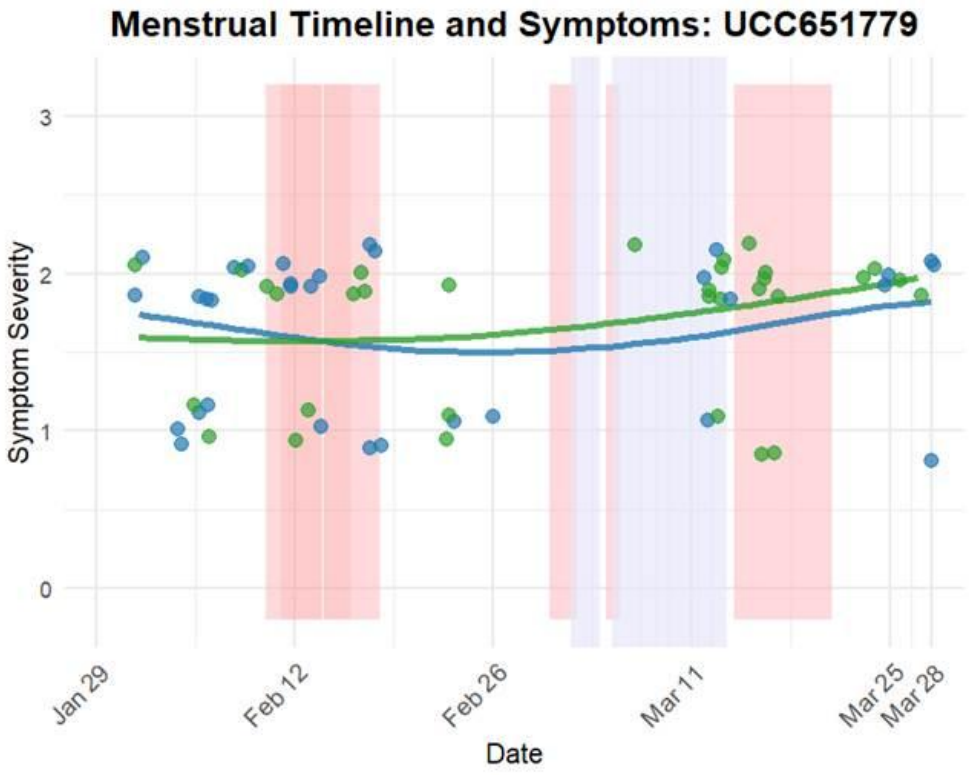
N


Menstrual Timeline and Symptoms: UCC6473P8
Symptom Severity
0
1
2
3
Jun 02
Jun 16
Jun 30
Jul 14
Jul 28
Aug 11
Aug 25
Sept 08
Sept 12
Date

O


Menstrual Timeline and Symptoms: UCC650148
Symptom Severity
0
1
2
3
Jan 29
Feb 12
Feb 26
Mar 11
Mar 16
Date

P


Menstrual Timeline and Symptoms: UCC651779
Symptom Severity
3
2
1
0
Jan 29
Feb 12
Feb 26
Mar 11
Mar 25
Mar 28
Date

Q

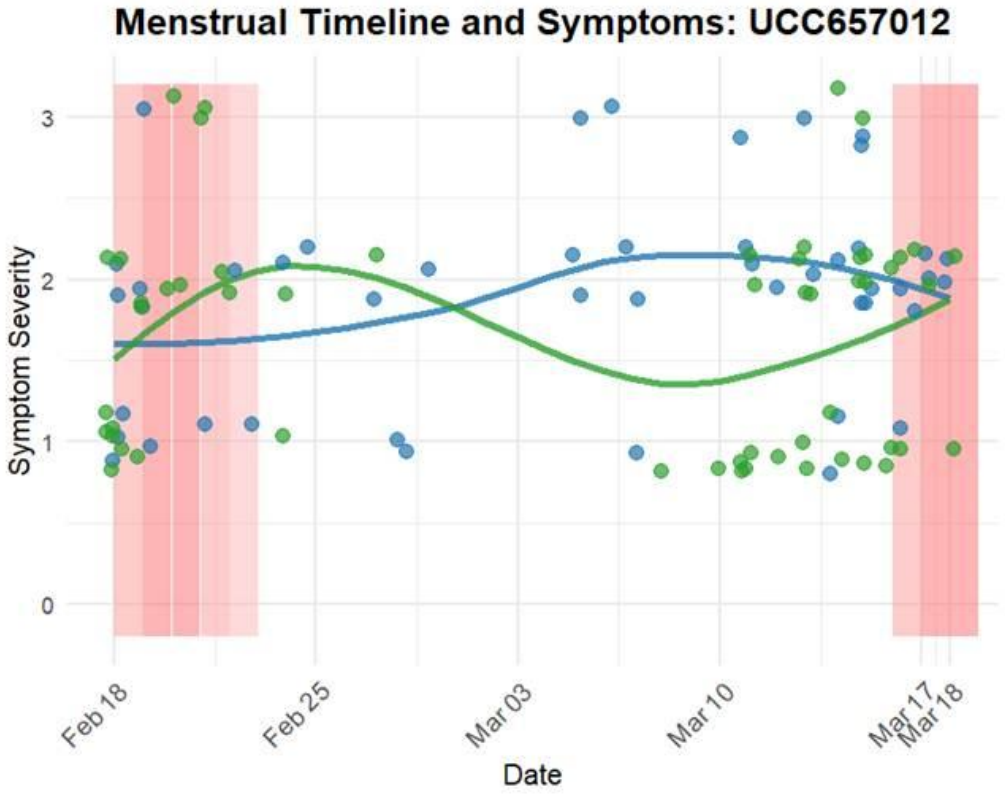

Menstrual Timeline and Symptoms: UCC657012
Symptom Severity
3
2
1
0
Feb 18
Feb 25
Mar 03
Mar 10
Mar 17
Mar 18
Date

Figure S4(A-Q): Individual timeline plots for n=17 menstruating women with Parkinson's disease showing motor (blue) and non-motor (green) symptom burden across menstrual cycles. Symptom severity scores plotted on 0-3 scale against date. Pink background indicates menstruation periods, with darker pink indicating heavier menstrual flow levels. Light purple indicates time points presumed to be associated with the same menstruation phase where no menstruation was explicitly logged.

Table S5: Motor Symptom Abbreviations in the Initial Survey

Original Label	**Abbreviated Label**
Tremors or shaking, usually in the hands, arms, or legs	Tremors
Stiffness or rigidity of the muscles	Rigidity
Slowness of movement (bradykinesia)	Slowness
Difficulty with balance and coordination	Imbalance
Postural instability, which can cause falls	Postural instability
Changes in speech and writing, such as slurring or small, cramped handwriting	Speech/Writing
Difficulty with swallowing	Swallowing

Table S5: Motor symptom descriptions used in the initial survey. Original symptom labels (left column) and corresponding abbreviated labels (right column).

Table S6: Non-motor Symptom Abbreviations in the Initial Survey

Original Label	**Abbreviated Label**
Depression and anxiety	Depression/Anxiety
Sleep disturbances, including insomnia and restless leg syndrome	Insomnia
Loss of sense of smell	Hyposmia
Constipation	Constipation
Cognitive changes, including memory problems, confusion, and dementia	Cognitive Changes
Difficulty Concentrating	Concentration
Fatigue and lack of energy	Fatigue
Excessive sweating	Sweating
Sexual dysfunction	Sexual dysfunction
Orthostatic hypotension (low blood pressure upon standing)	Hypotension
Visual disturbances, including blurry vision or double vision	Vision
Speech and swallowing difficulties	Speech
Pain and discomfort	Pain
Mood swings and emotional changes	Mood swings

Skin problems, including dry skin and rashes	Skin
Weight loss or gain	Weight
Urinary Problems	Urinary

Table S6: Non-motor symptom descriptions used in the initial survey. Original symptom labels (left column) and corresponding abbreviated labels (right column).

Table S7: Motor Symptom Options and Abbreviations in the MMM app

Original Label	**Abbreviated Label**
Tremor	Tremor
Rigidity (stiff muscles)	Rigidity
Slow movement (Bradykinesia)	Bradykinesia
Difficulty with balance and coordination	Balance loss
Clumsiness	Clumsiness
Small handwriting (Micrographia)	Micrographia
Swallow issues	Swallowing issues
Difficulty typing	Difficulty with typing
Facial masking	Facial masking
Freezing	Freezing
Jerky movements (Dyskinesia)	Dyskinesia
Muscle cramps (Dystonia)	Dystonia

Table S7: Motor symptom descriptions used in the MyMovesMatter (MMM) App. Original symptom labels (left column) and corresponding abbreviated labels (right column).

Table S8: Non-motor Symptom Options and Abbreviations in the MMM App

Original Label	**Abbreviated Label**
Anxiety attack, panic	Anxiety/Panic
Backache	Backache
Bloated	Bloating
Blurry Vision	Blurred vision
Brain fog (Slowed thinking)	Brain fog
Compulsive and impulsive behaviour	Compulsivity
Constipated	Constipation
Crying spells	Crying spells
Diarrhoea	Diarrhoea
Difficulty concentrating	Concentration issues
Difficulty finding words	Difficulty finding words
Difficulty in sleeping (Insomnia)	Insomnia
Difficulty multi-tasking	Difficulty multi-tasking
Dry eyes	Dry eyes

Dry skin (Seborrheic dermatitis)	Dry skin
Excessive daytime sleepiness	Daytime sleepiness
Excessive salivation (Sialorrhea)	Salivation
Excessive sweating	Sweating
Fatigue	Fatigue
Feeling anxious, tense or nervous	Tense/Nervous
Feeling depressed	Depression
Feeling dizzy or faint	Dizziness
Feeling irritable	Irritability
Food cravings	Cravings
Headache	Headache
Heart beating quickly or strongly	Fast heartbeat
Hot flushes	Hot flushes
Increased appetite	Increased appetite

Lack of interest (Apathy)	Apathy
Lack of joy (Anhedonia)	Anhedonia
Lack of sweating (hypo hidrosis)	Lack of sweating
Less interest in sex (Low Libido)	Low libido
Loss of taste (ageusia)	Loss of taste
Low blood pressure	Low BP
Low voice (Hypophonia)	Low voice
Lower tolerance for noise or light	Low tolerance to noise / light
Memory problems	Memory problems
Mood swings	Mood swings
Muscle and joint pains	Muscle / joint pains
Nausea	Nausea
Night Sweats	Night sweats
Pain in your body	Body pain

Pelvic cramps	Pelvic cramps
Pins and needles in any part of the body	Pins & needles
Reduced appetite	Low appetite
Restless leg syndrome	Restless legs
Shortness of breath	Shortness of breath
Sleep disordered breathing (sleep apnoea)	Sleep apnea
Sudden urge to urinate	Urinary urges
Talking, yelling, physically acting out while asleep	Movements during sleep
Tinnitus (ringing or buzzing in the ear)	Tinnitus
Vaginal dryness	Vaginal dryness
Weight gain	Weight gain
Weight loss	Weight loss

Loss of smell (hyposmia)	Hyposmia

Table S8: Non-motor symptom descriptions used in the MyMovesMatter (MMM) App. Original symptom labels (left column) and corresponding abbreviated labels (right column).

Figure S5: App Usability Follow-up Survey Analysis

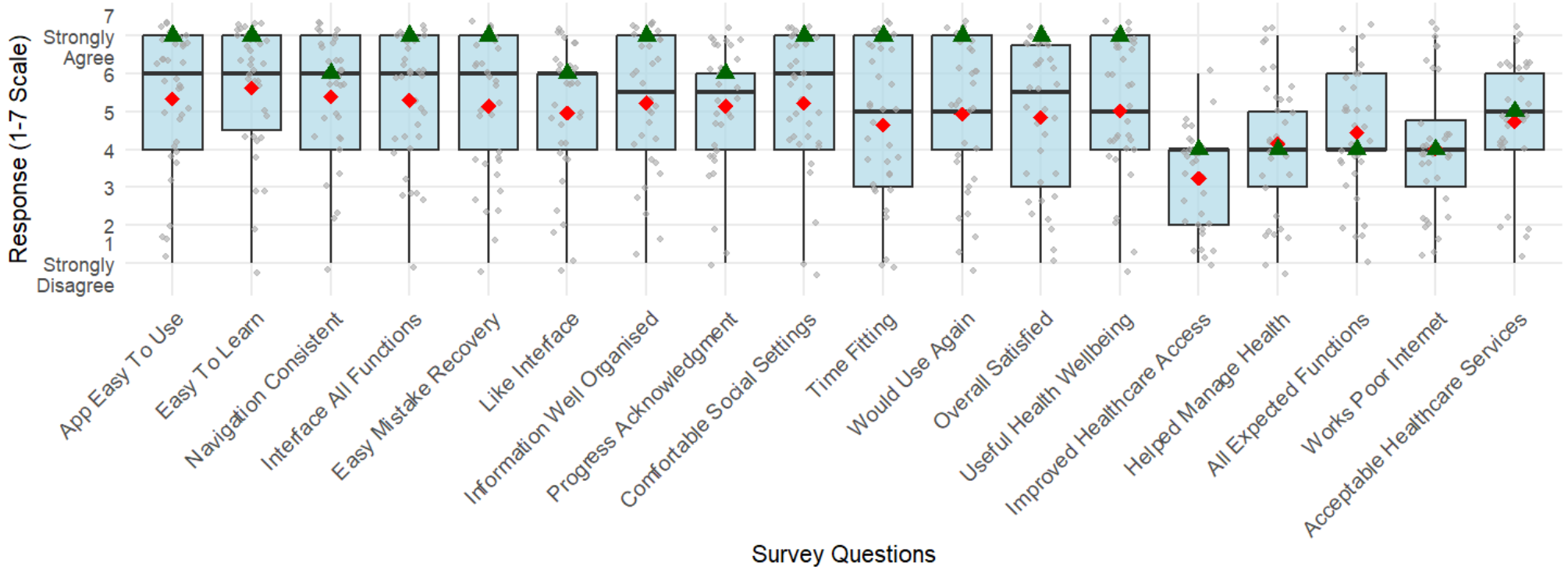


Figure S5: Individual question responses from the 18-question app usability survey. Box plots display response distributions for each usability question. Red diamonds indicate mean scores, green triangles indicate most common scores, grey dots represent individual participant responses. n=35 app users. Scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Somewhat Disagree, 4 = Neither Agree nor Disagree, 5 = Somewhat Agree, 6 = Agree, 7 = Strongly Agree.

Figure S6: Copy of Initial Survey

Unique Study Code

Your Unique Study Code is:

UCC[p][p][a]

Ensure you take note of the above unique study code and retain it for future use pertaining to the study.

Demographic Questionnaire

What is your age?	◯ 18 - 29 years old ◯ 30 - 39 years old ◯ 40 - 49 years old ◯ 50 - 54 years old ◯ 55 - 59 years old ◯ 60 - 64 years old ◯ 65 - 69 years old ◯ 70 - 74 years old ◯ 75 - 79 years old ◯ 80 - 84 years old ◯ 85 - 90 years old ◯ 90 years old ◯ Prefer not to say
What is your gender?	◯ Male (biological from birth) ◯ Female (biological from birth) ◯ Prefer not to say
What is your race/ethnicity?	◯ White ◯ Black ◯ Asian ◯ Other ◯ Prefer not to say
Please Specify "Other"	_____
What is your employment status?	◯ Employed (full-time) ◯ Employed (part-time) ◯ Self-employed ◯ Unemployed (actively seeking employment) ◯ Unemployed (not actively seeking employment) ◯ Student (full-time) ◯ Student (part-time) ◯ Homemaker ◯ Retired ◯ Disabled ◯ Other ◯ Prefer not to say
Please specify "Other"	_____
What is your marital status?	◯ Single ◯ In a relationship ◯ Married ◯ Widowed ◯ Divorced ◯ Separated ◯ Domestic partnership ◯ Civil union ◯ Prefer not to say
Have you ever been diagnosed with Parkinson's disease?	◯ Yes ◯ No ◯ Prefer not to say

This survey is directed towards women with Parkinson's Disease. Would you like to continue taking the survey?	◯ Yes ◯ No
What age were you when you were diagnosed with Parkinson's disease?	◯ 18 - 29 years old ◯ 30 - 39 years old ◯ 40 - 49 years old ◯ 50 - 54 years old ◯ 55 - 59 years old ◯ 60 - 64 years old ◯ 65 - 69 years old ◯ 70 - 74 years old ◯ 75 - 79 years old ◯ 80 - 84 years old ◯ 85 - 90 years old ◯ 90 years old ◯ Prefer not to say
What are the main motor symptoms you experience due to Parkinson's disease?	☐ Tremors or shaking, usually in the hands, arms, or legs ☐ Stiffness or rigidity of the muscles ☐ Slowness of movement (bradykinesia) ☐ Difficulty with balance and coordination ☐ Postural instability, which can cause falls ☐ Changes in speech and writing, such as slurring or small, cramped handwriting ☐ Reduced sense of smell ☐ Difficulty with swallowing ☐ Other _____ ☐ Prefer not to say
Do you experience any of the following?	☐ Dyskinesias ☐ Muscle Cramps ☐ Restless Leg Syndrome ☐ Other_____ ☐ Prefer not to say

Do you experience any non-motor symptoms associated with Parkinson's disease?	☐ Depression and anxiety ☐ Sleep disturbances, including insomnia and restless leg syndrome ☐ Loss of sense of smell ☐ Constipation ☐ Cognitive changes, including memory problems, confusion, and dementia ☐ Difficulty Concentrating ☐ Fatigue and lack of energy ☐ Excessive sweating ☐ Sexual dysfunction ☐ Orthostatic hypotension (low blood pressure upon standing) ☐ Visual disturbances, including blurry vision or double vision ☐ Speech and swallowing difficulties ☐ Pain and discomfort ☐ Mood swings and emotional changes ☐ Skin problems, including dry skin and rashes ☐ Weight loss or gain ☐ Urinary Problems ☐ Other_____ ☐ Prefer not to say
How long have you been experiencing Parkinson's symptoms?	◯ 0 - 3 years ◯ 4 - 6 years ◯ 7 - 9 years ◯ 10 - 12 years ◯ 13 - 15 years ◯ >15 years ◯ Prefer not to say

What medications are you currently taking for Parkinson's disease please select from below?

- ☐ Adartel
- ☐ Adodespan PR
- ☐ Amantadine
- ☐ Apogo
- ☐ Apokyn
- ☐ Apomorphine
- ☐ Artane
- ☐ Azilect
- ☐ Benztropine
- ☐ Bromocriptine
- ☐ Caramet CR
- ☐ Clonazepam
- ☐ Co-Beneldopa
- ☐ Co-Careldopa
- ☐ Comtan
- ☐ Comtess
- ☐ Cogentin
- ☐ Dacepton
- ☐ Duodopa
- ☐ Duopa
- ☐ Eldepryl
- ☐ Gocovri
- ☐ Inbrija
- ☐ Ipinnia
- ☐ Kemadrin
- ☐ Kynmobi
- ☐ Lecado
- ☐ Madopar
- ☐ Mirapex
- ☐ Mirapexin
- ☐ Neupro
- ☐ Nourianz
- ☐ Oprymea
- ☐ Ongentys
- ☐ Osmolex
- ☐ Opicapone
- ☐ Parlodel
- ☐ Parcopa
- ☐ Pipexus
- ☐ Pramipexole
- ☐ RalneaXL
- ☐ Rasagiline
- ☐ Requip
- ☐ Raponer
- ☐ Rivotril
- ☐ Ropilynz
- ☐ Ropinirole
- ☐ Rotigotine
- ☐ Rytary
- ☐ Sastravi
- ☐ Selegiline
- ☐ Slepark
- ☐ Sinemet
- ☐ Sinemet Plus
- ☐ Spirico
- ☐ Stalevo
- ☐ Stanek
- ☐ Symmetrel
- ☐ Tasmar
- ☐ Tolcapone
- ☐ Trihexyphenidryl/Benzhexol
- ☐ Xadago
- ☐ Zantiva
- ☐ Zelapar
- ☐ Other
- ☐ None of the above

What medications are you currently taking for Parkinson's disease please select from below?

Ardatel _____ Comtess _____ Neupro _____
Ropilynz _____ Trihexyphenidryl/Benzhexol _____
Adodespan PR _____ Cogentin _____ Nourianz _____
Ropinirole _____ Xadago _____
Amantadine _____ _____ Dacepton _____ Oprymea _____ Rotigotine _____ Zantiva _____
Apogo _____ Duodopa _____ Ongentys _____ Rytary _____ Zelapar _____
Apokyn _____ Duopa _____ Osmolex _____ Sastravi _____ Other _____
Apomorphine _____ Eldepryl _____ Opicapone _____ Selegiline _____ None of the above _____
Artane _____ Gocovri _____ Parlodel _____ Slepark _____
Azilect _____ Inbrija _____ Parcopa _____
Sinemet _____
Benztropine _____ Ipinnia _____ Pipexus _____
Sinemet Plus _____
Bromocriptine _____ Kemadrin _____ Pramipexole _____ Spirico _____
Caramet CR _____ Kynmobi _____ RalneaXL _____
Stalevo _____
Clonazepam _____ Lecado _____ Rasagiline _____
Stanek _____
Co-Beneldopa _____ Madopar _____ Requip _____ Symmetrel _____
Co-Careldopa _____ Mirapex _____ Raponer _____ Tasmar _____
Comtan _____ Mirapexin _____ Rivotril _____
Tolcapone _____

Please specify "Other"

_____

Are you currently taking any of the other medications listed below?

- ☐ Thyroid medication for underactive thyroid
- ☐ Thyroid medication for overactive thyroid
- ☐ HRT
- ☐ Other
- ☐ None of the above

Please specify "Other"

_____

Do you take an oral Hormonal Contraceptive? Please specify below:

- ☐ an implant
- ☐ an intrauterine device (IUD)
- ☐ injections,
- ☐ oral contraception
- ☐ vaginal rings
- ☐ skin patches
- ☐ If you take one of the above, please specify the name of the medication below _____
- ☐ I am not currently taking any of these medications

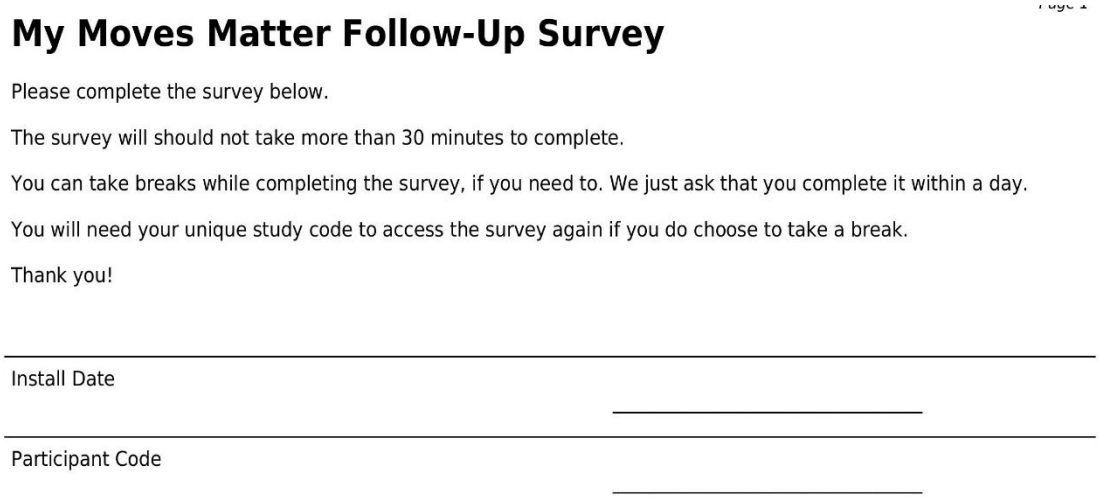
What treatments are you currently using for your Parkinson's Disease?	☐ Physical therapy ☐ Speech therapy ☐ Occupational therapy ☐ Nutrition therapy ☐ None of the above ☐ Prefer not to say ☐ Other_____
Have you participated in any clinical trials or research studies related to Parkinson's disease?	◯ Yes ◯ No ◯ Prefer not to say
	◯ Clinical Trial ◯ Research Study
If yes, please specify	_____
If yes, please specify	_____
Do you have any family members who also have Parkinson's disease?	◯ Yes ◯ No ◯ Prefer not to say ◯ I don't know
Have you ever had any surgeries or procedures related to your Parkinson's disease?	◯ Yes _____ ◯ No ◯ Prefer not to say
Do you have any of the following conditions:	☐ Hypothyroidism ☐ Hyperthyroidism ☐ Polycystic ovary Syndrome (PCOS) ☐ Premenstrual dysphoric disorder (PMDD) ☐ Other Menstrual disorders _____ ☐ Other _____ ☐ I do not have any of these conditions
Which category do you fall into from the below:	◯ Actively menstruating ◯ Post menstruation (no active period for 1 year) ◯ Pregnant ◯ Other ◯ Prefer not to say
Please specify "Other"	_____

Have you experienced any of the following in the past month?	☐ Dribbling of saliva during the daytime. ☐ Loss or change in your ability to taste or smell ☐ Difficulty swallowing food or drink or problems with choking. ☐ Vomiting or feelings of sickness (nausea) ☐ Constipation (less than three bowel movements a week) or having to strain to pass a stool. ☐ Bowel (faecal) incontinence. ☐ Feeling that your bowel emptying is incomplete after having been to the toilet. ☐ A sense of urgency to pass urine makes you rush to the toilet ☐ Getting up regularly at night to pass urine. ☐ Unexplained pains (not due to known conditions such as arthritis). ☐ Unexplained change in weight (not due to change in diet). ☐ Problems remembering things that have happened recently or forgetting to do things. ☐ Loss of interest in what is happening around you or in doing things. ☐ Seeing or hearing things that you know or are told are not there. ☐ Difficulty concentrating or staying focused. ☐ Feeling sad, 'low' or 'blue'. ☐ Feeling anxious, frightened or panicky. ☐ Feeling less interested in sex or more interested in sex. ☐ Finding it difficult to have sex when you try. ☐ Feeling light-headed, dizzy or weak standing from sitting or lying. ☐ Falling. ☐ Finding it difficult to stay awake during activities such as working, driving or eating. ☐ Difficulty getting to sleep at night or staying asleep at night. ☐ Intense, vivid or frightening dreams. ☐ Talking or moving about in your sleep, as if you are 'acting out' a dream. ☐ Unpleasant sensations in your legs at night or while resting, and a feeling that you need to move. ☐ Swelling of the legs. ☐ Excessive sweating. ☐ Double vision. ☐ Believing things are happening to you that other people say are not. ☐ I have not experienced any of these in the past month.
UUID	_____
Start Date	_____
End Date	_____
Schedule Date	_____

Status	◯ Deleted ◯ Completed ◯ Incomplete
Supplemental Data (JSON)	_____
Serialized Result	

Figure S6: Complete initial survey form created using REDCap showing all three pages: participant information leaflet and consent agreement form, unique identifier code generation, and questionnaire page.

Figure S7: Copy of Follow-up Survey



My Moves Matter Follow-Up Survey

Please complete the survey below.

The survey will should not take more than 30 minutes to complete.

You can take breaks while completing the survey, if you need to. We just ask that you complete it within a day.

You will need your unique study code to access the survey again if you do choose to take a break.

Thank you!

Install Date _____

Participant Code _____

Unique Study Code

Please enter the unique study code that you created in the initial survey at the beginning of the study and used the My Moves Matter app.
Your unique study code should comprise of the first three digits of your phone number and the last three digits/letters of your postcode.

1st three digits of phone number _____

Last three letters/numbers of postcode _____

UCC[p2][p2][a2]

Part 1: My Moves Matter Parkinson's Follow-Up Survey

What are the main motor symptoms you experience due to Parkinson's disease?	☐ Tremors or shaking, usually in the hands, arms, or legs ☐ Stiffness or rigidity of the muscles ☐ Slowness of movement (bradykinesia) ☐ Difficulty with balance and coordination ☐ Postural instability, which can cause falls ☐ Changes in speech and writing, such as slurring or small, cramped handwriting ☐ Reduced sense of smell ☐ Difficulty with swallowing ☐ Other _____ ☐ Prefer not to say
Do you experience any non-motor symptoms associated with Parkinson's disease?	☐ Depression ☐ Anxiety ☐ Sleep disturbances, including insomnia and restless leg syndrome ☐ Loss of sense of smell ☐ Constipation ☐ Cognitive changes, including memory problems, confusion, and dementia ☐ Difficulty Concentrating ☐ Fatigue and lack of energy ☐ Excessive sweating ☐ Sexual dysfunction ☐ Orthostatic hypotension (low blood pressure upon standing) ☐ Visual disturbances, including blurry vision or double vision ☐ Speech and swallowing difficulties ☐ Pain and discomfort ☐ Mood swings and emotional changes ☐ Skin problems, including dry skin and rashes ☐ Weight loss or gain ☐ Urinary Problems ☐ Other _____ ☐ Prefer not to say

Which non-motor symptoms do you feel should be prioritised in order to provide support for people with Parkinson's disease?

- ☐ Depression
- ☐ Anxiety
- ☐ Sleep disturbances, including insomnia and restless leg syndrome
- ☐ Loss of sense of smell
- ☐ Constipation
- ☐ Cognitive changes, including memory problems, confusion, and dementia
- ☐ Difficulty Concentrating
- ☐ Fatigue and lack of energy
- ☐ Excessive sweating
- ☐ Sexual dysfunction
- ☐ Orthostatic hypotension (low blood pressure upon standing)
- ☐ Visual disturbances, including blurry vision or double vision
- ☐ Speech and swallowing difficulties
- ☐ Pain and discomfort
- ☐ Mood swings and emotional changes
- ☐ Skin problems, including dry skin and rashes
- ☐ Weight loss or gain
- ☐ Urinary Problems
- ☐ Other _____
- ☐ Prefer not to say

What medications are you currently taking for Parkinson's disease please select from below?

Ardatel _____ Comtess _____ Neupro _____
Ropilynz _____ Trihexyphenidryl/Benzhexol _____
Adodespan PR _____ Cogentin _____ Nourianz _____
Ropinirole _____ Xadago _____
Amantadine _____ _____ Dacepton _____ Oprymea _____ Rotigotine _____ Zantiva _____
Apogo _____ Duodopa _____ Ongentys _____ Rytary _____ Zelapar _____
Apokyn _____ Duopa _____ Osmolex _____ Sastravi _____ Other _____
Apomorphine _____ Eldepryl _____ Opicapone _____ Selegiline _____ None of the above _____
Artane _____ Gocovri _____ Parlodel _____ Slepark _____
Azilect _____ Inbrija _____ Parcopa _____ Sinemet _____
Benztropine _____ Ipinnia _____ Pipexus _____ Sinemet Plus _____
Bromocriptine _____ Kemadrin _____ Pramipexole _____ Spirico _____
Caramet CR _____ Kynmobi _____ RalneaXL _____ Stalevo _____
Clonazepam _____ Lecado _____ Rasagiline _____ Stanek _____
Co-Beneldopa _____ Madopar _____ Requip _____ Symmetrel _____
Co-Careldopa _____ Mirapex _____ Raponer _____ Tasmar _____
Comtan _____ Mirapexin _____ Rivotril _____ Tolcapone _____

Are you currently taking any of the other medications listed below?

- ☐ Thyroid medication for underactive thyroid
- ☐ Thyroid medication for overactive thyroid
- ☐ HRT
- ☐ Other _____
- ☐ I am not currently taking any of these medications

Do you take an Hormonal Contraceptive? Please specify below:	☐ an implant ☐ an intrauterine device (IUD) ☐ injections, ☐ oral contraception ☐ vaginal rings ☐ skin patches ☐ I am not currently taking any of these medications
Please specify name of Hormaonal Contraceptive	_____
What treatments are you currently using for your Parkinson's Disease?	☐ Physical therapy ☐ Speech therapy ☐ Occupational therapy ☐ Nutrition therapy ☐ None of the above ☐ Prefer not to say ☐ Other _____
Have you ever had any surgeries or procedures related to your Parkinson's disease?	◯ Yes _____ ◯ No ◯ Prefer not to say
Have you experienced any of the following in the past month?	☐ Dyskinesias ☐ Muscle Cramps ☐ Restless Leg Syndrome ☐ Other _____ ☐ Prefer not to say
Which category do you fall into from the below:	◯ Actively menstruating ◯ Post menstruation (no active period for 1 year) ◯ Pregnant ◯ Other _____ ◯ Prefer not to say
Before being diagnosed with Parkinson's disease, did you have a history of poor mental health (e.g., depression, low mood, anxiety, or other)?	◯ Yes ◯ No ◯ Prefer not to say
How do you feel your mood or mental health symptoms compare to before your Parkinson's diagnosis?	◯ The same as before Parkinson's diagnosis ◯ Better ◯ Worse ◯ Prefer not to say
Have you availed of any mental health supports due to your Parkinson's disease?	☐ Counselling ☐ Psychology ☐ Psychotherapy ☐ Psychiatry ☐ Neuropsychology ☐ None of the above ☐ Prefer not to say
Have you experienced any significant life events or changes since you completed our initial survey?	_____
How did you find your experience tracking on the app overall?	_____

Have you experienced any of the following in the past month?	☐ Dribbling of saliva during the daytime. ☐ Loss or change in your ability to taste or smell ☐ Difficulty swallowing food or drink or problems with choking. ☐ Vomiting or feelings of sickness (nausea) ☐ Constipation (less than three bowel movements a week) or having to strain to pass a stool. ☐ Bowel (faecal) incontinence. ☐ Feeling that your bowel emptying is incomplete after having been to the toilet. ☐ A sense of urgency to pass urine makes you rush to the toilet ☐ Getting up regularly at night to pass urine. ☐ Unexplained pains (not due to known conditions such as arthritis). ☐ Unexplained change in weight (not due to change in diet). ☐ Problems remembering things that have happened recently or forgetting to do things. ☐ Loss of interest in what is happening around you or in doing things. ☐ Seeing or hearing things that you know or are told are not there. ☐ Difficulty concentrating or staying focused. ☐ Feeling sad, 'low' or 'blue'. ☐ Feeling anxious, frightened or panicky. ☐ Feeling less interested in sex or more interested in sex. ☐ Finding it difficult to have sex when you try. ☐ Feeling light-headed, dizzy or weak standing from sitting or lying. ☐ Falling. ☐ Finding it difficult to stay awake during activities such as working, driving or eating. ☐ Difficulty getting to sleep at night or staying asleep at night. ☐ Intense, vivid or frightening dreams. ☐ Talking or moving about in your sleep, as if you are 'acting out' a dream. ☐ Unpleasant sensations in your legs at night or while resting, and a feeling that you need to move. ☐ Swelling of the legs. ☐ Excessive sweating. ☐ Double vision. ☐ Believing things are happening to you that other people say are not. ☐ I have not experienced any of these in the past month.

Part 2: My Moves Matter Pilot Study App Usability Survey

In this questionnaire, 1 - strongly disagree, 2 - disagree, 3 - somewhat disagree, 4 - neither agree nor disagree, 5 - somewhat agree, 6 - agree, 7 - strongly agree

	1	2	3	4	5	6	7
The app was easy to use.	○	○	○	○	○	○	○
It was easy for me to learn to use the app.	○	○	○	○	○	○	○
The navigation was consistent when moving between screens.	○	○	○	○	○	○	○
The interface of the app allowed me to use all the functions (such as entering information, responding to reminders, viewing information) offered by the app.	○	○	○	○	○	○	○
Whenever I made a mistake using the app, I could recover easily and quickly.	○	○	○	○	○	○	○
I like the interface of the app.	○	○	○	○	○	○	○
The information in the app was well organized, so I could easily find the information I needed.	○	○	○	○	○	○	○
The app adequately acknowledged and provided information to let me know the progress of my action	○	○	○	○	○	○	○
I feel comfortable using this app in social settings.	○	○	○	○	○	○	○
The amount of time involved in using this app has been fitting for me.	○	○	○	○	○	○	○
I would use this app again.	○	○	○	○	○	○	○
Overall, I am satisfied with this app.	○	○	○	○	○	○	○
The app would be useful for my health and well-being.	○	○	○	○	○	○	○
The app improved my access to healthcare services.	○	○	○	○	○	○	○
The app helped me manage my health effectively.	○	○	○	○	○	○	○
This app has all the functions and capabilities I expected it to have.	○	○	○	○	○	○	○

I could use the app even when the Internet connection was poor or not available.	○	○	○	○	○	○	○
This MYMovesMatter app provides an acceptable way to receive healthcare services, such as accessing educational materials, tracking my own activities, and performing self-assessment.	○	○	○	○	○	○	○

UUID	_____
Start Date	_____
End Date	_____
Schedule Date	_____
Status	○ Deleted ○ Completed ○ Incomplete
Supplemental Data (JSON)	_____
Serialized Result	

Figure S7: Complete follow-up survey form created using REDCap showing both pages: unique identifier code input and questionnaire page.

Initial Survey Scripts

Script 03

```
library(Hmisc)
library(ggplot2)
library(dplyr)
library(tidyr)
library(lubridate)

data <- read.csv('PDMMM_DATA_2025-03-05_1211.csv')

initial_survey_rows <- data %>%
  filter(redcap_repeat_instrument == "my_moves_matter_ucc_parkinsons_study")

completed_surveys <- initial_survey_rows %>%
  filter(my_moves_matter_ucc_parkinsons_study_timestamp != "" &
         my_moves_matter_ucc_parkinsons_study_timestamp != "[not completed]")

data_UID <- completed_surveys
data_UID$p <- as.character(data_UID$p)
data_UID$a <- as.character(data_UID$a)

data_UID$original_p <- data_UID$p
data_UID$original_a <- data_UID$a

data_UID$p <- gsub(" ", "", data_UID$p)
data_UID$p <- trimws(data_UID$p)

data_UID$a <- gsub(" ", "", data_UID$a)
data_UID$a <- toupper(data_UID$a)
data_UID$a <- trimws(data_UID$a)

data_UID$UniqueID <- paste0("UCC", data_UID$p, data_UID$a)

test_ids <- c("UCC089WV4", "UCC123456", "UCC123ABC")
data_UID <- data_UID[!data_UID$UniqueID %in% test_ids, ]

valid_id_length <- nchar(data_UID$UniqueID) == 9 | nchar(data_UID$UniqueID) == 10
data_UID <- data_UID[valid_id_length, ]

data_UID$survey_timestamp <-
as.POSIXct(data_UID$my_moves_matter_ucc_parkinsons_study_timestamp,
                        format="%Y-%m-%d %H:%M:%S")

duplicate_ids <- data_UID$UniqueID[duplicated(data_UID$UniqueID)]

if (length(duplicate_ids) > 0) {
  data_UID <- data_UID %>%
    group_by(UniqueID) %>%
    arrange(desc(survey_timestamp)) %>%
    summarise(across(everything(), ~first(na.omit(.))), .groups = "drop") %>%
    ungroup()
}

data_UID <- data_UID[!is.na(data_UID$s01_gender) & (data_UID$s01_gender == 2 |
data_UID$s01_gender == "2"), ]
```

```
data_UID <- data_UID[data_UID$s01_pddiagnosis == 1 & !is.na(data_UID$s01_pddiagnosis), ]

data_UID <- data_UID[
 data_UID$s01_consent01 == 1 &
  data_UID$s01_consent02 == 1 &
  data_UID$s01_consent03 == 1 &
  data_UID$s01_consent04 == 1 &
  data_UID$s01_consent05 == 1,
]

data_UID <- data_UID[data_UID$redcap_event_name == 'initial_survey_arm_1', ]

data_UID <- data_UID[, !colnames(data_UID) %in% c("s01_consent01", "s01_consent02",
                                "s01_consent03", "s01_consent04",
                                "s01_consent05")]

data_UID <- data_UID[, !colnames(data_UID) %in% c('redcap_event_name', 'redcap_repeat_instrument',
                                'redcap_repeat_instance', 'redcap_data_access_group',
                                'redcap_survey_identifier', 'p', 'a', 'survey_timestamp',
                                'original_p', 'original_a')]

if("so1_prefernottosaymessage" %in% colnames(data_UID)) {
 data_UID <- subset(data_UID, select = -so1_prefernottosaymessage)
}
data_UID <- data_UID[, !grepl("^s01_pdmeds___", colnames(data_UID))]

data_UID <- data_UID[, c("record_id", "UniqueID",
                setdiff(colnames(data_UID),
                     c("record_id", "UniqueID")))]

create_demographic_table <- function(data, var_name, levels, labels, table_name = NULL) {
 result_table <- data.frame(
  RecordID = data$record_id,
  UniqueID = data$UniqueID,
  Code = data[[var_name]],
  Description = factor(data[[var_name]], levels = levels, labels = labels)
 )

 summary_table <- as.data.frame(table(result_table$Description, useNA = "ifany"))
 names(summary_table) <- c("Description", "Count")

 summary_table <- summary_table[!is.na(summary_table$Description) &
                      summary_table$Description != "Prefer not to say", ]

 summary_table$Percentage <- (summary_table$Count / sum(summary_table$Count)) * 100

 if (!is.null(table_name)) {
  assign(paste0(table_name, "_table"), result_table, envir = parent.frame())
  assign(paste0(table_name, "_summary_table"), summary_table, envir = parent.frame())
 }

 return(list(table = result_table, summary = summary_table))
}

create_checkbox_table <- function(data, column_pattern, labels, table_name = NULL) {
 cols <- grep(column_pattern, colnames(data), value = TRUE)
```

```
 result_table <- data[, cols]

 result_table <- lapply(result_table, function(x) as.numeric(as.character(x)))
 result_table <- as.data.frame(result_table)

 colnames(result_table) <- labels

 result_table <- cbind(data$record_id, data$UniqueID, result_table)
 colnames(result_table)[1:2] <- c("RecordID", "UniqueID")

 if (!is.null(table_name)) {
  assign(paste0(table_name, "_table"), result_table, envir = parent.frame())
 }

 return(result_table)
}

gender_result <- create_demographic_table(
 data_UID, "s01_gender",
 levels = c(1, 2, 3),
 labels = c("Male (biological from birth)",
        "Female (biological from birth)",
        "Prefer not to say"),
 table_name = "gender"
)

race_result <- create_demographic_table(
 data_UID, "s01_race",
 levels = c(1, 2, 3, 4, 5),
 labels = c("White", "Black", "Asian", "Other", "Prefer not to say"),
 table_name = "race"
)

employment_result <- create_demographic_table(
 data_UID, "s01_employment",
 levels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12),
 labels = c("Employed (full-time)", "Employed (part-time)", "Self-employed",
        "Unemployed (actively seeking employment)",
        "Unemployed (not actively seeking employment)",
        "Student (full-time)", "Student (part-time)",
        "Homemaker", "Retired", "Disabled", "Other", "Prefer not to say"),
 table_name = "employment"
)

marital_status_result <- create_demographic_table(
 data_UID, "s01_marital_status",
 levels = c(1, 2, 3, 4, 5, 6, 7, 8, 9),
 labels = c("Single", "In a relationship", "Married", "Widowed",
        "Divorced", "Separated", "Domestic partnership",
        "Civil union", "Prefer not to say"),
 table_name = "marital_status"
)

diagnosis_result <- create_demographic_table(
 data_UID, "s01_pddiagnosis",
 levels = c(1, 2, 3),
```

```
 labels = c("Yes", "No", "Prefer not to say"),
 table_name = "diagnosis"
)

age_result <- create_demographic_table(
 data_UID, "s01_age",
 levels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13),
 labels = c("18 - 29 years old", "30 - 39 years old", "40 - 49 years old",
        "50 - 54 years old", "55 - 59 years old", "60 - 64 years old",
        "65 - 69 years old", "70 - 74 years old", "75 - 79 years old",
        "80 - 84 years old", "85 - 90 years old", "90 years old",
        "Prefer not to say"),
 table_name = "age"
)

ageofdiagnosis_result <- create_demographic_table(
 data_UID, "s01_ageofdiagnosis",
 levels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13),
 labels = c("18 - 29 years old", "30 - 39 years old", "40 - 49 years old",
        "50 - 54 years old", "55 - 59 years old", "60 - 64 years old",
        "65 - 69 years old", "70 - 74 years old", "75 - 79 years old",
        "80 - 84 years old", "85 - 90 years old", "90 years old",
        "Prefer not to say"),
 table_name = "ageofdiagnosis"
)

symptomduration_result <- create_demographic_table(
 data_UID, "s01_durationofsymptoms",
 levels = c(1, 2, 3, 4, 5, 6, 7),
 labels = c("0 - 3 years", "4 - 6 years", "7 - 9 years",
        "10 - 12 years", "13 - 15 years", ">15 years",
        "Prefer not to say"),
 table_name = "symptomduration"
)

familydiagnosis_result <- create_demographic_table(
 data_UID, "s01_familymemberpd",
 levels = c(1, 2, 3, 4),
 labels = c("Yes", "No", "Prefer not to say", "I don't know"),
 table_name = "familydiagnosis"
)

procedures_result <- create_demographic_table(
 data_UID, "s01_surgeries",
 levels = c(1, 2, 3),
 labels = c("Yes", "No", "Prefer not to say"),
 table_name = "procedures"
)

motor_symptoms_labels <- c("Tremors or shaking",
                "Stiffness or rigidity",
                "Slowness of movement",
                "Balance and coordination issues",
                "Postural instability",
                "Changes in speech and writing",
                "Difficulty with swallowing",
                "Other",
```

```
                "Prefer not to say")

selected_motor_cols <- c(1:6, 8:10)
motor_cols <- paste0("s01_motor_symptoms___", selected_motor_cols)

motor_data <- data_UID[, motor_cols]
motor_data <- lapply(motor_data, function(x) as.numeric(as.character(x)))
motor_data <- as.data.frame(motor_data)

colnames(motor_data) <- motor_symptoms_labels

motorsymptoms_table <- cbind(data_UID$record_id, data_UID$UniqueID, motor_data)
colnames(motorsymptoms_table)[1:2] <- c("RecordID", "UniqueID")

nonmotor_labels <- c("Depression and anxiety",
            "Sleep disturbances, including insomnia and restless leg syndrome",
            "Loss of sense of smell",
            "Constipation",
            "Cognitive changes, including memory problems, confusion, and dementia",
            "Difficulty Concentrating",
            "Fatigue and lack of energy",
            "Excessive sweating",
            "Sexual dysfunction",
            "Orthostatic hypotension (low blood pressure upon standing)",
            "Visual disturbances, including blurry vision or double vision",
            "Speech and swallowing difficulties",
            "Pain and discomfort",
            "Mood swings and emotional changes",
            "Skin problems, including dry skin and rashes",
            "Weight loss or gain",
            "Urinary Problems",
            "Other",
            "Prefer not to say")

nonmotor_table <- create_checkbox_table(
 data_UID,
 "s01_nonmotor___",
 nonmotor_labels,
 "nonmotor"
)

pastmonth_labels <- c(
 "Dribbling of saliva during the daytime",
 "Loss or change in your ability to taste or smell",
 "Difficulty swallowing food or drink or problems with choking",
 "Vomiting or feelings of sickness (nausea)",
 "Constipation (less than three bowel movements a week) or having to strain to pass a stool",
 "Bowel (faecal) incontinence",
 "Feeling that your bowel emptying is incomplete after having been to the toilet",
 "A sense of urgency to pass urine makes you rush to the toilet",
 "Getting up regularly at night to pass urine",
 "Unexplained pains (not due to known conditions such as arthritis)",
 "Unexplained change in weight (not due to change in diet)",
 "Problems remembering things that have happened recently or forgetting to do things",
 "Loss of interest in what is happening around you or in doing things",
 "Seeing or hearing things that you know or are told are not there",
 "Difficulty concentrating or staying focused",
```

```
  "Feeling sad, 'low' or 'blue'",
  "Feeling anxious, frightened or panicky",
  "Feeling less interested in sex or more interested in sex",
  "Finding it difficult to have sex when you try",
  "Feeling light-headed, dizzy or weak standing from sitting or lying",
  "Falling",
  "Finding it difficult to stay awake during activities such as working, driving or eating",
  "Difficulty getting to sleep at night or staying asleep at night",
  "Intense, vivid or frightening dreams",
  "Talking or moving about in your sleep, as if you are 'acting out' a dream",
  "Unpleasant sensations in your legs at night or while resting, and a feeling that you need to move",
  "Swelling of the legs",
  "Excessive sweating",
  "Double vision",
  "Believing things are happening to you that other people say are not",
  "I have not experienced any of these in the past month"
)

pastmonth_table <- create_checkbox_table(
  data_UID,
  "s02_symptomspastmonth___",
  pastmonth_labels,
  "pastmonth"
)

othermotorsymptoms_labels <- c("Dyskinesias",
                               "Muscle Cramps",
                               "Restless Leg Syndrome",
                               "Other",
                               "Prefer not to say")

othermotorsymptoms_table <- create_checkbox_table(
  data_UID,
  "s01_motor_symptoms_2___",
  othermotorsymptoms_labels,
  "othermotorsymptoms"
)

pdmeds_labels <- c(
  "Adartel", "Adodespan PR", "Amantadine", "Apogo", "Apokyn",
  "Apomorphine", "Artane", "Azilect", "Benztropine", "Bromocriptine",
  "Caramet CR", "Clonazepam", "Co-Beneldopa", "Co-Careldopa", "Comtan",
  "Comtess", "Cogentin", "Dacepton", "Duodopa", "Duopa",
  "Eldepryl", "Gocovri", "Inbrija", "Ipinnia", "Kemadrin",
  "Kynmobi", "Lecado", "Madopar", "Mirapex", "Mirapexin",
  "Neupro", "Nourianz", "Oprymea", "Ongentys", "Osmolex",
  "Opicapone", "Parlodel", "Parcopa", "Pipexus", "Pramipexole",
  "RalneaXL", "Rasagiline", "Requip", "Raponer", "Rivotril",
  "Ropilynz", "Ropinirole", "Rotigotine", "Rytary", "Sastravi",
  "Selegiline", "Slepark", "Sinemet", "Sinemet Plus", "Spirico",
  "Stalevo", "Stanek", "Symmetrel", "Tasmar", "Tolcapone",
  "Trihexyphenidryl/Benzhexol", "Xadago", "Zantiva", "Zelapar",
  "Other", "None of the above"
)

pdmeds_cols <- grep("^s01t_", colnames(data_UID), value = TRUE)
pdmeds_cols <- pdmeds_cols[1:length(pdmeds_labels)]
```

```
pdmeds_table <- data_UID[, pdmeds_cols]
pdmeds_table <- lapply(pdmeds_table, function(x) as.numeric(as.character(x)))
pdmeds_table <- as.data.frame(pdmeds_table)
pdmeds_table <- cbind(data_UID$record_id, data_UID$UniqueID, pdmeds_table)
colnames(pdmeds_table) <- c("RecordID", "UniqueID", pdmeds_labels)

pdmeds_summary_table <- data.frame(
  Description = pdmeds_labels,
  Count = colSums(pdmeds_table[, -(1:2)], na.rm = TRUE)
)

total_responses <- nrow(data_UID) - sum(is.na(pdmeds_table$UniqueID))
pdmeds_summary_table$Percentage <- (pdmeds_summary_table$Count / total_responses) * 100

pdmeds_summary_table <- pdmeds_summary_table[pdmeds_summary_table$Count > 0, ]

othermeds_labels <- c(
  "Thyroid medication for underactive thyroid",
  "Thyroid medication for overactive thyroid",
  "HRT",
  "Other",
  "None of the above"
)

othermeds_table <- create_checkbox_table(
  data_UID,
  "s01_othermeds___",
  othermeds_labels,
  "othermeds"
)

contraceptive_labels <- c(
  "an implant",
  "an intrauterine device (IUD)",
  "injections",
  "oral contraception",
  "vaginal rings",
  "skin patches",
  "Specify medication",
  "I am not currently taking any of these medications"
)

contraceptive_table <- create_checkbox_table(
  data_UID,
  "s01_othermedshor_2___",
  contraceptive_labels,
  "contraceptive"
)

therapy_labels <- c(
  "Physical therapy",
  "Speech therapy",
  "Occupational therapy",
  "Nutrition therapy",
  "None of the above",
  "Prefer not to say",
```

```
  "Other"
)

therapy_table <- create_checkbox_table(
  data_UID,
  "s01_current_treatments___",
  therapy_labels,
  "therapy"
)

otherconditions_labels <- c(
  "Hypothyroidism",
  "Hyperthyroidism",
  "Polycystic ovary Syndrome (PCOS)",
  "Premenstrual dysphoric disorder (PMDD)",
  "Other Menstrual disorders",
  "Other",
  "I do not have any of these conditions"
)

otherconditions_table <- create_checkbox_table(
  data_UID,
  "s01_otherconditions___",
  otherconditions_labels,
  "otherconditions"
)

trial_table <- data.frame(
  RecordID = data_UID$record_id,
  UniqueID = data_UID$UniqueID,
  Code = data_UID$s01_trialtype,
  Description = factor(
    data_UID$s01_trialtype,
    levels = c(1, 2),
    labels = c("Clinical Trial", "Research Study")
  ),
  Specification = ifelse(
    data_UID$s01_trialtype == 1, data_UID$s01_study_yes_clinical,
    ifelse(data_UID$s01_trialtype == 2, data_UID$s01_study_yes_research, NA)
  )
)

levels(trial_table$Description) <- c(levels(trial_table$Description), "No")

trial_table$Description[is.na(trial_table$Description) & data_UID$s01_othertrials == 2] <- "No"

trial_summary_table <- data.frame(
  Description = c("Clinical Trial", "Research Study", "No"),
  Count = c(
    sum(trial_table$Description == "Clinical Trial", na.rm = TRUE),
    sum(trial_table$Description == "Research Study", na.rm = TRUE),
    sum(data_UID$s01_othertrials == 2, na.rm = TRUE)
  )
)

total_responses_trials <- sum(data_UID$s01_othertrials %in% c(1, 2), na.rm = TRUE)
trial_summary_table$Percentage <- (trial_summary_table$Count / total_responses_trials) * 100
```

```
menstruationstatus_table <- data.frame(
 RecordID = data_UID$record_id,
 UniqueID = data_UID$UniqueID,
 Code = data_UID$s01_menstruationstatus,
 Description = factor(data_UID$s01_menstruationstatus, levels = c(1, 2, 4),
                labels = c("Actively menstruating",
                           "Post menstruation (no active period for 1 year)",
                           "Other")),
 Specify = ifelse(data_UID$s01_menstruationstatus == 4, data_UID$s01_menstruationstatus_other, NA)
)

menstruationstatus_summary_table <- as.data.frame(table(menstruationstatus_table$Description))
names(menstruationstatus_summary_table) <- c("Description", "Count")

valid_responses <- sum(!is.na(menstruationstatus_table$Code))
menstruationstatus_summary_table$Percentage <- (menstruationstatus_summary_table$Count /
valid_responses) * 100

other_menstruation_filter_table <- menstruationstatus_table[
 menstruationstatus_table$Description == "Other" &
  !is.na(menstruationstatus_table$Specify),
 c("RecordID", "UniqueID", "Specify")
]

manual_categories <- data.frame(
 RecordID = c(
  101, 117, 118, 120, 145, 148, 165, 170, 182, 197, 207, 245, 252, 260, 267, 270, 271, 276, 280, 284, 314,
327, 328, 333, 357, 360, 368, 380, 389, 393, 433, 435, 442, 484, 491, 508, 512, 516, 521, 525, 547, 558,
567, 570, 573, 590, 600, 609, 612, 624, 626, 645, 651, 656, 681, 701,
  799, 363, 526, 318, 550, 579, 354, 597, 740, 486, 829, 375, 719, 474,
  785, 795, 796, 778, 762, 739, 765, 775
 ),
 Code = c(
  3, 1, 3, 3, 3, 1, 3, 3, 3, 2, 2, 3, 1, 1, 1, 2, 3, 2, 3, 2, 3, 3, 2, 2, 1, 3, 2, 2, 2, 3, 1, 3, 3, 3, 3, 2, 2, 2, 1, 1, 1, 1, 2,
1, 3, 1, 3, 2, 2, 3, 3, 1, 1, 3, 3, 3,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 3, 3, 3, 3, 3, 3
 ),
 Description = c(
  "Postmenopausal", "Menstruating", "Postmenopausal", "Postmenopausal", "Postmenopausal",
"Menstruating", "Postmenopausal", "Postmenopausal", "Postmenopausal", "Perimenopausal",
"Perimenopausal", "Postmenopausal", "Menstruating", "Menstruating", "Menstruating",
"Perimenopausal", "Postmenopausal", "Perimenopausal", "Postmenopausal", "Perimenopausal",
"Postmenopausal", "Postmenopausal", "Perimenopausal", "Perimenopausal", "Menstruating",
"Postmenopausal", "Perimenopausal", "Perimenopausal", "Perimenopausal", "Postmenopausal",
"Menstruating", "Postmenopausal", "Postmenopausal", "Postmenopausal", "Postmenopausal",
"Perimenopausal", "Perimenopausal", "Perimenopausal", "Menstruating", "Menstruating",
"Menstruating", "Menstruating", "Perimenopausal", "Menstruating", "Postmenopausal", "Menstruating",
"Postmenopausal", "Perimenopausal", "Perimenopausal", "Postmenopausal", "Postmenopausal",
"Menstruating", "Menstruating", "Postmenopausal", "Postmenopausal", "Postmenopausal",
  "Postmenopausal", "Postmenopausal", "Postmenopausal", "Postmenopausal", "Postmenopausal",
"Postmenopausal", "Postmenopausal", "Postmenopausal", "Postmenopausal", "Postmenopausal",
"Postmenopausal", "Postmenopausal", "Postmenopausal", "Postmenopausal",
  "Postmenopausal", "Postmenopausal", "Postmenopausal", "Postmenopausal", "Postmenopausal",
"Postmenopausal", "Postmenopausal", "Postmenopausal"
 ),
```

```
  stringsAsFactors = FALSE
)

valid_manual_categories <- manual_categories[manual_categories$RecordID %in%
data_UID$record_id, ]

other_menstruation_filter_table <- merge(
  other_menstruation_filter_table,
  valid_manual_categories,
  by = "RecordID",
  all.x = TRUE
)

other_remaining <- other_menstruation_filter_table[is.na(other_menstruation_filter_table$Code), ]

filtered_menstruationstatus_table <- menstruationstatus_table[
  menstruationstatus_table$Code %in% c(1, 2),
  c("RecordID", "UniqueID", "Code", "Description")
]

filtered_menstruationstatus_table$Description <-
as.character(filtered_menstruationstatus_table$Description)
filtered_menstruationstatus_table$Description[filtered_menstruationstatus_table$Description ==
"Actively menstruating"] <- "Menstruating"
filtered_menstruationstatus_table$Description[filtered_menstruationstatus_table$Description == "Post
menstruation (no active period for 1 year)"] <- "Post menstruation"

manually_recategorized_participants <- other_menstruation_filter_table[
  !is.na(other_menstruation_filter_table$Code),
  c("RecordID", "UniqueID", "Code", "Description")
]

remaining_other_participants <- other_menstruation_filter_table[
  is.na(other_menstruation_filter_table$Code),
  c("RecordID", "UniqueID", "Specify")
]

if (nrow(remaining_other_participants) > 0) {
  remaining_other_participants$Code <- 4
  remaining_other_participants$Description <- "Other"
  remaining_other_participants <- remaining_other_participants[, c("RecordID", "UniqueID", "Code",
"Description")]
} else {
  remaining_other_participants <- data.frame(
    RecordID = integer(0),
    UniqueID = character(0),
    Code = integer(0),
    Description = character(0)
  )
}

participants_in_data_uid <- data.frame(
  RecordID = data_UID$record_id,
  UniqueID = data_UID$UniqueID,
  stringsAsFactors = FALSE
)
```

```
participants_already_included <- rbind(
 filtered_menstruationstatus_table[, c("RecordID", "UniqueID")],
 manually_recategorized_participants[, c("RecordID", "UniqueID")],
 remaining_other_participants[, c("RecordID", "UniqueID")]
)

missing_participants_with_manual <- participants_in_data_uid %>%
 filter(!RecordID %in% participants_already_included$RecordID) %>%
 inner_join(valid_manual_categories[, c("RecordID", "Code", "Description")], by = "RecordID")

if(nrow(missing_participants_with_manual) > 0) {
 manually_recategorized_participants <- rbind(
  manually_recategorized_participants,
  missing_participants_with_manual[, c("RecordID", "UniqueID", "Code", "Description")]
 )
}

manually_recategorized_participants$Description <-
as.character(manually_recategorized_participants$Description)
remaining_other_participants$Description <- as.character(remaining_other_participants$Description)

combined_menstruation_table <- rbind(
 filtered_menstruationstatus_table,
 manually_recategorized_participants,
 remaining_other_participants
)

combined_categories <- table(combined_menstruation_table$Description, useNA = "ifany")

na_indices <- which(is.na(combined_menstruation_table$Code))
if (length(na_indices) > 0) {
 for (i in na_indices) {
  desc <- combined_menstruation_table$Description[i]
  if (!is.na(desc)) {
   if (desc == "Menstruating") combined_menstruation_table$Code[i] <- 1
   else if (desc == "Perimenopausal") combined_menstruation_table$Code[i] <- 2
   else if (desc %in% c("Postmenopausal", "Post menstruation")) combined_menstruation_table$Code[i]
<- 3
  }
 }
}

definitive_menstruationstatus_table <- combined_menstruation_table

definitive_menstruationstatus_table$Description <-
as.character(definitive_menstruationstatus_table$Description)

definitive_menstruationstatus_table$Description[definitive_menstruationstatus_table$Description ==
"Post menstruation"] <- "Postmenopausal"

definitive_menstruationstatus_table$Description <- factor(
 definitive_menstruationstatus_table$Description,
 levels = c("Menstruating", "Perimenopausal", "Postmenopausal", "Other"),
 labels = c("Menstruating", "Perimenopausal", "Postmenopausal", "Other")
)

definitive_menstruationstatus_table$Code <- ifelse(
```

```
 definitive_menstruationstatus_table$Description == "Other", 4,
 as.numeric(definitive_menstruationstatus_table$Description)
)

definitive_menstruationstatus_summary_table <-
as.data.frame(table(definitive_menstruationstatus_table$Description, useNA = "ifany"))
names(definitive_menstruationstatus_summary_table) <- c("Description", "Count")

valid_responses_final <- sum(!is.na(definitive_menstruationstatus_table$Code))
definitive_menstruationstatus_summary_table$Percentage <-
(definitive_menstruationstatus_summary_table$Count / valid_responses_final) * 100

new_menstruationstatus_table <- definitive_menstruationstatus_table
new_menstruationstatus_summary_table <- definitive_menstruationstatus_summary_table

applied_manual_ids <- intersect(new_menstruationstatus_table$RecordID,
valid_manual_categories$RecordID)
excluded_manual_ids <- setdiff(manual_categories$RecordID,
new_menstruationstatus_table$RecordID)

perimenopause_count <- sum(new_menstruationstatus_table$Description == "Perimenopausal", na.rm
= TRUE)

other_count <- sum(new_menstruationstatus_table$Description == "Other", na.rm = TRUE)

expected_manual_postmenopausal <- c(799, 363, 526, 318, 550, 579, 354, 597, 740, 486, 829, 375, 719,
474)
found_manual_postmenopausal <- sum(new_menstruationstatus_table$RecordID %in%
expected_manual_postmenopausal &
                      new_menstruationstatus_table$Description == "Postmenopausal")

if(found_manual_postmenopausal != length(expected_manual_postmenopausal)) {
 missing_manual <- setdiff(expected_manual_postmenopausal,
                  new_menstruationstatus_table$RecordID[new_menstruationstatus_table$Description ==
"Postmenopausal"])
}

raw_data_with_menstruation <- data %>%
 filter(redcap_repeat_instrument == "my_moves_matter_ucc_parkinsons_study") %>%
 filter(my_moves_matter_ucc_parkinsons_study_timestamp != "" &
       my_moves_matter_ucc_parkinsons_study_timestamp != "[not completed]") %>%
 mutate(
  p_clean = gsub(" ", "", trimws(as.character(p))),
  a_clean = gsub(" ", "", toupper(trimws(as.character(a)))),
  UniqueID_standardized = paste0("UCC", p_clean, a_clean)
 ) %>%
 filter(UniqueID_standardized != "UCC089WV4") %>%
 filter(!is.na(s01_menstruationstatus)) %>%
 filter(
  s01_pddiagnosis == 1 & !is.na(s01_pddiagnosis),
  s01_consent01 == 1 & s01_consent02 == 1 & s01_consent03 == 1 &
   s01_consent04 == 1 & s01_consent05 == 1,
  redcap_event_name == 'initial_survey_arm_1'
 ) %>%
 mutate(survey_timestamp = as.POSIXct(my_moves_matter_ucc_parkinsons_study_timestamp,
                     format="%Y-%m-%d %H:%M:%S")) %>%
 group_by(UniqueID_standardized) %>%
```

```
 arrange(desc(survey_timestamp)) %>%
 slice(1) %>%
 ungroup()

missing_from_data_uid <- raw_data_with_menstruation %>%
 filter(!UniqueID_standardized %in% data_UID$UniqueID) %>%
 select(record_id, UniqueID_standardized, s01_menstruationstatus, s01_menstruationstatus_other,
      s01_gender, p_clean, a_clean)

if (nrow(missing_from_data_uid) > 0) {
 supplementary_menstruation_entries <- missing_from_data_uid %>%
  mutate(
   RecordID = record_id,
   UniqueID = UniqueID_standardized,
   Code = s01_menstruationstatus,
   Description = case_when(
    Code == 1 ~ "Menstruating",
    Code == 2 ~ "Postmenopausal",
    Code == 3 ~ "Perimenopausal",
    Code == 4 ~ "Other",
    TRUE ~ "Other"
   )
  ) %>%
  select(RecordID, UniqueID, Code, Description)

 other_supplementary <- supplementary_menstruation_entries %>%
  filter(Description == "Other")

 if (nrow(other_supplementary) > 0) {
  for (i in 1:nrow(other_supplementary)) {
   record_id <- other_supplementary$RecordID[i]
   if (record_id %in% manual_categories$RecordID) {
    manual_cat <- manual_categories[manual_categories$RecordID == record_id, ]
    supplementary_menstruation_entries$Code[supplementary_menstruation_entries$RecordID ==
record_id] <- manual_cat$Code
     supplementary_menstruation_entries$Description[supplementary_menstruation_entries$RecordID
== record_id] <- manual_cat$Description
   }
  }
 }

 original_count <- nrow(new_menstruationstatus_table)

 expanded_menstruationstatus_table <- rbind(
  new_menstruationstatus_table,
  supplementary_menstruation_entries
 )

 expanded_menstruationstatus_table <- expanded_menstruationstatus_table %>%
  distinct(UniqueID, .keep_all = TRUE)

 final_count <- nrow(expanded_menstruationstatus_table)
 added_count <- final_count - original_count

 new_menstruationstatus_table <- expanded_menstruationstatus_table
```

```
 new_menstruationstatus_summary_table <-
as.data.frame(table(new_menstruationstatus_table$Description, useNA = "ifany"))
 names(new_menstruationstatus_summary_table) <- c("Description", "Count")

 valid_responses_final <- sum(!is.na(new_menstruationstatus_table$Code))
 new_menstruationstatus_summary_table$Percentage <-
(new_menstruationstatus_summary_table$Count / valid_responses_final) * 100

 supplementary_by_reason <- missing_from_data_uid %>%
  mutate(
   id_length = nchar(UniqueID_standardized),
   is_valid_length = id_length == 9 | id_length == 10,
   is_female = !is.na(s01_gender) & s01_gender == 2,
   filter_reason = case_when(
    !is_valid_length ~ paste("Invalid ID length (", id_length, ")"),
    !is_female & is.na(s01_gender) ~ "Missing gender",
    !is_female ~ "Not female",
    TRUE ~ "Other reason"
   )
  ) %>%
  count(filter_reason, name = "count")
}

label(data$record_id) <- "Record ID"
label(data$redcap_event_name) <- "Event Name"
label(data$redcap_repeat_instrument) <- "Repeat Instrument"
label(data$redcap_repeat_instance) <- "Repeat Instance"
label(data$redcap_data_access_group) <- "Data Access Group"
label(data$redcap_survey_identifier) <- "Survey Identifier"
label(data$my_moves_matter_ucc_parkinsons_study_timestamp) <- "Survey Timestamp"
label(data$s01_consent01) <- "I have read and understand the study"
label(data$s01_consent02) <- "I agree to participate in this research"
label(data$s01_consent03) <- "I consent for my anonymised information to be shared on a public
database for the purpose of this research and collaboration with other researchers"
label(data$s01_consent04) <- "I understand that my anonymised data will be stored at UCC for 15 years"
label(data$s01_consent05) <- "I confirm that I am over 18 years of age"
label(data$p) <- "1st three digits of phone number"
label(data$a) <- "Last three letters/numbers of postcode"
label(data$s01_age) <- "What is your age?"
label(data$s01_gender) <- "What is your gender?"
label(data$s01_race) <- "What is your race/ethnicity?"
label(data$s01_ethnicity_other) <- "Please Specify Other"
label(data$s01_employment) <- "What is your employment status? "
label(data$s01_employment_other) <- "Please specify Other"
label(data$s01_marital_status) <- "What is your marital status? "
label(data$s01_pddiagnosis) <- "Have you ever been diagnosed with Parkinsons disease? "
label(data$s01_ageofdiagnosis) <- "What age were you when you were diagnosed with Parkinsons
disease? "

for (i in 1:10) {
 field_name <- paste0("s01_motor_symptoms___", i)
 if (field_name %in% colnames(data)) {
  if (i == 9) {
   label(data[[field_name]]) <- paste0("What are the main motor symptoms you experience due to
Parkinsons disease?  (choice=", motor_symptoms_labels[i], " {s01_motor_other})")
  } else {
```

```
    label(data[[field_name]]) <- paste0("What are the main motor symptoms you experience due to
Parkinsons disease?  (choice=", motor_symptoms_labels[i], ")")
   }
  }
}

for (i in 1:19) {
  field_name <- paste0("s01_nonmotor___", i)
  if (field_name %in% colnames(data)) {
   if (i == 18) {
    label(data[[field_name]]) <- paste0("Do you experience any non-motor symptoms associated with
Parkinsons disease? (choice=", nonmotor_labels[i], "{s01_nonmotor_other})")
   } else {
    label(data[[field_name]]) <- paste0("Do you experience any non-motor symptoms associated with
Parkinsons disease? (choice=", nonmotor_labels[i], ")")
   }
  }
}

if ("s01_nonmotor_other" %in% colnames(data)) {
  label(data$s01_nonmotor_other) <- "Please specify Other"
}

if ("s01_durationofsymptoms" %in% colnames(data)) {
  label(data$s01_durationofsymptoms) <- "How long have you been experiencing Parkinsons symptoms?
"
}

data$redcap_event_name.factor =
factor(data$redcap_event_name,levels=c("initial_survey_arm_1","follow_up_survey_arm_1"))
data$redcap_repeat_instrument.factor = factor(data$redcap_repeat_instrument,
                                levels=c("my_moves_matter_ucc_parkinsons_study",
                                     "form_2_pwp_questionnaire_mmm_study_follow_up_surve",
                                     "my_moves_matter_followup_survey"))
data$redcap_data_access_group.factor = factor(data$redcap_data_access_group,levels=c("fyp"))
data$s01_consent01.factor = factor(data$s01_consent01,levels=c("1","0"))
data$s01_consent02.factor = factor(data$s01_consent02,levels=c("1","0"))
data$s01_consent03.factor = factor(data$s01_consent03,levels=c("1","0"))
data$s01_consent04.factor = factor(data$s01_consent04,levels=c("1","0"))
data$s01_consent05.factor = factor(data$s01_consent05,levels=c("1","0"))
data$s01_age.factor =
factor(data$s01_age,levels=c("1","2","3","4","5","6","7","8","9","10","11","12","13"))
data$s01_gender.factor = factor(data$s01_gender,levels=c("1","2","3"))
data$s01_race.factor = factor(data$s01_race,levels=c("1","2","3","4","5"))
data$s01_employment.factor =
factor(data$s01_employment,levels=c("1","2","3","4","5","6","7","8","9","10","11","12"))
data$s01_marital_status.factor =
factor(data$s01_marital_status,levels=c("1","2","3","4","5","6","7","8","9"))
data$s01_pddiagnosis.factor = factor(data$s01_pddiagnosis,levels=c("1","2","3"))
data$s01_ageofdiagnosis.factor =
factor(data$s01_ageofdiagnosis,levels=c("1","2","3","4","5","6","7","8","9","10","11","12","13"))

motorsymptoms_simple_table <- motorsymptoms_table
colnames(motorsymptoms_simple_table) <- c("RecordID", "UniqueID", "Tremors", "Rigidity", "Slowness",
"Imbalance",
                          "Postural instability", "Speech/Writing",
                          "Swallowing",
```

```
                        "Other", "Prefer not to say")

nonmotorsymptoms_simple_table <- nonmotor_table
colnames(nonmotorsymptoms_simple_table) <- c("RecordID", "UniqueID", "Depression/Anxiety",
"Insomnia", "Hyposmia",
                             "Constipation", "Cognitive Changes", "Concentration", "Fatigue",
                             "Sweating", "Sexual dysfunction", "Hypotension", "Vision",
                             "Speech", "Pain", "Mood swings", "Skin", "Weight",
                             "Urinary", "Other", "Prefer not to say")

motorsymptoms_table <- motorsymptoms_simple_table
nonmotorsymptoms_table <- nonmotorsymptoms_simple_table

hormonal_table <- data.frame(
 RecordID = contraceptive_table$RecordID,
 UniqueID = contraceptive_table$UniqueID,
 Yes = ifelse(rowSums(contraceptive_table[, 3:8], na.rm = TRUE) > 0, 1, 0),
 No = ifelse(contraceptive_table$`I am not currently taking any of these medications` == 1, 1, 0)
)

med_category_table <- data.frame(
 RecordID = pdmeds_table$RecordID,
 UniqueID = pdmeds_table$UniqueID,
 L_DOPA = ifelse(rowSums(pdmeds_table[, c("Madopar", "Sinemet", "Sinemet Plus",
                             "Co-Careldopa", "Co-Beneldopa", "Stalevo",
                             "Duodopa", "Duopa",
                             "Adodespan PR", "Caramet CR", "Inbrija",
                             "Lecado", "Parcopa", "Rytary")], na.rm = TRUE) > 0, 1, 0),

 DA_Agonist = ifelse(rowSums(pdmeds_table[, c("Mirapex", "Mirapexin", "Pramipexole",
                               "Requip", "Ropinirole", "Neupro", "Rotigotine",
                               "Apomorphine", "Apokyn", "Bromocriptine", "Parlodel",
                               "Adartel", "Apogo", "Dacepton", "Ipinnia", "Kynmobi",
                               "Oprymea", "Pipexus", "RalneaXL", "Raponer",
                               "Ropilynz", "Spirico", "Zantiva")], na.rm = TRUE) > 0, 1, 0),

 MAOB_Inhibitors = ifelse(rowSums(pdmeds_table[, c("Azilect", "Rasagiline", "Eldepryl",
                                 "Selegiline", "Zelapar", "Xadago")], na.rm = TRUE) > 0, 1, 0),

 Anticholinergics = ifelse(rowSums(pdmeds_table[, c("Artane", "Trihexyphenidryl/Benzhexol",
                                  "Benztropine", "Cogentin", "Kemadrin")], na.rm = TRUE) > 0, 1, 0),

 COMT_Inhibitors = ifelse(rowSums(pdmeds_table[, c("Comtan", "Comtess", "Tasmar",
                                 "Tolcapone", "Opicapone", "Ongentys",
                                 "Sastravi", "Stanek")], na.rm = TRUE) > 0, 1, 0),

 Benzodiazepines = ifelse(rowSums(pdmeds_table[, c("Clonazepam", "Rivotril")], na.rm = TRUE) > 0, 1,
0),

 Adenosine_Antagonist = ifelse(pdmeds_table$Nourianz > 0, 1, 0),

 NMDA_antagonist = ifelse(rowSums(pdmeds_table[, c("Amantadine", "Gocovri", "Osmolex",
"Symmetrel")], na.rm = TRUE) > 0, 1, 0)
)

colnames(med_category_table) <- c("RecordID", "UniqueID",
                      "L-DOPA", "DA Agonists", "MAO-B Inhibitors",
```

```
                    "COMT Inhibitors", "Anticholinergics",
                    "NMDA Antagonists", "Benzodiazepines",
                    "Adenosine Antagonists")

key_tables <- c("gender_table", "race_table", "employment_table", "marital_status_table",
          "age_table", "ageofdiagnosis_table", "symptomduration_table",
          "familydiagnosis_table", "procedures_table", "trial_table",
          "motorsymptoms_table", "nonmotor_table", "pastmonth_table",
          "pdmeds_table", "othermeds_table", "contraceptive_table",
          "therapy_table", "otherconditions_table", "hormonal_table",
          "med_category_table", "new_menstruationstatus_table")

master_symptom_orders <- list(

 motor_symptoms = c(
  "Tremors",
  "Rigidity",
  "Slowness",
  "Imbalance",
  "Postural instability",
  "Speech/Writing",
  "Swallowing"
 ),

 nonmotor_symptoms = c(
  "Depression/Anxiety",
  "Insomnia",
  "Hyposmia",
  "Constipation",
  "Cognitive Changes",
  "Concentration",
  "Fatigue",
  "Sweating",
  "Sexual dysfunction",
  "Hypotension",
  "Vision",
  "Speech",
  "Pain",
  "Mood swings",
  "Skin",
  "Weight",
  "Urinary"
 ),

 medication_categories = c(
  "L-DOPA",
  "DA Agonists",
  "MAO-B Inhibitors",
  "Anticholinergics",
  "COMT Inhibitors",
  "Benzodiazepines",
  "Adenosine Antagonists",
  "NMDA Antagonists"
 ),

 other_motor_symptoms = c(
  "Dyskinesias",
```

```
   "Muscle Cramps",
   "Restless Leg Syndrome"
  )
)

if(exists("motorsymptoms_table")) {
  motor_cols_actual <- setdiff(colnames(motorsymptoms_table), c("RecordID", "UniqueID"))
  motor_cols_master <- master_symptom_orders$motor_symptoms

  missing_from_master <- setdiff(motor_cols_actual, motor_cols_master)
  missing_from_actual <- setdiff(motor_cols_master, motor_cols_actual)
}

if(exists("nonmotorsymptoms_table")) {
  nonmotor_cols_actual <- setdiff(colnames(nonmotorsymptoms_table), c("RecordID", "UniqueID"))
  nonmotor_cols_master <- master_symptom_orders$nonmotor_symptoms

  missing_from_master <- setdiff(nonmotor_cols_actual, nonmotor_cols_master)
  missing_from_actual <- setdiff(nonmotor_cols_master, nonmotor_cols_actual)
}

if(exists("med_category_table")) {
  med_cols_actual <- setdiff(colnames(med_category_table), c("RecordID", "UniqueID"))
  med_cols_master <- master_symptom_orders$medication_categories

  missing_from_master <- setdiff(med_cols_actual, med_cols_master)
  missing_from_actual <- setdiff(med_cols_master, med_cols_actual)
}

if(exists("othermotorsymptoms_table")) {
  othermotorsymptoms_cols_actual <- setdiff(colnames(othermotorsymptoms_table), c("RecordID",
"UniqueID"))
  othermotorsymptoms_cols_master <- master_symptom_orders$other_motor_symptoms

  missing_from_master <- setdiff(othermotorsymptoms_cols_actual, othermotorsymptoms_cols_master)
  missing_from_actual <- setdiff(othermotorsymptoms_cols_master, othermotorsymptoms_cols_actual)
}

apply_standard_symptom_order <- function(results_data, symptom_type) {

  if(!"SymptomName" %in% colnames(results_data)) {
    return(results_data)
  }

  if(!symptom_type %in% names(master_symptom_orders)) {
    return(results_data)
  }

  standard_order <- master_symptom_orders[[symptom_type]]

  results_data$SymptomName <- factor(results_data$SymptomName,
                                     levels = standard_order)

  ordered_results <- results_data[order(results_data$SymptomName), ]

  ordered_results <- ordered_results[!is.na(ordered_results$SymptomName), ]
```

```
  return(ordered_results)
}

ensure_complete_symptom_coverage <- function(results_data, symptom_type) {

  if(!symptom_type %in% names(master_symptom_orders)) {
    return(results_data)
  }

  all_symptoms <- master_symptom_orders[[symptom_type]]

  present_symptoms <- unique(results_data$SymptomName)
  missing_symptoms <- setdiff(all_symptoms, present_symptoms)

  if(length(missing_symptoms) > 0) {
    sample_row <- results_data[1, ]

    missing_rows <- data.frame(
      SymptomName = missing_symptoms,
      stringsAsFactors = FALSE
    )

    for(col in setdiff(colnames(results_data), "SymptomName")) {
      if(is.numeric(results_data[[col]])) {
        missing_rows[[col]] <- NA
      } else {
        missing_rows[[col]] <- "No Data"
      }
    }

    complete_results <- rbind(results_data, missing_rows)

    complete_results <- apply_standard_symptom_order(complete_results, symptom_type)

    return(complete_results)
  }

  return(apply_standard_symptom_order(results_data, symptom_type))
}

get_symptom_display_labels <- function(symptom_type) {

  if(!symptom_type %in% names(master_symptom_orders)) {
    return(NULL)
  }

  return(master_symptom_orders[[symptom_type]])
}

save.image(file = "master_data_preparation_environment.RData")
```

Script 04

```
rm(list = ls(pattern = "^(summary_tables|table_1|filtered_data|menstruation_counts)"))

rm(list = ls(pattern = "^(filtered_contraceptive_table|hormonal_summary|kable_table)$"))

rm(list = ls(pattern = "^(filter_data_by_menstruation|create_menstruation_summary_table)$"))

rm(list = ls(pattern = "_labels$"))

required_vars <- c("data_UID", "new_menstruationstatus_table", "hormonal_table")

missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 loaded <- FALSE

 if (file.exists("master_data_preparation_environment.RData")) {
  load("master_data_preparation_environment.RData")
  loaded <- TRUE
 } else if (file.exists("03_menstruation_status_analysis.RData")) {
  load("03_menstruation_status_analysis.RData")
  loaded <- TRUE
 } else if (file.exists("survey_analysis_environment.RData")) {
  load("survey_analysis_environment.RData")
  loaded <- TRUE
 }

 if (!loaded) {
  stop("Required variables missing and no environment file found. Please run Script 03.0 first.\n",
       "Missing variables: ", paste(missing_vars, collapse = ", "))
 }

 still_missing <- c()
 for(var in required_vars) {
  if(!exists(var)) {
   still_missing <- c(still_missing, var)
  }
 }

 if(length(still_missing) > 0) {
  stop("Even after loading environment, some variables are still missing: ",
       paste(still_missing, collapse = ", "))
 }
}

if(!"Description" %in% colnames(new_menstruationstatus_table)) {
 stop("Description column not found in new_menstruationstatus_table. Please check Script 03.0
completed correctly.")
}

if(!"UniqueID" %in% colnames(data_UID)) {
```

```
 stop("UniqueID column not found in data_UID. Please check Script 03.0 completed correctly.")
}

required_demo_vars <- c("s01_age", "s01_gender", "s01_race", "s01_employment",
                        "s01_marital_status", "s01_ageofdiagnosis", "s01_durationofsymptoms")

missing_demo_vars <- setdiff(required_demo_vars, colnames(data_UID))
if(length(missing_demo_vars) > 0) {
 warning("Some demographic variables missing from data_UID: ",
         paste(missing_demo_vars, collapse = ", "))
}

menstruation_categories <- unique(new_menstruationstatus_table$Description)
required_categories <- c("Menstruating", "Postmenopausal")
available_categories <- intersect(required_categories, menstruation_categories)

if(length(available_categories) < 2) {
 stop("Insufficient menstruation categories for analysis. Available: ",
      paste(menstruation_categories, collapse = ", "))
}

library(dplyr)
library(knitr)
library(kableExtra)
library(tidyr)

valid_status <- new_menstruationstatus_table %>%
 filter(Description %in% c("Menstruating", "Postmenopausal"))

filtered_data <- merge(data_UID, valid_status, by = "UniqueID")

menstruation_counts <- valid_status %>%
 group_by(Description) %>%
 summarise(Count = n(), .groups = 'drop')

create_menstruation_summary_table <- function(df, variable, labels, category_name) {
 df <- df %>%
  filter(!is.na(!!sym(variable)))

 if (variable == "s01_gender") {
  df <- df %>% filter(s01_gender != 3)
 } else if (variable == "s01_race") {
  df <- df %>% filter(s01_race != 5)
 } else if (variable == "s01_marital_status") {
  df <- df %>% filter(s01_marital_status != 9)
 } else if (variable == "s01_age") {
  df <- df %>% filter(s01_age != 13)
 } else if (variable == "s01_ageofdiagnosis") {
  df <- df %>% filter(s01_ageofdiagnosis != 13)
 } else if (variable == "s01_employment") {
  df <- df %>% filter(s01_employment != 12)
 } else if (variable == "s01_durationofsymptoms") {
  df <- df %>% filter(s01_durationofsymptoms != 7)
 }

 actual_counts <- df %>%
  group_by(Description) %>%
```

```
  summarise(N = n(), .groups = 'drop')

 summary_table <- df %>%
  group_by(!!sym(variable), Description) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  group_by(Description) %>%
  mutate(Percentage = round((Count / sum(Count)) * 100, 2)) %>%
  ungroup() %>%
  select(-Count) %>%
  pivot_wider(
   names_from = Description,
   values_from = Percentage,
   values_fill = list(Percentage = 0)
  ) %>%
  filter(!grepl("prefer|Prefer|Other|other|NA", as.character(!!sym(variable)))) %>%
  mutate(Category = category_name)

 if (!is.null(labels)) {
  summary_table[[variable]] <- as.character(summary_table[[variable]])

  label_lookup <- setNames(labels, seq_along(labels))

  summary_table[[variable]] <- label_lookup[summary_table[[variable]]]

  summary_table <- summary_table %>% filter(!is.na(!!sym(variable)))
 }

 menstruation_categories <- c("Menstruating", "Postmenopausal")
 for (category in menstruation_categories) {
  if (!category %in% colnames(summary_table)) {
   summary_table[[category]] <- 0
  }
 }

 return(list(
  summary_table = summary_table,
  actual_counts = actual_counts
 ))
}

menstruation_counts <- valid_status %>%
 group_by(Description) %>%
 summarise(Count = n(), .groups = 'drop')

menstruation_counts_lookup <- setNames(
 menstruation_counts$Count,
 menstruation_counts$Description
)

age_labels <- c("18 - 29 years", "30 - 39 years", "40 - 49 years", "50 - 54 years",
          "55 - 59 years", "60 - 64 years", "65 - 69 years", "70 - 74 years",
          "75 - 79 years", "80 - 84 years", "85 - 90 years", "90 years and above")

gender_labels <- c("Male (biological from birth)", "Female (biological from birth)")
race_labels <- c("White", "Black", "Asian", "Other")
employment_labels <- c("Employed (full-time)", "Employed (part-time)", "Self-employed",
               "Unemployed (actively seeking employment)",
```

```
            "Unemployed (not actively seeking employment)",
            "Student (full-time)", "Student (part-time)",
            "Homemaker", "Retired", "Disabled", "Other")
marital_status_labels <- c("Single", "In a relationship", "Married", "Widowed",
              "Divorced", "Separated", "Domestic partnership", "Civil union")
ageofdiagnosis_labels <- age_labels
symptomduration_labels <- c("0 - 3 years", "4 - 6 years", "7 - 9 years",
               "10 - 12 years", "13 - 15 years", ">15 years")
familydiagnosis_labels <- c("Yes", "No")
procedures_labels <- c("Yes", "No")
hormonal_labels <- c("Yes", "No")

summary_tables <- list()

question_n_counts <- list()

age_result <- create_menstruation_summary_table(
 filtered_data, "s01_age", age_labels, "Age"
)
summary_tables$age <- age_result$summary_table
question_n_counts$age <- age_result$actual_counts

gender_result <- create_menstruation_summary_table(
 filtered_data, "s01_gender", gender_labels, "Gender"
)
summary_tables$gender <- gender_result$summary_table
question_n_counts$gender <- gender_result$actual_counts

race_result <- create_menstruation_summary_table(
 filtered_data, "s01_race", race_labels, "Race"
)
summary_tables$race <- race_result$summary_table
question_n_counts$race <- race_result$actual_counts

employment_result <- create_menstruation_summary_table(
 filtered_data, "s01_employment", employment_labels, "Employment"
)
summary_tables$employment <- employment_result$summary_table
question_n_counts$employment <- employment_result$actual_counts

marital_status_result <- create_menstruation_summary_table(
 filtered_data, "s01_marital_status", marital_status_labels, "Marital Status"
)
summary_tables$marital_status <- marital_status_result$summary_table
question_n_counts$marital_status <- marital_status_result$actual_counts

ageofdiagnosis_result <- create_menstruation_summary_table(
 filtered_data, "s01_ageofdiagnosis", ageofdiagnosis_labels, "Age of Diagnosis"
)
summary_tables$ageofdiagnosis <- ageofdiagnosis_result$summary_table
question_n_counts$ageofdiagnosis <- ageofdiagnosis_result$actual_counts

symptomduration_result <- create_menstruation_summary_table(
 filtered_data, "s01_durationofsymptoms", symptomduration_labels, "Symptom Duration"
)
summary_tables$symptomduration <- symptomduration_result$summary_table
question_n_counts$symptomduration <- symptomduration_result$actual_counts
```

```
familydiagnosis_result <- create_menstruation_summary_table(
 filtered_data, "s01_familymemberpd", familydiagnosis_labels, "Family Diagnosis"
)
summary_tables$familydiagnosis <- familydiagnosis_result$summary_table
question_n_counts$familydiagnosis <- familydiagnosis_result$actual_counts

procedures_result <- create_menstruation_summary_table(
 filtered_data, "s01_surgeries", procedures_labels, "Procedures"
)
summary_tables$procedures <- procedures_result$summary_table
question_n_counts$procedures <- procedures_result$actual_counts

filtered_contraceptive_table <- hormonal_table[
 hormonal_table$UniqueID %in% filtered_data$UniqueID,
]

filtered_contraceptive_table <- filtered_contraceptive_table %>%
 left_join(
  filtered_data[, c("UniqueID", "Description")],
  by = "UniqueID"
 )

hormonal_summary <- filtered_contraceptive_table %>%
 filter(!is.na(Yes)) %>%
 group_by(Yes, Description) %>%
 summarise(Count = n(), .groups = 'drop') %>%
 group_by(Description) %>%
 mutate(Percentage = round((Count / sum(Count)) * 100, 2)) %>%
 ungroup() %>%
 select(-Count) %>%
 pivot_wider(
  names_from = Description,
  values_from = Percentage,
  values_fill = list(Percentage = 0)
 ) %>%
 mutate(Category = "Hormonal Contraceptive Use")

hormonal_summary$Yes <- factor(
 hormonal_summary$Yes,
 levels = c(1, 0),
 labels = hormonal_labels
)

summary_tables$hormonal <- hormonal_summary
hormonal_n_counts <- filtered_contraceptive_table %>%
 filter(!is.na(Yes)) %>%
 group_by(Description) %>%
 summarise(N = n(), .groups = 'drop')

question_n_counts$hormonal <- hormonal_n_counts

question_names <- c("Age", "Gender", "Race", "Employment", "Marital Status",
            "Age of Diagnosis", "Symptom Duration", "Family Diagnosis",
            "Procedures", "Hormonal Contraceptive Use")

category_to_n_lookup <- list(
```

```
  "Age" = question_n_counts$age,
  "Gender" = question_n_counts$gender,
  "Race" = question_n_counts$race,
  "Employment" = question_n_counts$employment,
  "Marital Status" = question_n_counts$marital_status,
  "Age of Diagnosis" = question_n_counts$ageofdiagnosis,
  "Symptom Duration" = question_n_counts$symptomduration,
  "Family Diagnosis" = question_n_counts$familydiagnosis,
  "Procedures" = question_n_counts$procedures,
  "Hormonal Contraceptive Use" = question_n_counts$hormonal
)
table_1 <- bind_rows(
  summary_tables$age %>% rename("Option" = s01_age),
  summary_tables$gender %>% rename("Option" = s01_gender),
  summary_tables$race %>% rename("Option" = s01_race),
  summary_tables$employment %>% rename("Option" = s01_employment) %>%
    filter(!grepl("Other|Prefer not to say", Option)),
  summary_tables$marital_status %>% rename("Option" = s01_marital_status),
  summary_tables$ageofdiagnosis %>% rename("Option" = s01_ageofdiagnosis),
  summary_tables$symptomduration %>% rename("Option" = s01_durationofsymptoms),
  summary_tables$familydiagnosis %>% rename("Option" = s01_familymemberpd),
  summary_tables$procedures %>% rename("Option" = s01_surgeries),
  summary_tables$hormonal %>% rename("Option" = Yes)
)

table_1 <- table_1 %>%
  filter(!is.na(Option)) %>%
  filter(!grepl("prefer|Prefer|Other|other|NA|N/A|don't know|Don't know", Option))

for (cat_name in names(category_to_n_lookup)) {
  n_data <- category_to_n_lookup[[cat_name]]
  menstru_n <- n_data$N[n_data$Description == "Menstruating"]
  postmeno_n <- n_data$N[n_data$Description == "Postmenopausal"]

  if (length(menstru_n) > 0 && length(postmeno_n) > 0) {
    new_category_name <- paste0(cat_name, " (M: N=", menstru_n, ", PM: N=", postmeno_n, ")")
    table_1$Category[table_1$Category == cat_name] <- new_category_name
  }
}
table_1 <- table_1 %>%
  group_by(Category) %>%
  mutate(Category = ifelse(row_number() == 1, Category, "")) %>%
  ungroup()

table_1 <- table_1 %>% relocate(Category, Option)

new_column_names <- names(table_1)
for (col in c("Menstruating", "Postmenopausal")) {
  if (col %in% new_column_names) {
    names(table_1)[names(table_1) == col] <- col
  }
}

save(table_1, question_n_counts, file = "demographics_by_menstruation_UPDATED.RData")

display_table_1 <- table_1
```

```
current_category <- ""
for (i in 1:nrow(display_table_1)) {
 if (display_table_1$Category[i] != "") {
  current_category <- display_table_1$Category[i]

  if (current_category %in% names(category_to_n_lookup)) {
   n_counts <- category_to_n_lookup[[current_category]]

   for (j in 1:nrow(n_counts)) {
    status <- n_counts$Description[j]
    n_count <- n_counts$N[j]

    if (paste0(status, "_N") %in% colnames(display_table_1)) {
     old_col <- paste0(status, "_N")
     new_col <- paste0(status, " (N=", n_count, ")")
     colnames(display_table_1)[colnames(display_table_1) == old_col] <- new_col
    } else if (status %in% colnames(display_table_1)) {
     old_col <- status
     new_col <- paste0(status, " (N=", n_count, ")")
     colnames(display_table_1)[colnames(display_table_1) == old_col] <- new_col
    }
   }
  }
 }
}
kable_table <- kable(table_1,
             caption = "Table 1: Demographics by Menstruation Status (UPDATED with Question-Specific N
Counts)",
             format = "html") %>%
 kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
          full_width = FALSE, position = "center") %>%
 row_spec(0, background = "#f8f8f8", bold = TRUE) %>%
 column_spec(1:2, bold = TRUE) %>%
 collapse_rows(columns = 1, valign = "middle") %>%
 footnote(general = "N counts represent actual respondents to each question, excluding 'Prefer not to
say' and missing responses",
       general_title = "Note: ")

footnote_text <- "Question-specific sample sizes: "
for (cat_name in names(category_to_n_lookup)) {
 n_data <- category_to_n_lookup[[cat_name]]
 menstru_n <- n_data$N[n_data$Description == "Menstruating"]
 postmeno_n <- n_data$N[n_data$Description == "Postmenopausal"]
 if (length(menstru_n) > 0 && length(postmeno_n) > 0) {
  footnote_text <- paste0(footnote_text, cat_name, " (Menstruating N=", menstru_n,
                ", Postmenopausal N=", postmeno_n, "); ")
 }
}

kable_table <- kable_table %>%
 footnote(general = footnote_text, general_title = "Sample sizes: ")

cat(as.character(kable_table), file = "demographics_by_menstruation_UPDATED.html")

print(kable_table)
print(table_1)
```

Script 04.1

```
rm(list = ls(pattern = "^(filtered_menstruation|split_data|menstruation_counts|excluded_participants)"))
rm(list = ls(pattern = "^(split_.*_dist|split_.*[^_dist]$)"))
rm(list = ls(pattern = "^(result_table|long_data|cols)$"))
rm(list = ls(pattern = "^(create_split_|create_.*_visualization)"))
rm(list = ls(pattern = "_labels$"))
rm(list = ls(pattern = "split_colors"))

required_vars <- c("new_menstruationstatus_table", "data_UID", "motorsymptoms_table",
            "nonmotorsymptoms_table", "otherconditions_table", "therapy_table", "othermeds_table")

missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 loaded <- FALSE

 if (file.exists("master_data_preparation_environment.RData")) {
  load("master_data_preparation_environment.RData")
  loaded <- TRUE
 } else if (file.exists("03_menstruation_status_analysis.RData")) {
  load("03_menstruation_status_analysis.RData")
  loaded <- TRUE
 } else if (file.exists("survey_analysis_environment.RData")) {
  load("survey_analysis_environment.RData")
  loaded <- TRUE
 }
 if (!loaded) {
  stop("Required variables missing and no environment file found. Please run Script 03.0 first.\n",
      "Missing variables: ", paste(missing_vars, collapse = ", "))
 }
 still_missing <- c()
 for(var in required_vars) {
  if(!exists(var)) {
   still_missing <- c(still_missing, var)
  }
 }
 if(length(still_missing) > 0) {
  stop("Even after loading environment, some variables are still missing: ",
      paste(still_missing, collapse = ", "))
 }
}

if(!"Description" %in% colnames(new_menstruationstatus_table)) {
 stop("Description column not found in new_menstruationstatus_table. Please check Script 03.0
completed correctly.")
}

if(!"UniqueID" %in% colnames(data_UID)) {
 stop("UniqueID column not found in data_UID. Please check Script 03.0 completed correctly.")
}
```

```
required_demo_vars <- c("s01_age", "s01_race", "s01_employment", "s01_marital_status",
                "s01_ageofdiagnosis", "s01_durationofsymptoms", "s01_familymemberpd", "s01_surgeries")

missing_demo_vars <- setdiff(required_demo_vars, colnames(data_UID))
if(length(missing_demo_vars) > 0) {
 warning("Some demographic variables missing from data_UID: ",
       paste(missing_demo_vars, collapse = ", "))
}

menstruation_categories <- unique(new_menstruationstatus_table$Description)
required_categories <- c("Menstruating", "Postmenopausal")
available_categories <- intersect(required_categories, menstruation_categories)

if(length(available_categories) < 2) {
 stop("Insufficient menstruation categories for binary analysis. Available: ",
    paste(menstruation_categories, collapse = ", "))
}

menstruating_count <- sum(new_menstruationstatus_table$Description == "Menstruating", na.rm =
TRUE)
postmenopausal_count <- sum(new_menstruationstatus_table$Description == "Postmenopausal",
na.rm = TRUE)

if(menstruating_count < 5 || postmenopausal_count < 5) {
 warning("Small sample sizes detected for histograms: Menstruating=", menstruating_count,
       ", Postmenopausal=", postmenopausal_count)
}

library(Hmisc)
library(ggplot2)
library(dplyr)
library(tidyr)
library(forcats)
library(gridExtra)
library(patchwork)
filtered_menstruation <- new_menstruationstatus_table %>%
 filter(Description %in% c("Menstruating", "Postmenopausal"))
excluded_participants <- new_menstruationstatus_table %>%
 filter(!Description %in% c("Menstruating", "Postmenopausal"))
filtered_menstruation$BinaryStatus <- as.character(filtered_menstruation$Description)
split_data <- filtered_menstruation %>%
 left_join(data_UID, by = "UniqueID")
menstruation_counts <- table(split_data$BinaryStatus)
split_colors <- c("#E91E63", "#1565C0")
create_split_demographic_table <- function(data, var_name, levels, labels, var_label) {
 split_table <- data.frame(
  RecordID = data$record_id,
  UniqueID = data$UniqueID,
  Code = data[[var_name]],
  Description = factor(data[[var_name]], levels = levels, labels = labels),
  MenstruationStatus = data$BinaryStatus
 )

 split_table <- split_table[!is.na(split_table$Description) &
                 split_table$Description != "Prefer not to say", ]

 return(split_table)
```

```
}

create_split_visualization <- function(split_table, title, x_lab, y_lab = "No. of Participants", legend_title =
"Menstruation Status") {
 p <- ggplot(split_table, aes(x = Description, fill = MenstruationStatus)) +
  geom_bar(position = position_dodge()) +
  labs(title = title, x = x_lab, y = y_lab, fill = legend_title) +
  scale_fill_manual(values = split_colors) +
  theme_minimal() +
  theme(
   axis.text.x = element_text(angle = 45, hjust = 1),
   legend.position = "top",
   plot.title = element_text(hjust = 0.5)
  )

 return(p)
}

split_race <- create_split_demographic_table(
 split_data, "s01_race",
 levels = c(1, 2, 3, 4, 5),
 labels = c("White", "Black", "Asian", "Other", "Prefer not to say"),
 var_label = "Race"
)

split_race_dist <- create_split_visualization(
 split_race,
 "Race Distribution by Menstruation Status",
 "Race"
)

split_employment <- create_split_demographic_table(
 split_data, "s01_employment",
 levels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12),
 labels = c("Employed (full-time)", "Employed (part-time)", "Self-employed",
        "Unemployed (actively seeking)",
        "Unemployed (not seeking)",
        "Student (full-time)", "Student (part-time)",
        "Homemaker", "Retired", "Disabled", "Other", "Prefer not to say"),
 var_label = "Employment"
)

split_employment_dist <- create_split_visualization(
 split_employment,
 "Employment Distribution by Menstruation Status",
 "Employment"
)

split_marital_status <- create_split_demographic_table(
 split_data, "s01_marital_status",
 levels = c(1, 2, 3, 4, 5, 6, 7, 8, 9),
 labels = c("Single", "In a relationship", "Married", "Widowed",
        "Divorced", "Separated", "Domestic partnership",
        "Civil union", "Prefer not to say"),
 var_label = "Marital Status"
)
```

```
split_marital_status_dist <- create_split_visualization(
 split_marital_status,
 "Marital Status Distribution by Menstruation Status",
 "Marital Status"
)

split_age <- create_split_demographic_table(
 split_data, "s01_age",
 levels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13),
 labels = c("18 - 29 years old", "30 - 39 years old", "40 - 49 years old",
        "50 - 54 years old", "55 - 59 years old", "60 - 64 years old",
        "65 - 69 years old", "70 - 74 years old", "75 - 79 years old",
        "80 - 84 years old", "85 - 90 years old", "90 years old",
        "Prefer not to say"),
 var_label = "Age"
)

split_age_dist <- create_split_visualization(
 split_age,
 "Age Distribution by Menstruation Status",
 "Age Range"
)

split_ageofdiagnosis <- create_split_demographic_table(
 split_data, "s01_ageofdiagnosis",
 levels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13),
 labels = c("18 - 29 years old", "30 - 39 years old", "40 - 49 years old",
        "50 - 54 years old", "55 - 59 years old", "60 - 64 years old",
        "65 - 69 years old", "70 - 74 years old", "75 - 79 years old",
        "80 - 84 years old", "85 - 90 years old", "90 years old",
        "Prefer not to say"),
 var_label = "Age of Diagnosis"
)

split_ageofdiagnosis_dist <- create_split_visualization(
 split_ageofdiagnosis,
 "Age of Parkinson's Diagnosis by Menstruation Status",
 "Age Range"
)

split_symptomduration <- create_split_demographic_table(
 split_data, "s01_durationofsymptoms",
 levels = c(1, 2, 3, 4, 5, 6, 7),
 labels = c("0 - 3 years", "4 - 6 years", "7 - 9 years",
        "10 - 12 years", "13 - 15 years", ">15 years",
        "Prefer not to say"),
 var_label = "Symptom Duration"
)

split_symptomduration_dist <- create_split_visualization(
 split_symptomduration,
 "Symptom Duration Distribution by Menstruation Status",
 "Duration"
)

split_familydiagnosis <- create_split_demographic_table(
 split_data, "s01_familymemberpd",
```

```
  levels = c(1, 2, 3, 4),
  labels = c("Yes", "No", "Prefer not to say", "I don't know"),
  var_label = "Family Member with PD"
)

split_familydiagnosis_dist <- create_split_visualization(
  split_familydiagnosis,
  "Family History of Parkinson's by Menstruation Status",
  "Family Member with PD"
)

split_procedures <- create_split_demographic_table(
  split_data, "s01_surgeries",
  levels = c(1, 2, 3),
  labels = c("Yes", "No", "Prefer not to say"),
  var_label = "Procedures/Surgeries"
)

split_procedures_dist <- create_split_visualization(
  split_procedures,
  "Procedures/Surgeries by Menstruation Status",
  "Has Had Procedures/Surgeries"
)

create_processed_table_visualization <- function(data_table, table_name, title, legend_title =
"Menstruation Status") {
  merged_data <- data_table %>%
    inner_join(filtered_menstruation[, c("UniqueID", "BinaryStatus")], by = "UniqueID")

  symptom_cols <- setdiff(colnames(merged_data), c("RecordID", "UniqueID", "BinaryStatus"))

  long_data <- merged_data %>%
    select(-RecordID) %>%
    pivot_longer(
      cols = all_of(symptom_cols),
      names_to = "Option",
      values_to = "Value"
    ) %>%
    filter(Value == 1) %>%
    group_by(Option, BinaryStatus) %>%
    summarise(Count = n(), .groups = "drop") %>%
    mutate(Option = fct_reorder(Option, Count, .desc = TRUE))

  p <- ggplot(long_data, aes(x = Option, y = Count, fill = BinaryStatus)) +
    geom_bar(stat = "identity", position = position_dodge()) +
    labs(title = title, x = table_name, y = "Count", fill = legend_title) +
    scale_fill_manual(values = split_colors) +
    theme_minimal() +
    theme(
      axis.text.x = element_text(angle = 45, hjust = 1),
      legend.position = "top",
      plot.title = element_text(hjust = 0.5)
    )

  return(p)
}
```

```
split_motorsymptoms_dist <- create_processed_table_visualization(
 motorsymptoms_table,
 "Motor Symptoms",
 "Motor Symptoms by Menstruation Status"
)

split_secondarymotor_dist <- create_processed_table_visualization(
 othermotorsymptoms_table,
 "Secondary Motor Symptoms",
 "Secondary Motor Symptoms by Menstruation Status"
)

split_nonmotor_dist <- create_processed_table_visualization(
 nonmotorsymptoms_table,
 "Non-Motor Symptoms",
 "Non-Motor Symptoms by Menstruation Status"
)

split_otherconditions_dist <- create_processed_table_visualization(
 otherconditions_table,
 "Other Conditions",
 "Other Conditions by Menstruation Status"
)

split_therapy_dist <- create_processed_table_visualization(
 therapy_table,
 "Current Treatments",
 "Current Treatments by Menstruation Status"
)

split_othermeds_dist <- create_processed_table_visualization(
 othermeds_table,
 "Other Medications",
 "Other Medications by Menstruation Status"
)
```

Script 03.3.

```
rm(list = ls(pattern = "^(motor_plot|nonmotor_plot|medication_plot|results|plot_data)"))
rm(list = ls(pattern = "^(temp_data|ct|pct|test_result|symptom_order)$"))
rm(list = ls(pattern = "^create_diverging_bar_chart$"))
rm(list = ls(pattern = "_symptom_labels$|_category_labels$"))
required_vars <- c("new_menstruationstatus_table", "motorsymptoms_table",
                   "nonmotorsymptoms_table", "med_category_table", "master_symptom_orders")
missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 loaded <- FALSE
 if (file.exists("master_data_preparation_environment.RData")) {
  load("master_data_preparation_environment.RData")
  loaded <- TRUE
 } else if (file.exists("03_menstruation_status_analysis.RData")) {
  load("03_menstruation_status_analysis.RData")
  loaded <- TRUE
 } else if (file.exists("survey_analysis_environment.RData")) {
  load("survey_analysis_environment.RData")
  loaded <- TRUE
 }
 if (!loaded) {
  stop("Required variables missing and no environment file found. Please run Script 03.0 first.\n",
       "Missing variables: ", paste(missing_vars, collapse = ", "))
 }
 still_missing <- c()
 for(var in required_vars) {
  if(!exists(var)) {
   still_missing <- c(still_missing, var)
  }
 }

 if(length(still_missing) > 0) {
  stop("Even after loading environment, some variables are still missing: ",
       paste(still_missing, collapse = ", "))
 }
}
if(!"Description" %in% colnames(new_menstruationstatus_table)) {
 stop("Description column not found in new_menstruationstatus_table. Please check Script 03.0
completed correctly.")
}
menstruation_categories <- unique(new_menstruationstatus_table$Description)
required_categories <- c("Menstruating", "Postmenopausal")
missing_categories <- setdiff(required_categories, menstruation_categories)
```

```
if(length(missing_categories) > 0) {
  stop("Required menstruation categories missing: ", paste(missing_categories, collapse = ", "))
}
if(!exists("master_symptom_orders")) {
  stop("Master symptom ordering system not found. Please ensure Script 03.0 Section 15 completed
correctly.")
}
if(!exists("apply_standard_symptom_order")) {
  stop("Helper functions for symptom ordering not found. Please ensure Script 03.0 Section 15 completed
correctly.")
}
menstruating_count <- sum(new_menstruationstatus_table$Description == "Menstruating", na.rm =
TRUE)
postmenopausal_count <- sum(new_menstruationstatus_table$Description == "Postmenopausal",
na.rm = TRUE)

if(menstruating_count < 5 || postmenopausal_count < 5) {
  warning("Small sample sizes detected: Menstruating=", menstruating_count,
          ", Postmenopausal=", postmenopausal_count)
}
library(dplyr)
library(tidyr)
library(ggplot2)
library(forcats)

motorsymptoms_table <- motorsymptoms_table[rowSums(motorsymptoms_table[,-(1:2)]) > 0, ]
nonmotorsymptoms_table <- nonmotorsymptoms_table[rowSums(nonmotorsymptoms_table[,-(1:2)]) >
0, ]
med_category_table <- med_category_table[rowSums(med_category_table[,-(1:2)]) > 0, ]

create_diverging_bar_chart <- function(menstruation_status, symptoms_data, title, symptom_type) {
  if(!symptom_type %in% names(master_symptom_orders)) {
    return(NULL)
  }

  valid_status <- menstruation_status %>%
    filter(Description %in% c("Menstruating", "Postmenopausal"))

  n_menstruating <- nrow(valid_status[valid_status$Description == "Menstruating", ])
  n_postmenopausal <- nrow(valid_status[valid_status$Description == "Postmenopausal", ])

  total_in_menstruation_table <- nrow(menstruation_status)
  excluded_count <- total_in_menstruation_table - nrow(valid_status)
  if (excluded_count > 0) {
    excluded_breakdown <-
table(menstruation_status$Description[!menstruation_status$Description %in% c("Menstruating",
"Postmenopausal")])
  }
```

```
results <- data.frame(
 SymptomName = character(),
 Menstruating = numeric(),
 Postmenopausal = numeric(),
 Difference = numeric(),
 P_Value = numeric(),
 stringsAsFactors = FALSE
)

symptoms_to_analyze <- master_symptom_orders[[symptom_type]]

for (symptom in symptoms_to_analyze) {
 if (!symptom %in% colnames(symptoms_data)) {
  next
 }

 temp_data <- data.frame(
  UniqueID = symptoms_data$UniqueID,
  Symptom = symptoms_data[[symptom]]
 )

 temp_data <- temp_data %>%
  inner_join(valid_status[, c("UniqueID", "Description")], by = "UniqueID") %>%
  rename(MenstruationStatus = Description)

 ct <- table(temp_data$Symptom, temp_data$MenstruationStatus)
 ct <- ct[, colnames(ct) %in% c("Menstruating", "Postmenopausal")]

 if (ncol(ct) < 2) next

 if (!("1" %in% rownames(ct))) next

 pct <- prop.table(ct, margin = 2) * 100

 menstruating_pct <- pct["1", "Menstruating"]
 postmenopausal_pct <- pct["1", "Postmenopausal"]

 expected_counts <- chisq.test(ct)$expected
 test_result <- if (any(expected_counts < 5)) {
  fisher.test(ct)
 } else {
  chisq.test(ct, correct = FALSE)
 }

 results <- rbind(results, data.frame(
  SymptomName = symptom,
  Menstruating = menstruating_pct,
  Postmenopausal = postmenopausal_pct,
```

```
  Difference = menstruating_pct - postmenopausal_pct,
  P_Value = test_result$p.value,
  stringsAsFactors = FALSE
 ))
}

results <- apply_standard_symptom_order(results, symptom_type)

results <- ensure_complete_symptom_coverage(results, symptom_type)

plot_data <- results %>%
 pivot_longer(
  cols = c(Menstruating, Postmenopausal),
  names_to = "Status",
  values_to = "Percentage"
 ) %>%
 mutate(
  PlotValue = ifelse(Status == "Menstruating", -Percentage, Percentage),
  Significance = case_when(
   P_Value < 0.001 ~ "***",
   P_Value < 0.01 ~ "**",
   P_Value < 0.05 ~ "*",
   TRUE ~ ""
  ),
  Label = paste0(round(Percentage, 1), "%", Significance)
 )

symptom_order <- rev(results$SymptomName)
plot_data$SymptomName <- factor(plot_data$SymptomName, levels = symptom_order)

colors <- c("Menstruating" = "#E91E63", "Postmenopausal" = "#1565C0")

p <- ggplot(plot_data, aes(x = PlotValue, y = SymptomName, fill = Status)) +
 geom_bar(stat = "identity", width = 0.7, color = "black") +
 geom_text(
  aes(label = Label, x = ifelse(Status == "Menstruating", -Percentage - 2, Percentage + 2)),
  size = 3, hjust = ifelse(plot_data$Status == "Menstruating", 1, 0)
 ) +
 scale_fill_manual(
  values = colors,
  labels = c(
   paste0("Menstruating (n=", n_menstruating, ")"),
   paste0("Postmenopausal (n=", n_postmenopausal, ")")
  )
 ) +
 scale_x_continuous(
  limits = c(-100, 100),
  breaks = seq(-100, 100, 20),
  labels = abs(seq(-100, 100, 20))
```

```
  ) +
  labs(
    title = title,
    subtitle = "Percentage of participants reporting each symptom (consistent ordering)",
    x = "Percentage (%)",
    y = "",
    fill = "Menstruation Status",
    caption = "* p<0.05, ** p<0.01, *** p<0.001\nSymptoms ordered consistently for cross-analysis
comparison"
  ) +
  theme_minimal() +
  theme(
    panel.grid.major.y = element_blank(),
    panel.grid.minor = element_blank(),
    legend.position = "top",
    plot.title = element_text(size = 14, face = "bold"),
    plot.subtitle = element_text(size = 12),
    axis.text.y = element_text(size = 10),
    axis.text.x = element_text(size = 9)
  ) +
  geom_vline(xintercept = 0, color = "black", linetype = "solid")

 return(list(
  plot = p,
  data = results
 ))
}

motor_plot <- create_diverging_bar_chart(
 new_menstruationstatus_table,
 motorsymptoms_table,
 "Motor Symptoms by Menstruation Status",
 "motor_symptoms"
)

nonmotor_plot <- create_diverging_bar_chart(
 new_menstruationstatus_table,
 nonmotorsymptoms_table,
 "Non-Motor Symptoms by Menstruation Status",
 "nonmotor_symptoms"
)

medication_plot <- create_diverging_bar_chart(
 new_menstruationstatus_table,
 med_category_table,
 "Medication Categories by Menstruation Status",
 "medication_categories"
)
```

```
othermotorsymptoms_plot <- create_diverging_bar_chart(
  new_menstruationstatus_table,
  othermotorsymptoms_table,
  "Other Motor Symptoms by Menstruation Status",
  "other_motor_symptoms"
)

if(!is.null(motor_plot)) print(motor_plot$plot)
if(!is.null(nonmotor_plot)) print(nonmotor_plot$plot)
if(!is.null(medication_plot)) print(medication_plot$plot)
if(!is.null(othermotorsymptoms_plot)) print(othermotorsymptoms_plot$plot)

if(!is.null(motor_plot)) {
  ggsave("motor_symptoms_by_menstruation_UPDATED.png", motor_plot$plot, width = 10, height = 7,
dpi = 300)
}
if(!is.null(nonmotor_plot)) {
  ggsave("nonmotor_symptoms_by_menstruation_UPDATED.png", nonmotor_plot$plot, width = 10,
height = 10, dpi = 300)
}
if(!is.null(medication_plot)) {
  ggsave("medication_categories_by_menstruation_UPDATED.png", medication_plot$plot, width = 10,
height = 7, dpi = 300)
}
```

Script 05

```
rm(list = ls(pattern = "^(motor_results|nonmotor_results|med_results|all_significant_findings)"))

rm(list = ls(pattern = "^(analysis_data|contingency_table|test_result|percentage_table)$"))

rm(list = ls(pattern = "^(findings_data|analysis_participants)$"))

rm(list = ls(pattern =
"^(prepare_analysis_data|perform_statistical_test|display_analysis_results|analyze_variable)$"))

rm(list = ls(pattern = "^(extract_significant_findings)$"))

required_vars <- c("new_menstruationstatus_table", "motorsymptoms_table",
                   "nonmotorsymptoms_table", "med_category_table", "master_symptom_orders")

missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 loaded <- FALSE

 if (file.exists("master_data_preparation_environment.RData")) {
  load("master_data_preparation_environment.RData")
  loaded <- TRUE
 } else if (file.exists("03_menstruation_status_analysis.RData")) {
  load("03_menstruation_status_analysis.RData")
  loaded <- TRUE
 } else if (file.exists("survey_analysis_environment.RData")) {
  load("survey_analysis_environment.RData")
  loaded <- TRUE
 }

 if (!loaded) {
  stop("Required variables missing and no environment file found. Please run Script 03.0 first.\n",
       "Missing variables: ", paste(missing_vars, collapse = ", "))
 }

 still_missing <- c()
 for(var in required_vars) {
  if(!exists(var)) {
   still_missing <- c(still_missing, var)
  }
 }

 if(length(still_missing) > 0) {
```

```
   stop("Even after loading environment, some variables are still missing: ",
        paste(still_missing, collapse = ", "))
  }
}

if(!"Description" %in% colnames(new_menstruationstatus_table)) {
  stop("Description column not found in new_menstruationstatus_table. Please check Script 03.0
completed correctly.")
}

if(!exists("master_symptom_orders")) {
  stop("Master symptom ordering system not found. Please ensure Script 03.0 Section 15 completed
correctly.")
}

menstruation_categories <- unique(new_menstruationstatus_table$Description)
analysis_categories <- c("Menstruating", "Postmenopausal")
available_analysis_categories <- intersect(analysis_categories, menstruation_categories)

if(length(available_analysis_categories) < 2) {
  stop("Insufficient menstruation categories for correlation analysis. Available: ",
       paste(menstruation_categories, collapse = ", "))
}

menstruating_count <- sum(new_menstruationstatus_table$Description == "Menstruating", na.rm =
TRUE)
postmenopausal_count <- sum(new_menstruationstatus_table$Description == "Postmenopausal",
na.rm = TRUE)

if(menstruating_count < 10 || postmenopausal_count < 10) {
  warning("Small sample sizes for correlation analysis: Menstruating=", menstruating_count,
          ", Postmenopausal=", postmenopausal_count)
}

library(dplyr)
library(tidyr)
library(knitr)
library(kableExtra)

motorsymptoms_table <- motorsymptoms_table[rowSums(motorsymptoms_table[,-(1:2)]) > 0, ]
nonmotorsymptoms_table <- nonmotorsymptoms_table[rowSums(nonmotorsymptoms_table[,-(1:2)]) >
0, ]
med_category_table <- med_category_table[rowSums(med_category_table[,-(1:2)]) > 0, ]

prepare_analysis_data <- function(df, col1, col2, status_filter = c("Menstruating", "Postmenopausal")) {
  df <- df[df$Description %in% status_filter, ]
  df[[col1]] <- as.character(df[[col1]])
  df[[col2]] <- as.character(df[[col2]])
  return(df)
```

```
}

perform_statistical_test <- function(contingency_table) {
 if (nrow(contingency_table) < 2 || ncol(contingency_table) < 2) {
  return(list(p.value = NA, method = "Insufficient data"))
 }

 expected_counts <- chisq.test(contingency_table)$expected

 if (any(expected_counts < 5)) {
  return(fisher.test(contingency_table))
 } else {
  return(chisq.test(contingency_table, correct = FALSE))
 }
}

display_analysis_results <- function(variable_name, contingency_table, test_result) {
 col_names <- colnames(contingency_table)
 n_values <- colSums(contingency_table)

 percentage_table <- prop.table(contingency_table, margin = 2) * 100

 colnames(percentage_table) <- paste0(col_names, " (N=", n_values, ") (%)")

 test_name <- if(class(test_result)[1] == "htest") test_result$method else "Fisher's Exact Test"

 return(list(
  contingency_table = contingency_table,
  percentage_table = percentage_table,
  p_value = test_result$p.value,
  test = test_name
 ))
}

analyze_variable <- function(data_table_name, variable, variable_display_name) {
 df <- get(data_table_name)

 analysis_data <- merge(new_menstruationstatus_table, df, by = c("RecordID", "UniqueID"))
 analysis_data <- prepare_analysis_data(analysis_data, "Description", variable)

 contingency_table <- table(analysis_data[[variable]], analysis_data$Description)

 test_result <- perform_statistical_test(contingency_table)

 return(display_analysis_results(variable_display_name, contingency_table, test_result))
}

analysis_participants <- new_menstruationstatus_table[
 new_menstruationstatus_table$Description %in% c("Menstruating", "Postmenopausal"),
```

```
]

if(exists("master_symptom_orders")) {
 motor_symptoms <- master_symptom_orders$motor_symptoms
 nonmotor_symptoms <- master_symptom_orders$nonmotor_symptoms
 med_categories <- master_symptom_orders$medication_categories
} else {
 motor_symptoms <- c("Tremors", "Rigidity", "Slowness", "Imbalance",
                     "Postural instability", "Speech/Writing", "Swallowing")

 nonmotor_symptoms <- c("Depression/Anxiety", "Sleep", "Hyposmia", "Constipation",
                        "Cognitive Changes", "Concentration", "Fatigue", "Sweating",
                        "Sexual Dysfunction", "Hypotension", "Vision", "Speech",
                        "Pain", "Mood Swings", "Skin", "Weight", "Urinary")

 med_categories <- c("L_DOPA", "DA_Agonist", "MAOB_Inhibitors",
                     "Anticholinergics", "COMT_Inhibitors")
}

motor_results <- list()

for (symptom in motor_symptoms) {
 if(symptom %in% colnames(motorsymptoms_table)) {
  motor_results[[symptom]] <- analyze_variable("motorsymptoms_table", symptom, symptom)
 }
}

nonmotor_results <- list()

for (symptom in nonmotor_symptoms) {
 if(symptom %in% colnames(nonmotorsymptoms_table)) {
  nonmotor_results[[symptom]] <- analyze_variable("nonmotorsymptoms_table", symptom, symptom)
 }
}

med_results <- list()

for (category in med_categories) {
 if(category %in% colnames(med_category_table)) {
  med_results[[category]] <- analyze_variable("med_category_table", category, category)
 }
}

save(motor_results, nonmotor_results, med_results,
    file = "symptom_analysis.RData")

extract_significant_findings <- function(results_list, category_name, original_order) {
 findings_data <- data.frame(
  Category = character(),
```

```
  Variable = character(),
  P_Value = numeric(),
  Test = character(),
  Menstruating_Percent = numeric(),
  Postmenopausal_Percent = numeric(),
  Original_Order = numeric(),
  stringsAsFactors = FALSE
 )

 for (i in seq_along(original_order)) {
  var_name <- original_order[i]
  if (var_name %in% names(results_list)) {
   result <- results_list[[var_name]]
   if (!is.na(result$p_value) && result$p_value < 0.05) {
    ct <- result$contingency_table

    if ("1" %in% rownames(ct) && ncol(ct) >= 2) {
     menstru_count <- ct["1", "Menstruating"]
     postmeno_count <- ct["1", "Postmenopausal"]

     total_menstruating <- sum(ct[, "Menstruating"])
     total_postmenopausal <- sum(ct[, "Postmenopausal"])

     menstru_percent <- (menstru_count / total_menstruating) * 100
     postmeno_percent <- (postmeno_count / total_postmenopausal) * 100

     findings_data <- rbind(findings_data, data.frame(
      Category = category_name,
      Variable = var_name,
      P_Value = result$p_value,
      Test = result$test,
      Menstruating_Percent = menstru_percent,
      Postmenopausal_Percent = postmeno_percent,
      Original_Order = i,
      stringsAsFactors = FALSE
     ))
    }
   }
  }
 }

 return(findings_data)
}

all_significant_findings <- rbind(
 extract_significant_findings(motor_results, "Motor Symptoms", motor_symptoms),
 extract_significant_findings(nonmotor_results, "Non-Motor Symptoms", nonmotor_symptoms),
 extract_significant_findings(med_results, "Medication Categories", med_categories)
)
```

```
if (nrow(all_significant_findings) > 0) {
  category_order <- c("Motor Symptoms", "Non-Motor Symptoms", "Medication Categories")
  all_significant_findings$Category <- factor(all_significant_findings$Category, levels = category_order)

  all_significant_findings <- all_significant_findings[order(all_significant_findings$Category,
all_significant_findings$Original_Order), ]

  unique_tests <- unique(all_significant_findings$Test)

  if (length(unique_tests) == 1) {
    if (grepl("Chi-square", unique_tests[1], ignore.case = TRUE)) {
      legend_text <- "Statistical analyses performed using chi-squared tests."
    } else if (grepl("Fisher", unique_tests[1], ignore.case = TRUE)) {
      legend_text <- "Statistical analyses performed using Fisher's exact tests."
    } else {
      legend_text <- paste("Statistical analyses performed using", unique_tests[1], ".")
    }
  } else {
    chi_sq_used <- any(grepl("Chi-square", unique_tests, ignore.case = TRUE))
    fisher_used <- any(grepl("Fisher", unique_tests, ignore.case = TRUE))

    if (chi_sq_used && fisher_used) {
      legend_text <- "Statistical analyses performed using chi-squared tests and Fisher's exact tests (used
when expected cell counts < 5)."
    } else {
      legend_text <- paste("Statistical analyses performed using:", paste(unique_tests, collapse = ", "), ".")
    }
  }

  display_findings <- all_significant_findings[, c("Category", "Variable", "P_Value",
"Menstruating_Percent", "Postmenopausal_Percent")]
  display_findings$P_Value <- sprintf("%.4f", display_findings$P_Value)
  display_findings$Menstruating_Percent <- sprintf("%.1f%%", display_findings$Menstruating_Percent)
  display_findings$Postmenopausal_Percent <- sprintf("%.1f%%",
display_findings$Postmenopausal_Percent)

  colnames(display_findings) <- c(
    "Category", "Variable", "P-Value",
    "Menstruating %", "Postmenopausal %"
  )

  enhanced_table <- kable(display_findings,
                          format = "html",
                          row.names = FALSE,
                          escape = FALSE,
                          caption = paste("Significant Associations between Menstruation Status and
Symptoms/Medications<br><small>", legend_text, "</small>")) %>%
    kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
```

```
            full_width = FALSE, position = "center") %>%
   row_spec(0, background = "#f8f8f8", bold = TRUE) %>%
   column_spec(1:2, bold = TRUE) %>%
   collapse_rows(columns = 1, valign = "middle")

  print(enhanced_table)

  write.csv(all_significant_findings, "significant_findings.csv", row.names = FALSE)
}

save.image(file = "symptom_analysis_complete.RData")
```

Script 03.2.1.

```
rm(list = ls(pattern =
"^(postmenopausal_|othermeds_hrt|demographics_by_hrt|hrt_|motor_hrt|nonmotor_hrt|med_hrt)"))

rm(list = ls(pattern = "^(results|cont_table|percent_table|analysis_plots)$"))

rm(list = ls(pattern = "^(analyze_|create_hrt_)"))

required_vars <- c("othermeds_table", "new_menstruationstatus_table", "motorsymptoms_table",
                   "nonmotorsymptoms_table", "med_category_table", "data_UID", "master_symptom_orders")

missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 if (file.exists("master_data_preparation_environment.RData")) {
  load("master_data_preparation_environment.RData")
 } else {
  stop("Required variables missing and master environment file not found. Please run Script 03.0
first.\n",
       "Missing variables: ", paste(missing_vars, collapse = ", "))
 }

 still_missing <- c()
 for(var in required_vars) {
  if(!exists(var)) {
   still_missing <- c(still_missing, var)
  }
 }

 if(length(still_missing) > 0) {
  stop("Even after loading master environment, some variables are still missing: ",
       paste(still_missing, collapse = ", "))
 }
}

if(!"HRT" %in% colnames(othermeds_table)) {
 stop("HRT column not found in othermeds_table. Please check Script 03.0 completed correctly.")
}

if(!"Description" %in% colnames(new_menstruationstatus_table)) {
 stop("Description column not found in new_menstruationstatus_table. Please check Script 03.0
completed correctly.")
}
```

```
if(!exists("master_symptom_orders")) {
 stop("Master symptom ordering system not found. Please ensure Script 03.0 Section 15 completed
correctly.")
}

if(!exists("apply_standard_symptom_order")) {
 stop("Helper functions for symptom ordering not found. Please ensure Script 03.0 Section 15 completed
correctly.")
}

library(dplyr)
library(ggplot2)
library(tidyr)
library(knitr)
library(forcats)

motorsymptoms_table <- motorsymptoms_table[rowSums(motorsymptoms_table[,-(1:2)]) > 0, ]
nonmotorsymptoms_table <- nonmotorsymptoms_table[rowSums(nonmotorsymptoms_table[,-(1:2)]) >
0, ]
med_category_table <- med_category_table[rowSums(med_category_table[,-(1:2)]) > 0, ]

postmenopausal_participants <- new_menstruationstatus_table %>%
 filter(Description == "Postmenopausal")

postmenopausal_ids <- postmenopausal_participants$UniqueID

othermeds_hrt <- othermeds_table %>%
 select(RecordID, UniqueID, HRT) %>%
 mutate(HRT_User = ifelse(HRT == 1, "HRT User", "Non-HRT User"))

othermeds_hrt_postmenopausal <- othermeds_hrt %>%
 filter(UniqueID %in% postmenopausal_ids)

analyze_demographics_by_hrt_postmenopausal <- function() {
 othermeds_hrt_unique <- othermeds_hrt_postmenopausal %>%
  distinct(UniqueID, .keep_all = TRUE)

 demographics_by_hrt <- merge(
  othermeds_hrt_unique,
  data_UID,
  by = "UniqueID",
  all.x = TRUE
 )

 analyze_categorical <- function(variable_name, variable_label, group_var = "HRT_User") {
  if (!variable_name %in% names(demographics_by_hrt)) {
   return(NULL)
  }
```

```
  cont_table <- table(demographics_by_hrt[[group_var]], demographics_by_hrt[[variable_name]])

  percent_table <- prop.table(cont_table, margin = 1) * 100

  expected_counts <- tryCatch({
    chisq.test(cont_table, simulate.p.value = TRUE)$expected
  }, error = function(e) {
    return(matrix(0, nrow = nrow(cont_table), ncol = ncol(cont_table)))
  })

  if (any(expected_counts < 5)) {
    test_result <- tryCatch({
      fisher.test(cont_table)
    }, error = function(e) {
      return(list(p.value = NA))
    })
  } else {
    test_result <- tryCatch({
      chisq.test(cont_table, correct = TRUE)
    }, error = function(e) {
      tryCatch({
        fisher.test(cont_table)
      }, error = function(e2) {
        return(list(p.value = NA))
      })
    })
  }

  return(list(
    counts = cont_table,
    percentages = percent_table,
    p_value = test_result$p.value
  ))
}

  age_summary <- analyze_categorical("s01_age", "Age")
  race_summary <- analyze_categorical("s01_race", "Race/Ethnicity")
  employment_summary <- analyze_categorical("s01_employment", "Employment Status")
  marital_status_summary <- analyze_categorical("s01_marital_status", "Marital Status")

  duration_summary <- analyze_categorical("s01_durationofsymptoms", "PD Symptom Duration")

  diagnosis_age_summary <- analyze_categorical("s01_ageofdiagnosis", "Age at PD Diagnosis")

  hrt_percent <- round(prop.table(table(demographics_by_hrt$HRT_User)) * 100, 1)

  return(demographics_by_hrt)
}
```

```
demographics_by_hrt_postmenopausal <- analyze_demographics_by_hrt_postmenopausal()

create_hrt_diverging_barplot <- function(data_table, symptom_names, title, symptom_type) {
 n_hrt_user <- sum(othermeds_hrt_postmenopausal$HRT_User == "HRT User", na.rm = TRUE)
 n_non_hrt_user <- sum(othermeds_hrt_postmenopausal$HRT_User == "Non-HRT User", na.rm = TRUE)

 results <- data.frame(
  SymptomName = character(),
  HRT_User = numeric(),
  NonHRT_User = numeric(),
  Difference = numeric(),
  P_Value = numeric(),
  stringsAsFactors = FALSE
 )

 if(!symptom_type %in% names(master_symptom_orders)) {
  return(NULL)
 }

 symptoms_to_analyze <- master_symptom_orders[[symptom_type]]

 for (symptom in symptoms_to_analyze) {
  if (!symptom %in% colnames(data_table)) {
   next
  }

  temp_data <- data.frame(
   UniqueID = data_table$UniqueID,
   Symptom = data_table[[symptom]]
  )

  temp_data <- temp_data %>%
   filter(UniqueID %in% postmenopausal_ids)

  temp_data <- temp_data %>%
   inner_join(othermeds_hrt_postmenopausal[, c("UniqueID", "HRT_User")], by = "UniqueID")

  ct <- table(temp_data$Symptom, temp_data$HRT_User)

  if (nrow(ct) < 2 || ncol(ct) < 2) {
   next
  }

  pct <- prop.table(ct, margin = 2) * 100

  hrt_user_pct <- pct["1", "HRT User"]
  non_hrt_user_pct <- pct["1", "Non-HRT User"]

  expected_counts <- tryCatch({
```

```
    chisq.test(ct)$expected
  }, error = function(e) {
    return(matrix(0, nrow = nrow(ct), ncol = ncol(ct)))
  })

  if (any(expected_counts < 5)) {
    test_result <- tryCatch({
      fisher.test(ct)
    }, error = function(e) {
      return(list(p.value = NA))
    })
  } else {
    test_result <- tryCatch({
      chisq.test(ct)
    }, error = function(e) {
      tryCatch({
        fisher.test(ct)
      }, error = function(e2) {
        return(list(p.value = NA))
      })
    })
  }

  results <- rbind(results, data.frame(
    SymptomName = symptom,
    HRT_User = hrt_user_pct,
    NonHRT_User = non_hrt_user_pct,
    Difference = hrt_user_pct - non_hrt_user_pct,
    P_Value = test_result$p.value,
    stringsAsFactors = FALSE
  ))
}

results <- apply_standard_symptom_order(results, symptom_type)

results <- ensure_complete_symptom_coverage(results, symptom_type)

if (nrow(results) == 0) {
  return(NULL)
}

plot_data <- results %>%
  pivot_longer(
    cols = c(HRT_User, NonHRT_User),
    names_to = "Status",
    values_to = "Percentage"
  ) %>%
  mutate(
    PlotValue = ifelse(Status == "HRT_User", -Percentage, Percentage),
```

```
    Significance = case_when(
      P_Value < 0.001 ~ "***",
      P_Value < 0.01 ~ "**",
      P_Value < 0.05 ~ "*",
      TRUE ~ ""
    ),
    Label = paste0(round(Percentage, 1), "%", Significance)
  )

 symptom_order <- rev(results$SymptomName)
 plot_data$SymptomName <- factor(plot_data$SymptomName, levels = symptom_order)

 colors <- c("HRT_User" = "#7B68EE", "NonHRT_User" = "#808080")

 p <- ggplot(plot_data, aes(x = PlotValue, y = SymptomName, fill = Status)) +
  geom_bar(stat = "identity", width = 0.7) +
  geom_text(
    aes(label = Label, x = ifelse(Status == "HRT_User", -Percentage - 2, Percentage + 2)),
    size = 3, hjust = ifelse(plot_data$Status == "HRT_User", 1, 0)
  ) +
  scale_fill_manual(
    values = colors,
    labels = c(
      paste0("HRT Users (n=", n_hrt_user, ")"),
      paste0("Non-HRT Users (n=", n_non_hrt_user, ")")
    )
  ) +
  scale_x_continuous(
    limits = c(-100, 100),
    breaks = seq(-100, 100, 20),
    labels = abs(seq(-100, 100, 20))
  ) +
  labs(
    title = title,
    subtitle = "Percentage of postmenopausal participants reporting each symptom (consistent
ordering)",
    x = "Percentage (%)",
    y = "",
    fill = "Treatment Status:",
    caption = "* p<0.05, ** p<0.01, *** p<0.001\nSymptoms ordered consistently for cross-analysis
comparison"
  ) +
  theme_minimal() +
  theme(
    panel.grid.major.y = element_blank(),
    panel.grid.minor = element_blank(),
    legend.position = "top",
    plot.title = element_text(size = 14, face = "bold"),
    plot.subtitle = element_text(size = 12),
```

```
    axis.text.y = element_text(size = 10),
    axis.text.x = element_text(size = 9)
   ) +
   geom_vline(xintercept = 0, color = "black", linetype = "solid")

 return(list(
   plot = p,
   data = results
 ))
}

motor_hrt_plot <- create_hrt_diverging_barplot(
 motorsymptoms_table,
 NULL,
 "Motor Symptoms by HRT Status in Postmenopausal Women",
 "motor_symptoms"
)

nonmotor_hrt_plot <- create_hrt_diverging_barplot(
 nonmotorsymptoms_table,
 NULL,
 "Non-Motor Symptoms by HRT Status in Postmenopausal Women",
 "nonmotor_symptoms"
)

med_hrt_plot <- create_hrt_diverging_barplot(
 med_category_table,
 NULL,
 "Medication Categories by HRT Status in Postmenopausal Women",
 "medication_categories"
)

othermotorsymptoms_hrt_plot <- create_hrt_diverging_barplot(
 othermotorsymptoms_table,
 NULL,
 "Other Motor Symptoms by HRT Status in Postmenopausal Women",
 "other_motor_symptoms"
)

if (!is.null(motor_hrt_plot)) {
 print(motor_hrt_plot$plot)
 ggsave("motor_symptoms_by_hrt_postmenopausal.png", motor_hrt_plot$plot, width = 10, height = 7,
dpi = 300)
}

if (!is.null(nonmotor_hrt_plot)) {
 print(nonmotor_hrt_plot$plot)
 ggsave("nonmotor_symptoms_by_hrt_postmenopausal.png", nonmotor_hrt_plot$plot, width = 10,
height = 10, dpi = 300)
```

```
}

if (!is.null(med_hrt_plot)) {
  print(med_hrt_plot$plot)
  ggsave("med_categories_by_hrt_postmenopausal.png", med_hrt_plot$plot, width = 10, height = 7, dpi =
300)
}
if (!is.null(othermotorsymptoms_hrt_plot)) {
  print(othermotorsymptoms_hrt_plot$plot)
  ggsave("othermotorsymptoms_by_hrt_postmenopausal.png", othermotorsymptoms_hrt_plot$plot,
width = 10, height = 7, dpi = 300)
}

postmenopausal_hrt_analysis_plots <- list(
  motor_symptoms = if (!is.null(motor_hrt_plot)) motor_hrt_plot$data else NULL,
  nonmotor_symptoms = if (!is.null(nonmotor_hrt_plot)) nonmotor_hrt_plot$data else NULL,
  med_categories = if (!is.null(med_hrt_plot)) med_hrt_plot$data else NULL,
  othermotorsymptoms = if (!is.null(othermotorsymptoms_hrt_plot)) othermotorsymptoms_hrt_plot$data
else NULL
)
save(postmenopausal_hrt_analysis_plots, file = "hrt_postmenopausal_plots_data.RData")
```

Script 03.2.1.1.

```
rm(list = ls(pattern =
"^(postmenopausal_|othermeds_hrt|demographics_by_hrt|hrt_|motor_hrt|nonmotor_hrt|med_hrt)"))

rm(list = ls(pattern = "^(results|cont_table|percent_table|analysis_plots)$"))

rm(list = ls(pattern = "^(analyze_|create_hrt_)"))

required_vars <- c("othermeds_table", "new_menstruationstatus_table", "motorsymptoms_table",
                   "nonmotorsymptoms_table", "med_category_table", "data_UID", "master_symptom_orders")

missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 if (file.exists("master_data_preparation_environment.RData")) {
  load("master_data_preparation_environment.RData")
 } else {
  stop("Required variables missing and master environment file not found. Please run Script 03.0
first.\n",
       "Missing variables: ", paste(missing_vars, collapse = ", "))
 }

 still_missing <- c()
 for(var in required_vars) {
  if(!exists(var)) {
   still_missing <- c(still_missing, var)
  }
 }

 if(length(still_missing) > 0) {
  stop("Even after loading master environment, some variables are still missing: ",
       paste(still_missing, collapse = ", "))
 }
}

if(!"HRT" %in% colnames(othermeds_table)) {
 stop("HRT column not found in othermeds_table. Please check Script 03.0 completed correctly.")
}

if(!"Description" %in% colnames(new_menstruationstatus_table)) {
 stop("Description column not found in new_menstruationstatus_table. Please check Script 03.0
completed correctly.")
}
```

```
if(!exists("master_symptom_orders")) {
 stop("Master symptom ordering system not found. Please ensure Script 03.0 Section 15 completed
correctly.")
}

if(!exists("apply_standard_symptom_order")) {
 stop("Helper functions for symptom ordering not found. Please ensure Script 03.0 Section 15 completed
correctly.")
}

library(dplyr)
library(ggplot2)
library(tidyr)
library(knitr)
library(forcats)

motorsymptoms_table <- motorsymptoms_table[rowSums(motorsymptoms_table[,-(1:2)]) > 0, ]
nonmotorsymptoms_table <- nonmotorsymptoms_table[rowSums(nonmotorsymptoms_table[,-(1:2)]) >
0, ]
med_category_table <- med_category_table[rowSums(med_category_table[,-(1:2)]) > 0, ]

postmenopausal_participants <- new_menstruationstatus_table %>%
 filter(Description == "Postmenopausal")

postmenopausal_ids <- postmenopausal_participants$UniqueID

othermeds_hrt <- othermeds_table %>%
 select(RecordID, UniqueID, HRT) %>%
 mutate(HRT_User = ifelse(HRT == 1, "HRT User", "Non-HRT User"))

othermeds_hrt_postmenopausal <- othermeds_hrt %>%
 filter(UniqueID %in% postmenopausal_ids)

analyze_demographics_by_hrt_postmenopausal <- function() {
 othermeds_hrt_unique <- othermeds_hrt_postmenopausal %>%
  distinct(UniqueID, .keep_all = TRUE)

 demographics_by_hrt <- merge(
  othermeds_hrt_unique,
  data_UID,
  by = "UniqueID",
  all.x = TRUE
 )

 analyze_categorical <- function(variable_name, variable_label, group_var = "HRT_User") {
  if (!variable_name %in% names(demographics_by_hrt)) {
   return(NULL)
  }
```

```
  cont_table <- table(demographics_by_hrt[[group_var]], demographics_by_hrt[[variable_name]])

  percent_table <- prop.table(cont_table, margin = 1) * 100

  expected_counts <- tryCatch({
    chisq.test(cont_table, simulate.p.value = TRUE)$expected
  }, error = function(e) {
    return(matrix(0, nrow = nrow(cont_table), ncol = ncol(cont_table)))
  })

  if (any(expected_counts < 5)) {
    test_result <- tryCatch({
      fisher.test(cont_table)
    }, error = function(e) {
      return(list(p.value = NA))
    })
  } else {
    test_result <- tryCatch({
      chisq.test(cont_table, correct = TRUE)
    }, error = function(e) {
      tryCatch({
        fisher.test(cont_table)
      }, error = function(e2) {
        return(list(p.value = NA))
      })
    })
  }

  return(list(
    counts = cont_table,
    percentages = percent_table,
    p_value = test_result$p.value
  ))
}

  age_summary <- analyze_categorical("s01_age", "Age")
  race_summary <- analyze_categorical("s01_race", "Race/Ethnicity")
  employment_summary <- analyze_categorical("s01_employment", "Employment Status")
  marital_status_summary <- analyze_categorical("s01_marital_status", "Marital Status")

  duration_summary <- analyze_categorical("s01_durationofsymptoms", "PD Symptom Duration")

  diagnosis_age_summary <- analyze_categorical("s01_ageofdiagnosis", "Age at PD Diagnosis")

  hrt_percent <- round(prop.table(table(demographics_by_hrt$HRT_User)) * 100, 1)

  return(demographics_by_hrt)
}
```

```
demographics_by_hrt_postmenopausal <- analyze_demographics_by_hrt_postmenopausal()

create_hrt_diverging_barplot <- function(data_table, symptom_names, title, symptom_type) {
 n_hrt_user <- sum(othermeds_hrt_postmenopausal$HRT_User == "HRT User", na.rm = TRUE)
 n_non_hrt_user <- sum(othermeds_hrt_postmenopausal$HRT_User == "Non-HRT User", na.rm = TRUE)

 results <- data.frame(
  SymptomName = character(),
  HRT_User = numeric(),
  NonHRT_User = numeric(),
  Difference = numeric(),
  P_Value = numeric(),
  stringsAsFactors = FALSE
 )

 if(!symptom_type %in% names(master_symptom_orders)) {
  return(NULL)
 }

 symptoms_to_analyze <- master_symptom_orders[[symptom_type]]

 for (symptom in symptoms_to_analyze) {
  if (!symptom %in% colnames(data_table)) {
   next
  }

  temp_data <- data.frame(
   UniqueID = data_table$UniqueID,
   Symptom = data_table[[symptom]]
  )

  temp_data <- temp_data %>%
   filter(UniqueID %in% postmenopausal_ids)

  temp_data <- temp_data %>%
   inner_join(othermeds_hrt_postmenopausal[, c("UniqueID", "HRT_User")], by = "UniqueID")

  ct <- table(temp_data$Symptom, temp_data$HRT_User)

  if (nrow(ct) < 2 || ncol(ct) < 2) {
   next
  }

  pct <- prop.table(ct, margin = 2) * 100

  hrt_user_pct <- pct["1", "HRT User"]
  non_hrt_user_pct <- pct["1", "Non-HRT User"]

  expected_counts <- tryCatch({
```

```
    chisq.test(ct)$expected
  }, error = function(e) {
    return(matrix(0, nrow = nrow(ct), ncol = ncol(ct)))
  })

  if (any(expected_counts < 5)) {
    test_result <- tryCatch({
      fisher.test(ct)
    }, error = function(e) {
      return(list(p.value = NA))
    })
  } else {
    test_result <- tryCatch({
      chisq.test(ct)
    }, error = function(e) {
      tryCatch({
        fisher.test(ct)
      }, error = function(e2) {
        return(list(p.value = NA))
      })
    })
  }

  results <- rbind(results, data.frame(
    SymptomName = symptom,
    HRT_User = hrt_user_pct,
    NonHRT_User = non_hrt_user_pct,
    Difference = hrt_user_pct - non_hrt_user_pct,
    P_Value = test_result$p.value,
    stringsAsFactors = FALSE
  ))
}

results <- apply_standard_symptom_order(results, symptom_type)

results <- ensure_complete_symptom_coverage(results, symptom_type)

if (nrow(results) == 0) {
  return(NULL)
}

plot_data <- results %>%
  pivot_longer(
    cols = c(HRT_User, NonHRT_User),
    names_to = "Status",
    values_to = "Percentage"
  ) %>%
  mutate(
    PlotValue = ifelse(Status == "HRT_User", -Percentage, Percentage),
```

```
    Significance = case_when(
      P_Value < 0.001 ~ "***",
      P_Value < 0.01 ~ "**",
      P_Value < 0.05 ~ "*",
      TRUE ~ ""
    ),
    Label = paste0(round(Percentage, 1), "%", Significance)
  )

 symptom_order <- rev(results$SymptomName)
 plot_data$SymptomName <- factor(plot_data$SymptomName, levels = symptom_order)

 colors <- c("HRT_User" = "#7B68EE", "NonHRT_User" = "#808080")

 p <- ggplot(plot_data, aes(x = PlotValue, y = SymptomName, fill = Status)) +
  geom_bar(stat = "identity", width = 0.7) +
  geom_text(
    aes(label = Label, x = ifelse(Status == "HRT_User", -Percentage - 2, Percentage + 2)),
    size = 3, hjust = ifelse(plot_data$Status == "HRT_User", 1, 0)
  ) +
  scale_fill_manual(
    values = colors,
    labels = c(
      paste0("HRT Users (n=", n_hrt_user, ")"),
      paste0("Non-HRT Users (n=", n_non_hrt_user, ")")
    )
  ) +
  scale_x_continuous(
    limits = c(-100, 100),
    breaks = seq(-100, 100, 20),
    labels = abs(seq(-100, 100, 20))
  ) +
  labs(
    title = title,
    subtitle = "Percentage of postmenopausal participants reporting each symptom (consistent
ordering)",
    x = "Percentage (%)",
    y = "",
    fill = "Treatment Status:",
    caption = "* p<0.05, ** p<0.01, *** p<0.001\nSymptoms ordered consistently for cross-analysis
comparison"
  ) +
  theme_minimal() +
  theme(
    panel.grid.major.y = element_blank(),
    panel.grid.minor = element_blank(),
    legend.position = "top",
    plot.title = element_text(size = 14, face = "bold"),
    plot.subtitle = element_text(size = 12),
```

```
    axis.text.y = element_text(size = 10),
    axis.text.x = element_text(size = 9)
   ) +
   geom_vline(xintercept = 0, color = "black", linetype = "solid")

 return(list(
   plot = p,
   data = results
 ))
}

motor_hrt_plot <- create_hrt_diverging_barplot(
 motorsymptoms_table,
 NULL,
 "Motor Symptoms by HRT Status in Postmenopausal Women",
 "motor_symptoms"
)

nonmotor_hrt_plot <- create_hrt_diverging_barplot(
 nonmotorsymptoms_table,
 NULL,
 "Non-Motor Symptoms by HRT Status in Postmenopausal Women",
 "nonmotor_symptoms"
)

med_hrt_plot <- create_hrt_diverging_barplot(
 med_category_table,
 NULL,
 "Medication Categories by HRT Status in Postmenopausal Women",
 "medication_categories"
)

othermotorsymptoms_hrt_plot <- create_hrt_diverging_barplot(
 othermotorsymptoms_table,
 NULL,
 "Other Motor Symptoms by HRT Status in Postmenopausal Women",
 "other_motor_symptoms"
)

if (!is.null(motor_hrt_plot)) {
 print(motor_hrt_plot$plot)
 ggsave("motor_symptoms_by_hrt_postmenopausal.png", motor_hrt_plot$plot, width = 10, height = 7,
dpi = 300)
}

if (!is.null(nonmotor_hrt_plot)) {
 print(nonmotor_hrt_plot$plot)
 ggsave("nonmotor_symptoms_by_hrt_postmenopausal.png", nonmotor_hrt_plot$plot, width = 10,
height = 10, dpi = 300)
```

```
}

if (!is.null(med_hrt_plot)) {
  print(med_hrt_plot$plot)
  ggsave("med_categories_by_hrt_postmenopausal.png", med_hrt_plot$plot, width = 10, height = 7, dpi =
300)
}
if (!is.null(othermotorsymptoms_hrt_plot)) {
  print(othermotorsymptoms_hrt_plot$plot)
  ggsave("othermotorsymptoms_by_hrt_postmenopausal.png", othermotorsymptoms_hrt_plot$plot,
width = 10, height = 7, dpi = 300)
}

postmenopausal_hrt_analysis_plots <- list(
  motor_symptoms = if (!is.null(motor_hrt_plot)) motor_hrt_plot$data else NULL,
  nonmotor_symptoms = if (!is.null(nonmotor_hrt_plot)) nonmotor_hrt_plot$data else NULL,
  med_categories = if (!is.null(med_hrt_plot)) med_hrt_plot$data else NULL,
  othermotorsymptoms = if (!is.null(othermotorsymptoms_hrt_plot)) othermotorsymptoms_hrt_plot$data
else NULL
)
save(postmenopausal_hrt_analysis_plots, file = "hrt_postmenopausal_plots_data.RData")
```

Script 03.2.2.

```
rm(list = ls(pattern =
"^(menstruating_|contraceptive_|demographics_data|motor_contraceptive|nonmotor_contraceptive|med_contraceptive)"))

rm(list = ls(pattern = "^(results|cont_table|percent_table|analysis_plots|significant_)$"))

rm(list = ls(pattern = "^(analyze_|create_contraceptive_)"))

required_vars <- c("new_menstruationstatus_table", "hormonal_table", "motorsymptoms_table",
                   "nonmotorsymptoms_table", "med_category_table", "data_UID", "master_symptom_orders")

missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 loaded <- FALSE

 if (file.exists("master_data_preparation_environment.RData")) {
  load("master_data_preparation_environment.RData")
  loaded <- TRUE
 } else if (file.exists("03_menstruation_status_analysis.RData")) {
  load("03_menstruation_status_analysis.RData")
  loaded <- TRUE
 } else if (file.exists("survey_analysis_environment.RData")) {
  load("survey_analysis_environment.RData")
  loaded <- TRUE
 }

 if (!loaded) {
  stop("Required variables missing and no environment file found. Please run Script 03.0 first.\n",
       "Missing variables: ", paste(missing_vars, collapse = ", "))
 }

 still_missing <- c()
 for(var in required_vars) {
  if(!exists(var)) {
   still_missing <- c(still_missing, var)
  }
 }

 if(length(still_missing) > 0) {
  stop("Even after loading environment, some variables are still missing: ",
       paste(still_missing, collapse = ", "))
 }
```

```
}

if(!"Description" %in% colnames(new_menstruationstatus_table)) {
  stop("Description column not found in new_menstruationstatus_table. Please check Script 03.0
completed correctly.")
}

if(!all(c("Yes", "No") %in% colnames(hormonal_table))) {
  stop("Yes/No columns not found in hormonal_table. Please check Script 03.0 completed correctly.")
}

if(!exists("master_symptom_orders")) {
  stop("Master symptom ordering system not found. Please ensure Script 03.0 Section 15 completed
correctly.")
}

if(!exists("apply_standard_symptom_order")) {
  stop("Helper functions for symptom ordering not found. Please ensure Script 03.0 Section 15 completed
correctly.")
}

menstruating_count <- sum(new_menstruationstatus_table$Description == "Menstruating", na.rm =
TRUE)
if(menstruating_count == 0) {
  stop("No menstruating participants found in new_menstruationstatus_table. Please check Script 03.0
data.")
}

library(dplyr)
library(ggplot2)
library(tidyr)
library(knitr)
library(forcats)

motorsymptoms_table <- motorsymptoms_table[rowSums(motorsymptoms_table[,-(1:2)]) > 0, ]
nonmotorsymptoms_table <- nonmotorsymptoms_table[rowSums(nonmotorsymptoms_table[,-(1:2)]) >
0, ]
med_category_table <- med_category_table[rowSums(med_category_table[,-(1:2)]) > 0, ]

menstruating_ids_clean <- new_menstruationstatus_table$UniqueID[
  new_menstruationstatus_table$Description == "Menstruating" &
    !is.na(new_menstruationstatus_table$UniqueID)
]
menstruating_ids_clean <- unique(menstruating_ids_clean)

menstruating_hormonal <- hormonal_table %>%
  filter(
    UniqueID %in% menstruating_ids_clean,
    ((Yes == 1 & No == 0) | (Yes == 0 & No == 1))
```

```
 ) %>%
 mutate(
  Status = case_when(
   Yes == 1 & No == 0 ~ "Using Contraceptives",
   Yes == 0 & No == 1 ~ "Not Using Contraceptives"
  )
 )

n_users <- sum(menstruating_hormonal$Status == "Using Contraceptives")
n_nonusers <- sum(menstruating_hormonal$Status == "Not Using Contraceptives")

menstruating_hormonal$Contraceptive_Status <- factor(
 menstruating_hormonal$Status,
 levels = c("Using Contraceptives", "Not Using Contraceptives"),
 labels = c("Contraceptive User", "Non-Contraceptive User")
)

analyze_demographics_by_contraceptive <- function() {
 demographics_data <- data_UID %>%
  distinct(UniqueID, .keep_all = TRUE) %>%
  inner_join(
   menstruating_hormonal %>%
    select(UniqueID, Contraceptive_Status),
   by = "UniqueID"
  )

 analyze_categorical <- function(variable_name, variable_label, group_var = "Contraceptive_Status") {
  if (!variable_name %in% names(demographics_data)) {
   return(NULL)
  }

  cont_table <- table(demographics_data[[group_var]], demographics_data[[variable_name]])

  percent_table <- prop.table(cont_table, margin = 1) * 100

  expected_counts <- tryCatch({
   chisq.test(cont_table)$expected
  }, error = function(e) {
   return(matrix(0, nrow = nrow(cont_table), ncol = ncol(cont_table)))
  })

  if (any(expected_counts < 5)) {
   test_result <- tryCatch({
    fisher.test(cont_table)
   }, error = function(e) {
    return(list(p.value = NA))
   })
  } else {
   test_result <- tryCatch({
```

```
      chisq.test(cont_table, correct = TRUE)
    }, error = function(e) {
      tryCatch({
        fisher.test(cont_table)
      }, error = function(e2) {
        return(list(p.value = NA))
      })
    })
  }

  return(list(
    counts = cont_table,
    percentages = percent_table,
    p_value = test_result$p.value
  ))
 }

 age_summary <- analyze_categorical("s01_age", "Age")
 race_summary <- analyze_categorical("s01_race", "Race/Ethnicity")
 employment_summary <- analyze_categorical("s01_employment", "Employment Status")
 marital_status_summary <- analyze_categorical("s01_marital_status", "Marital Status")

 duration_summary <- analyze_categorical("s01_durationofsymptoms", "PD Symptom Duration")

 diagnosis_age_summary <- analyze_categorical("s01_ageofdiagnosis", "Age at PD Diagnosis")

 contraceptive_percent <- round(prop.table(table(demographics_data$Contraceptive_Status)) * 100, 1)

 return(demographics_data)
}

demographics_data <- analyze_demographics_by_contraceptive()

create_contraceptive_diverging_barplot <- function(data_table, symptom_names, title, symptom_type) {
 n_contraceptive_user <- sum(menstruating_hormonal$Contraceptive_Status == "Contraceptive User",
na.rm = TRUE)
 n_non_contraceptive_user <- sum(menstruating_hormonal$Contraceptive_Status == "Non-
Contraceptive User", na.rm = TRUE)

 results <- data.frame(
   SymptomName = character(),
   Contraceptive_User = numeric(),
   NonContraceptive_User = numeric(),
   Difference = numeric(),
   P_Value = numeric(),
   stringsAsFactors = FALSE
 )

 if(!symptom_type %in% names(master_symptom_orders)) {
```

```
 return(NULL)
}

symptoms_to_analyze <- master_symptom_orders[[symptom_type]]

for (symptom in symptoms_to_analyze) {
 if (!symptom %in% colnames(data_table)) {
  next
 }

 temp_data <- data.frame(
  UniqueID = data_table$UniqueID,
  Symptom = data_table[[symptom]]
 ) %>%
  distinct(UniqueID, .keep_all = TRUE)

 temp_data <- temp_data %>%
  filter(UniqueID %in% menstruating_hormonal$UniqueID)

 temp_data <- temp_data %>%
  inner_join(
   menstruating_hormonal %>%
    select(UniqueID, Contraceptive_Status) %>%
    distinct(UniqueID, .keep_all = TRUE),
   by = "UniqueID"
  )

 ct <- table(temp_data$Symptom, temp_data$Contraceptive_Status)

 if (nrow(ct) < 2 || ncol(ct) < 2) {
  next
 }

 pct <- prop.table(ct, margin = 2) * 100

 contraceptive_user_pct <- pct["1", "Contraceptive User"]
 non_contraceptive_user_pct <- pct["1", "Non-Contraceptive User"]

 expected_counts <- tryCatch({
  chisq.test(ct)$expected
 }, error = function(e) {
  return(matrix(0, nrow = nrow(ct), ncol = ncol(ct)))
 })

 if (any(expected_counts < 5)) {
  test_result <- tryCatch({
   fisher.test(ct)
  }, error = function(e) {
   return(list(p.value = NA))
```

```
  })
 } else {
  test_result <- tryCatch({
   chisq.test(ct)
  }, error = function(e) {
   tryCatch({
    fisher.test(ct)
   }, error = function(e2) {
    return(list(p.value = NA))
   })
  })
 }

 results <- rbind(results, data.frame(
  SymptomName = symptom,
  Contraceptive_User = contraceptive_user_pct,
  NonContraceptive_User = non_contraceptive_user_pct,
  Difference = contraceptive_user_pct - non_contraceptive_user_pct,
  P_Value = test_result$p.value,
  stringsAsFactors = FALSE
 ))
}

results <- apply_standard_symptom_order(results, symptom_type)

results <- ensure_complete_symptom_coverage(results, symptom_type)

if (nrow(results) == 0) {
 return(NULL)
}

plot_data <- results %>%
 pivot_longer(
  cols = c(Contraceptive_User, NonContraceptive_User),
  names_to = "Status",
  values_to = "Percentage"
 ) %>%
 mutate(
  PlotValue = ifelse(Status == "Contraceptive_User", -Percentage, Percentage),
  Significance = case_when(
   P_Value < 0.001 ~ "***",
   P_Value < 0.01 ~ "**",
   P_Value < 0.05 ~ "*",
   TRUE ~ ""
  ),
  Label = paste0(round(Percentage, 1), "%", Significance)
 )

symptom_order <- rev(results$SymptomName)
```

```
  plot_data$SymptomName <- factor(plot_data$SymptomName, levels = symptom_order)

  colors <- c("Contraceptive_User" = "#FF8C00", "NonContraceptive_User" = "#808080")

  p <- ggplot(plot_data, aes(x = PlotValue, y = SymptomName, fill = Status)) +
    geom_bar(stat = "identity", width = 0.7) +
    geom_text(
      aes(label = Label, x = ifelse(Status == "Contraceptive_User", -Percentage - 2, Percentage + 2)),
      size = 3, hjust = ifelse(plot_data$Status == "Contraceptive_User", 1, 0)
    ) +
    scale_fill_manual(
      values = colors,
      labels = c(
        paste0("Contraceptive User (n=", n_contraceptive_user, ")"),
        paste0("Non-Contraceptive User (n=", n_non_contraceptive_user, ")")
      )
    ) +
    scale_x_continuous(
      limits = c(-100, 100),
      breaks = seq(-100, 100, 20),
      labels = abs(seq(-100, 100, 20))
    ) +
    labs(
      title = title,
      subtitle = "Percentage of menstruating participants reporting each symptom (consistent ordering)",
      x = "Percentage (%)",
      y = "",
      fill = "Contraceptive Status:",
      caption = "* p<0.05, ** p<0.01, *** p<0.001\nSymptoms ordered consistently for cross-analysis
comparison"
    ) +
    theme_minimal() +
    theme(
      panel.grid.major.y = element_blank(),
      panel.grid.minor = element_blank(),
      legend.position = "top",
      plot.title = element_text(size = 14, face = "bold"),
      plot.subtitle = element_text(size = 12),
      axis.text.y = element_text(size = 10),
      axis.text.x = element_text(size = 9)
    ) +
    geom_vline(xintercept = 0, color = "black", linetype = "solid")

  return(list(
    plot = p,
    data = results
  ))
}
```

```
motor_contraceptive_plot <- create_contraceptive_diverging_barplot(
 motorsymptoms_table,
 NULL,
 "Motor Symptoms by Contraceptive Status in Menstruating Women",
 "motor_symptoms"
)

nonmotor_contraceptive_plot <- create_contraceptive_diverging_barplot(
 nonmotorsymptoms_table,
 NULL,
 "Non-Motor Symptoms by Contraceptive Status in Menstruating Women",
 "nonmotor_symptoms"
)

med_contraceptive_plot <- create_contraceptive_diverging_barplot(
 med_category_table,
 NULL,
 "Medication Categories by Contraceptive Status in Menstruating Women",
 "medication_categories"
)

othermotorsymptoms_contraceptive_plot <- create_contraceptive_diverging_barplot(
 othermotorsymptoms_table,
 NULL,
 "Other Motor Symptoms by Contraceptive Status in Menstruating Women",
 "other_motor_symptoms"
)

if (!is.null(motor_contraceptive_plot)) {
 print(motor_contraceptive_plot$plot)
 ggsave("motor_symptoms_by_contraceptive_menstruating.png", motor_contraceptive_plot$plot, width
= 10, height = 7, dpi = 300)
}

if (!is.null(nonmotor_contraceptive_plot)) {
 print(nonmotor_contraceptive_plot$plot)
 ggsave("nonmotor_symptoms_by_contraceptive_menstruating.png",
nonmotor_contraceptive_plot$plot, width = 10, height = 10, dpi = 300)
}

if (!is.null(med_contraceptive_plot)) {
 print(med_contraceptive_plot$plot)
 ggsave("med_categories_by_contraceptive_menstruating.png", med_contraceptive_plot$plot, width =
10, height = 7, dpi = 300)
}

if (!is.null(othermotorsymptoms_contraceptive_plot)) {
 print(othermotorsymptoms_contraceptive_plot$plot)
```

```
 ggsave("othermotorsymptoms_by_contraceptive_menstruating.png",
othermotorsymptoms_contraceptive_plot$plot, width = 10, height = 7, dpi = 300)
}

menstruating_contraceptive_analysis_plots <- list(
 motor_symptoms = if (!is.null(motor_contraceptive_plot)) motor_contraceptive_plot$data else NULL,
 nonmotor_symptoms = if (!is.null(nonmotor_contraceptive_plot)) nonmotor_contraceptive_plot$data
else NULL,
 med_categories = if (!is.null(med_contraceptive_plot)) med_contraceptive_plot$data else NULL,
 othermotorsymptoms = if (!is.null(othermotorsymptoms_contraceptive_plot))
othermotorsymptoms_contraceptive_plot$data else NULL
)
save(menstruating_contraceptive_analysis_plots, file = "contraceptive_menstruating_plots_data.RData")
```

App Scripts

Script 1

```
required_files <- c("Period_and_Symptoms_200525.xlsx", "Medication_Adherence_200525.xlsx")

missing_files <- c()
for(file in required_files) {
 if(!file.exists(file)) {
  missing_files <- c(missing_files, file)
 }
}

if(length(missing_files) > 0) {
 stop("Required Excel files missing: ", paste(missing_files, collapse = ", "), "\n",
    "Please ensure these files are in the working directory.")
}

required_packages <- c("dplyr", "ggplot2", "tidyr", "readxl")
missing_packages <- c()

for(pkg in required_packages) {
 if(!require(pkg, character.only = TRUE, quietly = TRUE)) {
  missing_packages <- c(missing_packages, pkg)
 }
}

if(length(missing_packages) > 0) {
 stop("Required packages missing: ", paste(missing_packages, collapse = ", "))
}

survey_vars_to_check <- c("data_UID", "new_menstruationstatus_table")

missing_survey_vars <- c()
for(var in survey_vars_to_check) {
 if(!exists(var)) {
  missing_survey_vars <- c(missing_survey_vars, var)
 }
}

if(length(missing_survey_vars) > 0) {
 cat("ℹ️ Note: Some survey variables not found:", paste(missing_survey_vars, collapse = ", "), "\n")
 cat("This is normal if running app scripts independently, but may affect later analyses.\n")
}

cat("✅ Prerequisites verified - ready to process app data\n")
cat("📁 Excel files available\n")
cat("📦 Required packages loaded\n")
cat("🚀 Starting fresh app data processing\n\n")
```

```
library(dplyr)
library(ggplot2)
library(readxl)
library(tidyr)

symptoms_data <- read_excel("Period_and_Symptoms_200525.xlsx", col_names = TRUE, skip = 1)
medication_data <- read_excel("Medication_Adherence_200525.xlsx", col_names = TRUE, skip = 1)

symptoms_data <- symptoms_data %>%
 mutate(research_id = gsub("\\s+", "", toupper(research_id)))

medication_data <- medication_data %>%
 mutate(research_id = gsub("\\s+", "", toupper(research_id)))

symptoms_data <- symptoms_data %>%
 filter(research_id != "UCC089WV4")

medication_data <- medication_data %>%
 filter(research_id != "UCC089WV4")

print(length(metric_name <- c(
 "Jerky movements (Dyskinesia)", "Muscle cramps (Dystonia)", "Rigidity (stiff muscles)",
 "Loss of Balance", "Slow movement (Bradykinesia)", "Small handwriting (Micrographia)",
 "Tremor", "Facial masking", "Freezing", "Backache", "Bloated", "Blurry Vision",
 "Clumsiness", "Constipated", "Diarrhoea", "Difficulty in sleeping (Insomnia)",
 "Difficulty typing", "Dry eyes", "Dry skin (Seborrheic dermatitis)",
 "Excessive daytime sleepiness", "Excessive salivation (Sialorrhea)", "Excessive sweating",
 "Fatigue", "Feeling dizzy or faint", "Food cravings", "Headache", "Heart beating quickly or strongly",
 "Hot flushes", "Increased appetite", "Loss of smell (hyposmia)", "Loss of taste (ageusia)",
 "Low blood pressure", "Low voice (Hypophonia)", "Muscle and joint pains", "Nausea",
 "Night Sweats", "Pain in your body", "Pelvic cramps", "Pins and needles in any part of the body",
 "Reduced appetite", "Restless leg syndrome", "Shortness of breath",
 "Sleep disordered breathing (sleep apnoea)", "Sudden urge to urinate", "Swallow issues",
 "Talking, yelling, physically acting out while asleep", "Tinnitus (ringing or buzzing in the ear)",
 "Vaginal dryness", "Weight gain", "Weight loss",
 "Anxiety attack, panic", "Brain fog (Slowed thinking)", "Compulsive and impulsive behaviour",
 "Crying spells", "Difficulty concentrating", "Difficulty finding words",
 "Difficulty multi-tasking", "Feeling anxious, tense or nervous", "Feeling depressed",
 "Feeling irritable", "Lack of interest (Apathy)", "Lack of joy (Anhedonia)",
 "Lack of sweating (hypo hidrosis)", "Less interest in sex (Low Libido)",
 "Lower tolerance for noise or light", "Memory problems", "Mood swings"
)))

print(length(metric_name_abbr <- c(
 "Dyskinesia", "Dystonia", "Rigidity", "Balance loss", "Bradykinesia", "Micrographia",
 "Tremor", "Facial masking", "Freezing", "Backache", "Bloating", "Blurred vision",
 "Clumsiness", "Constipation", "Diarrhoea", "Insomnia", "Difficulty with typing",
 "Dry eyes", "Dry skin", "Daytime sleepiness", "Salivation", "Sweating", "Fatigue",
 "Dizziness", "Cravings", "Headache", "Fast heartbeat", "Hot flushes", "Increased appetite",
```

```
  "Hyposmia", "Loss of taste", "Low BP", "Low voice", "Muscle / joint pains", "Nausea",
  "Night sweats", "Body pain", "Pelvic cramps", "Pins & needles", "Low appetite",
  "Restless legs", "Shortness of breath", "Sleep apnea", "Urinary urges", "Swallowing issues",
  "Movements during sleep", "Tinnitus", "Vaginal dryness", "Weight gain", "Weight loss",
  "Anxiety/Panic", "Brain fog", "Compulsivity", "Crying spells", "Concentration issues",
  "Difficulty finding words", "Difficulty multi-tasking", "Tense/Nervous", "Depression",
  "Irritability", "Apathy", "Anhedonia", "Lack of sweating", "Low libido",
  "Low tolerance to noise / light", "Memory problems", "Mood swings"
)))

is_motor <- c(
  TRUE,
  TRUE,
  TRUE,
  TRUE,
  TRUE,
  TRUE,
  TRUE,
  TRUE,
  TRUE,
  FALSE,
  FALSE,
  FALSE,
  TRUE,
  FALSE,
  FALSE,
  FALSE,
  TRUE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
```

```
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  TRUE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE,
  FALSE
)

symptom_type <- ifelse(is_motor, "Motor", "Non-Motor")

print(length(metric_name))
print(length(symptom_type))
print(length(metric_name_abbr))

symptom_categories <- data.frame(
  metric_name = metric_name,
  symptom_type = symptom_type,
  metric_name_abbr = metric_name_abbr
)

symptoms_data <- symptoms_data %>%
  left_join(symptom_categories, by = "metric_name") %>%
  mutate(
    metric_name_abbr = ifelse(is.na(metric_name_abbr), metric_name, metric_name_abbr)
  )
```

```
print("Updated symptoms_data with abbreviated names:")
print(head(symptoms_data))

additional_entry <- data.frame(
 metric_name = "Period Entry",
 symptom_type = "Period Log",
 metric_name_abbr = "Period Entry"
)

symptom_categories <- rbind(symptom_categories, additional_entry)

print(tail(symptom_categories, n = 1))

period_data <- symptoms_data %>%
 filter(type_name == "Period") %>%
 select(id, research_id, occurred_at_date, occurred_at_time, level) %>%
 distinct() %>%
 mutate(occurred_at_date = as.Date(occurred_at_date, format = "%Y-%m-%d"))

print("Updated period_data:")
print(head(period_data))

symptoms_data <- symptoms_data %>%
 mutate(research_id = gsub("\\s+", "", toupper(research_id)))

symptom_data_clean <- symptoms_data %>%
 filter(type_name != "Period") %>%
 select(
  research_id,
  symptom_type,
  metric_name_abbr,
  level,
  occurred_at_date,
  occurred_at_time
 )

dim(symptom_data_clean)
print("Updated symptom_data_clean:")
print(head(symptom_data_clean))

period_data_cleaned <- period_data %>%
 group_by(research_id, occurred_at_date) %>%
 arrange(desc(level), occurred_at_time) %>%
 slice(1) %>%
 ungroup()

period_data_clustered <- period_data_cleaned %>%
 arrange(research_id, occurred_at_date) %>%
 group_by(research_id) %>%
```

```
 mutate(day_gap = as.numeric(difftime(occurred_at_date, lag(occurred_at_date), units = "days"))) %>%
 ungroup()

print("Data with day gaps calculated:")
print(head(period_data_clustered))

period_data_phases <- period_data_clustered %>%
 group_by(research_id) %>%
 mutate(
  phase_id = cumsum(ifelse(is.na(day_gap) | day_gap >= 6, 1, 0))
 ) %>%
 ungroup()

print("Data with assigned menstruation phases:")
print(head(period_data_phases))

phase_summary <- period_data_phases %>%
 group_by(research_id, phase_id) %>%
 summarise(
  start_date = min(occurred_at_date),
  end_date = max(occurred_at_date),
  duration_days = as.numeric(end_date - start_date + 1),
  num_entries = n(),
  avg_severity = mean(level, na.rm = TRUE),
  max_severity = max(level, na.rm = TRUE)
 ) %>%
 ungroup()

print("Summarised menstruation phases:")
print(head(phase_summary))

assign_correct_phase_ids <- function(data, periods) {
 period_boundaries <- periods %>%
  arrange(occurred_at_date)

 last_period_phase <- period_boundaries %>%
  slice_tail(n = 1) %>%
  pull(phase_id)

 last_period_info <- period_boundaries %>%
  filter(phase_id == last_period_phase) %>%
  slice_tail(n = 1)

 last_period_date <- last_period_info %>% pull(occurred_at_date)

 amc_start_date <- last_period_date + lubridate::days(25)

 data %>%
  arrange(dates) %>%
```

```
  mutate(
    phase_id = sapply(dates, function(date) {
      is_period_date <- date %in% period_boundaries$occurred_at_date

      if(is_period_date) {
        period_phase <- period_boundaries %>%
          filter(occurred_at_date == date) %>%
          pull(phase_id)
        return(period_phase)
      }

      previous_period <- period_boundaries %>%
        filter(occurred_at_date <= date) %>%
        slice_tail(n = 1)

      if(nrow(previous_period) == 0) {
        -1
      } else if(previous_period$phase_id == last_period_phase &&
                date > amc_start_date) {
        last_period_phase + 1
      } else {
        previous_period$phase_id
      }
    })
  )
}

validate_phase_assignment <- function(research_id) {
  cat("\n=== Validation for participant:", research_id, "===\n")

  raw_symptoms <- symptom_data_clean %>%
    filter(research_id == !!research_id) %>%
    mutate(occurred_at_date = as.Date(occurred_at_date)) %>%
    arrange(occurred_at_date)

  raw_periods <- period_data_phases %>%
    filter(research_id == !!research_id) %>%
    mutate(occurred_at_date = as.Date(occurred_at_date)) %>%
    arrange(occurred_at_date)

  test_data <- raw_symptoms %>%
    select(research_id,
           dates = occurred_at_date,
           Symptom = metric_name_abbr,
           Severity = level)

  assigned_data <- assign_correct_phase_ids(test_data, raw_periods)

  cat("\n1. Data Completeness Check:\n")
```

```
cat("Raw symptom entries:", nrow(raw_symptoms), "\n")
cat("Assigned entries:", nrow(assigned_data), "\n")

cat("\n2. Period Entry Dates:\n")
print(raw_periods %>%
      select(occurred_at_date, phase_id))

cat("\n3. Sample of Phase Assignments:\n")
print(assigned_data %>%
      arrange(dates) %>%
      select(dates, phase_id, Symptom, Severity),
    n = Inf)

cat("\n4. Phase Transition Check:\n")
phase_changes <- assigned_data %>%
 arrange(dates) %>%
 select(dates, phase_id) %>%
 group_by(phase_id) %>%
 summarise(
  start_date = min(dates),
  end_date = max(dates)
 ) %>%
 rowwise() %>%
 mutate(
  dates = list(seq(start_date, end_date, by = "day"))
 ) %>%
 unnest(dates) %>%
 select(dates, phase_id)

print(phase_changes, n = Inf)

cat("\n5. Verify Phase Changes Match Period Dates:\n")
period_dates <- raw_periods %>%
 mutate(occurred_at_date = as.Date(occurred_at_date)) %>%
 pull(occurred_at_date)

phase_change_dates <- phase_changes %>%
 mutate(dates = as.Date(dates)) %>%
 pull(dates)

missing_transitions <- setdiff(period_dates, phase_change_dates)
if(length(missing_transitions) > 0) {
 cat("WARNING: Missing phase transitions for dates:",
    format(as.Date(missing_transitions), "%Y-%m-%d"), "\n")
} else {
 cat("All period dates correspond to phase changes ✓\n")
}

return(assigned_data)
```

```
}

test_result_1 <- validate_phase_assignment("UCC657012")
test_result_2 <- validate_phase_assignment("UCC0730NY")

phase_summary <- period_data_phases %>%
 group_by(research_id, phase_id) %>%
 summarise(
  start_date = min(occurred_at_date),
  end_date = max(occurred_at_date),
  duration_days = as.numeric(end_date - start_date + 1),
  num_entries = n(),
  avg_severity = mean(level, na.rm = TRUE),
  max_severity = max(level, na.rm = TRUE)
 ) %>%
 ungroup()

print("Summarised menstruation phases:")
print(head(phase_summary))

single_cycle_participants <- phase_summary %>%
 group_by(research_id) %>%
 summarise(total_phases = n()) %>%
 filter(total_phases == 1) %>%
 select(research_id)

single_cycle_data <- phase_summary %>%
 filter(research_id %in% single_cycle_participants$research_id)

print("Participants with only one cycle:")
print(head(single_cycle_participants))

multi_cycle_participants <- phase_summary %>%
 group_by(research_id) %>%
 summarise(total_phases = n()) %>%
 filter(total_phases > 1) %>%
 select(research_id)

multi_cycle_data <- phase_summary %>%
 filter(research_id %in% multi_cycle_participants$research_id)

print("Participants with more than one cycle:")
print(head(multi_cycle_participants))

multi_cycle_research_ids <- multi_cycle_participants %>%
 pull(research_id)

print("Multi-cycle participant research IDs:")
print(multi_cycle_research_ids)
```

```
total_single_cycle = n_distinct(single_cycle_participants$research_id)
total_multi_cycle = n_distinct(multi_cycle_participants$research_id)
total_participants_in_summary = n_distinct(phase_summary$research_id)

print(paste("Total participants with one cycle:", total_single_cycle))
print(paste("Total participants with more than one cycle:", total_multi_cycle))
print(paste("Total unique participants in phase_summary:", total_participants_in_summary))

if (total_single_cycle + total_multi_cycle == total_participants_in_summary) {
  print("The numbers match: All participants accounted for.")
} else {
  print("Mismatch in counts: Investigate further.")
}

multi_cycle_summary <- phase_summary %>%
  filter(!(research_id %in% single_cycle_participants$research_id))

print("Multi-cycle participants' data prepared for average cycle length calculation:")
print(head(multi_cycle_summary))

multi_cycle_lengths <- multi_cycle_summary %>%
  arrange(research_id, start_date) %>%
  group_by(research_id) %>%
  mutate(
    next_start_date = lead(start_date),
    cycle_length = as.numeric(next_start_date - end_date),
    adjusted_cycle_length = ifelse(cycle_length > 45, 28, cycle_length)
  ) %>%
  ungroup()

print("Cycle lengths calculated for multi-cycle participants:")
print(head(multi_cycle_lengths))

average_cycle_lengths <- multi_cycle_lengths %>%
  group_by(research_id) %>%
  summarise(
    avg_cycle_length = round(mean(adjusted_cycle_length, na.rm = TRUE))
  ) %>%
  ungroup()

print("Average menstrual cycle lengths per participant:")
print(head(average_cycle_lengths))

cycle_graph_data <- multi_cycle_lengths %>%
  select(research_id, phase_id, start_date, end_date, cycle_length, adjusted_cycle_length)

print("Data prepared for individual menstrual cycle graphs:")
print(head(cycle_graph_data))
```

```
single_cycle_length <- single_cycle_data %>%
 mutate(
  duration_days = as.numeric(difftime(end_date, start_date, units="days")) + 1,
  presumed_cycle_length = 28,
  adjusted_cycle_length = presumed_cycle_length
 ) %>%
 select(research_id, phase_id, start_date, end_date, duration_days, presumed_cycle_length,
adjusted_cycle_length)

sc_average_cycle_lengths <- single_cycle_length %>%
 group_by(research_id) %>%
 summarise(
  avg_cycle_length = first(presumed_cycle_length)
 ) %>%
 ungroup()

sc_individual_phase_durations <- single_cycle_length %>%
 select(research_id, duration_days) %>%
 mutate(
  avg_cycle_length = 28,
  avg_menstruation_length = duration_days,
  remaining_days = avg_cycle_length - avg_menstruation_length,
  follicular_days = round(remaining_days * 0.5),
  ovulation_days = round(remaining_days * 0.05),
  luteal_days = avg_cycle_length - (avg_menstruation_length + follicular_days + ovulation_days)
 )

single_cycle_graph_data <- single_cycle_length %>%
 select(research_id, phase_id, start_date, end_date, adjusted_cycle_length)

period_criteria <- period_data_clustered %>%
 group_by(research_id) %>%
 summarise(
  distinct_period_days = n_distinct(occurred_at_date)
 ) %>%
 mutate(
  period_log = ifelse(distinct_period_days >= 3, "MET", NA)
 ) %>%
 select(research_id, period_log)

cat("\nPeriod Criteria:\n")
print(period_criteria)

symptom_criteria <- symptoms_data %>%
 group_by(research_id) %>%
 summarise(
  distinct_symptom_days = n_distinct(occurred_at_date)
 ) %>%
```

```
 mutate(
  symptom_log = ifelse(distinct_symptom_days >= 3, "MET", NA)
 ) %>%
 select(research_id, symptom_log)

cat("\nSymptom Criteria:\n")
print(symptom_criteria)

scheduled_valid <- medication_data %>%
 filter(
  !is.na(scheduled_drug_name) &
   medication_status_name %in% c("Taken On Time", "Taken Early", "Taken Late")
 ) %>%
 mutate(log_date = medication_due_at_date)

adhoc_valid <- medication_data %>%
 filter(
  !is.na(adhoc_drug_name) &
   medication_status_name == "Taken Now"
 ) %>%
 mutate(log_date = medication_due_at_date)

valid_medication_logs <- bind_rows(scheduled_valid, adhoc_valid)

medication_criteria <- valid_medication_logs %>%
 group_by(research_id) %>%
 summarise(
  distinct_medication_days = n_distinct(log_date)
 ) %>%
 mutate(
  medication_log = ifelse(distinct_medication_days >= 3, "MET", NA)
 ) %>%
 select(research_id, medication_log)

cat("\nMedication Criteria:\n")
print(medication_criteria)

combined_criteria <- period_criteria %>%
 full_join(symptom_criteria, by = "research_id") %>%
 full_join(medication_criteria, by = "research_id")

cat("\nCombined Criteria Table:\n")
print(combined_criteria)

period_met <- combined_criteria %>%
 filter(period_log == "MET") %>%
 select(research_id)

total_period_met <- nrow(period_met)
```

```
cat("\nParticipants Meeting Period Data Criterion:\n")
print(period_met)
cat("\nTotal Participants Meeting Period Data Criterion:", total_period_met, "\n")

any_criteria_met <- combined_criteria %>%
 filter(!is.na(period_log) | !is.na(symptom_log) | !is.na(medication_log)) %>%
 select(research_id)

total_any_criteria_met <- nrow(any_criteria_met)

cat("\nParticipants Meeting Any Criterion:\n")
print(any_criteria_met)
cat("\nTotal Participants Meeting Any Criterion:", total_any_criteria_met, "\n")

Unique_Participants <- bind_rows(
 period_data_clustered %>% select(research_id),
 symptoms_data %>% select(research_id),
 medication_data %>% select(research_id)
) %>%
 distinct()

total_unique_participants <- nrow(Unique_Participants)

cat("\nTotal Unique Participants in the Dataset:", total_unique_participants, "\n")

participants_not_met <- Unique_Participants %>%
 anti_join(any_criteria_met, by = "research_id")

total_not_met <- nrow(participants_not_met)

cat("\nParticipants NOT Meeting Any Criterion:\n")
print(participants_not_met)
cat("\nTotal Participants NOT Meeting Any Criterion:", total_not_met, "\n")

phase_colours <- c(
 "Menstruation" = "#FFB3B3",
 "Follicular" = "#FFCCCC",
 "Ovulation" = "#D3D3D3",
 "Luteal" = "#ADD8E6"
)

participant_data <- cycle_graph_data %>%
 filter(research_id == "UCC0778AS") %>%
 mutate(
  phase = case_when(
   phase_id == 1 ~ "Menstruation",
   TRUE ~ as.character(NA)
  )
```

```
 )

participant_data$phase <- factor(
 participant_data$phase,
 levels = c("Menstruation", "Follicular", "Ovulation", "Luteal")
)

ggplot(participant_data, aes(x = start_date, xend = end_date, y = 0, yend = 0)) +
 annotate("rect", xmin = 0, xmax = 5, ymin = 0, ymax = 3, fill = phase_colours["Menstruation"], alpha = 0.5)
+
 annotate("rect", xmin = 5, xmax = 15, ymin = 0, ymax = 3, fill = phase_colours["Follicular"], alpha = 0.5) +
 annotate("rect", xmin = 15, xmax = 17, ymin = 0, ymax = 3, fill = phase_colours["Ovulation"], alpha = 0.5)
+
 annotate("rect", xmin = 17, xmax = 28, ymin = 0, ymax = 3, fill = phase_colours["Luteal"], alpha = 0.5) +
 labs(
  title = "Participant: UCC0778AS",
  x = "Menstrual cycle days",
  y = "Severity Levels"
 ) +
 scale_x_continuous(breaks = c(0, 28)) +
 theme_minimal()

participant_data <- cycle_graph_data %>%
 filter(research_id == "UCC0778AS") %>%
 arrange(start_date)

average_cycle_length <- average_cycle_lengths %>%
 filter(research_id == "UCC0778AS") %>%
 pull(avg_cycle_length)

phase_durations <- participant_data %>%
 summarise(
  average_cycle_length = average_cycle_length,
  menstruation_days = 5,
  remaining_days = average_cycle_length - 5,
  follicular_days = round((remaining_days) * 0.5),
  ovulation_days = round((remaining_days) * 0.05),
  luteal_days = round((remaining_days) * 0.45)
 )

phase_durations <- phase_durations %>%
 mutate(
  total_days = menstruation_days + follicular_days + ovulation_days + luteal_days,
  adjustment = average_cycle_length - total_days,
  luteal_days = luteal_days + adjustment
 ) %>%
 select(-total_days, -adjustment)

print("Phase durations calculated for participant UCC0778AS:")
```

```
print(phase_durations)

phase_boundaries <- phase_durations %>%
 mutate(
  follicular_end = menstruation_days + follicular_days,
  ovulation_end = follicular_end + ovulation_days,
  luteal_end = average_cycle_length
 ) %>%
 select(menstruation_days, follicular_end, ovulation_end, luteal_end)

print("Phase boundaries calculated for participant UCC0778AS:")
print(phase_boundaries)

phase_colours <- c(
 "Menstruation" = "#FFB3B3",
 "Follicular" = "#FFCCCC",
 "Ovulation" = "#D3D3D3",
 "Luteal" = "#ADD8E6"
)

library(ggplot2)
ggplot() +
 annotate("rect", xmin = 0, xmax = phase_boundaries$menstruation_days, ymin = 0, ymax = 3, fill =
phase_colours["Menstruation"], alpha = 0.5) +
 annotate("rect", xmin = phase_boundaries$menstruation_days, xmax =
phase_boundaries$follicular_end, ymin = 0, ymax = 3, fill = phase_colours["Follicular"], alpha = 0.5) +
 annotate("rect", xmin = phase_boundaries$follicular_end, xmax = phase_boundaries$ovulation_end,
ymin = 0, ymax = 3, fill = phase_colours["Ovulation"], alpha = 0.5) +
 annotate("rect", xmin = phase_boundaries$ovulation_end, xmax = phase_boundaries$luteal_end, ymin
= 0, ymax = 3, fill = phase_colours["Luteal"], alpha = 0.5) +
 labs(
  title = "Participant: UCC0778AS",
  x = "Menstrual cycle days",
  y = "Severity Levels (Mild, Moderate, Severe)"
 ) +
 scale_x_continuous(breaks = c(0, round(average_cycle_length, 0))) +
 theme_minimal()

menstruation_phase_data <- multi_cycle_lengths %>%
 select(research_id, duration_days)

average_menstruation_lengths <- menstruation_phase_data %>%
 group_by(research_id) %>%
 summarise(
  avg_menstruation_length_unrounded = mean(duration_days, na.rm = TRUE),
  avg_menstruation_length = round(avg_menstruation_length_unrounded)
 ) %>%
 ungroup()
```

```
cat("Individual Average Menstruation Lengths (Unrounded and Rounded):\n")
print(average_menstruation_lengths)

overall_avg_menstruation_length_unrounded <-
mean(average_menstruation_lengths$avg_menstruation_length_unrounded, na.rm = TRUE)
overall_avg_menstruation_length <- round(overall_avg_menstruation_length_unrounded)

cat("\nOverall Average Menstruation Phase Length (Unrounded):",
overall_avg_menstruation_length_unrounded, "\n")
cat("Overall Average Menstruation Phase Length (Rounded):", overall_avg_menstruation_length, "\n")

individual_phase_durations <- average_cycle_lengths %>%
 left_join(average_menstruation_lengths, by = "research_id") %>%
 rowwise() %>%
 mutate(
  remaining_days_unrounded = avg_cycle_length - avg_menstruation_length_unrounded,
  remaining_days = avg_cycle_length - avg_menstruation_length,
  follicular_days_unrounded = (remaining_days_unrounded * 0.5),
  follicular_days = round(follicular_days_unrounded),
  ovulation_days_unrounded = (remaining_days_unrounded * 0.05),
  ovulation_days = round(ovulation_days_unrounded),
  luteal_days_unadjusted = remaining_days - follicular_days - ovulation_days,
  luteal_days = luteal_days_unadjusted + (avg_cycle_length - (
   avg_menstruation_length + follicular_days + ovulation_days + luteal_days_unadjusted
  ))
 ) %>%
 ungroup() %>%
 select(research_id, avg_cycle_length, avg_menstruation_length_unrounded, avg_menstruation_length,
      follicular_days_unrounded, follicular_days, ovulation_days_unrounded, ovulation_days,
      luteal_days_unadjusted, luteal_days)

cat("\nIndividual Phase Durations (Unrounded and Rounded):\n")
print(individual_phase_durations)

overall_phase_durations_multi_cycle <- individual_phase_durations %>%
 summarise(
  avg_cycle_length = mean(avg_cycle_length, na.rm = TRUE),
  avg_menstruation_length = mean(avg_menstruation_length, na.rm = TRUE),
  avg_follicular_days = mean(follicular_days, na.rm = TRUE),
  avg_ovulation_days = mean(ovulation_days, na.rm = TRUE),
  avg_luteal_days = mean(luteal_days, na.rm = TRUE),
  n_participants = n()
 )

overall_phase_durations_multi_cycle$avg_cycle_length <-
round(overall_phase_durations_multi_cycle$avg_cycle_length)

overall_phase_durations_multi_cycle <- overall_phase_durations_multi_cycle %>%
 mutate(
```

```
  menstruation_days = round(avg_menstruation_length),
  follicular_days = round(avg_follicular_days),
  ovulation_days = round(avg_ovulation_days),
  luteal_days = round(avg_luteal_days)
 )

overall_phase_durations_multi_cycle <- overall_phase_durations_multi_cycle %>%
 mutate(
  sum_of_phases = menstruation_days + follicular_days + ovulation_days + luteal_days
 )

overall_phase_durations_multi_cycle <- overall_phase_durations_multi_cycle %>%
 mutate(
  discrepancy = avg_cycle_length - sum_of_phases,
  largest_phase = case_when(
   menstruation_days >= follicular_days & menstruation_days >= ovulation_days & menstruation_days
>= luteal_days ~ "menstruation",
   follicular_days >= menstruation_days & follicular_days >= ovulation_days & follicular_days >=
luteal_days ~ "follicular",
   ovulation_days >= menstruation_days & ovulation_days >= follicular_days & ovulation_days >=
luteal_days ~ "ovulation",
   TRUE ~ "luteal"
  )
 )

overall_phase_durations_multi_cycle <- overall_phase_durations_multi_cycle %>%
 mutate(
  menstruation_days = if_else(largest_phase == "menstruation", menstruation_days + discrepancy,
menstruation_days),
  follicular_days = if_else(largest_phase == "follicular", follicular_days + discrepancy, follicular_days),
  ovulation_days = if_else(largest_phase == "ovulation", ovulation_days + discrepancy, ovulation_days),
  luteal_days = if_else(largest_phase == "luteal", luteal_days + discrepancy, luteal_days)
 )

print("Overall Average Phase Durations (Original 4-Phase Model, Whole Numbers):")
print(overall_phase_durations_multi_cycle %>%
    select(
     avg_cycle_length,
     menstruation_days,
     follicular_days,
     ovulation_days,
     luteal_days,
     n_participants
    ))

overall_phase_durations_multi_cycle <- overall_phase_durations_multi_cycle %>%
 mutate(
  menstruation_pct = round(menstruation_days / avg_cycle_length * 100, 1),
  follicular_pct = round(follicular_days / avg_cycle_length * 100, 1),
```

```
  ovulation_pct = round(ovulation_days / avg_cycle_length * 100, 1),
  luteal_pct = round(luteal_days / avg_cycle_length * 100, 1)
 )

print("\nPhase Percentages:")
print(overall_phase_durations_multi_cycle %>%
      select(
       avg_cycle_length,
       menstruation_pct,
       follicular_pct,
       ovulation_pct,
       luteal_pct
      ))

sum_of_phases <- with(overall_phase_durations_multi_cycle,
                menstruation_days + follicular_days + ovulation_days + luteal_days)

cat("\nSum of phase durations:", sum_of_phases, "days\n")
cat("Average cycle length:", overall_phase_durations_multi_cycle$avg_cycle_length, "days\n")
cat("Difference:", overall_phase_durations_multi_cycle$avg_cycle_length - sum_of_phases, "days\n")
```

Script 2

```
rm(list = ls(pattern = "^(multi_cycle_merged_data|single_cycle_merged_data|no_period_final)$"))

rm(list = ls(pattern = "^(mini_.*_data|base_data|med_entries|temp_data)$"))

rm(list = ls(pattern = "^(.*_verification.*|.*_summary|final_combined_data)"))

rm(list = ls(pattern = "^(verify_.*|get_participant_data|add_scheduled_meds)"))

required_script1_vars <- c("symptoms_data", "medication_data", "period_data",
                           "multi_cycle_participants", "single_cycle_participants")

missing_script1_vars <- c()
for(var in required_script1_vars) {
 if(!exists(var)) {
  missing_script1_vars <- c(missing_script1_vars, var)
 }
}

if(length(missing_script1_vars) > 0) {
 stop("Required variables from Script 1 missing: ", paste(missing_script1_vars, collapse = ", "), "\n",
    "Please run Script 1 (MMM App Data) first.")
}

cat(" 🧹 Environment cleaned - Script 2 variables cleared, Script 1 data preserved\n")
cat(" 📊 Script 1 variables found:", length(required_script1_vars) - length(missing_script1_vars), "of",
length(required_script1_vars), "\n")
cat(" 📄 Ready to create participant data tables\n\n")

library(dplyr)
library(ggplot2)
library(readxl)
library(tidyr)
library(purrr)
library(lubridate)

symptoms_data <- read_excel("Period_and_Symptoms_200525.xlsx", col_names = TRUE, skip = 1)
medication_data <- read_excel("Medication_Adherence_200525.xlsx", col_names = TRUE, skip = 1)

verify_data_step <- function(data, step_name, research_id = "UCC0778AS") {
 cat("\n=== Verification for", step_name, "===\n")
 cat("Structure:\n")
 str(data)

 cat("\nGeneral Statistics:\n")
 cat("Total rows:", nrow(data), "\n")
 cat("Total unique participants:", n_distinct(data$research_id), "\n")
 cat("Columns:", paste(names(data), collapse=", "), "\n")
```

```
critical_cols <- names(data)[names(data) != "row_num"]
na_counts <- sapply(data[critical_cols], function(x) sum(is.na(x)))
if(any(na_counts > 0)) {
 cat("\nWARNING: NA values found in columns:\n")
 print(na_counts[na_counts > 0])
}

if("research_id" %in% names(data)) {
 participant_data <- data %>% filter(research_id == !!research_id)
 cat("\nFor participant", research_id, ":\n")
 cat("Total rows:", nrow(participant_data), "\n")

 if(nrow(participant_data) > 0) {
  cat("Sample data:\n")
  print(head(participant_data))

  if("dates" %in% names(participant_data)) {
   date_gaps <- participant_data %>%
    arrange(dates) %>%
    mutate(date_diff = as.numeric(dates - lag(dates))) %>%
    filter(date_diff > 1) %>%
    select(dates, date_diff)

   if(nrow(date_gaps) > 0) {
    cat("\nWARNING: Date gaps found:\n")
    print(date_gaps)
   }
  }
 }
}

if(nrow(data) > 0) {
 exact_dupes <- sum(duplicated(data))
 if(exact_dupes > 0) {
  cat("\nWARNING:", exact_dupes, "exact duplicated rows found!\n")
  cat("Example of duplicated data:\n")
  print(head(data[duplicated(data), ]))
 }

 if(all(c("research_id", "dates") %in% names(data))) {
  semantic_dupes <- data %>%
   group_by(research_id, dates) %>%
   filter(n() > 1) %>%
   arrange(research_id, dates)

  if(nrow(semantic_dupes) > 0) {
   cat("\nWARNING: Potential semantic duplicates found (same research_id and date):\n")
   print(head(semantic_dupes))
```

```
  }
 }
}

if(all(c("research_id", "Symptom") %in% names(data))) {
 cat("\nSymptom Statistics for", research_id, ":\n")

 symptom_counts <- data %>%
  filter(research_id == !!research_id) %>%
  count(Symptom, sort = TRUE)

 cat("\nTop 5 most frequent symptoms:\n")
 print(head(symptom_counts, 5))

 if("Symptom Date" %in% names(data)) {
  date_summary <- data %>%
   filter(research_id == !!research_id) %>%
   summarise(
    total_days = n_distinct(`Symptom Date`),
    date_range = paste(min(`Symptom Date`), "to", max(`Symptom Date`)),
    avg_symptoms_per_day = n() / n_distinct(`Symptom Date`),
    max_symptoms_per_day = max(table((`Symptom Date`)))
   )

  cat("\nTracking Period Information:\n")
  cat("Date Range:", date_summary$date_range, "\n")
  cat("Total Days with Records:", date_summary$total_days, "\n")
  cat("Average Symptoms per Day:", round(date_summary$avg_symptoms_per_day, 2), "\n")
  cat("Maximum Symptoms in a Single Day:", date_summary$max_symptoms_per_day, "\n")
 }

 if("Severity" %in% names(data)) {
  severity_summary <- data %>%
   filter(research_id == !!research_id) %>%
   group_by(Symptom) %>%
   summarise(
    avg_severity = mean(Severity, na.rm = TRUE),
    min_severity = min(Severity, na.rm = TRUE),
    max_severity = max(Severity, na.rm = TRUE),
    severity_range = max_severity - min_severity,
    n_severity_changes = sum(diff(Severity) != 0) + 1
   ) %>%
   arrange(desc(avg_severity))

  cat("\nEnhanced Severity Analysis (Top 5 by average severity):\n")
  print(head(severity_summary, 5))
 }
}
```

```
 if(any(c("Adhoc med", "scheduled_meds_info") %in% names(data))) {
  cat("\nMedication Statistics for", research_id, ":\n")

  if("Adhoc med" %in% names(data)) {
   adhoc_summary <- data %>%
    filter(research_id == !!research_id, `Adhoc med` != "No Entry") %>%
    group_by(`Adhoc med`) %>%
    summarise(
     count = n(),
     avg_dosage = mean(Dosage, na.rm = TRUE),
     avg_tablets = mean(`No. of Tablets`, na.rm = TRUE)
    )

   cat("\nAdhoc Medication Summary:\n")
   print(adhoc_summary)
  }

  if("scheduled_meds_info" %in% names(data)) {
   scheduled_summary <- data %>%
    filter(research_id == !!research_id,
         scheduled_meds_info != "No Scheduled Meds",
         scheduled_meds_info != "End of Scheduled Meds") %>%
    select(scheduled_meds_info) %>%
    distinct()

   cat("\nScheduled Medications:\n")
   print(scheduled_summary)
  }
 }

 cat("\n=====================================\n")

 invisible(list(
  total_rows = nrow(data),
  unique_participants = n_distinct(data$research_id),
  na_counts = na_counts,
  duplicates = exact_dupes
 ))
}

verify_data_consistency <- function(previous_data, current_data, step_name,
                        expected_row_change = 0,
                        expected_participant_change = 0) {
 cat("\n=== Consistency Check for", step_name, "===\n")

 prev_rows <- nrow(previous_data)
 curr_rows <- nrow(current_data)
 row_diff <- curr_rows - prev_rows
```

```
  cat("Row count change:", row_diff,
      "(Expected:", expected_row_change, ")\n")

  if(row_diff != expected_row_change) {
    cat("WARNING: Unexpected change in row count!\n")
  }

  prev_participants <- n_distinct(previous_data$research_id)
  curr_participants <- n_distinct(current_data$research_id)
  participant_diff <- curr_participants - prev_participants

  cat("Participant count change:", participant_diff,
      "(Expected:", expected_participant_change, ")\n")

  if(participant_diff != expected_participant_change) {
    cat("WARNING: Unexpected change in participant count!\n")
  }

  common_cols <- intersect(names(previous_data), names(current_data))
  cat("\nCommon columns:", paste(common_cols, collapse=", "), "\n")

  invisible(list(
    row_diff = row_diff,
    participant_diff = participant_diff,
    common_cols = common_cols
  ))
}

symptoms_data <- symptoms_data %>%
  mutate(research_id = gsub("\\s+", "", toupper(research_id)))
verify_data_step(symptoms_data, "Step 1a: Symptoms data standardization")

medication_data <- medication_data %>%
  mutate(research_id = gsub("\\s+", "", toupper(research_id)))
verify_data_step(medication_data, "Step 1b: Medication data standardization")

mini_period_data <- period_data_phases %>%
  mutate(
    research_id = gsub("\\s+", "", toupper(research_id)),
    `Period Entry Date` = as.Date(occurred_at_date)
  ) %>%
  select(research_id, phase_id, `Period Entry Date`, level) %>%
  rename(Flow = level)
verify_data_step(mini_period_data, "Step 2: Mini-period table creation")

verify_symptom_deduplication <- function(data, original_data) {
  cat("\n=== Symptom Deduplication Verification ===\n")

  duplicates <- data %>%
```

```
 group_by(research_id, `Symptom Date`, Symptom) %>%
 filter(n() > 1) %>%
 arrange(research_id, `Symptom Date`, Symptom)

if(nrow(duplicates) > 0) {
 cat("\nWARNING: Found duplicate symptoms on same date:\n")
 print(duplicates)
} else {
 cat("\nCheck 1 Passed: No duplicate symptoms found on same date\n")
}

same_date_symptoms <- data %>%
 group_by(research_id, `Symptom Date`) %>%
 filter(n() > 1) %>%
 arrange(research_id, `Symptom Date`, Symptom)

cat("\nExample of preserved different symptoms on same date:\n")
print(head(same_date_symptoms))

severity_check <- original_data %>%
 mutate(occurred_at_date = as.Date(occurred_at_date)) %>%
 group_by(research_id, occurred_at_date, metric_name_abbr) %>%
 summarise(
  max_original_severity = max(level, na.rm = TRUE),
  .groups = 'drop'
 ) %>%
 left_join(
  data %>%
   rename(occurred_at_date = `Symptom Date`,
       metric_name_abbr = Symptom) %>%
   select(research_id, occurred_at_date, metric_name_abbr, Severity),
  by = c("research_id", "occurred_at_date", "metric_name_abbr")
 ) %>%
 filter(max_original_severity != as.numeric(Severity)) %>%
 arrange(research_id, occurred_at_date)

if(nrow(severity_check) > 0) {
 cat("\nWARNING: Found cases where highest severity was not kept:\n")
 print(severity_check)
} else {
 cat("\nCheck 3 Passed: Highest severity was correctly preserved\n")
}

dates_comparison <- bind_rows(
 original_data %>%
  mutate(occurred_at_date = as.Date(occurred_at_date)) %>%
  group_by(research_id) %>%
  summarise(
   original_dates = n_distinct(occurred_at_date),
```

```
      .groups = 'drop'
    ),
   data %>%
    group_by(research_id) %>%
    summarise(
     processed_dates = n_distinct(`Symptom Date`),
     .groups = 'drop'
    )
  ) %>%
   group_by(research_id) %>%
   summarise(
    original_dates = max(original_dates, na.rm = TRUE),
    processed_dates = max(processed_dates, na.rm = TRUE),
    date_difference = original_dates - processed_dates
   ) %>%
   filter(date_difference != 0)

  if(nrow(dates_comparison) > 0) {
   cat("\nWARNING: Found differences in unique dates before and after processing:\n")
   print(dates_comparison)
  } else {
   cat("\nCheck 4 Passed: All unique dates were preserved\n")
  }

  cat("\nSummary Statistics:\n")
  summary_stats <- data %>%
   summarise(
    total_participants = n_distinct(research_id),
    total_symptoms = n(),
    total_dates = n_distinct(`Symptom Date`),
    avg_symptoms_per_date = round(total_symptoms / total_dates, 2)
   )
  print(summary_stats)

  invisible(list(
   has_duplicates = nrow(duplicates) > 0,
   severity_issues = nrow(severity_check) > 0,
   date_differences = nrow(dates_comparison) > 0
  ))
}

mini_symptom_data <- symptom_data_clean %>%
  mutate(
   research_id = gsub("\\s+", "", toupper(research_id)),
   `Symptom Date` = as.Date(occurred_at_date)
  ) %>%
  distinct() %>%
  group_by(research_id, `Symptom Date`, symptom_type, metric_name_abbr) %>%
  slice_max(level, n = 1, with_ties = FALSE) %>%
```

```
 ungroup() %>%
 select(research_id, symptom_type, metric_name_abbr, `Symptom Date`, level) %>%
 rename(
  `Symptom Type` = symptom_type,
  Symptom = metric_name_abbr,
  Severity = level
 ) %>%
 distinct(research_id, `Symptom Date`, Symptom, .keep_all = TRUE)

verification_results <- verify_symptom_deduplication(mini_symptom_data, symptom_data_clean)

verify_data_step(mini_symptom_data, "Step 3: Mini-symptom table creation")

mini_adhoc_medication_data <- medication_data %>%
 mutate(
  research_id = gsub("\\s+", "", toupper(research_id)),
  `Adhoc Date` = as.Date(medication_due_at_date),
  Dosage = ifelse(is.na(adhoc_drug_name), NA, medication_dosage),
  `No. of Tablets` = ifelse(is.na(adhoc_drug_name), NA, medication_no_of_tablets)
 ) %>%
 select(research_id, adhoc_drug_name, Dosage, `No. of Tablets`, `Adhoc Date`) %>%
 distinct() %>%
 rename(`Adhoc med` = adhoc_drug_name)

verify_data_step(mini_adhoc_medication_data, "Step 4: Mini adhoc medication table")

mini_scheduled_medication_data <- medication_data %>%
 mutate(research_id = gsub("\\s+", "", toupper(research_id))) %>%
 filter(!is.na(scheduled_drug_name)) %>%
 select(research_id, scheduled_drug_name, medication_dosage, medication_no_of_tablets,
medication_due_at_time) %>%
 distinct() %>%
 rename(
  `Scheduled Med` = scheduled_drug_name,
  Dosage = medication_dosage,
  `No. of Tablets` = medication_no_of_tablets,
  Time = medication_due_at_time
 )

verify_data_step(mini_scheduled_medication_data, "Step 5: Mini scheduled medication table")

multi_cycle_master_date_range <- multi_cycle_lengths %>%
 select(research_id, phase_id, start_date, end_date)

multi_cycle_merged_data <- multi_cycle_participants %>%
 pull(research_id) %>%
 map_df(function(id) {
  participant_dates <- bind_rows(
   symptom_data_clean %>%
```

```
    mutate(dates = as.Date(occurred_at_date)) %>%
    select(research_id, dates),
   mini_adhoc_medication_data %>%
    select(research_id, dates = `Adhoc Date`)
) %>%
  filter(research_id == id) %>%
  distinct()

if(nrow(participant_dates) == 0) {
  return(tibble())
}

period_data <- mini_period_data %>%
  filter(research_id == id) %>%
  mutate(`Period Entry Date` = as.Date(`Period Entry Date`)) %>%
  rename(occurred_at_date = `Period Entry Date`) %>%
  select(research_id, occurred_at_date, phase_id, Flow)

if(nrow(period_data) > 0) {
  period_only_dates <- period_data %>%
    select(research_id, dates = occurred_at_date)

  participant_dates <- bind_rows(participant_dates, period_only_dates) %>%
    distinct()
}

if(nrow(participant_dates) == 0) {
  return(tibble())
}

if(nrow(period_data) > 0) {
  first_period_date <- min(period_data$occurred_at_date)
  last_phase_id <- max(period_data$phase_id)
  last_period_date <- period_data %>%
    filter(phase_id == last_phase_id) %>%
    pull(occurred_at_date) %>%
    max()

  dates_with_phases <- participant_dates %>%
    mutate(phase_id = -1)

  for(i in 1:nrow(dates_with_phases)) {
    current_date <- dates_with_phases$dates[i]

    if(current_date < first_period_date) next

    if(current_date %in% period_data$occurred_at_date) {
      period_phase <- period_data %>%
        filter(occurred_at_date == current_date) %>%
```

```
      pull(phase_id) %>%
      first()

    dates_with_phases$phase_id[i] <- period_phase
    next
   }

   prior_periods <- period_data %>%
    filter(occurred_at_date < current_date)

   if(nrow(prior_periods) > 0) {
    most_recent <- prior_periods %>%
     arrange(desc(occurred_at_date)) %>%
     slice(1)

    if(current_date > last_period_date + 28 &&
      most_recent$phase_id == last_phase_id) {
     dates_with_phases$phase_id[i] <- last_phase_id + 10
    } else {
     dates_with_phases$phase_id[i] <- most_recent$phase_id
    }
   }
  }
 } else {
  dates_with_phases <- participant_dates %>%
   mutate(phase_id = -1)
 }

 result <- dates_with_phases %>%
  left_join(
   mini_period_data %>%
    mutate(`Period Entry Date` = as.Date(`Period Entry Date`)),
   by = c("research_id", "dates" = "Period Entry Date")
  )

 if("phase_id.x" %in% names(result) && "phase_id.y" %in% names(result)) {
  result <- result %>%
   mutate(
    phase_id = ifelse(!is.na(phase_id.y), phase_id.y, phase_id.x),
    `Period Entry Date` = if_else(!is.na(Flow), dates, as.Date(NA))
   ) %>%
   select(-phase_id.x, -phase_id.y)
 } else {
  result <- result %>%
   mutate(`Period Entry Date` = if_else(!is.na(Flow), dates, as.Date(NA)))
 }

 result <- result %>%
  mutate(
```

```
    Flow = if_else(is.na(Flow), "No Entry", as.character(Flow)),
    `Period Entry Date` = if_else(is.na(`Period Entry Date`), "No Entry", as.character(`Period Entry
Date`))
   )

  return(result)
 })

multi_cycle_merged_data <- multi_cycle_merged_data %>%
 left_join(
  mini_symptom_data %>%
   group_by(research_id, `Symptom Date`, `Symptom Type`, Symptom) %>%
   slice_max(Severity, n = 1) %>%
   ungroup(),
  by = c(
   "research_id",
   "dates" = "Symptom Date"
  )
 ) %>%
 group_by(research_id, dates) %>%
 mutate(
  is_period = Flow != "No Entry",
  row_num = row_number(),
  Flow = if_else(is_period & row_num == 1, Flow, NA_character_),
  `Period Entry Date` = if_else(is_period & row_num == 1, `Period Entry Date`, NA_character_),
  Flow = if_else(is.na(Flow),
           if_else(is_period, "---", "No Entry"),
           as.character(Flow)),
  `Period Entry Date` = if_else(is.na(`Period Entry Date`),
                    if_else(is_period, "---", "No Entry"),
                    as.character(`Period Entry Date`)),
  `Symptom Type` = if_else(is.na(`Symptom Type`), "No Entry", `Symptom Type`),
  Symptom = if_else(is.na(Symptom), "No Entry", Symptom),
  Severity = if_else(is.na(Severity), "No Entry", as.character(Severity))
 ) %>%
 ungroup() %>%
 select(-is_period, -row_num) %>%
 arrange(research_id, phase_id, dates, desc(Flow != "No Entry"))

base_data <- multi_cycle_merged_data %>%
 mutate(
  `Adhoc med` = "No Entry",
  Dosage = NA_real_,
  `No. of Tablets` = NA_real_
 )

adhoc_meds_clean <- mini_adhoc_medication_data %>%
 filter(
  !is.na(`Adhoc med`),
```

```
  research_id %in% multi_cycle_participants$research_id
 ) %>%
 distinct(research_id, `Adhoc Date`, `Adhoc med`, Dosage, `No. of Tablets`)

date_phases <- base_data %>%
 select(research_id, phase_id, dates) %>%
 distinct()

med_entries <- adhoc_meds_clean %>%
 left_join(
  date_phases,
  by = c("research_id", "Adhoc Date" = "dates")
 ) %>%
 mutate(
  dates = `Adhoc Date`,
  phase_id = ifelse(is.na(phase_id), -1, phase_id),
  Flow = "No Entry",
  `Period Entry Date` = "No Entry",
  `Symptom Type` = "No Entry",
  Symptom = "No Entry",
  Severity = "No Entry"
 ) %>%
 select(
  research_id,
  phase_id,
  dates,
  Flow,
  `Period Entry Date`,
  `Symptom Type`,
  Symptom,
  Severity,
  `Adhoc med`,
  Dosage,
  `No. of Tablets`
 )

multi_cycle_merged_data <- bind_rows(
 base_data,
 med_entries
) %>%
 filter(research_id %in% multi_cycle_participants$research_id) %>%
 distinct() %>%
 group_by(research_id, dates, `Adhoc med`) %>%
 mutate(
  row_num = if_else(`Adhoc med` != "No Entry", row_number(), 0)
 ) %>%
 ungroup() %>%
 filter(row_num <= 1 | row_num == 0) %>%
 select(-row_num) %>%
```

```
 arrange(
  research_id,
  phase_id,
  dates,
  desc(Flow != "No Entry"),
  desc(Symptom != "No Entry"),
  desc(`Adhoc med` != "No Entry")
 )

scheduled_meds_formatted <- mini_scheduled_medication_data %>%
 arrange(research_id, `Scheduled Med`) %>%
 group_by(research_id) %>%
 summarise(
  scheduled_meds_info = paste(
   paste(
    `Scheduled Med`,
    Dosage,
    `No. of Tablets`,
    Time,
    sep = "|"
   ),
   collapse = "; "
  ),
  .groups = 'drop'
 )

add_scheduled_meds_multi <- function(input_table) {
 input_table %>%
  group_by(research_id) %>%
  mutate(
   row_num = row_number(),
   scheduled_meds_info = case_when(
    row_num == 1 & research_id %in% scheduled_meds_formatted$research_id ~
     scheduled_meds_formatted$scheduled_meds_info[match(research_id,
scheduled_meds_formatted$research_id)],
    row_num == 1 ~ "No Scheduled Meds",
    TRUE ~ "End of Scheduled Meds"
   )
  ) %>%
  select(-row_num) %>%
  ungroup()
}

multi_cycle_merged_data <- add_scheduled_meds_multi(multi_cycle_merged_data) %>%
 group_by(research_id) %>%
 mutate(
  is_first_row = row_number() == 1,
  has_data = Flow != "No Entry" |
   Symptom != "No Entry" |
```

```
    `Adhoc med` != "No Entry" |
    (is_first_row & scheduled_meds_info != "No Scheduled Meds")
  ) %>%
  ungroup() %>%
  filter(has_data) %>%
  select(-is_first_row) %>%
  select(
    research_id,
    phase_id,
    dates,
    Flow,
    `Period Entry Date`,
    `Symptom Type`,
    Symptom,
    Severity,
    `Adhoc med`,
    Dosage,
    `No. of Tablets`,
    scheduled_meds_info
  ) %>%
  arrange(
    research_id,
    phase_id,
    dates,
    desc(Flow != "No Entry"),
    desc(Severity != "No Entry")
  )

valid_single_cycle_participants <- single_cycle_participants %>%
  distinct(research_id) %>%
  pull(research_id)

single_cycle_merged_data <- valid_single_cycle_participants %>%
  map_df(function(id) {
    period_data <- mini_period_data %>%
      filter(research_id == id) %>%
      mutate(`Period Entry Date` = as.Date(`Period Entry Date`)) %>%
      group_by(`Period Entry Date`) %>%
      slice_max(Flow, n = 1) %>%
      ungroup() %>%
      rename(occurred_at_date = `Period Entry Date`) %>%
      select(research_id, occurred_at_date, phase_id, Flow)

    participant_dates <- bind_rows(
      symptom_data_clean %>%
        mutate(dates = as.Date(occurred_at_date)) %>%
        select(research_id, dates),
      mini_adhoc_medication_data %>%
        select(research_id, dates = `Adhoc Date`),
```

```
  period_data %>%
    select(research_id, dates = occurred_at_date)
) %>%
  filter(research_id == id) %>%
  distinct()

if(nrow(period_data) > 0) {
  period_only_dates <- period_data %>%
    select(research_id, dates = occurred_at_date)

  participant_dates <- bind_rows(participant_dates, period_only_dates) %>%
    distinct()
}

if(nrow(participant_dates) == 0) {
  return(tibble())
}

period_data <- mini_period_data %>%
  filter(research_id == id) %>%
  mutate(`Period Entry Date` = as.Date(`Period Entry Date`)) %>%
  rename(occurred_at_date = `Period Entry Date`) %>%
  select(research_id, occurred_at_date, phase_id, Flow)

if(nrow(period_data) > 0) {
  first_period_date <- min(period_data$occurred_at_date)
  last_phase_id <- max(period_data$phase_id)
  last_period_date <- period_data %>%
    filter(phase_id == last_phase_id) %>%
    pull(occurred_at_date) %>%
    max()

  dates_with_phases <- participant_dates %>%
    mutate(phase_id = -1)

  for(i in 1:nrow(dates_with_phases)) {
    current_date <- dates_with_phases$dates[i]

    if(current_date < first_period_date) next

    if(current_date %in% period_data$occurred_at_date) {
      dates_with_phases$phase_id[i] <- last_phase_id
      next
    }

    if(current_date > first_period_date) {
      if(current_date <= last_period_date + 28) {
        dates_with_phases$phase_id[i] <- last_phase_id
      } else {
```

```
        dates_with_phases$phase_id[i] <- last_phase_id + 10
       }
      }
     }
    } else {
     dates_with_phases <- participant_dates %>%
      mutate(phase_id = -1)
    }

    result <- dates_with_phases %>%
     left_join(
      mini_period_data %>%
       mutate(`Period Entry Date` = as.Date(`Period Entry Date`)),
      by = c("research_id", "dates" = "Period Entry Date")
     )

    if("phase_id.x" %in% names(result) && "phase_id.y" %in% names(result)) {
     result <- result %>%
      mutate(
       phase_id = ifelse(!is.na(phase_id.y), phase_id.y, phase_id.x),
       `Period Entry Date` = if_else(!is.na(Flow), dates, as.Date(NA))
      ) %>%
      select(-phase_id.x, -phase_id.y)
    } else {
     result <- result %>%
      mutate(`Period Entry Date` = if_else(!is.na(Flow), dates, as.Date(NA)))
    }

    result <- result %>%
     mutate(
      Flow = if_else(is.na(Flow), "No Entry", as.character(Flow)),
      `Period Entry Date` = if_else(is.na(`Period Entry Date`), "No Entry", as.character(`Period Entry
Date`))
     )

    return(result)
  })

single_cycle_merged_data <- single_cycle_merged_data %>%
  left_join(
   mini_symptom_data %>%
    group_by(research_id, `Symptom Date`, `Symptom Type`, Symptom) %>%
    slice_max(Severity, n = 1) %>%
    ungroup(),
   by = c(
    "research_id",
    "dates" = "Symptom Date"
   )
  ) %>%
```

```
 group_by(research_id, dates) %>%
 mutate(
  is_period = Flow != "No Entry",
  row_num = row_number(),
  Flow = if_else(is_period & row_num == 1, Flow, NA_character_),
  `Period Entry Date` = if_else(is_period & row_num == 1, `Period Entry Date`, NA_character_),
  Flow = if_else(is.na(Flow),
           if_else(is_period, "---", "No Entry"),
           as.character(Flow)),
  `Period Entry Date` = if_else(is.na(`Period Entry Date`),
                    if_else(is_period, "---", "No Entry"),
                    as.character(`Period Entry Date`)),
  `Symptom Type` = if_else(is.na(`Symptom Type`), "No Entry", `Symptom Type`),
  Symptom = if_else(is.na(Symptom), "No Entry", Symptom),
  Severity = if_else(is.na(Severity), "No Entry", as.character(Severity))
 ) %>%
 ungroup() %>%
 select(-is_period, -row_num)

base_data <- single_cycle_merged_data %>%
 mutate(
  `Adhoc med` = "No Entry",
  Dosage = NA_real_,
  `No. of Tablets` = NA_real_
 )

adhoc_meds_clean <- mini_adhoc_medication_data %>%
 filter(
  !is.na(`Adhoc med`),
  research_id %in% valid_single_cycle_participants
 ) %>%
 distinct(research_id, `Adhoc Date`, `Adhoc med`, Dosage, `No. of Tablets`)

date_phases <- base_data %>%
 select(research_id, phase_id, dates) %>%
 distinct()

med_entries <- adhoc_meds_clean %>%
 left_join(
  date_phases,
  by = c("research_id", "Adhoc Date" = "dates")
 ) %>%
 mutate(
  dates = `Adhoc Date`,
  phase_id = ifelse(is.na(phase_id), -1, phase_id),
  Flow = "No Entry",
  `Period Entry Date` = "No Entry",
  `Symptom Type` = "No Entry",
  Symptom = "No Entry",
```

```
  Severity = "No Entry"
 ) %>%
 select(
  research_id,
  phase_id,
  dates,
  Flow,
  `Period Entry Date`,
  `Symptom Type`,
  Symptom,
  Severity,
  `Adhoc med`,
  Dosage,
  `No. of Tablets`
 )

single_cycle_merged_data <- bind_rows(
 base_data,
 med_entries
) %>%
 filter(research_id %in% valid_single_cycle_participants) %>%
 distinct() %>%
 group_by(research_id, dates, `Adhoc med`) %>%
 mutate(
  row_num = if_else(`Adhoc med` != "No Entry", row_number(), 0)
 ) %>%
 ungroup() %>%
 filter(row_num <= 1 | row_num == 0) %>%
 select(-row_num)

single_cycle_merged_data <- add_scheduled_meds_multi(single_cycle_merged_data)

single_cycle_merged_data <- add_scheduled_meds_multi(single_cycle_merged_data) %>%
 group_by(research_id) %>%
 mutate(
  is_first_row = row_number() == 1,
  has_data = Flow != "No Entry" |
   Symptom != "No Entry" |
   `Adhoc med` != "No Entry" |
   (is_first_row & scheduled_meds_info != "No Scheduled Meds")
 ) %>%
 ungroup() %>%
 filter(has_data) %>%
 select(-is_first_row) %>%
 select(
  research_id,
  phase_id,
  dates,
  Flow,
```

```
    `Period Entry Date`,
    `Symptom Type`,
    Symptom,
    Severity,
    `Adhoc med`,
    Dosage,
    `No. of Tablets`,
    scheduled_meds_info
  ) %>%
  arrange(
    research_id,
    phase_id,
    dates,
    desc(Flow != "No Entry"),
    desc(Severity != "No Entry")
  )

no_period_participants <- setdiff(
  union(
    mini_symptom_data %>% distinct(research_id) %>% pull(research_id),
    mini_adhoc_medication_data %>% distinct(research_id) %>% pull(research_id)
  ),
  mini_period_data %>% distinct(research_id) %>% pull(research_id)
)

no_period_base <- tibble(research_id = no_period_participants) %>%
  left_join(
    bind_rows(
      mini_symptom_data %>%
        select(research_id, date = `Symptom Date`),
      mini_adhoc_medication_data %>%
        select(research_id, date = `Adhoc Date`)
    ) %>%
      distinct(),
    by = "research_id"
  ) %>%
  filter(!is.na(date)) %>%
  arrange(research_id, date) %>%
  mutate(dates = date) %>%
  select(research_id, dates)

no_period_with_symptoms <- no_period_base %>%
  left_join(
    mini_symptom_data %>%
      group_by(research_id, `Symptom Date`, `Symptom Type`, Symptom) %>%
      slice_max(Severity, n = 1) %>%
      ungroup(),
    by = c(
      "research_id",
```

```
    "dates" = "Symptom Date"
   )
  ) %>%
  mutate(
   `Symptom Type` = coalesce(`Symptom Type`, "No Entry"),
   Symptom = coalesce(Symptom, "No Entry"),
   Severity = coalesce(as.character(Severity), "No Entry")
  )

base_data <- no_period_with_symptoms %>%
  mutate(
   `Adhoc med` = "No Entry",
   Dosage = NA_real_,
   `No. of Tablets` = NA_real_
  )

adhoc_meds_clean <- mini_adhoc_medication_data %>%
  filter(
   !is.na(`Adhoc med`),
   research_id %in% no_period_participants
  ) %>%
  distinct(research_id, `Adhoc Date`, `Adhoc med`, Dosage, `No. of Tablets`)

med_entries <- adhoc_meds_clean %>%
  mutate(
   dates = `Adhoc Date`,
   `Symptom Type` = "No Entry",
   Symptom = "No Entry",
   Severity = "No Entry"
  ) %>%
  select(
   research_id,
   dates,
   `Symptom Type`,
   Symptom,
   Severity,
   `Adhoc med`,
   Dosage,
   `No. of Tablets`
  )

no_period_complete <- bind_rows(
  base_data,
  med_entries
) %>%
  filter(research_id %in% no_period_participants) %>%
  distinct() %>%
  group_by(research_id, dates, `Adhoc med`) %>%
  mutate(
```

```
   row_num = if_else(`Adhoc med` != "No Entry", row_number(), 0)
  ) %>%
  ungroup() %>%
  filter(row_num <= 1 | row_num == 0) %>%
  select(-row_num)

no_period_complete <- add_scheduled_meds_multi(no_period_complete)

no_period_final <- no_period_complete %>%
  group_by(research_id) %>%
  mutate(
    is_first_row = row_number() == 1,
    has_data = Symptom != "No Entry" |
      `Adhoc med` != "No Entry" |
      (is_first_row & scheduled_meds_info != "No Scheduled Meds")
  ) %>%
  filter(has_data) %>%
  select(-is_first_row) %>%
  ungroup() %>%
  select(
    research_id,
    dates,
    `Symptom Type`,
    Symptom,
    Severity,
    `Adhoc med`,
    Dosage,
    `No. of Tablets`,
    scheduled_meds_info
  ) %>%
  arrange(
    research_id,
    dates,
    desc(Symptom != "No Entry")
  ) %>%
  distinct()

period_tracked_data <- bind_rows(
  multi_cycle_merged_data,
  single_cycle_merged_data
)

final_combined_data_for_display <- bind_rows(
  period_tracked_data,
  no_period_final
)

save(no_period_final, file="no_period_data.RData")
```

```
verify_no_period_data <- function() {
 cat("\n=== NO-PERIOD DATA VERIFICATION ===\n")

 cat("Participants without logged periods:", length(no_period_participants), "\n")
 cat("Participants in no_period_final table:", n_distinct(no_period_final$research_id), "\n")

 period_tracked_ids <- mini_period_data %>%
  distinct(research_id) %>%
  pull(research_id)

 wrong_participants <- intersect(
  no_period_final$research_id,
  period_tracked_ids
 )

 if(length(wrong_participants) > 0) {
  cat("\nWARNING: Found", length(wrong_participants),
      "participants in no_period_final who have logged period entries!\n")
  print(wrong_participants)
 } else {
  cat("\nVerified: No participants who have logged periods are in this table ✓\n")
 }

 cat("\n=== SYMPTOM DATA VERIFICATION ===\n")

 symptom_verification <- lapply(no_period_participants, function(id) {
  raw_symptoms <- symptom_data_clean %>%
   filter(research_id == id) %>%
   mutate(
    occurred_at_date = as.Date(occurred_at_date),
    level = as.character(level)
   ) %>%
   select(research_id, date = occurred_at_date, symptom = metric_name_abbr, severity = level)

  table_symptoms <- no_period_final %>%
   filter(research_id == id, Symptom != "No Entry") %>%
   select(research_id, date = dates, symptom = Symptom, severity = Severity)

  raw_symptom_days <- raw_symptoms %>%
   distinct(date, symptom) %>%
   nrow()

  table_symptom_days <- table_symptoms %>%
   distinct(date, symptom) %>%
   nrow()

  distinct_raw <- raw_symptoms %>% distinct(date, symptom)
  distinct_table <- table_symptoms %>% distinct(date, symptom)
```

```
  missing_symptom_days <- anti_join(
    distinct_raw,
    distinct_table,
    by = c("date", "symptom")
  )

  list(
    research_id = id,
    raw_count = raw_symptom_days,
    table_count = table_symptom_days,
    missing_count = nrow(missing_symptom_days),
    missing_examples = if(nrow(missing_symptom_days) > 0) head(missing_symptom_days, 3) else NULL
  )
 })

 symptom_verification <- symptom_verification[!sapply(symptom_verification, is.null)]

 if(length(symptom_verification) > 0) {
  symptom_summary <- do.call(rbind, lapply(symptom_verification, function(x) {
    data.frame(
      research_id = x$research_id,
      raw_count = x$raw_count,
      table_count = x$table_count,
      missing_count = x$missing_count
    )
  }))

  total_missing_symptoms <- sum(symptom_summary$missing_count)

  cat("Total original symptom-date combinations:", sum(symptom_summary$raw_count), "\n")
  cat("Total symptom-date combinations in table:", sum(symptom_summary$table_count), "\n")
  cat("Missing symptom-date combinations:", total_missing_symptoms, "\n")

  if(total_missing_symptoms > 0) {
    cat("\nWARNING: Missing symptoms detected for these participants:\n")
    print(symptom_summary[symptom_summary$missing_count > 0, ])

    cat("\nExamples of missing symptom entries:\n")
    for(i in which(symptom_summary$missing_count > 0)) {
      id <- symptom_summary$research_id[i]
      cat("\nParticipant", id, "missing symptom examples:\n")
      print(symptom_verification[[which(sapply(symptom_verification, function(x) x$research_id ==
id))]]$missing_examples)
    }
  } else {
    cat("\nAll symptom entries preserved in final table ✓\n")
  }
 } else {
  cat("No symptom data found for no-period participants.\n")
```

```
}

cat("\n=== MEDICATION DATA VERIFICATION ===\n")

med_verification <- lapply(no_period_participants, function(id) {
 raw_meds <- mini_adhoc_medication_data %>%
  filter(research_id == id, `Adhoc med` != "No Entry") %>%
  mutate(`Adhoc Date` = as.Date(`Adhoc Date`)) %>%
  select(research_id, date = `Adhoc Date`, med = `Adhoc med`)

 if(nrow(raw_meds) == 0) return(NULL)

 table_meds <- no_period_final %>%
  filter(research_id == id, `Adhoc med` != "No Entry") %>%
  select(research_id, date = dates, med = `Adhoc med`)

 raw_med_days <- raw_meds %>%
  distinct(date, med) %>%
  nrow()

 table_med_days <- table_meds %>%
  distinct(date, med) %>%
  nrow()

 distinct_raw <- raw_meds %>% distinct(date, med)
 distinct_table <- table_meds %>% distinct(date, med)

 missing_med_days <- anti_join(
  distinct_raw,
  distinct_table,
  by = c("date", "med")
 )

 table_dupes <- table_meds %>%
  group_by(date, med) %>%
  filter(n() > 1) %>%
  ungroup()

 list(
  research_id = id,
  raw_count = raw_med_days,
  table_count = table_med_days,
  missing_count = nrow(missing_med_days),
  duplicate_count = nrow(table_dupes),
  missing_examples = if(nrow(missing_med_days) > 0) head(missing_med_days, 3) else NULL
 )
})

med_verification <- med_verification[!sapply(med_verification, is.null)]
```

```
 if(length(med_verification) > 0) {
  med_summary <- do.call(rbind, lapply(med_verification, function(x) {
   data.frame(
    research_id = x$research_id,
    raw_count = x$raw_count,
    table_count = x$table_count,
    missing_count = x$missing_count,
    duplicate_count = x$duplicate_count
   )
  }))

  total_missing_meds <- sum(med_summary$missing_count)
  total_duplicate_meds <- sum(med_summary$duplicate_count)

  cat("Total original medication-date combinations:", sum(med_summary$raw_count), "\n")
  cat("Total medication-date combinations in table:", sum(med_summary$table_count), "\n")
  cat("Missing medication-date combinations:", total_missing_meds, "\n")
  cat("Duplicate medication entries:", total_duplicate_meds, "\n")

  if(total_missing_meds > 0) {
   cat("\nWARNING: Missing medications detected for these participants:\n")
   print(med_summary[med_summary$missing_count > 0, ])

   cat("\nExamples of missing medication entries:\n")
   for(i in which(med_summary$missing_count > 0)) {
    id <- med_summary$research_id[i]
    cat("\nParticipant", id, "missing medication examples:\n")
    print(med_verification[[which(sapply(med_verification, function(x) x$research_id ==
id))]]$missing_examples)
   }
  } else {
   cat("\nAll medication entries preserved in final table ✓\n")
  }

  if(total_duplicate_meds > 0) {
   cat("\nWARNING: Duplicate medications detected for these participants:\n")
   print(med_summary[med_summary$duplicate_count > 0, ])
  } else {
   cat("\nNo medication duplicates found in final table ✓\n")
  }
 } else {
  cat("No medication data found for no-period participants.\n")
 }

 return(list(
  wrong_participants = wrong_participants,
  symptom_verification = symptom_verification,
  med_verification = med_verification
```

```
 ))
}

no_period_verification_results <- verify_no_period_data()

get_participant_data <- function(research_id, combined_data = final_combined_data_for_display) {
 research_id <- gsub("\\s+", "", toupper(research_id))

 participant_data <- combined_data %>%
  filter(research_id == !!research_id)

 if(nrow(participant_data) == 0) {
  message("No data found for participant ", research_id)
  return(NULL)
 }

 is_period_tracker <- "Flow" %in% colnames(participant_data)
 has_phase_id <- "phase_id" %in% colnames(participant_data)

 participant_info <- list(
  research_id = research_id,
  data_source = case_when(
   is_period_tracker & has_phase_id ~ "Period tracker (multi-cycle)",
   is_period_tracker & !has_phase_id ~ "Period tracker (single-cycle)",
   !is_period_tracker ~ "Non-period tracker",
   TRUE ~ "Unknown"
  ),
  date_range = list(
   start_date = min(participant_data$dates),
   end_date = max(participant_data$dates)
  ),
  total_entries = nrow(participant_data),
  data = participant_data
 )

 if(is_period_tracker) {
  participant_info$period_stats <- participant_data %>%
   filter(Flow != "No Entry") %>%
   summarise(
    total_period_days = n(),
    unique_flow_levels = n_distinct(Flow)
   )
 }

 participant_info$symptom_stats <- participant_data %>%
  filter(Symptom != "No Entry") %>%
  group_by(`Symptom Type`, Symptom) %>%
  summarise(
   count = n(),
```

```
    avg_severity = mean(as.numeric(Severity), na.rm = TRUE),
    max_severity = max(as.numeric(Severity), na.rm = TRUE),
    .groups = 'drop'
   )

  participant_info$medication_stats <- list(
   adhoc_meds = participant_data %>%
    filter(`Adhoc med` != "No Entry") %>%
    group_by(`Adhoc med`) %>%
    summarise(
     count = n(),
     avg_dosage = mean(Dosage, na.rm = TRUE),
     avg_tablets = mean(`No. of Tablets`, na.rm = TRUE),
     .groups = 'drop'
    ),
   scheduled_meds = participant_data %>%
    filter(scheduled_meds_info != "No Scheduled Meds",
         scheduled_meds_info != "End of Scheduled Meds") %>%
    select(scheduled_meds_info) %>%
    distinct()
  )

  return(participant_info)
}

print_participant_summary <- function(participant_info) {
  if(is.null(participant_info)) return(invisible())

  cat("\nParticipant Summary for", participant_info$research_id, "\n")
  cat("----------------------------------------\n")
  cat("Data Source:", participant_info$data_source, "\n")
  cat("Date Range:", format(participant_info$date_range$start_date), "to",
    format(participant_info$date_range$end_date), "\n")
  cat("Total Entries:", participant_info$total_entries, "\n\n")

  if(!is.null(participant_info$period_stats)) {
   cat("Period Statistics:\n")
   print(participant_info$period_stats)
   cat("\n")
  }

  cat("Symptom Statistics:\n")
  print(participant_info$symptom_stats)
  cat("\n")

  cat("Medication Statistics:\n")
  cat("Adhoc Medications:\n")
  print(participant_info$medication_stats$adhoc_meds)
  cat("\nScheduled Medications:\n")
```

```
  print(participant_info$medication_stats$scheduled_meds)
}

view_participant_table <- function(research_id, combined_data = final_combined_data_for_display) {
  research_id <- gsub("\\s+", "", toupper(research_id))

  participant_data <- combined_data %>%
    filter(research_id == !!research_id)

  if(nrow(participant_data) == 0) {
    message("No data found for participant ", research_id)
    return(NULL)
  }

  participant_data <- participant_data %>%
    mutate(
      dates = format(dates, "%Y-%m-%d")
    )

  participant_data <- participant_data %>%
    mutate(
      scheduled_meds_info = ifelse(
        scheduled_meds_info == "End of Scheduled Meds",
        "",
        scheduled_meds_info
      )
    )

  View(participant_data, title = paste("Data for participant", research_id))

  return(invisible(participant_data))
}

view_participant_table("UCC0576AA")

participant_data <- get_participant_data("UCC0730NY")

print_participant_summary(participant_data)

participant_data <- get_participant_data("UCC0730NY")

print_participant_summary(participant_data)

participant_data$data
participant_data$symptom_stats
participant_data$medication_stats
```

Script 14.3

```
cat("=== FLOW CHART LOGGING STATISTICS BY MENSTRUATION STATUS ===\n")

required_vars <- c("mini_period_data", "mini_symptom_data", "mini_adhoc_medication_data",
                   "mini_scheduled_medication_data", "new_menstruationstatus_table")

missing_vars <- c()
for(var in required_vars) {
  if(!exists(var)) {
    missing_vars <- c(missing_vars, var)
    cat("❌ Missing:", var, "\n")
  } else {
    cat("✅ Found:", var, "\n")
  }
}

if(length(missing_vars) > 0) {
  stop("ERROR: Missing required variables: ", paste(missing_vars, collapse = ", "))
}

cat("\n=== STEP 1: BASIC LOGGING LISTS ===\n")

logged_periods <- mini_period_data %>%
  distinct(research_id) %>%
  pull(research_id)

cat("Participants who logged periods:", length(logged_periods), "\n")

logged_symptoms <- mini_symptom_data %>%
  distinct(research_id) %>%
  pull(research_id)

cat("Participants who logged symptoms:", length(logged_symptoms), "\n")

logged_adhoc_meds <- mini_adhoc_medication_data %>%
  filter(!is.na(`Adhoc med`) & `Adhoc med` != "" & `Adhoc med` != "No Entry") %>%
  distinct(research_id) %>%
  pull(research_id)

logged_scheduled_meds <- mini_scheduled_medication_data %>%
  filter(!is.na(`Scheduled Med`) & `Scheduled Med` != "" & `Scheduled Med` != "No Entry") %>%
  distinct(research_id) %>%
  pull(research_id)

logged_meds <- unique(c(logged_adhoc_meds, logged_scheduled_meds))

cat("Participants who logged adhoc medications:", length(logged_adhoc_meds), "\n")
cat("Participants who logged scheduled medications:", length(logged_scheduled_meds), "\n")
cat("Participants who logged any medications:", length(logged_meds), "\n")
```

```
cat("\n=== STEP 2: OVERLAP CATEGORIES ===\n")

all_app_participants <- unique(c(logged_periods, logged_symptoms, logged_meds))

logged_all_3 <- intersect(intersect(logged_periods, logged_symptoms), logged_meds)

logged_symptoms_and_periods <- intersect(logged_symptoms, logged_periods)

logged_both_symptoms_meds <- intersect(logged_symptoms, logged_meds)

logged_both_periods_meds <- intersect(logged_periods, logged_meds)

cat("Total participants with any app data:", length(all_app_participants), "\n")
cat("Logged all 3 (periods + symptoms + meds):", length(logged_all_3), "\n")
cat("Logged symptoms and periods:", length(logged_symptoms_and_periods), "\n")
cat("Logged both symptoms and meds:", length(logged_both_symptoms_meds), "\n")
cat("Logged both periods and meds:", length(logged_both_periods_meds), "\n")

cat("\n=== STEP 3: MATCH WITH MENSTRUATION STATUS ===\n")

all_participants_with_status <- new_menstruationstatus_table %>%
 select(UniqueID, Description) %>%
 rename(research_id = UniqueID, menstruation_status = Description)

cat("Total participants with menstruation status:", nrow(all_participants_with_status), "\n")
cat("Menstruation status breakdown:\n")
print(table(all_participants_with_status$menstruation_status, useNA = "ifany"))

cat("\n=== STEP 4: FLOW CHART STATISTICS BY MENSTRUATION STATUS ===\n")

calculate_logging_by_status <- function(participant_list, category_name) {

 participants_with_status <- all_participants_with_status %>%
  filter(research_id %in% participant_list) %>%
  group_by(menstruation_status) %>%
  summarise(count = n(), .groups = 'drop') %>%
  mutate(category = category_name)

 return(participants_with_status)
}

logging_stats <- list()

logging_stats[["Total with App Data"]] <- calculate_logging_by_status(all_app_participants, "Total with
App Data")
logging_stats[["Logged Periods"]] <- calculate_logging_by_status(logged_periods, "Logged Periods")
logging_stats[["Logged Symptoms"]] <- calculate_logging_by_status(logged_symptoms, "Logged
Symptoms")
```

```
logging_stats[["Logged Medications"]] <- calculate_logging_by_status(logged_meds, "Logged
Medications")
logging_stats[["Logged All 3"]] <- calculate_logging_by_status(logged_all_3, "Logged All 3")
logging_stats[["Logged Symptoms and Periods"]] <-
calculate_logging_by_status(logged_symptoms_and_periods, "Logged Symptoms and Periods")
logging_stats[["Logged Both (Symptoms + Meds)"]] <-
calculate_logging_by_status(logged_both_symptoms_meds, "Logged Both (Symptoms + Meds)")

flow_chart_summary <- do.call(rbind, logging_stats)

flow_chart_wide <- flow_chart_summary %>%
  pivot_wider(names_from = menstruation_status, values_from = count, values_fill = 0)

cat("\n=== STEP 5: FLOW CHART RESULTS ===\n")

print(flow_chart_wide)

cat("\n=== STEP 6: DETAILED BREAKDOWN BY MENSTRUATION STATUS ===\n")

menstruation_statuses <- unique(all_participants_with_status$menstruation_status)
menstruation_statuses <- menstruation_statuses[!is.na(menstruation_statuses)]

for(status in menstruation_statuses) {
  cat("\n", toupper(status), " GROUP:\n")
  cat("=" , rep("=", nchar(status) + 6), "=\n", sep="")

  status_participants <- all_participants_with_status %>%
    filter(menstruation_status == status) %>%
    pull(research_id)

  cat("Total participants:", length(status_participants), "\n")

  cat("- Logged periods:", sum(status_participants %in% logged_periods), "\n")
  cat("- Logged symptoms:", sum(status_participants %in% logged_symptoms), "\n")
  cat("- Logged medications:", sum(status_participants %in% logged_meds), "\n")
  cat("- Logged all 3:", sum(status_participants %in% logged_all_3), "\n")
  cat("- Logged symptoms and periods:", sum(status_participants %in% logged_symptoms_and_periods),
"\n")
  cat("- Logged both (symptoms + meds):", sum(status_participants %in%
logged_both_symptoms_meds), "\n")
}

cat("\n=== STEP 7: SAVING RESULTS ===\n")

flow_chart_logging_summary <- flow_chart_wide
flow_chart_detailed_stats <- flow_chart_summary

flow_chart_participant_lists <- list(
  all_app_participants = all_app_participants,
```

```
  logged_periods = logged_periods,
  logged_symptoms = logged_symptoms,
  logged_meds = logged_meds,
  logged_all_3 = logged_all_3,
  logged_symptoms_and_periods = logged_symptoms_and_periods,
  logged_both_symptoms_meds = logged_both_symptoms_meds
)

cat(" ✅ flow_chart_logging_summary created (wide format table)\n")
cat(" ✅ flow_chart_detailed_stats created (long format table)\n")
cat(" ✅ flow_chart_participant_lists created (participant IDs for each category)\n")

cat("\n=== FLOW CHART LOGGING ANALYSIS COMPLETE ===\n")
cat("Results show simple counts of who logged what, split by menstruation status\n")
cat("No minimum requirements or data quality filters applied - just basic logging behavior\n")
```

Script 2.4

```
library(dplyr)
app_users <- unique(c(
 multi_cycle_participants$research_id,
 single_cycle_participants$research_id,
 unique(no_period_final$research_id)
))

cat("Total unique app users:", length(app_users), "\n")

survey_unique_ids <- data_UID$UniqueID

cat("Total survey participants with UniqueIDs:", length(survey_unique_ids), "\n")

standardized_app_users <- gsub("\\s+", "", toupper(app_users))

known_matches <- list()
known_matches[["UCCC264580"]] <- "UCC264580"

for (unmatched_id in names(known_matches)) {
 idx <- which(standardized_app_users == unmatched_id)
 if (length(idx) > 0) {
  standardized_app_users[idx] <- known_matches[[unmatched_id]]
 }
}

direct_matches <- standardized_app_users %in% survey_unique_ids
matched_users <- standardized_app_users[direct_matches]
unmatched_users <- standardized_app_users[!direct_matches]

cat("Number of app users matched to survey:", length(matched_users), "\n")
cat("Number of app users NOT matched to survey:", length(unmatched_users), "\n")

unmatched_details <- data.frame(
 research_id = app_users[!direct_matches],
 standardized_id = unmatched_users
)

cat("\nFirst 10 unmatched research IDs (if any):\n")
if(length(unmatched_users) > 0) {
 print(head(unmatched_details, 10))
} else {
 cat("No unmatched users found.\n")
}

cat("\n=== KNOWN EXACT MATCHES ===\n")
for (unmatched_id in names(known_matches)) {
 cat(unmatched_id, "is the same participant as", known_matches[[unmatched_id]], "\n")
}
```

```
if(length(unmatched_users) > 0) {
 cat("\nAttempting alternative matching methods...\n")

 extract_numeric <- function(id) {
  num_part <- gsub("UCC", "", id, fixed = TRUE)
  num_part <- gsub("[^0-9]", "", num_part)
  return(num_part)
 }

 create_zero_o_variants <- function(id) {
  variants <- id

  if (grepl("0", id)) {
   zero_positions <- gregexpr("0", id)[[1]]

   for (pos in zero_positions) {
    variant <- id
    substr(variant, pos, pos) <- "O"
    variants <- c(variants, variant)
   }
  }

  if (grepl("O", id)) {
   o_positions <- gregexpr("O", id)[[1]]

   for (pos in o_positions) {
    variant <- id
    substr(variant, pos, pos) <- "0"
    variants <- c(variants, variant)
   }
  }

  return(unique(variants))
 }

 unmatched_details$numeric_part <- sapply(unmatched_details$standardized_id, extract_numeric)

 potential_matches <- data.frame(
  unmatched_id = character(),
  potential_survey_id = character(),
  match_method = character(),
  stringsAsFactors = FALSE
 )

 for(i in 1:nrow(unmatched_details)) {
  current_id <- unmatched_details$standardized_id[i]
  current_numeric <- unmatched_details$numeric_part[i]
```

```
  if(nchar(current_numeric) > 0) {
   for(survey_id in survey_unique_ids) {
    survey_numeric <- extract_numeric(survey_id)
    if(current_numeric == survey_numeric) {
     potential_matches <- rbind(potential_matches, data.frame(
      unmatched_id = current_id,
      potential_survey_id = survey_id,
      match_method = "Numeric portion match",
      stringsAsFactors = FALSE
     ))
    }
   }
  }

  partial_matches <- survey_unique_ids[grepl(substr(current_id, 4, nchar(current_id)),
survey_unique_ids, fixed = TRUE)]
  if(length(partial_matches) > 0) {
   for(match_id in partial_matches) {
    if(!any(potential_matches$unmatched_id == current_id &
         potential_matches$potential_survey_id == match_id)) {
     potential_matches <- rbind(potential_matches, data.frame(
      unmatched_id = current_id,
      potential_survey_id = match_id,
      match_method = "Partial string match",
      stringsAsFactors = FALSE
     ))
    }
   }
  }

  id_variants <- create_zero_o_variants(current_id)
  id_variants <- setdiff(id_variants, current_id)

  for (variant in id_variants) {
   if (variant %in% survey_unique_ids) {
    potential_matches <- rbind(potential_matches, data.frame(
     unmatched_id = current_id,
     potential_survey_id = variant,
     match_method = "0/O confusion match",
     stringsAsFactors = FALSE
    ))
   }
  }
 }

 if(nrow(potential_matches) > 0) {
  cat("\nFound", nrow(potential_matches), "potential matches using alternative methods:\n")
  print(potential_matches)
 } else {
```

```
   cat("\nNo potential matches found using alternative methods.\n")
 }
}

unmatched_result <- list(
 app_users = app_users,
 matched_users = matched_users,
 unmatched_users = unmatched_users,
 unmatched_details = unmatched_details,
 known_matches = known_matches
)

if(exists("potential_matches") && nrow(potential_matches) > 0) {
 unmatched_result$potential_matches <- potential_matches
}

if(exists("potential_matches") && nrow(potential_matches) > 0) {
 zero_o_matches <- potential_matches[potential_matches$match_method == "0/O confusion match", ]
 if(nrow(zero_o_matches) > 0) {
  cat("\n=== ADDITIONAL CONFIRMED MATCHES (0/O confusion) ===\n")
  for(i in 1:nrow(zero_o_matches)) {
   cat(zero_o_matches$unmatched_id[i], "and", zero_o_matches$potential_survey_id[i],
       "are the same participant (0/O confusion)\n")
  }
 }
}

save(unmatched_result, file = "unmatched_app_users.RData")

cat("\nAnalysis complete. Results saved to 'unmatched_app_users.RData'\n")

library(dplyr)

if(!exists("unmatched_result")) {
 load("unmatched_app_users.RData")
}

unmatched_ids <- unmatched_result$unmatched_users

cat("=== FOCUSED ANALYSIS OF UNMATCHED APP USERS ===\n")
cat("Total app users:", length(unmatched_result$app_users), "\n")
cat("Directly matched users:", length(unmatched_result$matched_users),
   sprintf("(%.1f%%)",
length(unmatched_result$matched_users)/length(unmatched_result$app_users)*100), "\n")
cat("Users needing further investigation:", length(unmatched_ids), "\n\n")

cat("Unmatched user IDs:\n")
print(unmatched_ids)
```

```
unmatched_analysis <- data.frame(
 id = unmatched_ids,
 length = nchar(unmatched_ids),
 numeric_portion = gsub("[^0-9]", "", unmatched_ids),
 alpha_portion = gsub("[^A-Za-z]", "", unmatched_ids)
)

unmatched_analysis$numeric_length <- nchar(unmatched_analysis$numeric_portion)
unmatched_analysis$alpha_length <- nchar(unmatched_analysis$alpha_portion)

cat("\nStructure analysis of unmatched IDs:\n")
print(unmatched_analysis)

survey_ids <- data_UID$UniqueID
survey_analysis <- data.frame(
 id = survey_ids,
 length = nchar(survey_ids),
 numeric_portion = gsub("[^0-9]", "", survey_ids),
 alpha_portion = gsub("[^A-Za-z]", "", survey_ids)
)

survey_analysis$numeric_length <- nchar(survey_analysis$numeric_portion)
survey_analysis$alpha_length <- nchar(survey_analysis$alpha_portion)

cat("\nSurvey ID pattern summary:\n")
survey_patterns <- survey_analysis %>%
 group_by(numeric_length, alpha_length) %>%
 summarise(count = n(), .groups = 'drop') %>%
 arrange(desc(count))
print(survey_patterns)

close_matches <- data.frame(
 unmatched_id = character(),
 potential_match = character(),
 match_reason = character(),
 stringsAsFactors = FALSE
)

for(i in 1:nrow(unmatched_analysis)) {
 id <- unmatched_analysis$id[i]
 num_portion <- unmatched_analysis$numeric_portion[i]
 alpha_portion <- unmatched_analysis$alpha_portion[i]

 if(nchar(num_portion) >= 4) {
  numeric_matches <- survey_analysis$id[survey_analysis$numeric_portion == num_portion]

  if(length(numeric_matches) > 0) {
   for(match in numeric_matches) {
    close_matches <- rbind(close_matches, data.frame(
```

```
        unmatched_id = id,
        potential_match = match,
        match_reason = "Exact numeric match",
        stringsAsFactors = FALSE
      ))
    }
   }
  }

  if(nchar(num_portion) >= 4) {
   for(j in 1:nrow(survey_analysis)) {
    survey_num <- survey_analysis$numeric_portion[j]

    if(nchar(survey_num) == nchar(num_portion)) {
     diff_count <- sum(strsplit(num_portion, "")[[1]] != strsplit(survey_num, "")[[1]])

     if(diff_count <= 2) {
      close_matches <- rbind(close_matches, data.frame(
       unmatched_id = id,
       potential_match = survey_analysis$id[j],
       match_reason = paste("Similar numeric portion", diff_count, "different digits"),
       stringsAsFactors = FALSE
      ))
     }
    }
   }
  }

  if(nchar(alpha_portion) >= 2) {
   alpha_matches <- survey_analysis$id[toupper(survey_analysis$alpha_portion) ==
toupper(alpha_portion) &
                       survey_analysis$numeric_length >= 4]

   if(length(alpha_matches) > 0 && length(alpha_matches) < 20) {
    for(match in alpha_matches) {
     close_matches <- rbind(close_matches, data.frame(
      unmatched_id = id,
      potential_match = match,
      match_reason = "Alpha portion match",
      stringsAsFactors = FALSE
     ))
    }
   }
  }
}

close_matches <- close_matches %>% distinct()

if(nrow(close_matches) > 0) {
```

```
 cat("\n=== FILTERED POTENTIAL MATCHES ===\n")
 cat("Found", nrow(close_matches), "potential matches for",
length(unique(close_matches$unmatched_id)), "unmatched IDs\n\n")

 for(id in unique(close_matches$unmatched_id)) {
  matches <- close_matches %>% filter(unmatched_id == id)
  cat("Potential matches for", id, ":\n")
  print(matches)
  cat("\n")
 }

 no_matches <- setdiff(unmatched_ids, close_matches$unmatched_id)
 if(length(no_matches) > 0) {
  cat("\nRemaining", length(no_matches), "unmatched IDs with no potential matches:\n")
  print(no_matches)
 }
} else {
 cat("\nNo filtered potential matches found.\n")
}

cat("\n=== POTENTIAL DATA ENTRY ISSUES ===\n")

unusual_structure <- unmatched_analysis %>%
 filter(!(paste(numeric_length, alpha_length) %in% paste(survey_patterns$numeric_length,
survey_patterns$alpha_length)))

if(nrow(unusual_structure) > 0) {
 cat("Found", nrow(unusual_structure), "unmatched IDs with unusual structure:\n")
 print(unusual_structure)

 cat("\nThese IDs may have incorrect format or data entry errors.\n")
} else {
 cat("All unmatched IDs have structure patterns that exist in the survey data.\n")
}

cat("\n=== FINAL SUMMARY ===\n")
cat("1. Successfully matched", length(unmatched_result$matched_users), "app users with survey
data.\n")
cat("2. Found potential matches for", length(unique(close_matches$unmatched_id)), "additional
users.\n")
cat("3. Unable to find matches for", length(no_matches), "users.\n\n")

cat("Possible explanations for unmatched IDs:\n")
cat("- Data entry errors when entering unique codes in the app\n")
cat("- Transposition or substitution of digits\n")
cat("- Inconsistent formatting between systems\n")
cat("- Use of different identifiers in the two systems\n\n")

cat("Recommended next steps:\n")
```

```
cat("1. Manually review the promising potential matches identified\n")
cat("2. For remaining unmatched IDs, check original data collection records\n")
cat("3. Consider implementing a more standardized ID format for future studies\n")

library(dplyr)
library(writexl)

create_simplified_verification <- function() {
 unmatched_ids <- unmatched_result$unmatched_users

 verification_df <- data.frame(
  unmatched_id = character(),
  potential_match = character(),
  match_reason = character(),
  confidence = character(),
  verification_status = character(),
  notes = character(),
  stringsAsFactors = FALSE
 )

 for(id in unmatched_ids) {
  if(exists("close_matches")) {
   id_matches <- close_matches[close_matches$unmatched_id == id, ]

   if(nrow(id_matches) > 0) {
    exact_matches <- id_matches[id_matches$match_reason == "Exact numeric match", ]
    if(nrow(exact_matches) > 0) {
     for(i in 1:nrow(exact_matches)) {
      verification_df <- rbind(verification_df, data.frame(
       unmatched_id = id,
       potential_match = exact_matches$potential_match[i],
       match_reason = exact_matches$match_reason[i],
       confidence = "High",
       verification_status = "Pending",
       notes = "",
       stringsAsFactors = FALSE
      ))
     }
    }

    similar_matches <- id_matches[grepl("0 different digits|1 different digits",
id_matches$match_reason), ]
    if(nrow(similar_matches) > 0) {
     if(nrow(similar_matches) > 5) similar_matches <- similar_matches[1:5, ]

     for(i in 1:nrow(similar_matches)) {
      if(similar_matches$potential_match[i] %in%
       verification_df$potential_match[verification_df$unmatched_id == id]) {
       next
```

```
    }

    verification_df <- rbind(verification_df, data.frame(
      unmatched_id = id,
      potential_match = similar_matches$potential_match[i],
      match_reason = similar_matches$match_reason[i],
      confidence = "Medium",
      verification_status = "Pending",
      notes = "",
      stringsAsFactors = FALSE
    ))
  }
}

alpha_matches <- id_matches[id_matches$match_reason == "Alpha portion match", ]
if(nrow(alpha_matches) > 0) {
  if(nrow(alpha_matches) > 3) alpha_matches <- alpha_matches[1:3, ]

  for(i in 1:nrow(alpha_matches)) {
    if(alpha_matches$potential_match[i] %in%
       verification_df$potential_match[verification_df$unmatched_id == id]) {
      next
    }

    verification_df <- rbind(verification_df, data.frame(
      unmatched_id = id,
      potential_match = alpha_matches$potential_match[i],
      match_reason = alpha_matches$match_reason[i],
      confidence = "Low",
      verification_status = "Pending",
      notes = "",
      stringsAsFactors = FALSE
    ))
  }
}

if(nrow(verification_df[verification_df$unmatched_id == id, ]) == 0) {
  sorted_matches <- id_matches[order(grepl("0 different", id_matches$match_reason),
                                     grepl("1 different", id_matches$match_reason),
                                     grepl("2 different", id_matches$match_reason),
                                     decreasing = TRUE), ]

  top_matches <- sorted_matches[1:min(3, nrow(sorted_matches)), ]

  for(i in 1:nrow(top_matches)) {
    verification_df <- rbind(verification_df, data.frame(
      unmatched_id = id,
      potential_match = top_matches$potential_match[i],
      match_reason = top_matches$match_reason[i],
```

```
          confidence = "Low",
          verification_status = "Pending",
          notes = "",
          stringsAsFactors = FALSE
        ))
       }
      }
     } else {
      verification_df <- rbind(verification_df, data.frame(
       unmatched_id = id,
       potential_match = "NO MATCHES FOUND",
       match_reason = "None",
       confidence = "N/A",
       verification_status = "No matches",
       notes = "No potential matches identified",
       stringsAsFactors = FALSE
      ))
     }
    } else {
     verification_df <- rbind(verification_df, data.frame(
      unmatched_id = id,
      potential_match = "ERROR: close_matches not found",
      match_reason = "Error",
      confidence = "N/A",
      verification_status = "Error",
      notes = "close_matches data frame not available",
      stringsAsFactors = FALSE
     ))
    }
   }

   verification_df <- verification_df %>%
    mutate(confidence_rank = case_when(
     confidence == "High" ~ 1,
     confidence == "Medium" ~ 2,
     confidence == "Low" ~ 3,
     TRUE ~ 4
    )) %>%
    arrange(unmatched_id, confidence_rank) %>%
    select(-confidence_rank)

   return(verification_df)
  }

  create_simple_verification_workbook <- function() {
   cat("Creating simplified verification workbook...\n")

   verification_df <- create_simplified_verification()
```

```
 if(exists("unusual_structure")) {
  unusual_ids <- unusual_structure$id

  for(i in 1:nrow(verification_df)) {
   if(verification_df$unmatched_id[i] %in% unusual_ids) {
    verification_df$notes[i] <- paste0(
     verification_df$notes[i],
     "ID has unusual structure that doesn't match survey ID patterns"
    )
   }
  }
 }

 write_xlsx(list("Verification" = verification_df), "unmatched_user_verification.xlsx")

 cat("Verification workbook created: unmatched_user_verification.xlsx\n")
 cat("- Contains verification table for reviewing matches\n")

 return(verification_df)
}

verification_result <- create_simple_verification_workbook()

library(dplyr)
library(writexl)

multi_cycle_ids <- multi_cycle_participants %>%
 select(research_id)

multi_cycle_ages <- multi_cycle_ids %>%
 left_join(data_UID %>% select(UniqueID, s01_age), by = c("research_id" = "UniqueID"))

age_descriptions <- c(
 "18 - 29 years old", "30 - 39 years old", "40 - 49 years old",
 "50 - 54 years old", "55 - 59 years old", "60 - 64 years old",
 "65 - 69 years old", "70 - 74 years old", "75 - 79 years old",
 "80 - 84 years old", "85 - 90 years old", "90 years old",
 "Prefer not to say"
)

multi_cycle_ages <- multi_cycle_ages %>%
 mutate(Age_Description = case_when(
  !is.na(s01_age) ~ age_descriptions[as.numeric(s01_age)],
  TRUE ~ "Missing data"
 )) %>%
 select(research_id, Age_Description) %>%
 rename(`Research ID` = research_id, Age = Age_Description)

timestamp <- format(Sys.time(), "%Y%m%d_%H%M")
```

```
excel_filename <- paste0("multi_cycle_participants_ages_", timestamp, ".xlsx")
write_xlsx(multi_cycle_ages, excel_filename)

cat("Excel file created:", excel_filename, "\n")
cat("Total multi-cycle participants:", nrow(multi_cycle_ages), "\n")
cat("Participants with age data:", sum(!is.na(multi_cycle_ages$Age) & multi_cycle_ages$Age != "Missing
data"), "\n")
```

Script 14.1

```
library(dplyr)
cat("=== MOTOR SYMPTOM FREQUENCIES AND AVERAGE SEVERITY SCORES ===\n")
cat("Analysis equivalent to initial survey motor symptoms tables\n\n")

cat("MOTOR SYMPTOM ANALYSIS\n")
cat("======================\n")

all_motor_symptoms <- data.frame()

if(exists("multi_cycle_merged_data") && nrow(multi_cycle_merged_data) > 0) {
 multi_motor <- multi_cycle_merged_data %>%
  filter(
   `Symptom Type` == "Motor",
   Symptom != "No Entry",
   !is.na(Symptom),
   !is.na(Severity)
  ) %>%
  select(research_id, Symptom, Severity) %>%
  mutate(
   dataset_type = "Multi-cycle",
   Severity = as.numeric(Severity)
  )

 all_motor_symptoms <- rbind(all_motor_symptoms, multi_motor)
}

if(exists("single_cycle_merged_data") && nrow(single_cycle_merged_data) > 0) {
 single_motor <- single_cycle_merged_data %>%
  filter(
   `Symptom Type` == "Motor",
   Symptom != "No Entry",
   !is.na(Symptom),
   !is.na(Severity)
  ) %>%
  select(research_id, Symptom, Severity) %>%
  mutate(
   dataset_type = "Single-cycle",
   Severity = as.numeric(Severity)
  )

 all_motor_symptoms <- rbind(all_motor_symptoms, single_motor)
}

if(exists("no_period_final") && nrow(no_period_final) > 0) {
 no_period_motor <- no_period_final %>%
  filter(
   `Symptom Type` == "Motor",
   Symptom != "No Entry",
```

```
    !is.na(Symptom),
    !is.na(Severity)
   ) %>%
   select(research_id, Symptom, Severity) %>%
   mutate(
    dataset_type = "No-period",
    Severity = as.numeric(Severity)
   )

 all_motor_symptoms <- rbind(all_motor_symptoms, no_period_motor)
}

if(nrow(all_motor_symptoms) > 0) {

 cat("Total motor symptom entries found:", nrow(all_motor_symptoms), "\n")
 cat("Unique participants reporting motor symptoms:",
length(unique(all_motor_symptoms$research_id)), "\n")
 cat("Unique motor symptoms reported:", length(unique(all_motor_symptoms$Symptom)), "\n\n")

 total_app_participants <- 148

 motor_symptom_summary <- all_motor_symptoms %>%
  group_by(Symptom) %>%
  summarise(
   participants_reporting = n_distinct(research_id),
   total_reports = n(),
   mean_severity = round(mean(Severity, na.rm = TRUE), 2),
   median_severity = round(median(Severity, na.rm = TRUE), 2),
   min_severity = min(Severity, na.rm = TRUE),
   max_severity = max(Severity, na.rm = TRUE),
   sd_severity = round(sd(Severity, na.rm = TRUE), 2),
   sem_severity = round(sd(Severity, na.rm = TRUE) / sqrt(n_distinct(research_id)), 2),
   .groups = 'drop'
  ) %>%
  mutate(
   prevalence_percentage = round((participants_reporting / total_app_participants) * 100, 1),
   mean_sem_formatted = ifelse(is.na(mean_severity) | is.na(sem_severity),
                               "-",
                               paste0(mean_severity, " ± ", sem_severity))
  ) %>%
  arrange(desc(participants_reporting))

 cat("MOTOR SYMPTOM FREQUENCIES AND SEVERITY SCORES:\n")
 cat("(Ordered by number of participants reporting)\n\n")
 print(motor_symptom_summary)

 cat("\n\nSUMMARY STATISTICS:\n")
 cat("==================\n")
 cat("Most commonly reported motor symptom:", motor_symptom_summary$Symptom[1],
```

```
    "(", motor_symptom_summary$participants_reporting[1], "participants,",
    motor_symptom_summary$prevalence_percentage[1], "%)\n")

 cat("Least commonly reported motor symptom:",
motor_symptom_summary$Symptom[nrow(motor_symptom_summary)],
    "(", motor_symptom_summary$participants_reporting[nrow(motor_symptom_summary)],
"participants,",
    motor_symptom_summary$prevalence_percentage[nrow(motor_symptom_summary)], "%)\n")

 cat("Highest average severity:",
    motor_symptom_summary$Symptom[which.max(motor_symptom_summary$mean_severity)],
    "(mean =", max(motor_symptom_summary$mean_severity), ")\n")

 cat("Lowest average severity:",
    motor_symptom_summary$Symptom[which.min(motor_symptom_summary$mean_severity)],
    "(mean =", min(motor_symptom_summary$mean_severity), ")\n")

 cat("\n\nMOTOR SYMPTOM REPORTS BY MENSTRUATION STATUS:\n")
 cat("============================================\n")

 if(exists("new_menstruationstatus_table")) {
  motor_with_status <- all_motor_symptoms %>%
    left_join(
      new_menstruationstatus_table %>%
        select(UniqueID, Description) %>%
        rename(research_id = UniqueID, menstruation_status = Description),
      by = "research_id"
    ) %>%
    filter(!is.na(menstruation_status))

  menstruation_breakdown <- motor_with_status %>%
    group_by(menstruation_status) %>%
    summarise(
      participants = n_distinct(research_id),
      total_reports = n(),
      unique_symptoms = n_distinct(Symptom),
      mean_severity_overall = round(mean(Severity, na.rm = TRUE), 2),
      .groups = 'drop'
    )

  print(menstruation_breakdown)
 } else {
  cat("Menstruation status data not available for comparison.\n")
 }

 cat("\n\nTOP 5 MOST COMMONLY REPORTED MOTOR SYMPTOMS:\n")
 cat("========================================\n")

 top_5_motor <- motor_symptom_summary %>%
```

```
  head(5) %>%
  select(Symptom, participants_reporting, prevalence_percentage, mean_severity)

 print(top_5_motor)

 cat("\n\nMOTOR SYMPTOM BREAKDOWN BY MENSTRUATION STATUS:\n")
 cat("==========================================\n")

 if(exists("motor_with_status")) {
  detailed_menstruation_breakdown <- motor_with_status %>%
   group_by(Symptom, menstruation_status) %>%
   summarise(
    participants = n_distinct(research_id),
    reports = n(),
    mean_severity = round(mean(Severity, na.rm = TRUE), 2),
    .groups = 'drop'
   ) %>%
   arrange(Symptom, menstruation_status)

  top_symptoms <- motor_symptom_summary$Symptom[1:min(5, nrow(motor_symptom_summary))]
  detailed_top_5_menstruation <- detailed_menstruation_breakdown %>%
   filter(Symptom %in% top_symptoms)

  print(detailed_top_5_menstruation)
 } else {
  cat("Cannot create breakdown by menstruation status - data not available.\n")
 }

} else {
 cat("No motor symptom data found in any dataset.\n")
 cat("Please check that the following exist and contain motor symptom data:\n")
 cat("- multi_cycle_merged_data\n")
 cat("- single_cycle_merged_data\n")
 cat("- no_period_final\n")
}

cat("\n\n=== SAVING RESULTS ===\n")

if(exists("motor_symptom_summary")) {
 app_motor_symptom_summary <- motor_symptom_summary
 cat(" ✅ app_motor_symptom_summary created\n")
}

if(exists("dataset_breakdown")) {
 app_motor_dataset_breakdown <- dataset_breakdown
 cat(" ✅ app_motor_dataset_breakdown created\n")
}

if(exists("top_5_motor")) {
```

```
 app_top_5_motor_symptoms <- top_5_motor
 cat("✅ app_top_5_motor_symptoms created\n")
}

if(exists("detailed_breakdown")) {
 app_motor_detailed_breakdown <- detailed_breakdown
 cat("✅ app_motor_detailed_breakdown created\n")
}

if(exists("all_motor_symptoms")) {
 app_all_motor_symptoms_data <- all_motor_symptoms
 cat("✅ app_all_motor_symptoms_data created (raw data)\n")
}

cat("\n=== MOTOR SYMPTOM ANALYSIS COMPLETE ===\n")
cat("All motor symptom summary tables saved with 'app_' prefix\n")

library(dplyr)

cat("=== NON-MOTOR SYMPTOM FREQUENCIES AND AVERAGE SEVERITY SCORES ===\n")
cat("Analysis equivalent to initial survey non-motor symptoms tables\n\n")

cat("NON-MOTOR SYMPTOM ANALYSIS\n")
cat("============================\n")

all_nonmotor_symptoms <- data.frame()

if(exists("multi_cycle_merged_data") && nrow(multi_cycle_merged_data) > 0) {
 multi_nonmotor <- multi_cycle_merged_data %>%
  filter(
   `Symptom Type` == "Non-Motor",
   Symptom != "No Entry",
   !is.na(Symptom),
   !is.na(Severity)
  ) %>%
  select(research_id, Symptom, Severity) %>%
  mutate(
   dataset_type = "Multi-cycle",
   Severity = as.numeric(Severity)
  )

 all_nonmotor_symptoms <- rbind(all_nonmotor_symptoms, multi_nonmotor)
}

if(exists("single_cycle_merged_data") && nrow(single_cycle_merged_data) > 0) {
 single_nonmotor <- single_cycle_merged_data %>%
  filter(
   `Symptom Type` == "Non-Motor",
```

```
    Symptom != "No Entry",
    !is.na(Symptom),
    !is.na(Severity)
   ) %>%
   select(research_id, Symptom, Severity) %>%
   mutate(
    dataset_type = "Single-cycle",
    Severity = as.numeric(Severity)
   )

 all_nonmotor_symptoms <- rbind(all_nonmotor_symptoms, single_nonmotor)
}

if(exists("no_period_final") && nrow(no_period_final) > 0) {
 no_period_nonmotor <- no_period_final %>%
   filter(
    `Symptom Type` == "Non-Motor",
    Symptom != "No Entry",
    !is.na(Symptom),
    !is.na(Severity)
   ) %>%
   select(research_id, Symptom, Severity) %>%
   mutate(
    dataset_type = "No-period",
    Severity = as.numeric(Severity)
   )

 all_nonmotor_symptoms <- rbind(all_nonmotor_symptoms, no_period_nonmotor)
}

if(nrow(all_nonmotor_symptoms) > 0) {

 cat("Total non-motor symptom entries found:", nrow(all_nonmotor_symptoms), "\n")
 cat("Unique participants reporting non-motor symptoms:",
length(unique(all_nonmotor_symptoms$research_id)), "\n")
 cat("Unique non-motor symptoms reported:", length(unique(all_nonmotor_symptoms$Symptom)),
"\n\n")

 total_app_participants <- 148

 nonmotor_symptom_summary <- all_nonmotor_symptoms %>%
   group_by(Symptom) %>%
   summarise(
    participants_reporting = n_distinct(research_id),
    total_reports = n(),
    mean_severity = round(mean(Severity, na.rm = TRUE), 2),
    median_severity = round(median(Severity, na.rm = TRUE), 2),
    min_severity = min(Severity, na.rm = TRUE),
    max_severity = max(Severity, na.rm = TRUE),
```

```
    sd_severity = round(sd(Severity, na.rm = TRUE), 2),
    sem_severity = round(sd(Severity, na.rm = TRUE) / sqrt(n_distinct(research_id)), 2),
    .groups = 'drop'
   ) %>%
   mutate(
    prevalence_percentage = round((participants_reporting / total_app_participants) * 100, 1),
    mean_sem_formatted = ifelse(is.na(mean_severity) | is.na(sem_severity),
                                "-",
                                paste0(mean_severity, " ± ", sem_severity))
   ) %>%
   arrange(desc(participants_reporting))

 cat("NON-MOTOR SYMPTOM FREQUENCIES AND SEVERITY SCORES:\n")
 cat("(Ordered by number of participants reporting)\n\n")
 print(nonmotor_symptom_summary)

 cat("\n\nSUMMARY STATISTICS:\n")
 cat("==================\n")
 cat("Most commonly reported non-motor symptom:", nonmotor_symptom_summary$Symptom[1],
    "(", nonmotor_symptom_summary$participants_reporting[1], "participants,",
    nonmotor_symptom_summary$prevalence_percentage[1], "%)\n")

 cat("Least commonly reported non-motor symptom:",
nonmotor_symptom_summary$Symptom[nrow(nonmotor_symptom_summary)],
    "(", nonmotor_symptom_summary$participants_reporting[nrow(nonmotor_symptom_summary)],
"participants,",
    nonmotor_symptom_summary$prevalence_percentage[nrow(nonmotor_symptom_summary)],
"%)\n")

 cat("Highest average severity:",

nonmotor_symptom_summary$Symptom[which.max(nonmotor_symptom_summary$mean_severity)],
    "(mean =", max(nonmotor_symptom_summary$mean_severity), ")\n")

 cat("Lowest average severity:",

nonmotor_symptom_summary$Symptom[which.min(nonmotor_symptom_summary$mean_severity)],
    "(mean =", min(nonmotor_symptom_summary$mean_severity), ")\n")

 cat("\n\nNON-MOTOR SYMPTOM REPORTS BY MENSTRUATION STATUS:\n")
 cat("================================================\n")

 if(exists("new_menstruationstatus_table")) {
  nonmotor_with_status <- all_nonmotor_symptoms %>%
   left_join(
    new_menstruationstatus_table %>%
     select(UniqueID, Description) %>%
     rename(research_id = UniqueID, menstruation_status = Description),
    by = "research_id"
```

```
    ) %>%
    filter(!is.na(menstruation_status))

   nonmotor_menstruation_breakdown <- nonmotor_with_status %>%
    group_by(menstruation_status) %>%
    summarise(
     participants = n_distinct(research_id),
     total_reports = n(),
     unique_symptoms = n_distinct(Symptom),
     mean_severity_overall = round(mean(Severity, na.rm = TRUE), 2),
     .groups = 'drop'
    )

   print(nonmotor_menstruation_breakdown)
  } else {
   cat("Menstruation status data not available for comparison.\n")
  }

  cat("\n\nTOP 5 MOST COMMONLY REPORTED NON-MOTOR SYMPTOMS:\n")
  cat("================================================\n")

  top_5_nonmotor <- nonmotor_symptom_summary %>%
   head(5) %>%
   select(Symptom, participants_reporting, prevalence_percentage, mean_severity)

  print(top_5_nonmotor)

  cat("\n\nNON-MOTOR SYMPTOM BREAKDOWN BY MENSTRUATION STATUS:\n")
  cat("================================================\n")

  if(exists("nonmotor_with_status")) {
   detailed_nonmotor_menstruation_breakdown <- nonmotor_with_status %>%
    group_by(Symptom, menstruation_status) %>%
    summarise(
     participants = n_distinct(research_id),
     reports = n(),
     mean_severity = round(mean(Severity, na.rm = TRUE), 2),
     .groups = 'drop'
    ) %>%
    arrange(Symptom, menstruation_status)

   top_nonmotor_symptoms <- nonmotor_symptom_summary$Symptom[1:min(5,
nrow(nonmotor_symptom_summary))]
   detailed_top_5_nonmotor_menstruation <- detailed_nonmotor_menstruation_breakdown %>%
    filter(Symptom %in% top_nonmotor_symptoms)

   print(detailed_top_5_nonmotor_menstruation)
  } else {
   cat("Cannot create breakdown by menstruation status - data not available.\n")
```

```
 }

} else {
 cat("No non-motor symptom data found in any dataset.\n")
 cat("Please check that the following exist and contain non-motor symptom data:\n")
 cat("- multi_cycle_merged_data\n")
 cat("- single_cycle_merged_data\n")
 cat("- no_period_final\n")
}

cat("\n\n=== SAVING RESULTS ===\n")

if(exists("nonmotor_symptom_summary")) {
 app_nonmotor_symptom_summary <- nonmotor_symptom_summary
 cat(" ✅ app_nonmotor_symptom_summary created\n")
}

if(exists("nonmotor_menstruation_breakdown")) {
 app_nonmotor_menstruation_breakdown <- nonmotor_menstruation_breakdown
 cat(" ✅ app_nonmotor_menstruation_breakdown created\n")
}

if(exists("top_5_nonmotor")) {
 app_top_5_nonmotor_symptoms <- top_5_nonmotor
 cat(" ✅ app_top_5_nonmotor_symptoms created\n")
}

if(exists("detailed_nonmotor_menstruation_breakdown")) {
 app_nonmotor_detailed_menstruation_breakdown <- detailed_nonmotor_menstruation_breakdown
 cat(" ✅ app_nonmotor_detailed_menstruation_breakdown created\n")
}

if(exists("all_nonmotor_symptoms")) {
 app_all_nonmotor_symptoms_data <- all_nonmotor_symptoms
 cat(" ✅ app_all_nonmotor_symptoms_data created (raw data)\n")
}

cat("\n=== NON-MOTOR SYMPTOM ANALYSIS COMPLETE ===\n")
cat("All non-motor symptom summary tables saved with 'app_' prefix\n")

library(dplyr)

cat("=== TOP 5 MOST COMMONLY REPORTED SYMPTOMS ===\n")
cat("Combined Motor and Non-Motor Symptoms Analysis\n")
cat("Equivalent to initial survey most common symptoms summary\n\n")

if(exists("app_motor_symptom_summary") && exists("app_nonmotor_symptom_summary")) {
```

```
top_5_motor_combined <- app_motor_symptom_summary %>%
  head(5) %>%
  select(Symptom, participants_reporting, prevalence_percentage, mean_severity) %>%
  mutate(symptom_type = "Motor") %>%
  arrange(desc(participants_reporting))

top_5_nonmotor_combined <- app_nonmotor_symptom_summary %>%
  head(5) %>%
  select(Symptom, participants_reporting, prevalence_percentage, mean_severity) %>%
  mutate(symptom_type = "Non-Motor") %>%
  arrange(desc(participants_reporting))

top_symptoms_combined <- rbind(top_5_motor_combined, top_5_nonmotor_combined) %>%
  arrange(symptom_type, desc(participants_reporting))

cat("TOP 5 MOTOR SYMPTOMS:\n")
cat("====================\n")
print(top_5_motor_combined)

cat("\n\nTOP 5 NON-MOTOR SYMPTOMS:\n")
cat("========================\n")
print(top_5_nonmotor_combined)

cat("\n\nCOMBINED TOP SYMPTOMS SUMMARY:\n")
cat("============================\n")
print(top_symptoms_combined)

cat("\n\nCOMPARISON STATISTICS:\n")
cat("======================\n")

motor_stats <- list(
  total_participants = max(app_motor_symptom_summary$participants_reporting),
  mean_prevalence = round(mean(top_5_motor_combined$prevalence_percentage), 1),
  mean_severity = round(mean(top_5_motor_combined$mean_severity), 2),
  most_common = top_5_motor_combined$Symptom[1],
  highest_severity =
top_5_motor_combined$Symptom[which.max(top_5_motor_combined$mean_severity)]
)

nonmotor_stats <- list(
  total_participants = max(app_nonmotor_symptom_summary$participants_reporting),
  mean_prevalence = round(mean(top_5_nonmotor_combined$prevalence_percentage), 1),
  mean_severity = round(mean(top_5_nonmotor_combined$mean_severity), 2),
  most_common = top_5_nonmotor_combined$Symptom[1],
  highest_severity =
top_5_nonmotor_combined$Symptom[which.max(top_5_nonmotor_combined$mean_severity)]
)

cat("MOTOR SYMPTOMS (Top 5):\n")
```

```
 cat("- Most common:", motor_stats$most_common, "(",
top_5_motor_combined$participants_reporting[1], "participants)\n")
 cat("- Average prevalence among top 5:", motor_stats$mean_prevalence, "%\n")
 cat("- Average severity among top 5:", motor_stats$mean_severity, "\n")
 cat("- Highest severity in top 5:", motor_stats$highest_severity,
    "(", top_5_motor_combined$mean_severity[which.max(top_5_motor_combined$mean_severity)],
")\n")

 cat("\nNON-MOTOR SYMPTOMS (Top 5):\n")
 cat("- Most common:", nonmotor_stats$most_common, "(",
top_5_nonmotor_combined$participants_reporting[1], "participants)\n")
 cat("- Average prevalence among top 5:", nonmotor_stats$mean_prevalence, "%\n")
 cat("- Average severity among top 5:", nonmotor_stats$mean_severity, "\n")
 cat("- Highest severity in top 5:", nonmotor_stats$highest_severity,
    "(",
top_5_nonmotor_combined$mean_severity[which.max(top_5_nonmotor_combined$mean_severity)],
")\n")

 cat("\nCROSS-CATEGORY COMPARISON:\n")
 cat("==========================\n")

 overall_most_common <- rbind(
  app_motor_symptom_summary %>%
   select(Symptom, participants_reporting, prevalence_percentage, mean_severity) %>%
   mutate(symptom_type = "Motor"),
  app_nonmotor_symptom_summary %>%
   select(Symptom, participants_reporting, prevalence_percentage, mean_severity) %>%
   mutate(symptom_type = "Non-Motor")
 ) %>%
  arrange(desc(participants_reporting)) %>%
  head(1)

 cat("Overall most reported symptom:", overall_most_common$Symptom,
    "(", overall_most_common$symptom_type, ",",
    overall_most_common$participants_reporting, "participants,",
    overall_most_common$prevalence_percentage, "%)\n")

 if(motor_stats$mean_prevalence > nonmotor_stats$mean_prevalence) {
  cat("Motor symptoms have higher average prevalence in top 5 (",
     motor_stats$mean_prevalence, "% vs ", nonmotor_stats$mean_prevalence, "%)\n")
 } else {
  cat("Non-motor symptoms have higher average prevalence in top 5 (",
     nonmotor_stats$mean_prevalence, "% vs ", motor_stats$mean_prevalence, "%)\n")
 }

 if(motor_stats$mean_severity > nonmotor_stats$mean_severity) {
  cat("Motor symptoms have higher average severity in top 5 (",
     motor_stats$mean_severity, " vs ", nonmotor_stats$mean_severity, ")\n")
 } else {
```

```
   cat("Non-motor symptoms have higher average severity in top 5 (",
       nonmotor_stats$mean_severity, " vs ", motor_stats$mean_severity, ")\n")
  }

  cat("\n\nTOP SYMPTOMS BY MENSTRUATION STATUS:\n")
  cat("====================================\n")

  if(exists("app_motor_detailed_menstruation_breakdown") &&
exists("app_nonmotor_detailed_menstruation_breakdown")) {

   top_motor_by_status <- app_motor_detailed_menstruation_breakdown %>%
     filter(Symptom == motor_stats$most_common) %>%
     arrange(desc(participants))

   top_nonmotor_by_status <- app_nonmotor_detailed_menstruation_breakdown %>%
     filter(Symptom == nonmotor_stats$most_common) %>%
     arrange(desc(participants))

   cat("Most common MOTOR symptom (", motor_stats$most_common, ") by menstruation status:\n")
   print(top_motor_by_status)

   cat("\nMost common NON-MOTOR symptom (", nonmotor_stats$most_common, ") by menstruation
status:\n")
   print(top_nonmotor_by_status)
  }

} else {
  cat("ERROR: Required summary data not found.\n")
  cat("Please ensure the following analyses have been run:\n")
  cat("- Motor symptom analysis (app_motor_symptom_summary)\n")
  cat("- Non-motor symptom analysis (app_nonmotor_symptom_summary)\n")
}

cat("\n\n=== SAVING RESULTS ===\n")

if(exists("top_symptoms_combined")) {
  app_top_symptoms_combined <- top_symptoms_combined
  cat(" ✅ app_top_symptoms_combined created\n")
}

if(exists("top_5_motor_combined")) {
  app_top_5_motor_combined <- top_5_motor_combined
  cat(" ✅ app_top_5_motor_combined created\n")
}

if(exists("top_5_nonmotor_combined")) {
  app_top_5_nonmotor_combined <- top_5_nonmotor_combined
  cat(" ✅ app_top_5_nonmotor_combined created\n")
```

```
}

if(exists("overall_most_common")) {
 app_overall_most_common_symptom <- overall_most_common
 cat("✅ app_overall_most_common_symptom created\n")
}

if(exists("motor_stats") && exists("nonmotor_stats")) {
 app_symptom_comparison_stats <- list(
  motor = motor_stats,
  nonmotor = nonmotor_stats
 )
 cat("✅ app_symptom_comparison_stats created\n")
}

cat("\n=== TOP SYMPTOMS COMBINED ANALYSIS COMPLETE ===\n")
cat("All combined symptom summary tables saved with 'app_' prefix\n")

cat("\n\n")
cat(strrep("=", 80), "\n")
cat("MENSTRUATION STATUS STRATIFICATION ANALYSIS - SYMPTOM PREVALENCE & SEVERITY\n")
cat(strrep("=", 80), "\n")
cat("Re-running all symptom analyses stratified by menstruation status\n")
cat("Comparing: Menstruating vs. Postmenopausal participants\n")
cat("Excluding: Perimenopausal participants from stratified analysis\n\n")

cat("STEP 1: Preparing stratified participant groups and symptom data\n")
cat("============================================================\n")

if(!exists("new_menstruationstatus_table")) {
 stop("ERROR: new_menstruationstatus_table not found. Please run Script 03 first.")
}

required_objects <- c("app_motor_symptom_summary", "app_nonmotor_symptom_summary",
                      "all_motor_symptoms", "all_nonmotor_symptoms")
missing_objects <- required_objects[!sapply(required_objects, exists)]

if(length(missing_objects) > 0) {
 stop("ERROR: Missing required objects from main analysis: ", paste(missing_objects, collapse = ", "))
}

all_motor_participants <- unique(all_motor_symptoms$research_id)
all_nonmotor_participants <- unique(all_nonmotor_symptoms$research_id)
all_symptom_participants <- unique(c(all_motor_participants, all_nonmotor_participants))

cat("Total participants in symptom data:", length(all_symptom_participants), "\n")

symptom_participants_menstruation <- new_menstruationstatus_table %>%
 filter(UniqueID %in% all_symptom_participants) %>%
```

```
 rename(research_id = UniqueID) %>%
 filter(Description %in% c("Menstruating", "Postmenopausal"))

menstruating_symptom_participants <- symptom_participants_menstruation %>%
 filter(Description == "Menstruating") %>%
 pull(research_id)

postmenopausal_symptom_participants <- symptom_participants_menstruation %>%
 filter(Description == "Postmenopausal") %>%
 pull(research_id)

cat("Sample sizes for stratified symptom analysis:\n")
cat("- Menstruating participants with symptom data:", length(menstruating_symptom_participants), "\n")
cat("- Postmenopausal participants with symptom data:",
length(postmenopausal_symptom_participants), "\n")
cat("- Excluded (Perimenopausal):", sum(new_menstruationstatus_table$UniqueID %in%
all_symptom_participants &
                        new_menstruationstatus_table$Description == "Perimenopausal"), "\n")
cat("- Total for stratified analysis:", length(menstruating_symptom_participants) +
length(postmenopausal_symptom_participants), "\n\n")

if(length(menstruating_symptom_participants) < 5) {
 cat(" ⚠️ WARNING: Menstruating group has n < 5. Statistical comparisons may be unreliable.\n")
}
if(length(postmenopausal_symptom_participants) < 5) {
 cat(" ⚠️ WARNING: Postmenopausal group has n < 5. Statistical comparisons may be unreliable.\n")
}

cat("\nSTEP 2: Stratified motor symptom analysis\n")
cat("=========================================\n")

create_stratified_motor_analysis <- function(participants, group_name, total_group_participants) {
 cat("\n--- ", group_name, " MOTOR SYMPTOM ANALYSIS ---\n")

 group_motor_symptoms <- all_motor_symptoms %>%
  filter(research_id %in% participants)

 if(nrow(group_motor_symptoms) == 0) {
  cat("No motor symptom data found for", group_name, "participants\n")
  return(data.frame())
 }

 cat("Total motor symptom entries for", group_name, ":", nrow(group_motor_symptoms), "\n")
 cat("Unique", group_name, "participants reporting motor symptoms:",
length(unique(group_motor_symptoms$research_id)), "\n")
 cat("Unique motor symptoms reported by", group_name, ":",
length(unique(group_motor_symptoms$Symptom)), "\n")

 motor_symptom_summary <- group_motor_symptoms %>%
```

```
  group_by(Symptom) %>%
  summarise(
   participants_reporting = n_distinct(research_id),
   total_reports = n(),
   mean_severity = round(mean(Severity, na.rm = TRUE), 2),
   median_severity = round(median(Severity, na.rm = TRUE), 2),
   min_severity = min(Severity, na.rm = TRUE),
   max_severity = max(Severity, na.rm = TRUE),
   sd_severity = round(sd(Severity, na.rm = TRUE), 2),
   sem_severity = round(sd(Severity, na.rm = TRUE) / sqrt(n_distinct(research_id)), 2),
   .groups = 'drop'
  ) %>%
  mutate(
   prevalence_percentage = round((participants_reporting / total_group_participants) * 100, 1),
   mean_sem_formatted = ifelse(is.na(mean_severity) | is.na(sem_severity),
                               "-",
                               paste0(mean_severity, " ± ", sem_severity))
  ) %>%
  arrange(desc(participants_reporting))

 cat("\nTop 5 motor symptoms for", group_name, ":\n")
 print(head(motor_symptom_summary, 5))

 return(motor_symptom_summary)
}

motor_symptom_summary_menstruating <- create_stratified_motor_analysis(
 menstruating_symptom_participants, "MENSTRUATING", length(menstruating_symptom_participants)
)

motor_symptom_summary_postmenopausal <- create_stratified_motor_analysis(
 postmenopausal_symptom_participants, "POSTMENOPAUSAL",
length(postmenopausal_symptom_participants)
)

cat("\nSTEP 3: Stratified non-motor symptom analysis\n")
cat("==============================================\n")

create_stratified_nonmotor_analysis <- function(participants, group_name, total_group_participants) {
 cat("\n--- ", group_name, " NON-MOTOR SYMPTOM ANALYSIS ---\n")

 group_nonmotor_symptoms <- all_nonmotor_symptoms %>%
  filter(research_id %in% participants)

 if(nrow(group_nonmotor_symptoms) == 0) {
  cat("No non-motor symptom data found for", group_name, "participants\n")
  return(data.frame())
 }
```

```
  cat("Total non-motor symptom entries for", group_name, ":", nrow(group_nonmotor_symptoms), "\n")
  cat("Unique", group_name, "participants reporting non-motor symptoms:",
length(unique(group_nonmotor_symptoms$research_id)), "\n")
  cat("Unique non-motor symptoms reported by", group_name, ":",
length(unique(group_nonmotor_symptoms$Symptom)), "\n")

  nonmotor_symptom_summary <- group_nonmotor_symptoms %>%
    group_by(Symptom) %>%
    summarise(
      participants_reporting = n_distinct(research_id),
      total_reports = n(),
      mean_severity = round(mean(Severity, na.rm = TRUE), 2),
      median_severity = round(median(Severity, na.rm = TRUE), 2),
      min_severity = min(Severity, na.rm = TRUE),
      max_severity = max(Severity, na.rm = TRUE),
      sd_severity = round(sd(Severity, na.rm = TRUE), 2),
      sem_severity = round(sd(Severity, na.rm = TRUE) / sqrt(n_distinct(research_id)), 2),
      .groups = 'drop'
    ) %>%
    mutate(
      prevalence_percentage = round((participants_reporting / total_group_participants) * 100, 1),
      mean_sem_formatted = ifelse(is.na(mean_severity) | is.na(sem_severity),
                                  "-",
                                  paste0(mean_severity, " ± ", sem_severity))
    ) %>%
    arrange(desc(participants_reporting))

  cat("\nTop 5 non-motor symptoms for", group_name, ":\n")
  print(head(nonmotor_symptom_summary, 5))

  return(nonmotor_symptom_summary)
}

nonmotor_symptom_summary_menstruating <- create_stratified_nonmotor_analysis(
  menstruating_symptom_participants, "MENSTRUATING", length(menstruating_symptom_participants)
)

nonmotor_symptom_summary_postmenopausal <- create_stratified_nonmotor_analysis(
  postmenopausal_symptom_participants, "POSTMENOPAUSAL",
length(postmenopausal_symptom_participants)
)

cat("\nSTEP 4: Creating side-by-side comparison tables\n")
cat("================================================\n")

create_symptom_comparison_table <- function(summary_menstru, summary_postmeno,
symptom_type) {
  all_symptoms <- unique(c(summary_menstru$Symptom, summary_postmeno$Symptom))
```

```
 if(length(all_symptoms) == 0) {
  cat("No symptoms found for comparison in", symptom_type, "category\n")
  return(data.frame())
 }

 comparison_table <- data.frame(
  Symptom = all_symptoms,
  stringsAsFactors = FALSE
 )

 comparison_table <- comparison_table %>%
  left_join(
   summary_menstru %>% select(Symptom, participants_reporting, prevalence_percentage,
mean_sem_formatted),
   by = "Symptom"
  ) %>%
  rename(
   Menstruating_n = participants_reporting,
   Menstruating_prevalence = prevalence_percentage,
   Menstruating_severity = mean_sem_formatted
  ) %>%
  left_join(
   summary_postmeno %>% select(Symptom, participants_reporting, prevalence_percentage,
mean_sem_formatted),
   by = "Symptom",
   suffix = c("", "_postmeno")
  ) %>%
  rename(
   Postmenopausal_n = participants_reporting,
   Postmenopausal_prevalence = prevalence_percentage,
   Postmenopausal_severity = mean_sem_formatted
  ) %>%
  mutate(
   Menstruating_n = ifelse(is.na(Menstruating_n), 0, Menstruating_n),
   Menstruating_prevalence = ifelse(is.na(Menstruating_prevalence), 0, Menstruating_prevalence),
   Menstruating_severity = ifelse(is.na(Menstruating_severity), "-", Menstruating_severity),
   Postmenopausal_n = ifelse(is.na(Postmenopausal_n), 0, Postmenopausal_n),
   Postmenopausal_prevalence = ifelse(is.na(Postmenopausal_prevalence), 0,
Postmenopausal_prevalence),
   Postmenopausal_severity = ifelse(is.na(Postmenopausal_severity), "-", Postmenopausal_severity),
   Prevalence_Difference = Menstruating_prevalence - Postmenopausal_prevalence
  ) %>%
  arrange(desc(pmax(Menstruating_n, Postmenopausal_n, na.rm = TRUE)))

 cat("\n", symptom_type, "SYMPTOM COMPARISON TABLE (Top 10):\n")
 cat(strrep("=", 50), "\n")
 print(head(comparison_table, 10))

 return(comparison_table)
```

```
}

motor_comparison_table <- create_symptom_comparison_table(
 motor_symptom_summary_menstruating,
 motor_symptom_summary_postmenopausal,
 "MOTOR"
)

nonmotor_comparison_table <- create_symptom_comparison_table(
 nonmotor_symptom_summary_menstruating,
 nonmotor_symptom_summary_postmenopausal,
 "NON-MOTOR"
)

cat("\nSTEP 5: Statistical comparisons for ALL qualifying symptoms\n")
cat("============================================================\n")

perform_symptom_statistical_test <- function(symptom_name, menstru_n, menstru_total, postmeno_n,
postmeno_total) {
 contingency_table <- matrix(c(
  menstru_n, postmeno_n,
  menstru_total - menstru_n, postmeno_total - postmeno_n
 ), nrow = 2, ncol = 2,
 dimnames = list(
  c("Has_symptom", "No_symptom"),
  c("Menstruating", "Postmenopausal")
 ))

 if (any(colSums(contingency_table) == 0) || any(rowSums(contingency_table) == 0)) {
  return(list(p.value = NA, method = "No variation in data", test_used = "None"))
 }

 expected_counts <- chisq.test(contingency_table, simulate.p.value = TRUE)$expected

 if (any(expected_counts < 5)) {
  test_result <- tryCatch({
   fisher.test(contingency_table)
  }, error = function(e) {
   return(list(p.value = NA, method = paste("Fisher's test failed:", e$message), test_used = "Failed"))
  })
  test_result$test_used <- "Fisher's Exact Test"
 } else {
  test_result <- tryCatch({
   chisq.test(contingency_table, correct = TRUE)
  }, error = function(e) {
   return(list(p.value = NA, method = paste("Chi-square test failed:", e$message), test_used = "Failed"))
  })
  test_result$test_used <- "Chi-square Test"
 }
```

```
 return(test_result)
}

test_all_qualifying_symptoms <- function(comparison_table, symptom_type, n_menstruating,
n_postmenopausal) {
 cat("\n--- Testing ALL qualifying", symptom_type, "symptoms ---\n")

 if(nrow(comparison_table) == 0) {
  cat("No symptoms available for testing in", symptom_type, "category\n")
  return(data.frame())
 }

 qualifying_symptoms <- comparison_table %>%
  filter(Menstruating_n >= 5 | Postmenopausal_n >= 5)

 cat("Total symptoms in", symptom_type, "category:", nrow(comparison_table), "\n")
 cat("Symptoms meeting sample size criteria (n>=5):", nrow(qualifying_symptoms), "\n")

 if(nrow(qualifying_symptoms) == 0) {
  cat("No symptoms meet the minimum sample size criteria for statistical testing\n")
  return(data.frame())
 }

 test_results <- data.frame()

 for(i in 1:nrow(qualifying_symptoms)) {
  symptom <- qualifying_symptoms$Symptom[i]
  menstru_n <- qualifying_symptoms$Menstruating_n[i]
  postmeno_n <- qualifying_symptoms$Postmenopausal_n[i]

  cat("\nTesting:", symptom, "(", i, "of", nrow(qualifying_symptoms), ")\n")
  cat("Menstruating:", menstru_n, "/", n_menstruating,
      "(", round(qualifying_symptoms$Menstruating_prevalence[i], 1), "%)\n")
  cat("Postmenopausal:", postmeno_n, "/", n_postmenopausal,
      "(", round(qualifying_symptoms$Postmenopausal_prevalence[i], 1), "%)\n")

  test_result <- perform_symptom_statistical_test(
   symptom, menstru_n, n_menstruating, postmeno_n, n_postmenopausal
  )

  cat("Test:", test_result$test_used, "\n")
  if(!is.na(test_result$p.value)) {
   cat("p-value:", format(test_result$p.value, digits = 4), "\n")
   significance <- ifelse(test_result$p.value < 0.05, "SIGNIFICANT", "Not significant")
   cat("Result:", significance, "\n")
  } else {
   cat("p-value: Not available\n")
   significance <- "Not testable"
```

```
  }

  test_results <- rbind(test_results, data.frame(
    Symptom_Type = symptom_type,
    Symptom = symptom,
    Menstruating_n = menstru_n,
    Menstruating_prevalence = qualifying_symptoms$Menstruating_prevalence[i],
    Postmenopausal_n = postmeno_n,
    Postmenopausal_prevalence = qualifying_symptoms$Postmenopausal_prevalence[i],
    Prevalence_Difference = qualifying_symptoms$Prevalence_Difference[i],
    Test_Used = test_result$test_used,
    P_Value = test_result$p.value,
    Significance = significance,
    stringsAsFactors = FALSE
  ))
 }

 return(test_results)
}

motor_test_results <- test_all_qualifying_symptoms(
 motor_comparison_table, "Motor",
 length(menstruating_symptom_participants),
 length(postmenopausal_symptom_participants)
)

nonmotor_test_results <- test_all_qualifying_symptoms(
 nonmotor_comparison_table, "Non-Motor",
 length(menstruating_symptom_participants),
 length(postmenopausal_symptom_participants)
)

all_symptom_test_results <- rbind(motor_test_results, nonmotor_test_results)

cat("\n=== STATISTICAL TESTING SUMMARY ===\n")
cat("Total motor symptoms tested:", nrow(motor_test_results), "\n")
cat("Total non-motor symptoms tested:", nrow(nonmotor_test_results), "\n")
cat("Total symptoms with statistical tests:", nrow(all_symptom_test_results), "\n")

if(nrow(all_symptom_test_results) > 0) {
 testable_results <- all_symptom_test_results %>% filter(!is.na(P_Value))
 cat("Tests that could be performed:", nrow(testable_results), "\n")
 cat("Tests that failed or had no variation:", nrow(all_symptom_test_results) - nrow(testable_results),
"\n")
}

cat("\nSTEP 6: Summary of significant findings\n")
cat("=====================================\n")
```

```
if(nrow(all_symptom_test_results) > 0) {
 significant_findings <- all_symptom_test_results %>%
  filter(Significance == "SIGNIFICANT") %>%
  arrange(P_Value)

 cat("SIGNIFICANT DIFFERENCES IN SYMPTOM PREVALENCE:\n")
 cat(strrep("-", 50), "\n")

 if(nrow(significant_findings) > 0) {
  for(i in 1:nrow(significant_findings)) {
   finding <- significant_findings[i, ]
   cat(i, ". ", finding$Symptom, " (", finding$Symptom_Type, ")\n", sep = "")
   cat("   Menstruating: ", finding$Menstruating_prevalence, "% | ",
       "Postmenopausal: ", finding$Postmenopausal_prevalence, "%\n", sep = "")
   cat("   Difference: ", round(finding$Prevalence_Difference, 1), "% | ",
       "p-value: ", format(finding$P_Value, digits = 4), "\n\n", sep = "")
  }

  cat("Total significant findings:", nrow(significant_findings), "\n")
 } else {
  cat("No statistically significant differences found in symptom prevalence.\n")
 }

 cat("\nOVERALL TESTING SUMMARY:\n")
 cat("- Total symptoms tested:", nrow(all_symptom_test_results), "\n")
 cat("- Motor symptoms tested:", sum(all_symptom_test_results$Symptom_Type == "Motor"), "\n")
 cat("- Non-motor symptoms tested:", sum(all_symptom_test_results$Symptom_Type == "Non-Motor"),
"\n")
 cat("- Significant findings:", nrow(significant_findings), "\n")
 cat("- Fisher's exact tests used:", sum(all_symptom_test_results$Test_Used == "Fisher's Exact Test",
na.rm = TRUE), "\n")
 cat("- Chi-square tests used:", sum(all_symptom_test_results$Test_Used == "Chi-square Test", na.rm =
TRUE), "\n")

} else {
 cat("No statistical tests could be performed.\n")
}

cat("\nSTEP 7: Saving stratified results\n")
cat("================================\n")

app_motor_symptom_summary_menstruating <- motor_symptom_summary_menstruating
app_motor_symptom_summary_postmenopausal <- motor_symptom_summary_postmenopausal
app_motor_symptom_comparison_table <- motor_comparison_table

app_nonmotor_symptom_summary_menstruating <- nonmotor_symptom_summary_menstruating
app_nonmotor_symptom_summary_postmenopausal <-
nonmotor_symptom_summary_postmenopausal
app_nonmotor_symptom_comparison_table <- nonmotor_comparison_table
```

```
app_symptom_statistical_test_results <- all_symptom_test_results
app_symptom_significant_findings <- if(exists("significant_findings")) significant_findings else
data.frame()

app_symptom_stratification_info <- list(
 menstruating_participants = menstruating_symptom_participants,
 postmenopausal_participants = postmenopausal_symptom_participants,
 sample_sizes = list(
  menstruating = length(menstruating_symptom_participants),
  postmenopausal = length(postmenopausal_symptom_participants)
 ),
 total_tested_symptoms = nrow(all_symptom_test_results),
 significant_findings_count = if(exists("significant_findings")) nrow(significant_findings) else 0
)

cat("✅ app_motor_symptom_summary_menstruating created\n")
cat("✅ app_motor_symptom_summary_postmenopausal created\n")
cat("✅ app_motor_symptom_comparison_table created\n")
cat("✅ app_nonmotor_symptom_summary_menstruating created\n")
cat("✅ app_nonmotor_symptom_summary_postmenopausal created\n")
cat("✅ app_nonmotor_symptom_comparison_table created\n")
cat("✅ app_symptom_statistical_test_results created\n")
cat("✅ app_symptom_significant_findings created\n")
cat("✅ app_symptom_stratification_info created\n")

cat("\nSTEP 8: Final summary and clinical interpretation\n")
cat("================================================\n")

cat("📍 STRATIFIED SYMPTOM ANALYSIS SUMMARY:\n")
cat("• Total participants with symptom data analyzed:",
  length(menstruating_symptom_participants) + length(postmenopausal_symptom_participants), "\n")
cat("• Menstruating participants:", length(menstruating_symptom_participants),
  "(", round(length(menstruating_symptom_participants) /
        (length(menstruating_symptom_participants) + length(postmenopausal_symptom_participants))
* 100, 1), "%)\n")
cat("• Postmenopausal participants:", length(postmenopausal_symptom_participants),
  "(", round(length(postmenopausal_symptom_participants) /
        (length(menstruating_symptom_participants) + length(postmenopausal_symptom_participants))
* 100, 1), "%)\n")

cat("\n📊 KEY SYMPTOM PATTERNS:\n")

if(nrow(motor_symptom_summary_menstruating) > 0 &&
nrow(motor_symptom_summary_postmenopausal) > 0) {
 top_motor_menstru <- motor_symptom_summary_menstruating$Symptom[1]
 top_motor_postmeno <- motor_symptom_summary_postmenopausal$Symptom[1]
 cat("• Most common motor symptom in menstruating:", top_motor_menstru, "\n")
```

```
  cat("• Most common motor symptom in postmenopausal:", top_motor_postmeno, "\n")
}

if(nrow(nonmotor_symptom_summary_menstruating) > 0 &&
nrow(nonmotor_symptom_summary_postmenopausal) > 0) {
  top_nonmotor_menstru <- nonmotor_symptom_summary_menstruating$Symptom[1]
  top_nonmotor_postmeno <- nonmotor_symptom_summary_postmenopausal$Symptom[1]
  cat("• Most common non-motor symptom in menstruating:", top_nonmotor_menstru, "\n")
  cat("• Most common non-motor symptom in postmenopausal:", top_nonmotor_postmeno, "\n")
}

cat("\n 📈 STATISTICAL SUMMARY:\n")
if(nrow(all_symptom_test_results) > 0) {
  cat("• Total symptoms compared:", nrow(all_symptom_test_results), "\n")
  cat("• Statistically significant differences:",
      if(exists("significant_findings")) nrow(significant_findings) else 0, "\n")

  if(exists("significant_findings") && nrow(significant_findings) > 0) {
    cat("• Most significant finding:", significant_findings$Symptom[1],
        " (p =", format(significant_findings$P_Value[1], digits = 4), ")\n")
  }
} else {
  cat("• No statistical comparisons possible due to insufficient data\n")
}

cat("\n ⚠️ SAMPLE SIZE ASSESSMENT:\n")
if(length(menstruating_symptom_participants) >= 5 && length(postmenopausal_symptom_participants)
>= 5) {
  cat("• Sample sizes adequate for statistical analysis\n")
} else {
  cat("• Sample sizes may be insufficient for robust statistical conclusions\n")
}

cat("\n", strrep("=", 80), "\n")
cat("MENSTRUATION STATUS STRATIFICATION ANALYSIS - SYMPTOM PREVALENCE & SEVERITY
COMPLETE\n")
cat(strrep("=", 80), "\n")
cat("All stratified symptom results saved with appropriate suffixes:\n")
cat("- '_menstruating' for menstruating participant results\n")
cat("- '_postmenopausal' for postmenopausal participant results\n")
cat("- '_comparison_table' for side-by-side comparisons\n")
cat("- '_statistical_test_results' for all test results\n")
cat("- '_significant_findings' for significant differences only\n")
cat("- '_stratification_info' for participant and analysis metadata\n")

library(knitr)
library(kableExtra)
library(dplyr)
```

```
cat("=== EXTRACTING MENSTRUATION STATUS SYMPTOM COMPARISON DATA ===\n")

cat("\n=== EXTRACTING SAMPLE SIZES ===\n")

menstruating_n <- app_symptom_stratification_info$sample_sizes$menstruating
postmenopausal_n <- app_symptom_stratification_info$sample_sizes$postmenopausal

cat("Sample sizes:\n")
cat("- Menstruating participants:", menstruating_n, "\n")
cat("- Postmenopausal participants:", postmenopausal_n, "\n")

menstruating_header <- paste0("Menstruating (n=", menstruating_n, ")")
postmenopausal_header <- paste0("Postmenopausal (n=", postmenopausal_n, ")")

format_pvalue <- function(p_value) {
 if(is.na(p_value)) {
  return("N/A")
 }

 rounded_p <- round(p_value, 3)

 if(p_value < 0.001) {
  return(paste0(rounded_p, "***"))
 } else if(p_value < 0.01) {
  return(paste0(rounded_p, "**"))
 } else if(p_value < 0.05) {
  return(paste0(rounded_p, "*"))
 } else {
  return(as.character(rounded_p))
 }
}

cat("\n=== PROCESSING MOTOR SYMPTOMS DATA ===\n")

motor_symptoms_clean <- app_motor_symptom_comparison_table %>%
 filter(Menstruating_n >= 5 | Postmenopausal_n >= 5) %>%
 select(
  Symptom,
  Menstruating_n,
  Postmenopausal_n,
  Menstruating_severity,
  Postmenopausal_severity
 ) %>%
 mutate(
  Menstruating_severity = ifelse(is.na(Menstruating_severity), "-", as.character(Menstruating_severity)),
  Postmenopausal_severity = ifelse(is.na(Postmenopausal_severity), "-",
as.character(Postmenopausal_severity))
 )
```

```
motor_with_pvalues <- motor_symptoms_clean %>%
 left_join(
  app_symptom_statistical_test_results %>%
   filter(Symptom_Type == "Motor") %>%
   select(Symptom, P_Value),
  by = "Symptom"
 ) %>%
 mutate(
  P_Value_Formatted = sapply(P_Value, format_pvalue)
 ) %>%
 select(-P_Value)

motor_symptoms_kable <- motor_with_pvalues %>%
 kable(
  col.names = c("Symptom", menstruating_header, postmenopausal_header, "Menstruating Mean ± SEM", "Postmenopausal Mean ± SEM", "P-value"),
  caption = "Motor Symptoms: Menstruating vs Postmenopausal Comparison",
  align = c("l", "c", "c", "c", "c", "c")
 ) %>%
 kable_styling(
  bootstrap_options = c("striped", "hover", "condensed", "responsive"),
  full_width = FALSE,
  position = "center"
 ) %>%
 add_header_above(c(" " = 1, "Sample Size" = 2, "Mean ± SEM" = 2, "Statistical Test" = 1)) %>%
 column_spec(1, bold = TRUE) %>%
 column_spec(2:3, color = "blue") %>%
 column_spec(4:5, color = "darkgreen") %>%
 column_spec(6, color = "red") %>%
 footnote(general = "* p < 0.05, ** p < 0.01, *** p < 0.001; SEM = Standard Error of the Mean")

cat("✅ Motor symptoms table created\n")
cat("   - Rows included:", nrow(motor_with_pvalues), "\n")
cat("   - Symptoms with adequate sample sizes (n>=5 in either group)\n")

cat("\n=== PROCESSING NON-MOTOR SYMPTOMS DATA ===\n")

nonmotor_symptoms_clean <- app_nonmotor_symptom_comparison_table %>%
 filter(Menstruating_n >= 5 | Postmenopausal_n >= 5) %>%
 select(
  Symptom,
  Menstruating_n,
  Postmenopausal_n,
  Menstruating_severity,
  Postmenopausal_severity
 ) %>%
 mutate(
  Menstruating_severity = ifelse(is.na(Menstruating_severity), "-", as.character(Menstruating_severity)),
```

```
  Postmenopausal_severity = ifelse(is.na(Postmenopausal_severity), "-",
as.character(Postmenopausal_severity))
 )

nonmotor_with_pvalues <- nonmotor_symptoms_clean %>%
 left_join(
  app_symptom_statistical_test_results %>%
   filter(Symptom_Type == "Non-Motor") %>%
   select(Symptom, P_Value),
  by = "Symptom"
 ) %>%
 mutate(
  P_Value_Formatted = sapply(P_Value, format_pvalue)
 ) %>%
 select(-P_Value)

nonmotor_symptoms_kable <- nonmotor_with_pvalues %>%
 kable(
  col.names = c("Symptom", menstruating_header, postmenopausal_header, "Menstruating Mean ±
SEM", "Postmenopausal Mean ± SEM", "P-value"),
  caption = "Non-Motor Symptoms: Menstruating vs Postmenopausal Comparison",
  align = c("l", "c", "c", "c", "c", "c")
 ) %>%
 kable_styling(
  bootstrap_options = c("striped", "hover", "condensed", "responsive"),
  full_width = FALSE,
  position = "center"
 ) %>%
 add_header_above(c(" " = 1, "Sample Size" = 2, "Mean ± SEM" = 2, "Statistical Test" = 1)) %>%
 column_spec(1, bold = TRUE) %>%
 column_spec(2:3, color = "blue") %>%
 column_spec(4:5, color = "darkgreen") %>%
 column_spec(6, color = "red") %>%
 footnote(general = "* p < 0.05, ** p < 0.01, *** p < 0.001; SEM = Standard Error of the Mean")

cat("✅ Non-motor symptoms table created\n")
cat("   - Rows included:", nrow(nonmotor_with_pvalues), "\n")
cat("   - Symptoms with adequate sample sizes (n>=5 in either group)\n")

cat("\n=== SAVING NEW VARIABLES ===\n")

app_motor_symptoms_menstruation_table <- motor_with_pvalues
app_nonmotor_symptoms_menstruation_table <- nonmotor_with_pvalues

app_motor_symptoms_kable_display <- motor_symptoms_kable
app_nonmotor_symptoms_kable_display <- nonmotor_symptoms_kable

cat("✅ app_motor_symptoms_menstruation_table created (with p-values)\n")
```

```
cat("✅ app_nonmotor_symptoms_menstruation_table created (with p-values)\n")
cat("✅ app_motor_symptoms_kable_display created\n")
cat("✅ app_nonmotor_symptoms_kable_display created\n")

cat("\n=== DISPLAYING FORMATTED TABLES ===\n")

cat("\n🔵 MOTOR SYMPTOMS COMPARISON:\n")
cat("============================\n")
print(motor_symptoms_kable)

cat("\n\n🟢 NON-MOTOR SYMPTOMS COMPARISON:\n")
cat("================================\n")
print(nonmotor_symptoms_kable)

cat("\n=== SUMMARY STATISTICS ===\n")

cat("📊 MOTOR SYMPTOMS SUMMARY:\n")
cat("- Total symptoms included:", nrow(motor_with_pvalues), "\n")
cat("- Symptoms with menstruating data:", sum(!is.na(motor_with_pvalues$Menstruating_n) &
motor_with_pvalues$Menstruating_n > 0), "\n")
cat("- Symptoms with postmenopausal data:", sum(!is.na(motor_with_pvalues$Postmenopausal_n) &
motor_with_pvalues$Postmenopausal_n > 0), "\n")
cat("- Symptoms with statistical tests:", sum(!is.na(motor_with_pvalues$P_Value_Formatted) &
motor_with_pvalues$P_Value_Formatted != "N/A"), "\n")

motor_significant <- motor_with_pvalues %>%
 filter(!is.na(P_Value_Formatted), P_Value_Formatted != "N/A") %>%
 mutate(
  sig_level = case_when(
   grepl("\\*\\*\\*", P_Value_Formatted) ~ "p < 0.001",
   grepl("\\*\\*", P_Value_Formatted) ~ "p < 0.01",
   grepl("\\*", P_Value_Formatted) ~ "p < 0.05",
   TRUE ~ "Not significant"
  )
 )

if(nrow(motor_significant) > 0) {
 cat("- Significance breakdown:\n")
 print(table(motor_significant$sig_level))
}

cat("\n📊 NON-MOTOR SYMPTOMS SUMMARY:\n")
cat("- Total symptoms included:", nrow(nonmotor_with_pvalues), "\n")
cat("- Symptoms with menstruating data:", sum(!is.na(nonmotor_with_pvalues$Menstruating_n) &
nonmotor_with_pvalues$Menstruating_n > 0), "\n")
cat("- Symptoms with postmenopausal data:", sum(!is.na(nonmotor_with_pvalues$Postmenopausal_n)
& nonmotor_with_pvalues$Postmenopausal_n > 0), "\n")
```

```
cat("- Symptoms with statistical tests:", sum(!is.na(nonmotor_with_pvalues$P_Value_Formatted) &
nonmotor_with_pvalues$P_Value_Formatted != "N/A"), "\n")

nonmotor_significant <- nonmotor_with_pvalues %>%
 filter(!is.na(P_Value_Formatted), P_Value_Formatted != "N/A") %>%
 mutate(
  sig_level = case_when(
   grepl("\\*\\*\\*", P_Value_Formatted) ~ "p < 0.001",
   grepl("\\*\\*", P_Value_Formatted) ~ "p < 0.01",
   grepl("\\*", P_Value_Formatted) ~ "p < 0.05",
   TRUE ~ "Not significant"
  )
 )

if(nrow(nonmotor_significant) > 0) {
 cat("- Significance breakdown:\n")
 print(table(nonmotor_significant$sig_level))
}

cat("\n=== MENSTRUATION STATUS SYMPTOM TABLES COMPLETE ===\n")
cat("All data extracted from existing Script 14 variables\n")
cat("Tables filtered for adequate sample sizes (n>=5)\n")
cat("Data maintains original prevalence-based ordering\n")
cat("Statistical significance marked with standard notation\n")
```

Script 2.5

```
library(writexl)
library(dplyr)

export_multi_cycle_participants <- function(output_file = "multi_cycle_participants.xlsx") {
 sheets_list <- list()

 multi_ids <- multi_cycle_participants$research_id

 cat("Exporting data for", length(multi_ids), "multi-cycle participants...\n")

 for (i in seq_along(multi_ids)) {
  id <- multi_ids[i]

  cat("Processing participant", i, "of", length(multi_ids), ":", id, "\n")

  participant_data <- final_combined_data_for_display %>%
   filter(research_id == id) %>%
   mutate(
    dates = format(dates, "%Y-%m-%d")
   )

  if (nrow(participant_data) == 0) {
   cat("  Warning: No data found for participant", id, "\n")
   next
  }

  med_info <- participant_data %>%
   filter(!is.na(scheduled_meds_info) &
         scheduled_meds_info != "" &
         scheduled_meds_info != "No Scheduled Meds" &
         scheduled_meds_info != "End of Scheduled Meds") %>%
   pull(scheduled_meds_info) %>%
   first()

  filtered_data <- participant_data %>%
   filter(!is.na(phase_id) & phase_id >= 0 & phase_id < 10)

  if (nrow(filtered_data) == 0) {
   cat("  Warning: No rows left after filtering phase_ids for participant", id, "\n")
   next
  }

  if (!is.null(med_info) && nrow(filtered_data) > 0) {
   filtered_data$scheduled_meds_info[1] <- med_info
   if (nrow(filtered_data) > 1) {
    filtered_data$scheduled_meds_info[2:nrow(filtered_data)] <- ""
   }
   cat("  Added medication information to filtered data\n")
```

```
  } else {
    cat("  No medication information found\n")
  }

  sheets_list[[id]] <- filtered_data

  cat("  Added", nrow(filtered_data), "rows for participant", id, "\n")
 }

 if (length(sheets_list) == 0) {
  cat("No data to export. Please check that multi_cycle_participants contains valid IDs.\n")
  return(NULL)
 }

 cat("\nWriting data to Excel file:", output_file, "\n")
 write_xlsx(sheets_list, output_file)

 cat("Export complete! File saved at:", output_file, "\n")
 cat("Total participants exported:", length(sheets_list), "\n")

 return(output_file)
}

export_multi_cycle_participants()
```

Script 8

```
rm(list = ls(pattern = "^(bleeding_.*|non_bleeding_.*|menstruating_vs_|cohort_.*_analysis)"))

rm(list = ls(pattern = "^(comparison_.*|vs_.*_data|cohort_comparison|group_.*_analysis)$"))

rm(list = ls(pattern = "^(t_test_.*|wilcox_.*|comparison_stats|statistical_comparison)$"))

rm(list = ls(pattern = "^(compare_.*_cohorts|analyze_bleeding|calculate_.*_differences)$"))

rm(list = ls(pattern = "^(bleeding_.*_plot|cohort_.*_plot|comparison_.*_chart)$"))

required_vars <- c("multi_cycle_merged_data", "single_cycle_merged_data", "symptom_categories")

missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 stop("Required variables from previous scripts missing: ", paste(missing_vars, collapse = ", "), "\n",
     "Please run Scripts 1-7 first.")
}

if(!exists("multi_cycle_participants") || !exists("single_cycle_participants")) {
 stop("Participant classification variables missing. Please run Scripts 1 and 2 first.")
}

required_packages <- c("ggplot2", "dplyr", "tidyr")
missing_packages <- c()

for(pkg in required_packages) {
 if(!require(pkg, character.only = TRUE, quietly = TRUE)) {
  missing_packages <- c(missing_packages, pkg)
 }
}

if(length(missing_packages) > 0) {
 stop("Required packages for cohort comparison missing: ", paste(missing_packages, collapse = ", "))
}

multi_flow_count <- sum(multi_cycle_merged_data$Flow != "No Entry", na.rm = TRUE)
single_flow_count <- sum(single_cycle_merged_data$Flow != "No Entry", na.rm = TRUE)

total_flow_entries <- multi_flow_count + single_flow_count

if(total_flow_entries == 0) {
 stop("No menstrual flow data found. Cannot classify bleeding vs non-bleeding cohorts.")
```

```
}

multi_symptom_count <- sum(multi_cycle_merged_data$Symptom != "No Entry", na.rm = TRUE)
single_symptom_count <- sum(single_cycle_merged_data$Symptom != "No Entry", na.rm = TRUE)

total_symptom_entries <- multi_symptom_count + single_symptom_count

if(total_symptom_entries == 0) {
 stop("No symptom data found in period-tracking participants. Cannot perform cohort comparison.")
}

participants_with_flow_and_symptoms <- length(intersect(
 unique(c(
  multi_cycle_merged_data$research_id[multi_cycle_merged_data$Flow != "No Entry"],
  single_cycle_merged_data$research_id[single_cycle_merged_data$Flow != "No Entry"]
 )),
 unique(c(
  multi_cycle_merged_data$research_id[multi_cycle_merged_data$Symptom != "No Entry"],
  single_cycle_merged_data$research_id[single_cycle_merged_data$Symptom != "No Entry"]
 ))
))

if(participants_with_flow_and_symptoms < 3) {
 warning("Very few participants (", participants_with_flow_and_symptoms,
      ") have both flow and symptom data. Cohort comparison may be limited.")
}

original_multi_cycle <- length(unique(multi_cycle_merged_data$research_id))
original_single_cycle <- length(unique(single_cycle_merged_data$research_id))
original_no_period <- length(unique(no_period_final$research_id))

multi_cycle_merged_data <- multi_cycle_merged_data %>%
 filter(research_id %in% consented_participants)

single_cycle_merged_data <- single_cycle_merged_data %>%
 filter(research_id %in% consented_participants)

no_period_final <- no_period_final %>%
 filter(research_id %in% consented_participants)

multi_cycle_participants <- multi_cycle_participants %>%
 filter(research_id %in% consented_participants)

single_cycle_participants <- single_cycle_participants %>%
 filter(research_id %in% consented_participants)

consent_multi_cycle <- length(unique(multi_cycle_merged_data$research_id))
consent_single_cycle <- length(unique(single_cycle_merged_data$research_id))
consent_no_period <- length(unique(no_period_final$research_id))
```

```
cat("\n=== CONSENT FILTERING RESULTS ===\n")
cat("Multi-cycle participants: ", consent_multi_cycle, " (was ", original_multi_cycle, ", excluded ",
original_multi_cycle - consent_multi_cycle, ")\n", sep="")
cat("Single-cycle participants: ", consent_single_cycle, " (was ", original_single_cycle, ", excluded ",
original_single_cycle - consent_single_cycle, ")\n", sep="")
cat("No-period participants: ", consent_no_period, " (was ", original_no_period, ", excluded ",
original_no_period - consent_no_period, ")\n", sep="")

total_original <- original_multi_cycle + original_single_cycle + original_no_period
total_consented <- consent_multi_cycle + consent_single_cycle + consent_no_period
total_excluded <- total_original - total_consented

cat("TOTAL: ", total_consented, " participants (was ", total_original, ", excluded ", total_excluded, " = ",
round(total_excluded/total_original*100, 1), "%)\n", sep="")

if(total_excluded > 0) {


cat("=== SCRIPT 8: BLEEDING VS NON-BLEEDING POPULATION ANALYSIS ===\n\n")

check_consecutive_period_days <- function(research_id, min_days = 3) {
 participant_periods <- mini_period_data %>%
  filter(research_id == !!research_id) %>%
  arrange(`Period Entry Date`)

 if(nrow(participant_periods) < min_days) {
  return(FALSE)
 }

 dates <- participant_periods$`Period Entry Date`

 date_diffs <- as.numeric(diff(dates))

 consecutive_count <- 1
 max_consecutive <- 1

 for(i in 1:length(date_diffs)) {
  if(date_diffs[i] == 1) {
   consecutive_count <- consecutive_count + 1
   max_consecutive <- max(max_consecutive, consecutive_count)
  } else {
   consecutive_count <- 1
  }
 }

 return(max_consecutive >= min_days)
}
```

```
check_consecutive_symptom_days <- function(research_id, min_days = 3) {
 if(research_id %in% c(multi_cycle_participants$research_id, single_cycle_participants$research_id)) {
  all_symptom_data <- bind_rows(
   multi_cycle_merged_data %>% filter(Symptom != "No Entry"),
   single_cycle_merged_data %>% filter(Symptom != "No Entry")
  )

  participant_symptoms <- all_symptom_data %>%
   filter(research_id == !!research_id) %>%
   distinct(dates) %>%
   arrange(dates)

 } else {
  participant_symptoms <- no_period_final %>%
   filter(research_id == !!research_id) %>%
   filter(Symptom != "No Entry") %>%
   distinct(dates) %>%
   arrange(dates)
 }

 if(nrow(participant_symptoms) < min_days) {
  return(FALSE)
 }

 dates <- participant_symptoms$dates

 date_diffs <- as.numeric(diff(dates))

 consecutive_count <- 1
 max_consecutive <- 1

 for(i in 1:length(date_diffs)) {
  if(date_diffs[i] == 1) {
   consecutive_count <- consecutive_count + 1
   max_consecutive <- max(max_consecutive, consecutive_count)
  } else {
   consecutive_count <- 1
  }
 }

 return(max_consecutive >= min_days)
}

cat("Step 1: Consecutive day checking functions defined\n")

cat("\nStep 2: Identifying eligible participants...\n")

potential_bleeding_participants <- c(
 multi_cycle_participants$research_id,
```

```
 single_cycle_participants$research_id
)

cat("Potential bleeding participants (before filtering):", length(potential_bleeding_participants), "\n")
cat("No-period participants (before filtering):", length(no_period_participants), "\n")

bleeding_consecutive_periods <- sapply(potential_bleeding_participants,
check_consecutive_period_days)
eligible_bleeding_period_check <- potential_bleeding_participants[bleeding_consecutive_periods]

cat("Participants with 3+ consecutive period days:", length(eligible_bleeding_period_check), "\n")

bleeding_consecutive_symptoms <- sapply(eligible_bleeding_period_check,
check_consecutive_symptom_days)
final_bleeding_participants <- eligible_bleeding_period_check[bleeding_consecutive_symptoms]

cat("Bleeding participants with both period and symptom criteria:", length(final_bleeding_participants),
"\n")

nonbleeding_consecutive_symptoms <- sapply(no_period_participants,
check_consecutive_symptom_days)
final_nonbleeding_participants <- no_period_participants[nonbleeding_consecutive_symptoms]

cat("Non-bleeding participants with symptom criteria:", length(final_nonbleeding_participants), "\n")

cat("\n=== FINAL PARTICIPANT COUNTS ===\n")
cat("Bleeding group (3+ consecutive period days + 3+ consecutive symptom days):",
length(final_bleeding_participants), "\n")
cat("Non-bleeding group (3+ consecutive symptom days, no period data):",
length(final_nonbleeding_participants), "\n")
cat("Total participants for analysis:", length(final_bleeding_participants) +
length(final_nonbleeding_participants), "\n")

cat("\nStep 2: Participant filtering complete\n")

cat("\nStep 3: Calculating average daily symptom severity for each participant...\n")

calculate_avg_daily_severity <- function(research_id) {
 if(research_id %in% final_bleeding_participants) {
  all_symptom_data <- bind_rows(
   multi_cycle_merged_data %>% filter(Symptom != "No Entry"),
   single_cycle_merged_data %>% filter(Symptom != "No Entry")
  )

  participant_symptoms <- all_symptom_data %>%
   filter(research_id == !!research_id) %>%
   mutate(Severity = as.numeric(Severity))

 } else if(research_id %in% final_nonbleeding_participants) {
```

```
   participant_symptoms <- no_period_final %>%
    filter(research_id == !!research_id) %>%
    filter(Symptom != "No Entry") %>%
    mutate(Severity = as.numeric(Severity))

  } else {
   return(NA)
  }

  if(nrow(participant_symptoms) == 0) {
   return(NA)
  }

  total_severity <- sum(participant_symptoms$Severity, na.rm = TRUE)
  unique_symptom_days <- n_distinct(participant_symptoms$dates)

  avg_daily_severity <- total_severity / unique_symptom_days

  return(avg_daily_severity)
}

cat("Calculating for bleeding participants...\n")
bleeding_severity_data <- data.frame(
  research_id = final_bleeding_participants,
  group = "Bleeding",
  avg_daily_severity = sapply(final_bleeding_participants, calculate_avg_daily_severity),
  stringsAsFactors = FALSE
)

cat("Calculating for non-bleeding participants...\n")
nonbleeding_severity_data <- data.frame(
  research_id = final_nonbleeding_participants,
  group = "Non-Bleeding",
  avg_daily_severity = sapply(final_nonbleeding_participants, calculate_avg_daily_severity),
  stringsAsFactors = FALSE
)

combined_severity_data <- rbind(bleeding_severity_data, nonbleeding_severity_data)

combined_severity_data <- combined_severity_data %>%
  filter(!is.na(avg_daily_severity))

cat("\n=== AVERAGE DAILY SYMPTOM SEVERITY SUMMARY ===\n")

bleeding_summary <- bleeding_severity_data %>%
  summarise(
    n = n(),
    mean = mean(avg_daily_severity, na.rm = TRUE),
    median = median(avg_daily_severity, na.rm = TRUE),
```

```
  sd = sd(avg_daily_severity, na.rm = TRUE),
  se_mean = sd / sqrt(n),
  min = min(avg_daily_severity, na.rm = TRUE),
  max = max(avg_daily_severity, na.rm = TRUE)
 )

nonbleeding_summary <- nonbleeding_severity_data %>%
 summarise(
  n = n(),
  mean = mean(avg_daily_severity, na.rm = TRUE),
  median = median(avg_daily_severity, na.rm = TRUE),
  sd = sd(avg_daily_severity, na.rm = TRUE),
  se_mean = sd / sqrt(n),
  min = min(avg_daily_severity, na.rm = TRUE),
  max = max(avg_daily_severity, na.rm = TRUE)
 )

cat("BLEEDING GROUP:\n")
cat("  n =", bleeding_summary$n, "\n")
cat("  Mean:", round(bleeding_summary$mean, 3), "\n")
cat("  SE of Mean:", round(bleeding_summary$se_mean, 3), "\n")
cat("  Median:", round(bleeding_summary$median, 3), "\n")
cat("  SD:", round(bleeding_summary$sd, 3), "\n")
cat("  Range:", round(bleeding_summary$min, 3), "to", round(bleeding_summary$max, 3), "\n\n")

cat("NON-BLEEDING GROUP:\n")
cat("  n =", nonbleeding_summary$n, "\n")
cat("  Mean:", round(nonbleeding_summary$mean, 3), "\n")
cat("  SE of Mean:", round(nonbleeding_summary$se_mean, 3), "\n")
cat("  Median:", round(nonbleeding_summary$median, 3), "\n")
cat("  SD:", round(nonbleeding_summary$sd, 3), "\n")
cat("  Range:", round(nonbleeding_summary$min, 3), "to", round(nonbleeding_summary$max, 3), "\n\n")

cat("Step 3: Average daily symptom severity calculation complete\n")

cat("\nStep 4: Creating boxplot visualization...\n")

combined_severity_data$group <- factor(combined_severity_data$group,
                        levels = c("Bleeding", "Non-Bleeding"))

bleeding_vs_nonbleeding_plot <- ggplot(combined_severity_data,
                        aes(x = group, y = avg_daily_severity, fill = group)) +
 geom_boxplot(alpha = 0.7, outlier.shape = NA, width = 0.6) +

 geom_jitter(width = 0.2, height = 0, alpha = 0.6, size = 2.5, color = "black") +

 annotate("text", x = 1, y = max(combined_severity_data$avg_daily_severity) * 0.95,
       label = paste("n =", bleeding_summary$n), size = 4, fontface = "bold") +
 annotate("text", x = 2, y = max(combined_severity_data$avg_daily_severity) * 0.95,
```

```
      label = paste("n =", nonbleeding_summary$n), size = 4, fontface = "bold") +

 scale_fill_manual(values = c("Bleeding" = "#FFB3B3", "Non-Bleeding" = "#ADD8E6")) +

 labs(
  title = "Average Daily Symptom Severity: Bleeding vs Non-Bleeding Participants",
  subtitle = "Comparison of participants who track menstruation vs those who don't",
  x = "Participant Group",
  y = "Average Daily Symptom Severity",
  caption = paste("Bleeding group: Mean =", round(bleeding_summary$mean, 2),
            ", Median =", round(bleeding_summary$median, 2),
            "\nNon-Bleeding group: Mean =", round(nonbleeding_summary$mean, 2),
            ", Median =", round(nonbleeding_summary$median, 2))
 ) +

 theme_minimal() +
 theme(
  plot.title = element_text(face = "bold", size = 14, hjust = 0.5),
  plot.subtitle = element_text(size = 12, hjust = 0.5),
  legend.position = "none",
  axis.text.x = element_text(size = 12, face = "bold"),
  axis.title = element_text(size = 12),
  plot.caption = element_text(size = 10, hjust = 0.5)
 )

print(bleeding_vs_nonbleeding_plot)

cat("Step 4: Boxplot visualization complete\n")

cat("\nStep 5: Performing statistical analysis...\n")

bleeding_values <- bleeding_severity_data$avg_daily_severity
nonbleeding_values <- nonbleeding_severity_data$avg_daily_severity

bleeding_values <- bleeding_values[!is.na(bleeding_values)]
nonbleeding_values <- nonbleeding_values[!is.na(nonbleeding_values)]

mann_whitney_result <- wilcox.test(
 bleeding_values,
 nonbleeding_values,
 alternative = "two.sided",
 paired = FALSE,
 exact = FALSE
)

welch_t_result <- t.test(
 bleeding_values,
 nonbleeding_values,
 alternative = "two.sided",
```

```
 paired = FALSE,
 var.equal = FALSE
)

pooled_sd <- sqrt(((length(bleeding_values) - 1) * var(bleeding_values) +
                   (length(nonbleeding_values) - 1) * var(nonbleeding_values)) /
                  (length(bleeding_values) + length(nonbleeding_values) - 2))

cohens_d <- (mean(bleeding_values) - mean(nonbleeding_values)) / pooled_sd

cat("\n=== STATISTICAL TEST RESULTS ===\n")

cat("\nMann-Whitney U Test (non-parametric):\n")
cat("  W statistic:", mann_whitney_result$statistic, "\n")
cat("  p-value:", format(mann_whitney_result$p.value, digits = 4), "\n")

if(mann_whitney_result$p.value < 0.001) {
  significance_mw <- "*** (highly significant)"
} else if(mann_whitney_result$p.value < 0.01) {
  significance_mw <- "** (very significant)"
} else if(mann_whitney_result$p.value < 0.05) {
  significance_mw <- "* (significant)"
} else {
  significance_mw <- "(not significant)"
}
cat("  Significance:", significance_mw, "\n")

cat("\nWelch's t-test (parametric):\n")
cat("  t statistic:", round(welch_t_result$statistic, 4), "\n")
cat("  df:", round(welch_t_result$parameter, 2), "\n")
cat("  p-value:", format(welch_t_result$p.value, digits = 4), "\n")

if(welch_t_result$p.value < 0.001) {
  significance_t <- "*** (highly significant)"
} else if(welch_t_result$p.value < 0.01) {
  significance_t <- "** (very significant)"
} else if(welch_t_result$p.value < 0.05) {
  significance_t <- "* (significant)"
} else {
  significance_t <- "(not significant)"
}
cat("  Significance:", significance_t, "\n")

cat("\nEffect Size:\n")
cat("  Cohen's d:", round(cohens_d, 4), "\n")

if(abs(cohens_d) < 0.2) {
  effect_interpretation <- "(negligible effect)"
} else if(abs(cohens_d) < 0.5) {
```

```
  effect_interpretation <- "(small effect)"
} else if(abs(cohens_d) < 0.8) {
  effect_interpretation <- "(medium effect)"
} else {
  effect_interpretation <- "(large effect)"
}
cat("  Interpretation:", effect_interpretation, "\n")

cat("\n=== INTERPRETATION ===\n")
if(mann_whitney_result$p.value < 0.05 || welch_t_result$p.value < 0.05) {
  direction <- ifelse(mean(bleeding_values) > mean(nonbleeding_values), "higher", "lower")
  cat("There IS a statistically significant difference between groups.\n")
  cat("Participants who track menstruation have", direction, "average daily symptom severity.\n")
} else {
  cat("There is NO statistically significant difference in average daily symptom severity\n")
  cat("between participants who track menstruation and those who don't.\n")
}

cat("\nBoth groups show similar symptom burden patterns, suggesting that menstrual\n")
cat("tracking behavior is not strongly associated with overall symptom severity.\n")

cat("\nStep 5: Statistical analysis complete\n")

bleeding_vs_nonbleeding_results <- list(
  data = combined_severity_data,
  plot = bleeding_vs_nonbleeding_plot,
  mann_whitney = mann_whitney_result,
  welch_t = welch_t_result,
  cohens_d = cohens_d,
  bleeding_summary = bleeding_summary,
  nonbleeding_summary = nonbleeding_summary
)

cat("\n=== SCRIPT 8 COMPLETE ===\n")
cat("Results stored in 'bleeding_vs_nonbleeding_results' for future reference\n")
```

Script 8.1

```
rm(list = ls(pattern = "^(motor_.*_bleeding|nonmotor_.*_bleeding|ms_.*_cohort|nms_.*_cohort)"))
rm(list = ls(pattern = "^(motor_vs_nonmotor|symptom_type_.*|ms_nms_.*_analysis)$"))
rm(list = ls(pattern = "^(bleeding_.*_motor|bleeding_.*_nonmotor|cohort_.*_breakdown)$"))
rm(list = ls(pattern = "^(analyze_.*_by_bleeding|compare_motor_nonmotor|breakdown_.*_cohorts)$"))
rm(list = ls(pattern = "^(ms_.*_plot|nms_.*_plot|motor_nonmotor_.*_chart)$"))
required_vars <- c("multi_cycle_merged_data", "single_cycle_merged_data", "symptom_categories")

missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 stop("Required variables from previous scripts missing: ", paste(missing_vars, collapse = ", "), "\n",
    "Please run Scripts 1-8 first.")
}

if(!exists("multi_cycle_participants") || !exists("single_cycle_participants")) {
 stop("Participant classification variables missing. Please run Scripts 1 and 2 first.")
}

required_packages <- c("ggplot2", "dplyr", "tidyr")
missing_packages <- c()

for(pkg in required_packages) {
 if(!require(pkg, character.only = TRUE, quietly = TRUE)) {
  missing_packages <- c(missing_packages, pkg)
 }
}

if(length(missing_packages) > 0) {
 stop("Required packages for motor/non-motor cohort analysis missing: ", paste(missing_packages,
collapse = ", "))
}

if(!"symptom_type" %in% colnames(symptom_categories)) {
 stop("Symptom categories missing symptom_type column. Cannot classify motor vs non-motor
symptoms.")
}

motor_symptoms_available <- sum(symptom_categories$symptom_type == "Motor", na.rm = TRUE)
nonmotor_symptoms_available <- sum(symptom_categories$symptom_type == "Non-Motor", na.rm =
TRUE)

if(motor_symptoms_available == 0 || nonmotor_symptoms_available == 0) {
 stop("Insufficient motor or non-motor symptoms in categories. Motor: ", motor_symptoms_available,
```

```
    ", Non-Motor: ", nonmotor_symptoms_available)
}

multi_symptom_type_count <- sum(multi_cycle_merged_data$`Symptom Type` != "No Entry", na.rm =
TRUE)
single_symptom_type_count <- sum(single_cycle_merged_data$`Symptom Type` != "No Entry", na.rm =
TRUE)

total_symptom_type_entries <- multi_symptom_type_count + single_symptom_type_count

if(total_symptom_type_entries == 0) {
 stop("No symptom type data found. Cannot perform motor vs non-motor analysis.")
}

motor_entries <- sum(c(multi_cycle_merged_data$`Symptom Type`,
single_cycle_merged_data$`Symptom Type`) == "Motor", na.rm = TRUE)
nonmotor_entries <- sum(c(multi_cycle_merged_data$`Symptom Type`,
single_cycle_merged_data$`Symptom Type`) == "Non-Motor", na.rm = TRUE)

if(motor_entries == 0 || nonmotor_entries == 0) {
 stop("Insufficient motor or non-motor symptom entries. Motor: ", motor_entries,
    ", Non-Motor: ", nonmotor_entries)
}

set.seed(12345)
cat(" 🎯 Random seed set for reproducible statistical tests with tied values\n\n")

cat("=== SCRIPT 8.1: MOTOR VS NON-MOTOR BETWEEN PERIOD TRACKING GROUPS ===\n\n")

cat("Step 1: Checking participant eligibility and symptom type availability...\n")

eligible_participants_8_1 <- c(final_bleeding_participants, final_nonbleeding_participants)

cat("Starting with", length(eligible_participants_8_1), "participants from Script 8\n")
cat("- Participants who tracked periods:", length(final_bleeding_participants), "\n")
cat("- Participants who did not track periods:", length(final_nonbleeding_participants), "\n")

check_motor_nonmotor_availability <- function(research_id) {
 if(research_id %in% final_bleeding_participants) {
  all_symptom_data <- bind_rows(
   multi_cycle_merged_data %>% filter(Symptom != "No Entry"),
   single_cycle_merged_data %>% filter(Symptom != "No Entry")
  )

  participant_symptoms <- all_symptom_data %>%
   filter(research_id == !!research_id)

 } else if(research_id %in% final_nonbleeding_participants) {
  participant_symptoms <- no_period_final %>%
```

```
    filter(research_id == !!research_id) %>%
    filter(Symptom != "No Entry")

  } else {
   return(list(has_motor = FALSE, has_nonmotor = FALSE, both_types = FALSE))
  }

  if(nrow(participant_symptoms) == 0) {
   return(list(has_motor = FALSE, has_nonmotor = FALSE, both_types = FALSE))
  }

  categorized_symptoms <- participant_symptoms %>%
   left_join(symptom_categories, by = c("Symptom" = "metric_name_abbr")) %>%
   filter(symptom_type %in% c("Motor", "Non-Motor"))

  has_motor <- any(categorized_symptoms$symptom_type == "Motor")
  has_nonmotor <- any(categorized_symptoms$symptom_type == "Non-Motor")
  both_types <- has_motor & has_nonmotor

  return(list(
   has_motor = has_motor,
   has_nonmotor = has_nonmotor,
   both_types = both_types
  ))
}

cat("Checking motor and non-motor symptom availability for all participants...\n")

symptom_availability_results <- map(eligible_participants_8_1, check_motor_nonmotor_availability)
names(symptom_availability_results) <- eligible_participants_8_1

availability_df <- map_dfr(symptom_availability_results, ~as.data.frame(.x), .id = "research_id")

cat("\n=== STEP 1 AVAILABILITY CHECK RESULTS ===\n")

summary_availability <- availability_df %>%
  summarise(
   total_participants = n(),
   participants_with_motor = sum(has_motor),
   participants_with_nonmotor = sum(has_nonmotor),
   participants_with_both = sum(both_types),
   participants_motor_only = sum(has_motor & !has_nonmotor),
   participants_nonmotor_only = sum(!has_motor & has_nonmotor),
   participants_with_neither = sum(!has_motor & !has_nonmotor)
  )

print(summary_availability)

final_participants_8_1 <- availability_df %>%
```

```
 filter(both_types == TRUE) %>%
 pull(research_id)

cat("\nFinal eligible participants (with both motor and non-motor symptoms):",
length(final_participants_8_1), "\n")

removed_participants <- setdiff(eligible_participants_8_1, final_participants_8_1)
if(length(removed_participants) > 0) {
 cat("\nParticipants removed and reasons:\n")

 removal_reasons <- availability_df %>%
  filter(research_id %in% removed_participants) %>%
  mutate(
   reason = case_when(
    !has_motor & !has_nonmotor ~ "No motor or non-motor symptoms",
    !has_motor ~ "No motor symptoms",
    !has_nonmotor ~ "No non-motor symptoms",
    TRUE ~ "Unknown reason"
   )
  ) %>%
  select(research_id, reason)

 print(removal_reasons)
} else {
 cat("No participants removed - all have both motor and non-motor symptoms ✓\n")
}

cat("\nStep 1: Participant eligibility check complete\n")

cat("\nStep 2: Calculating average motor symptom burden for each participant...\n")

calculate_avg_motor_severity <- function(research_id) {
 if(research_id %in% final_bleeding_participants) {
  all_symptom_data <- bind_rows(
   multi_cycle_merged_data %>% filter(Symptom != "No Entry"),
   single_cycle_merged_data %>% filter(Symptom != "No Entry")
  )

  participant_symptoms <- all_symptom_data %>%
   filter(research_id == !!research_id)

 } else if(research_id %in% final_nonbleeding_participants) {
  participant_symptoms <- no_period_final %>%
   filter(research_id == !!research_id) %>%
   filter(Symptom != "No Entry")

 } else {
  return(NA)
 }
```

```
 if(nrow(participant_symptoms) == 0) {
  return(NA)
 }

 motor_symptoms <- participant_symptoms %>%
  left_join(symptom_categories, by = c("Symptom" = "metric_name_abbr")) %>%
  filter(symptom_type == "Motor") %>%
  mutate(Severity = as.numeric(Severity))

 if(nrow(motor_symptoms) == 0) {
  return(NA)
 }

 total_motor_severity <- sum(motor_symptoms$Severity, na.rm = TRUE)
 unique_motor_symptom_days <- n_distinct(motor_symptoms$dates)

 avg_motor_severity <- total_motor_severity / unique_motor_symptom_days

 return(avg_motor_severity)
}

cat("Calculating motor symptom averages...\n")
motor_averages <- sapply(final_participants_8_1, calculate_avg_motor_severity)

motor_results <- data.frame(
 research_id = final_participants_8_1,
 avg_severity_motor = motor_averages,
 stringsAsFactors = FALSE
)

motor_results <- motor_results %>%
 filter(!is.na(avg_severity_motor))

cat("Motor symptom averages calculated for", nrow(motor_results), "participants\n")
cat("Motor burden range:", round(min(motor_results$avg_severity_motor), 3), "to",
round(max(motor_results$avg_severity_motor), 3), "\n")

cat("\nStep 2: Motor symptom burden calculation complete\n")

cat("\nStep 3: Calculating average non-motor symptom burden for each participant...\n")

calculate_avg_nonmotor_severity <- function(research_id) {
 if(research_id %in% final_bleeding_participants) {
  all_symptom_data <- bind_rows(
   multi_cycle_merged_data %>% filter(Symptom != "No Entry"),
   single_cycle_merged_data %>% filter(Symptom != "No Entry")
  )
```

```
   participant_symptoms <- all_symptom_data %>%
    filter(research_id == !!research_id)

  } else if(research_id %in% final_nonbleeding_participants) {
   participant_symptoms <- no_period_final %>%
    filter(research_id == !!research_id) %>%
    filter(Symptom != "No Entry")

  } else {
   return(NA)
  }

  if(nrow(participant_symptoms) == 0) {
   return(NA)
  }

  nonmotor_symptoms <- participant_symptoms %>%
   left_join(symptom_categories, by = c("Symptom" = "metric_name_abbr")) %>%
   filter(symptom_type == "Non-Motor") %>%
   mutate(Severity = as.numeric(Severity))

  if(nrow(nonmotor_symptoms) == 0) {
   return(NA)
  }

  total_nonmotor_severity <- sum(nonmotor_symptoms$Severity, na.rm = TRUE)
  unique_nonmotor_symptom_days <- n_distinct(nonmotor_symptoms$dates)

  avg_nonmotor_severity <- total_nonmotor_severity / unique_nonmotor_symptom_days

  return(avg_nonmotor_severity)
}

cat("Calculating non-motor symptom averages...\n")
nonmotor_averages <- sapply(final_participants_8_1, calculate_avg_nonmotor_severity)

nonmotor_results <- data.frame(
  research_id = final_participants_8_1,
  avg_severity_nonmotor = nonmotor_averages,
  stringsAsFactors = FALSE
)

nonmotor_results <- nonmotor_results %>%
  filter(!is.na(avg_severity_nonmotor))

cat("Non-motor symptom averages calculated for", nrow(nonmotor_results), "participants\n")
cat("Non-motor burden range:", round(min(nonmotor_results$avg_severity_nonmotor), 3), "to",
round(max(nonmotor_results$avg_severity_nonmotor), 3), "\n")
```

```
cat("\nStep 3: Non-motor symptom burden calculation complete\n")

cat("\nStep 4: Combining motor and non-motor averages...\n")

participant_averages_wide <- motor_results %>%
 inner_join(nonmotor_results, by = "research_id")

cat("Combined averages for", nrow(participant_averages_wide), "participants\n")

cat("\nExample participant averages (first 5):\n")
print(participant_averages_wide %>%
     head(5) %>%
     mutate(across(where(is.numeric), ~ round(.x, 3))))

summary_stats_combined <- participant_averages_wide %>%
 summarise(
  n = n(),
  mean_motor = mean(avg_severity_motor, na.rm = TRUE),
  median_motor = median(avg_severity_motor, na.rm = TRUE),
  sd_motor = sd(avg_severity_motor, na.rm = TRUE),
  mean_nonmotor = mean(avg_severity_nonmotor, na.rm = TRUE),
  median_nonmotor = median(avg_severity_nonmotor, na.rm = TRUE),
  sd_nonmotor = sd(avg_severity_nonmotor, na.rm = TRUE)
 )

cat("\n=== OVERALL SUMMARY STATISTICS ===\n")
cat("Total participants:", summary_stats_combined$n, "\n")
cat("Motor symptoms - Mean:", round(summary_stats_combined$mean_motor, 3), ", Median:",
round(summary_stats_combined$median_motor, 3), ", SD:",
round(summary_stats_combined$sd_motor, 3), "\n")
cat("Non-motor symptoms - Mean:", round(summary_stats_combined$mean_nonmotor, 3), ", Median:",
round(summary_stats_combined$median_nonmotor, 3), ", SD:",
round(summary_stats_combined$sd_nonmotor, 3), "\n")

cat("\nStep 4: Data combination complete\n")

cat("\nStep 5: Splitting participants by period tracking status and performing comparisons...\n")

bleeding_participants_8_1 <- intersect(final_participants_8_1, final_bleeding_participants)
nonbleeding_participants_8_1 <- intersect(final_participants_8_1, final_nonbleeding_participants)

cat("Splitting", length(final_participants_8_1), "eligible participants by period tracking status:\n")
cat("- Participants who tracked periods (with both motor & non-motor symptoms):",
length(bleeding_participants_8_1), "\n")
cat("- Participants who did not track periods (with both motor & non-motor symptoms):",
length(nonbleeding_participants_8_1), "\n")
cat("- Total check:", length(bleeding_participants_8_1) + length(nonbleeding_participants_8_1), "\n")
```

```
if(length(bleeding_participants_8_1) + length(nonbleeding_participants_8_1) !=
length(final_participants_8_1)) {
 cat("WARNING: Split doesn't add up to total!\n")
} else {
 cat("✓ Split verified: All participants accounted for\n")
}

bleeding_averages <- participant_averages_wide %>%
 filter(research_id %in% bleeding_participants_8_1) %>%
 mutate(group = "Tracked Periods")

nonbleeding_averages <- participant_averages_wide %>%
 filter(research_id %in% nonbleeding_participants_8_1) %>%
 mutate(group = "Did Not Track Periods")

combined_averages <- bind_rows(bleeding_averages, nonbleeding_averages)

summary_by_group <- combined_averages %>%
 group_by(group) %>%
 summarise(
  n = n(),
  mean_motor = mean(avg_severity_motor, na.rm = TRUE),
  median_motor = median(avg_severity_motor, na.rm = TRUE),
  sd_motor = sd(avg_severity_motor, na.rm = TRUE),
  mean_nonmotor = mean(avg_severity_nonmotor, na.rm = TRUE),
  median_nonmotor = median(avg_severity_nonmotor, na.rm = TRUE),
  sd_nonmotor = sd(avg_severity_nonmotor, na.rm = TRUE),
  .groups = 'drop'
 )

cat("\n=== SUMMARY STATISTICS BY GROUP ===\n")
print(summary_by_group)

cat("\nStep 5: Group splitting complete\n")

cat("\nStep 6: Performing between-group statistical comparisons...\n")

bleeding_motor <- bleeding_averages$avg_severity_motor
bleeding_nonmotor <- bleeding_averages$avg_severity_nonmotor

nonbleeding_motor <- nonbleeding_averages$avg_severity_motor
nonbleeding_nonmotor <- nonbleeding_averages$avg_severity_nonmotor

cat("Sample sizes for statistical testing:\n")
cat("- Tracked periods group: n =", length(bleeding_motor), "\n")
cat("- Did not track periods group: n =", length(nonbleeding_motor), "\n")

cat("\n=== COMPARISON 1: MOTOR SYMPTOMS (Tracked Periods vs Did Not Track Periods) ===\n")
```

```
motor_t_test <- t.test(
  bleeding_motor,
  nonbleeding_motor,
  alternative = "two.sided",
  paired = FALSE,
  var.equal = FALSE
)

motor_wilcox_test <- wilcox.test(
  bleeding_motor,
  nonbleeding_motor,
  alternative = "two.sided",
  paired = FALSE,
  exact = FALSE
)

cat("Descriptive statistics:\n")
cat("Tracked periods group - Motor: Mean =", round(mean(bleeding_motor), 3), ", Median =",
round(median(bleeding_motor), 3), "\n")
cat("Did not track periods group - Motor: Mean =", round(mean(nonbleeding_motor), 3), ", Median =",
round(median(nonbleeding_motor), 3), "\n")
cat("Difference in means:", round(mean(bleeding_motor) - mean(nonbleeding_motor), 3), "\n")

cat("\nWelch's t-test results:\n")
cat("t =", round(motor_t_test$statistic, 3), ", df =", round(motor_t_test$parameter, 1), ", p =",
format(motor_t_test$p.value, digits = 4), "\n")

cat("Mann-Whitney U test results:\n")
cat("W =", motor_wilcox_test$statistic, ", p =", format(motor_wilcox_test$p.value, digits = 4), "\n")

cat("\n=== COMPARISON 2: NON-MOTOR SYMPTOMS (Tracked Periods vs Did Not Track Periods) ===\n")

nonmotor_t_test <- t.test(
  bleeding_nonmotor,
  nonbleeding_nonmotor,
  alternative = "two.sided",
  paired = FALSE,
  var.equal = FALSE
)

nonmotor_wilcox_test <- wilcox.test(
  bleeding_nonmotor,
  nonbleeding_nonmotor,
  alternative = "two.sided",
  paired = FALSE,
  exact = FALSE
)

cat("Descriptive statistics:\n")
```

```
cat("Tracked periods group - Non-motor: Mean =", round(mean(bleeding_nonmotor), 3), ", Median =",
round(median(bleeding_nonmotor), 3), "\n")
cat("Did not track periods group - Non-motor: Mean =", round(mean(nonbleeding_nonmotor), 3), ",
Median =", round(median(nonbleeding_nonmotor), 3), "\n")
cat("Difference in means:", round(mean(bleeding_nonmotor) - mean(nonbleeding_nonmotor), 3), "\n")

cat("\nWelch's t-test results:\n")
cat("t =", round(nonmotor_t_test$statistic, 3), ", df =", round(nonmotor_t_test$parameter, 1), ", p =",
format(nonmotor_t_test$p.value, digits = 4), "\n")

cat("Mann-Whitney U test results:\n")
cat("W =", nonmotor_wilcox_test$statistic, ", p =", format(nonmotor_wilcox_test$p.value, digits = 4), "\n")

pooled_sd_motor <- sqrt(((length(bleeding_motor) - 1) * var(bleeding_motor) +
                         (length(nonbleeding_motor) - 1) * var(nonbleeding_motor)) /
                        (length(bleeding_motor) + length(nonbleeding_motor) - 2))

cohens_d_motor <- (mean(bleeding_motor) - mean(nonbleeding_motor)) / pooled_sd_motor

pooled_sd_nonmotor <- sqrt(((length(bleeding_nonmotor) - 1) * var(bleeding_nonmotor) +
                            (length(nonbleeding_nonmotor) - 1) * var(nonbleeding_nonmotor)) /
                           (length(bleeding_nonmotor) + length(nonbleeding_nonmotor) - 2))

cohens_d_nonmotor <- (mean(bleeding_nonmotor) - mean(nonbleeding_nonmotor)) /
pooled_sd_nonmotor

cat("\n=== EFFECT SIZES ===\n")
cat("Motor symptoms - Cohen's d:", round(cohens_d_motor, 3), "\n")
cat("Non-motor symptoms - Cohen's d:", round(cohens_d_nonmotor, 3), "\n")

cat("\n=== SIGNIFICANCE SUMMARY ===\n")

motor_sig <- ifelse(motor_t_test$p.value < 0.05, "SIGNIFICANT", "NOT SIGNIFICANT")
nonmotor_sig <- ifelse(nonmotor_t_test$p.value < 0.05, "SIGNIFICANT", "NOT SIGNIFICANT")

cat("Motor symptoms comparison:", motor_sig, "(p =", format(motor_t_test$p.value, digits = 3), ")\n")
cat("Non-motor symptoms comparison:", nonmotor_sig, "(p =", format(nonmotor_t_test$p.value, digits =
3), ")\n")

cat("\nStep 6: Between-group statistical comparisons complete\n")

cat("\nStep 7: Creating side-by-side boxplot visualization...\n")

plot_data_motor <- data.frame(
  group = rep(c("Tracked Periods", "Did Not Track Periods"), c(length(bleeding_motor),
length(nonbleeding_motor))),
  severity = c(bleeding_motor, nonbleeding_motor),
  symptom_type = "Motor Symptoms"
)
```

```
plot_data_nonmotor <- data.frame(
 group = rep(c("Tracked Periods", "Did Not Track Periods"), c(length(bleeding_nonmotor),
length(nonbleeding_nonmotor))),
 severity = c(bleeding_nonmotor, nonbleeding_nonmotor),
 symptom_type = "Non-Motor Symptoms"
)

plot_data_combined <- bind_rows(plot_data_motor, plot_data_nonmotor)

plot_data_combined$group <- factor(plot_data_combined$group, levels = c("Tracked Periods", "Did Not
Track Periods"))
plot_data_combined$symptom_type <- factor(plot_data_combined$symptom_type, levels = c("Motor
Symptoms", "Non-Motor Symptoms"))

summary_stats_combined <- plot_data_combined %>%
 group_by(symptom_type, group) %>%
 summarise(
  n = n(),
  mean_severity = mean(severity, na.rm = TRUE),
  se_mean_severity = sd(severity, na.rm = TRUE) / sqrt(n()),
  median_severity = median(severity, na.rm = TRUE),
  sd_severity = sd(severity, na.rm = TRUE),
  .groups = 'drop'
 )

cat("Summary statistics for visualization:\n")
print(summary_stats_combined)

comparison_plot <- ggplot(plot_data_combined, aes(x = group, y = severity, fill = group)) +
 facet_wrap(~ symptom_type, scales = "free_y") +

 geom_boxplot(alpha = 0.7, outlier.shape = NA, width = 0.6) +

 geom_jitter(width = 0.2, height = 0, alpha = 0.6, size = 2.5, color = "black") +

 geom_point(
  data = summary_stats_combined,
  aes(x = group, y = mean_severity),
  shape = 18, size = 5, color = "black"
 ) +

 geom_text(
  data = summary_stats_combined,
  aes(x = group, y = max(plot_data_combined$severity) * 0.95, label = paste("n =", n)),
  size = 4, fontface = "bold"
 ) +

 scale_fill_manual(values = c("Tracked Periods" = "#FFB3B3", "Did Not Track Periods" = "#ADD8E6")) +
```

```
  labs(
    title = "Motor vs Non-Motor Symptom Burden by Period Tracking Status",
    subtitle = "Between-group comparison of average daily symptom burden",
    x = "Participant Group",
    y = "Average Daily Symptom Burden",
    caption = paste0("Motor symptoms: Tracked vs Did Not Track Periods, p = ",
round(motor_wilcox_test$p.value, 3),
                     "\nNon-motor symptoms: Tracked vs Did Not Track Periods, p = ",
round(nonmotor_wilcox_test$p.value, 3),
                     "\nBlack diamonds = group means")
  ) +

  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 14, hjust = 0.5),
    plot.subtitle = element_text(size = 12, hjust = 0.5),
    plot.caption = element_text(size = 10, hjust = 0.5),
    legend.position = "none",
    axis.text.x = element_text(size = 12, face = "bold"),
    axis.title = element_text(size = 12),
    strip.text = element_text(size = 12, face = "bold")
  )

print(comparison_plot)

bleeding_vs_nonbleeding_motor_nonmotor_results <- list(
  final_participants = final_participants_8_1,
  participant_averages = participant_averages_wide,
  bleeding_averages = bleeding_averages,
  nonbleeding_averages = nonbleeding_averages,
  plot_data = plot_data_combined,
  plot = comparison_plot,
  motor_t_test = motor_t_test,
  motor_wilcox_test = motor_wilcox_test,
  nonmotor_t_test = nonmotor_t_test,
  nonmotor_wilcox_test = nonmotor_wilcox_test,
  cohens_d_motor = cohens_d_motor,
  cohens_d_nonmotor = cohens_d_nonmotor,
  summary_stats = summary_stats_combined
)
```

Script 8.2

```
rm(list = ls(pattern = "^(bleeding_days_|non_bleeding_days_|daily_.*_comparison|day_.*_analysis)"))
rm(list = ls(pattern =
"^(within_participant_|paired_.*|participant_.*_comparison|individual_.*_analysis)$"))
rm(list = ls(pattern = "^(daily_burden|day_severity|burden_.*_days|severity_.*_days)$"))
rm(list = ls(pattern = "^(analyze_.*_days|compare_bleeding_days|calculate_daily_burden)$"))
rm(list = ls(pattern = "^(paired_.*_test|wilcox_.*_result|day_.*_stats|comparison_.*_results)$"))
rm(list = ls(pattern = "^(days_.*_plot|bleeding_.*_chart|daily_.*_visualization)$"))
required_vars <- c("multi_cycle_merged_data", "single_cycle_merged_data", "symptom_categories")
missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 stop("Required variables from previous scripts missing: ", paste(missing_vars, collapse = ", "), "\n",
    "Please run Scripts 1-8.1 first.")
}

if(!exists("multi_cycle_participants") || !exists("single_cycle_participants")) {
 stop("Participant classification variables missing. Please run Scripts 1 and 2 first.")
}

required_packages <- c("ggplot2", "dplyr", "tidyr", "lubridate")
missing_packages <- c()

for(pkg in required_packages) {
 if(!require(pkg, character.only = TRUE, quietly = TRUE)) {
  missing_packages <- c(missing_packages, pkg)
 }
}

if(length(missing_packages) > 0) {
 stop("Required packages for daily bleeding analysis missing: ", paste(missing_packages, collapse = ", "))
}

multi_flow_data <- multi_cycle_merged_data[multi_cycle_merged_data$Flow != "No Entry", ]
single_flow_data <- single_cycle_merged_data[single_cycle_merged_data$Flow != "No Entry", ]

bleeding_days <- nrow(multi_flow_data) + nrow(single_flow_data)
non_bleeding_days <- sum(multi_cycle_merged_data$Flow == "No Entry", na.rm = TRUE) +
 sum(single_cycle_merged_data$Flow == "No Entry", na.rm = TRUE)

if(bleeding_days == 0) {
 stop("No bleeding days found. Cannot perform bleeding vs non-bleeding days analysis.")
}
```

```
if(non_bleeding_days == 0) {
  stop("No non-bleeding days found. Cannot perform bleeding vs non-bleeding days analysis.")
}

participants_with_bleeding <- unique(c(
  multi_cycle_merged_data$research_id[multi_cycle_merged_data$Flow != "No Entry"],
  single_cycle_merged_data$research_id[single_cycle_merged_data$Flow != "No Entry"]
))

participants_with_non_bleeding <- unique(c(
  multi_cycle_merged_data$research_id[multi_cycle_merged_data$Flow == "No Entry"],
  single_cycle_merged_data$research_id[single_cycle_merged_data$Flow == "No Entry"]
))

participants_with_both_day_types <- intersect(participants_with_bleeding,
participants_with_non_bleeding)

if(length(participants_with_both_day_types) < 3) {
  warning("Very few participants (", length(participants_with_both_day_types),
          ") have both bleeding and non-bleeding days. Within-participant analysis may be limited.")
}

symptom_data_bleeding_days <- sum(c(
  multi_cycle_merged_data$Symptom[multi_cycle_merged_data$Flow != "No Entry"],
  single_cycle_merged_data$Symptom[single_cycle_merged_data$Flow != "No Entry"]
) != "No Entry", na.rm = TRUE)

symptom_data_non_bleeding_days <- sum(c(
  multi_cycle_merged_data$Symptom[multi_cycle_merged_data$Flow == "No Entry"],
  single_cycle_merged_data$Symptom[single_cycle_merged_data$Flow == "No Entry"]
) != "No Entry", na.rm = TRUE)

if(symptom_data_bleeding_days == 0 || symptom_data_non_bleeding_days == 0) {
  stop("Insufficient symptom data on bleeding or non-bleeding days. Bleeding days: ",
       symptom_data_bleeding_days, ", Non-bleeding days: ", symptom_data_non_bleeding_days)
}

cat("=== SCRIPT 8.2: MENSTRUATING VS NON-MENSTRUATING DAYS ANALYSIS ===\n\n")

cat("Step 1: Further filtering bleeding participants...\n")

check_consecutive_nonperiod_symptom_days <- function(research_id, min_days = 3) {
  period_dates <- mini_period_data %>%
    filter(research_id == !!research_id) %>%
    pull(`Period Entry Date`)

  if(research_id %in% multi_cycle_participants$research_id) {
    symptom_dates <- multi_cycle_merged_data %>%
      filter(research_id == !!research_id, Symptom != "No Entry") %>%
```

```
    distinct(dates) %>%
    pull(dates)
  } else if(research_id %in% single_cycle_participants$research_id) {
   symptom_dates <- single_cycle_merged_data %>%
    filter(research_id == !!research_id, Symptom != "No Entry") %>%
    distinct(dates) %>%
    pull(dates)
  } else {
   return(FALSE)
  }

  nonperiod_symptom_dates <- symptom_dates[!symptom_dates %in% period_dates]

  if(length(nonperiod_symptom_dates) < min_days) {
   return(FALSE)
  }

  sorted_dates <- sort(nonperiod_symptom_dates)
  date_diffs <- as.numeric(diff(sorted_dates))

  consecutive_count <- 1
  max_consecutive <- 1

  for(i in 1:length(date_diffs)) {
   if(date_diffs[i] == 1) {
    consecutive_count <- consecutive_count + 1
    max_consecutive <- max(max_consecutive, consecutive_count)
   } else {
    consecutive_count <- 1
   }
  }

  return(max_consecutive >= min_days)
}

cat("Starting with", length(final_bleeding_participants), "bleeding participants from Script 8\n")

nonperiod_consecutive_check <- sapply(final_bleeding_participants,
check_consecutive_nonperiod_symptom_days)

eligible_bleeding_participants_8_1 <- final_bleeding_participants[nonperiod_consecutive_check]

cat("Participants with 3+ consecutive non-period symptom days:",
length(eligible_bleeding_participants_8_1), "\n")

cat("\n=== STEP 1 FILTERING RESULTS ===\n")
cat("Started with:", length(final_bleeding_participants), "bleeding participants\n")
cat("After additional filter:", length(eligible_bleeding_participants_8_1), "participants\n")
```

```
cat("Participants removed:", length(final_bleeding_participants) -
length(eligible_bleeding_participants_8_1), "\n")

removed_participants <- setdiff(final_bleeding_participants, eligible_bleeding_participants_8_1)
if(length(removed_participants) > 0) {
 cat("\nParticipants removed (insufficient consecutive non-period symptom days):\n")
 cat(paste(removed_participants, collapse = ", "), "\n")
}

cat("\nStep 1: Additional filtering complete\n")

cat("\nStep 2: Classifying menstruating vs non-menstruating days for each participant...\n")

classify_participant_days <- function(research_id) {
 period_dates <- mini_period_data %>%
  filter(research_id == !!research_id) %>%
  pull(`Period Entry Date`)

 if(research_id %in% multi_cycle_participants$research_id) {
  participant_symptoms <- multi_cycle_merged_data %>%
   filter(research_id == !!research_id, Symptom != "No Entry") %>%
   mutate(Severity = as.numeric(Severity))
 } else if(research_id %in% single_cycle_participants$research_id) {
  participant_symptoms <- single_cycle_merged_data %>%
   filter(research_id == !!research_id, Symptom != "No Entry") %>%
   mutate(Severity = as.numeric(Severity))
 } else {
  return(NULL)
 }

 participant_symptoms <- participant_symptoms %>%
  mutate(
   day_type = ifelse(dates %in% period_dates, "Menstruating", "Non-Menstruating")
  )

 return(participant_symptoms)
}

cat("Classifying days for", length(eligible_bleeding_participants_8_1), "participants...\n")

all_classified_data <- map_dfr(eligible_bleeding_participants_8_1, classify_participant_days)

cat("\n=== STEP 2 CLASSIFICATION RESULTS ===\n")

day_type_summary <- all_classified_data %>%
 group_by(day_type) %>%
 summarise(
  total_symptom_entries = n(),
  unique_participant_days = n_distinct(paste(research_id, dates)),
```

```
  unique_participants = n_distinct(research_id),
  .groups = 'drop'
 )

print(day_type_summary)

participant_day_type_check <- all_classified_data %>%
 group_by(research_id) %>%
 summarise(
  has_menstruating_days = any(day_type == "Menstruating"),
  has_nonmenstruating_days = any(day_type == "Non-Menstruating"),
  both_types = has_menstruating_days & has_nonmenstruating_days,
  menstruating_days_count = length(unique(dates[day_type == "Menstruating"])),
  nonmenstruating_days_count = length(unique(dates[day_type == "Non-Menstruating"])),
  .groups = 'drop'
 )

participants_with_both_types <- participant_day_type_check %>%
 filter(both_types == TRUE)

cat("\nParticipants with data for BOTH day types:", nrow(participants_with_both_types), "\n")

participants_missing_type <- participant_day_type_check %>%
 filter(both_types == FALSE)

if(nrow(participants_missing_type) > 0) {
 cat("\nParticipants missing one day type (will be excluded):\n")
 print(participants_missing_type %>% select(research_id, has_menstruating_days,
has_nonmenstruating_days))
} else {
 cat("All participants have both menstruating and non-menstruating symptom days ✓\n")
}

cat("\nSummary for participants with both day types:\n")
summary_stats <- participants_with_both_types %>%
 summarise(
  total_participants = n(),
  avg_menstruating_days = mean(menstruating_days_count),
  avg_nonmenstruating_days = mean(nonmenstruating_days_count),
  min_menstruating_days = min(menstruating_days_count),
  max_menstruating_days = max(menstruating_days_count),
  min_nonmenstruating_days = min(nonmenstruating_days_count),
  max_nonmenstruating_days = max(nonmenstruating_days_count)
 )

print(summary_stats)

cat("\nStep 2: Day classification complete\n")
```

```
cat("\nStep 3: Calculating per-participant average daily symptom burden...\n")

calculate_participant_averages <- function(research_id) {
 participant_data <- all_classified_data %>%
  filter(research_id == !!research_id)

 averages <- participant_data %>%
  group_by(day_type) %>%
  summarise(
   total_severity = sum(Severity, na.rm = TRUE),
   unique_days = n_distinct(dates),
   avg_daily_severity = total_severity / unique_days,
   total_symptom_entries = n(),
   .groups = 'drop'
  ) %>%
  mutate(research_id = research_id)

 return(averages)
}

cat("Calculating averages for all", length(eligible_bleeding_participants_8_1), "participants...\n")

participant_averages_list <- map(eligible_bleeding_participants_8_1, calculate_participant_averages)
participant_averages <- bind_rows(participant_averages_list)

participant_averages_wide <- participant_averages %>%
 select(research_id, day_type, avg_daily_severity) %>%
 pivot_wider(
  names_from = day_type,
  values_from = avg_daily_severity,
  names_prefix = "avg_severity_"
 ) %>%
 rename(
  avg_severity_menstruating = `avg_severity_Menstruating`,
  avg_severity_nonmenstruating = `avg_severity_Non-Menstruating`
 ) %>%
 mutate(
  difference = avg_severity_menstruating - avg_severity_nonmenstruating
 )

cat("\n=== STEP 3 CALCULATION RESULTS ===\n")

cat("Participants with averages calculated:", nrow(participant_averages_wide), "\n")

summary_by_day_type <- participant_averages %>%
 group_by(day_type) %>%
 summarise(
  n_participants = n(),
  mean_avg_severity = mean(avg_daily_severity, na.rm = TRUE),
```

```
  median_avg_severity = median(avg_daily_severity, na.rm = TRUE),
  sd_avg_severity = sd(avg_daily_severity, na.rm = TRUE),
  min_avg_severity = min(avg_daily_severity, na.rm = TRUE),
  max_avg_severity = max(avg_daily_severity, na.rm = TRUE),
  .groups = 'drop'
 )

cat("\nSummary statistics by day type:\n")
print(summary_by_day_type)

difference_summary <- participant_averages_wide %>%
 summarise(
  n_participants = n(),
  mean_difference = mean(difference, na.rm = TRUE),
  median_difference = median(difference, na.rm = TRUE),
  sd_difference = sd(difference, na.rm = TRUE),
  min_difference = min(difference, na.rm = TRUE),
  max_difference = max(difference, na.rm = TRUE),
  participants_higher_menstruating = sum(difference > 0, na.rm = TRUE),
  participants_higher_nonmenstruating = sum(difference < 0, na.rm = TRUE),
  participants_equal = sum(difference == 0, na.rm = TRUE)
 )

cat("\nDifference summary (Menstruating - Non-Menstruating):\n")
print(difference_summary)

cat("\nExample participant averages (first 5):\n")
print(participant_averages_wide %>%
    head(5) %>%
    mutate(across(where(is.numeric), ~ round(.x, 3))))

cat("\nStep 3: Per-participant average calculation complete\n")

cat("\nStep 4: Performing statistical analysis...\n")

menstruating_values <- participant_averages_wide$avg_severity_menstruating
nonmenstruating_values <- participant_averages_wide$avg_severity_nonmenstruating

complete_cases <- !is.na(menstruating_values) & !is.na(nonmenstruating_values)
menstruating_clean <- menstruating_values[complete_cases]
nonmenstruating_clean <- nonmenstruating_values[complete_cases]

cat("Complete paired observations:", length(menstruating_clean), "\n")

paired_t_result <- t.test(
 menstruating_clean,
 nonmenstruating_clean,
 paired = TRUE,
 alternative = "two.sided"
```

```
)

paired_wilcox_result <- wilcox.test(
  menstruating_clean,
  nonmenstruating_clean,
  paired = TRUE,
  alternative = "two.sided",
  exact = FALSE
)

differences <- menstruating_clean - nonmenstruating_clean
mean_difference <- mean(differences)
sd_difference <- sd(differences)
cohens_d_paired <- mean_difference / sd_difference

n <- length(differences)
se_difference <- sd_difference / sqrt(n)
t_critical <- qt(0.975, df = n - 1)
ci_lower <- mean_difference - (t_critical * se_difference)
ci_upper <- mean_difference + (t_critical * se_difference)

cat("\n=== STEP 4 STATISTICAL ANALYSIS RESULTS ===\n")

cat("\nDescriptive Statistics:\n")
cat("Mean difference (Menstruating - Non-Menstruating):", round(mean_difference, 4), "\n")
cat("Standard deviation of differences:", round(sd_difference, 4), "\n")
cat("95% CI for mean difference: [", round(ci_lower, 4), ",", round(ci_upper, 4), "]\n")

cat("\nPaired t-test:\n")
cat("t-statistic:", round(paired_t_result$statistic, 4), "\n")
cat("df:", paired_t_result$parameter, "\n")
cat("p-value:", format(paired_t_result$p.value, digits = 4), "\n")

if(paired_t_result$p.value < 0.001) {
  significance_t <- "*** (highly significant)"
} else if(paired_t_result$p.value < 0.01) {
  significance_t <- "** (very significant)"
} else if(paired_t_result$p.value < 0.05) {
  significance_t <- "* (significant)"
} else {
  significance_t <- "(not significant)"
}
cat("Significance:", significance_t, "\n")

cat("\nPaired Wilcoxon signed-rank test:\n")
cat("V statistic:", paired_wilcox_result$statistic, "\n")
cat("p-value:", format(paired_wilcox_result$p.value, digits = 4), "\n")

if(paired_wilcox_result$p.value < 0.001) {
```

```
 significance_w <- "*** (highly significant)"
} else if(paired_wilcox_result$p.value < 0.01) {
 significance_w <- "** (very significant)"
} else if(paired_wilcox_result$p.value < 0.05) {
 significance_w <- "* (significant)"
} else {
 significance_w <- "(not significant)"
}
cat("Significance:", significance_w, "\n")

cat("\nEffect Size:\n")
cat("Cohen's d (paired):", round(cohens_d_paired, 4), "\n")

if(abs(cohens_d_paired) < 0.2) {
 effect_interpretation <- "(negligible effect)"
} else if(abs(cohens_d_paired) < 0.5) {
 effect_interpretation <- "(small effect)"
} else if(abs(cohens_d_paired) < 0.8) {
 effect_interpretation <- "(medium effect)"
} else {
 effect_interpretation <- "(large effect)"
}
cat("Interpretation:", effect_interpretation, "\n")

cat("\nDirection of differences:\n")
cat("Participants with higher severity during menstruation:", sum(differences > 0), "\n")
cat("Participants with higher severity during non-menstruation:", sum(differences < 0), "\n")
cat("Participants with equal severity:", sum(differences == 0), "\n")

cat("\nStep 4: Statistical analysis complete\n")

cat("\nStep 5: Creating paired boxplot visualization...\n")

plot_data <- participant_averages_wide %>%
 select(research_id, avg_severity_menstruating, avg_severity_nonmenstruating) %>%
 pivot_longer(
  cols = c(avg_severity_menstruating, avg_severity_nonmenstruating),
  names_to = "day_type",
  values_to = "avg_severity"
 ) %>%
 mutate(
  day_type = case_when(
   day_type == "avg_severity_menstruating" ~ "Menstruating",
   day_type == "avg_severity_nonmenstruating" ~ "Non-Menstruating",
   TRUE ~ day_type
  ),
  day_type = factor(day_type, levels = c("Menstruating", "Non-Menstruating"))
 )
```

```
summary_stats_plot <- plot_data %>%
 group_by(day_type) %>%
 summarise(
  n = n(),
  mean_severity = mean(avg_severity, na.rm = TRUE),
  median_severity = median(avg_severity, na.rm = TRUE),
  .groups = 'drop'
 )

menstruating_vs_nonmenstruating_plot <- ggplot(plot_data, aes(x = day_type, y = avg_severity, fill =
day_type)) +
 geom_boxplot(alpha = 0.7, outlier.shape = NA, width = 0.6) +
 geom_line(aes(group = research_id), alpha = 0.3, color = "gray50") +
 geom_jitter(width = 0.2, height = 0, alpha = 0.6, size = 2, color = "black") +
 scale_fill_manual(values = c("Menstruating" = "#FFB3B3", "Non-Menstruating" = "#ADD8E6")) +
 labs(
  title = "Average Daily Symptom Burden: Menstruating vs Non-Menstruating Days",
  subtitle = "Period-tracking cohort: Symptom burden on bleeding vs non-bleeding days",
  x = "Day Type",
  y = "Average Daily Symptom Burden",
  caption = paste0("Paired t-test: p = ", round(paired_t_result$p.value, 3),
            " | n = ", nrow(participant_averages_wide))
 ) +
 theme_minimal() +
 theme(
  plot.title = element_text(face = "bold", size = 12, hjust = 0.5),
  plot.subtitle = element_text(size = 11, hjust = 0.5),
  legend.position = "none",
  axis.text.x = element_text(size = 10),
  axis.title = element_text(size = 12)
 )

print(menstruating_vs_nonmenstruating_plot)

menstruating_vs_nonmenstruating_results <- list(
 participant_data = participant_averages_wide,
 plot_data = plot_data,
 plot = menstruating_vs_nonmenstruating_plot,
 paired_t_test = paired_t_result,
 paired_wilcox_test = paired_wilcox_result,
 cohens_d = cohens_d_paired,
 summary_stats = summary_stats_plot,
 difference_summary = difference_summary
)

check_motor_both_day_types <- function(research_id) {
 participant_data <- all_classified_data %>%
  filter(research_id == !!research_id)
```

```
 if(nrow(participant_data) == 0) {
  return(FALSE)
 }

 categorized_data <- participant_data %>%
  left_join(symptom_categories, by = c("Symptom" = "metric_name_abbr")) %>%
  filter(symptom_type == "Motor")

 motor_day_types <- categorized_data %>%
  distinct(day_type) %>%
  pull(day_type)

 has_motor_both <- all(c("Menstruating", "Non-Menstruating") %in% motor_day_types)

 return(has_motor_both)
}

check_nonmotor_both_day_types <- function(research_id) {
 participant_data <- all_classified_data %>%
  filter(research_id == !!research_id)

 if(nrow(participant_data) == 0) {
  return(FALSE)
 }

 categorized_data <- participant_data %>%
  left_join(symptom_categories, by = c("Symptom" = "metric_name_abbr")) %>%
  filter(symptom_type == "Non-Motor")

 nonmotor_day_types <- categorized_data %>%
  distinct(day_type) %>%
  pull(day_type)

 has_nonmotor_both <- all(c("Menstruating", "Non-Menstruating") %in% nonmotor_day_types)

 return(has_nonmotor_both)
}

cat("Checking motor and non-motor availability separately for",
length(eligible_bleeding_participants_8_1), "participants...\n")

motor_availability <- sapply(eligible_bleeding_participants_8_1, check_motor_both_day_types)
eligible_participants_motor <- eligible_bleeding_participants_8_1[motor_availability]

nonmotor_availability <- sapply(eligible_bleeding_participants_8_1, check_nonmotor_both_day_types)
eligible_participants_nonmotor <- eligible_bleeding_participants_8_1[nonmotor_availability]

cat("\n=== SEPARATE FILTERING RESULTS ===\n")
cat("Started with:", length(eligible_bleeding_participants_8_1), "participants from previous analysis\n")
```

```
overlap_participants <- intersect(eligible_participants_motor, eligible_participants_nonmotor)
cat("Participants eligible for BOTH analyses:", length(overlap_participants), "\n")

motor_only <- setdiff(eligible_participants_motor, eligible_participants_nonmotor)
nonmotor_only <- setdiff(eligible_participants_nonmotor, eligible_participants_motor)

cat("\nStep 2: Calculating motor and non-motor symptom burden separately by day type...\n")

calculate_motor_averages_separate <- function(research_id) {
  participant_data <- all_classified_data %>%
    filter(research_id == !!research_id)

  motor_data <- participant_data %>%
    left_join(symptom_categories, by = c("Symptom" = "metric_name_abbr")) %>%
    filter(symptom_type == "Motor") %>%
    mutate(Severity = as.numeric(Severity))

  averages <- motor_data %>%
    group_by(day_type) %>%
    summarise(
      total_severity = sum(Severity, na.rm = TRUE),
      unique_days = n_distinct(dates),
      avg_daily_severity = total_severity / unique_days,
      .groups = 'drop'
    ) %>%
    mutate(
      research_id = research_id,
      symptom_type = "Motor"
    )

  return(averages)
}

calculate_nonmotor_averages_separate <- function(research_id) {
  participant_data <- all_classified_data %>%
    filter(research_id == !!research_id)

  nonmotor_data <- participant_data %>%
    left_join(symptom_categories, by = c("Symptom" = "metric_name_abbr")) %>%
    filter(symptom_type == "Non-Motor") %>%
    mutate(Severity = as.numeric(Severity))

  averages <- nonmotor_data %>%
    group_by(day_type) %>%
    summarise(
      total_severity = sum(Severity, na.rm = TRUE),
      unique_days = n_distinct(dates),
      avg_daily_severity = total_severity / unique_days,
```

```
    .groups = 'drop'
  ) %>%
  mutate(
    research_id = research_id,
    symptom_type = "Non-Motor"
  )

 return(averages)
}

cat("Calculating motor symptom averages for", length(eligible_participants_motor), "participants...\n")
motor_averages_separate_list <- map(eligible_participants_motor, calculate_motor_averages_separate)
motor_averages_separate <- bind_rows(motor_averages_separate_list)

cat("Calculating non-motor symptom averages for", length(eligible_participants_nonmotor),
"participants...\n")
nonmotor_averages_separate_list <- map(eligible_participants_nonmotor,
calculate_nonmotor_averages_separate)
nonmotor_averages_separate <- bind_rows(nonmotor_averages_separate_list)

combined_averages_separate <- bind_rows(motor_averages_separate, nonmotor_averages_separate)

motor_averages_wide_separate <- motor_averages_separate %>%
 select(research_id, day_type, avg_daily_severity) %>%
 pivot_wider(
  names_from = day_type,
  values_from = avg_daily_severity,
  names_prefix = "motor_"
 ) %>%
 rename(
  motor_menstruating = `motor_Menstruating`,
  motor_nonmenstruating = `motor_Non-Menstruating`
 ) %>%
 mutate(
  motor_difference = motor_menstruating - motor_nonmenstruating
 )

nonmotor_averages_wide_separate <- nonmotor_averages_separate %>%
 select(research_id, day_type, avg_daily_severity) %>%
 pivot_wider(
  names_from = day_type,
  values_from = avg_daily_severity,
  names_prefix = "nonmotor_"
 ) %>%
 rename(
  nonmotor_menstruating = `nonmotor_Menstruating`,
  nonmotor_nonmenstruating = `nonmotor_Non-Menstruating`
 ) %>%
 mutate(
```

```
  nonmotor_difference = nonmotor_menstruating - nonmotor_nonmenstruating
 )

cat("\n=== STEP 2 CALCULATION RESULTS ===\n")
cat("Motor analysis - participants with averages calculated:", nrow(motor_averages_wide_separate),
"\n")
cat("Non-motor analysis - participants with averages calculated:",
nrow(nonmotor_averages_wide_separate), "\n")

motor_summary_separate <- motor_averages_separate %>%
 group_by(day_type) %>%
 summarise(
  n_participants = n(),
  mean_avg_severity = mean(avg_daily_severity, na.rm = TRUE),
  se_mean_severity = sd(avg_daily_severity, na.rm = TRUE) / sqrt(n()),
  median_avg_severity = median(avg_daily_severity, na.rm = TRUE),
  sd_avg_severity = sd(avg_daily_severity, na.rm = TRUE),
  .groups = 'drop'
 ) %>%
 mutate(symptom_type = "Motor")

nonmotor_summary_separate <- nonmotor_averages_separate %>%
 group_by(day_type) %>%
 summarise(
  n_participants = n(),
  mean_avg_severity = mean(avg_daily_severity, na.rm = TRUE),
  se_mean_severity = sd(avg_daily_severity, na.rm = TRUE) / sqrt(n()),
  median_avg_severity = median(avg_daily_severity, na.rm = TRUE),
  sd_avg_severity = sd(avg_daily_severity, na.rm = TRUE),
  .groups = 'drop'
 ) %>%
 mutate(symptom_type = "Non-Motor")

summary_motor_nonmotor_separate <- bind_rows(motor_summary_separate,
nonmotor_summary_separate) %>%
 arrange(symptom_type, day_type)

cat("\nSummary statistics by symptom type and day type:\n")
print(summary_motor_nonmotor_separate)

motor_differences_summary <- motor_averages_wide_separate %>%
 summarise(
  n_participants = n(),
  mean_motor_diff = mean(motor_difference, na.rm = TRUE),
  median_motor_diff = median(motor_difference, na.rm = TRUE),
  sd_motor_diff = sd(motor_difference, na.rm = TRUE)
 ) %>%
 mutate(symptom_type = "Motor")
```

```
nonmotor_differences_summary <- nonmotor_averages_wide_separate %>%
 summarise(
  n_participants = n(),
  mean_nonmotor_diff = mean(nonmotor_difference, na.rm = TRUE),
  median_nonmotor_diff = median(nonmotor_difference, na.rm = TRUE),
  sd_nonmotor_diff = sd(nonmotor_difference, na.rm = TRUE)
 ) %>%
 mutate(symptom_type = "Non-Motor")

cat("\nDifference summary (Menstruating - Non-Menstruating):\n")
cat("MOTOR SYMPTOMS:\n")
print(motor_differences_summary)
cat("\nNON-MOTOR SYMPTOMS:\n")
print(nonmotor_differences_summary)

cat("\nExample motor averages (first 5):\n")
print(motor_averages_wide_separate %>%
    head(5) %>%
    mutate(across(where(is.numeric), ~ round(.x, 3))))

cat("\nExample non-motor averages (first 5):\n")
print(nonmotor_averages_wide_separate %>%
    head(5) %>%
    mutate(across(where(is.numeric), ~ round(.x, 3))))

cat("\nStep 2: Separate motor/non-motor average calculation complete\n")

cat("\nStep 3: Performing separate statistical analyses for motor and non-motor symptoms...\n")

cat("\n=== MOTOR SYMPTOMS ANALYSIS ===\n")

motor_menstruating_separate <- motor_averages_wide_separate$motor_menstruating
motor_nonmenstruating_separate <- motor_averages_wide_separate$motor_nonmenstruating

motor_complete_cases <- !is.na(motor_menstruating_separate)
& !is.na(motor_nonmenstruating_separate)
motor_men_clean_separate <- motor_menstruating_separate[motor_complete_cases]
motor_nonmen_clean_separate <- motor_nonmenstruating_separate[motor_complete_cases]

cat("Motor symptoms - complete paired observations:", length(motor_men_clean_separate), "\n")

motor_paired_t_separate <- t.test(motor_men_clean_separate, motor_nonmen_clean_separate,
                  paired = TRUE, alternative = "two.sided")

motor_paired_wilcox_separate <- wilcox.test(motor_men_clean_separate,
motor_nonmen_clean_separate,
                       paired = TRUE, alternative = "two.sided", exact = FALSE)

motor_differences_separate <- motor_men_clean_separate - motor_nonmen_clean_separate
```

```
motor_mean_diff_separate <- mean(motor_differences_separate)
motor_sd_diff_separate <- sd(motor_differences_separate)
motor_cohens_d_separate <- motor_mean_diff_separate / motor_sd_diff_separate

cat("Motor symptoms (Menstruating vs Non-Menstruating):\n")
cat("Mean difference:", round(motor_mean_diff_separate, 4), "\n")
cat("Paired t-test: t =", round(motor_paired_t_separate$statistic, 3),
  ", df =", motor_paired_t_separate$parameter,
  ", p =", format(motor_paired_t_separate$p.value, digits = 3), "\n")
cat("Paired Wilcoxon: V =", motor_paired_wilcox_separate$statistic,
  ", p =", format(motor_paired_wilcox_separate$p.value, digits = 3), "\n")
cat("Cohen's d:", round(motor_cohens_d_separate, 3), "\n")

cat("\n=== NON-MOTOR SYMPTOMS ANALYSIS ===\n")

nonmotor_menstruating_separate <- nonmotor_averages_wide_separate$nonmotor_menstruating
nonmotor_nonmenstruating_separate <-
nonmotor_averages_wide_separate$nonmotor_nonmenstruating

nonmotor_complete_cases <- !is.na(nonmotor_menstruating_separate)
& !is.na(nonmotor_nonmenstruating_separate)
nonmotor_men_clean_separate <- nonmotor_menstruating_separate[nonmotor_complete_cases]
nonmotor_nonmen_clean_separate <-
nonmotor_nonmenstruating_separate[nonmotor_complete_cases]

cat("Non-motor symptoms - complete paired observations:", length(nonmotor_men_clean_separate),
"\n")

nonmotor_paired_t_separate <- t.test(nonmotor_men_clean_separate,
nonmotor_nonmen_clean_separate,
                    paired = TRUE, alternative = "two.sided")

nonmotor_paired_wilcox_separate <- wilcox.test(nonmotor_men_clean_separate,
nonmotor_nonmen_clean_separate,
                        paired = TRUE, alternative = "two.sided", exact = FALSE)

nonmotor_differences_separate <- nonmotor_men_clean_separate -
nonmotor_nonmen_clean_separate
nonmotor_mean_diff_separate <- mean(nonmotor_differences_separate)
nonmotor_sd_diff_separate <- sd(nonmotor_differences_separate)
nonmotor_cohens_d_separate <- nonmotor_mean_diff_separate / nonmotor_sd_diff_separate

cat("Non-motor symptoms (Menstruating vs Non-Menstruating):\n")
cat("Mean difference:", round(nonmotor_mean_diff_separate, 4), "\n")
cat("Paired t-test: t =", round(nonmotor_paired_t_separate$statistic, 3),
  ", df =", nonmotor_paired_t_separate$parameter,
  ", p =", format(nonmotor_paired_t_separate$p.value, digits = 3), "\n")
cat("Paired Wilcoxon: V =", nonmotor_paired_wilcox_separate$statistic,
  ", p =", format(nonmotor_paired_wilcox_separate$p.value, digits = 3), "\n")
```

```
cat("Cohen's d:", round(nonmotor_cohens_d_separate, 3), "\n")

cat("\n=== STATISTICAL SIGNIFICANCE SUMMARY ===\n")

motor_sig_separate <- ifelse(motor_paired_t_separate$p.value < 0.05, "SIGNIFICANT", "NOT
SIGNIFICANT")
nonmotor_sig_separate <- ifelse(nonmotor_paired_t_separate$p.value < 0.05, "SIGNIFICANT", "NOT
SIGNIFICANT")

cat("Motor symptoms (n =", length(motor_men_clean_separate), "):", motor_sig_separate,
   "(p =", format(motor_paired_t_separate$p.value, digits = 3), ")\n")
cat("Non-motor symptoms (n =", length(nonmotor_men_clean_separate), "):", nonmotor_sig_separate,
   "(p =", format(nonmotor_paired_t_separate$p.value, digits = 3), ")\n")

cat("\n=== EFFECT SIZE SUMMARY ===\n")

motor_effect_interpretation <- case_when(
 abs(motor_cohens_d_separate) < 0.2 ~ "(negligible effect)",
 abs(motor_cohens_d_separate) < 0.5 ~ "(small effect)",
 abs(motor_cohens_d_separate) < 0.8 ~ "(medium effect)",
 TRUE ~ "(large effect)"
)

nonmotor_effect_interpretation <- case_when(
 abs(nonmotor_cohens_d_separate) < 0.2 ~ "(negligible effect)",
 abs(nonmotor_cohens_d_separate) < 0.5 ~ "(small effect)",
 abs(nonmotor_cohens_d_separate) < 0.8 ~ "(medium effect)",
 TRUE ~ "(large effect)"
)

cat("Motor symptoms Cohen's d:", round(motor_cohens_d_separate, 3), motor_effect_interpretation,
"\n")
cat("Non-motor symptoms Cohen's d:", round(nonmotor_cohens_d_separate, 3),
nonmotor_effect_interpretation, "\n")

cat("\nStep 3: Separate statistical analysis complete\n")

cat("\nStep 4: Creating motor/non-motor comparison visualization with separate samples...\n")

plot_data_motor_nonmotor_separate <- bind_rows(
 motor_averages_wide_separate %>%
  select(research_id, motor_menstruating, motor_nonmenstruating) %>%
  pivot_longer(
   cols = c(motor_menstruating, motor_nonmenstruating),
   names_to = "condition",
   values_to = "avg_severity"
  ) %>%
  separate(condition, into = c("symptom_type", "day_type"), sep = "_") %>%
  mutate(symptom_type = "Motor"),
```

```
  nonmotor_averages_wide_separate %>%
   select(research_id, nonmotor_menstruating, nonmotor_nonmenstruating) %>%
   pivot_longer(
    cols = c(nonmotor_menstruating, nonmotor_nonmenstruating),
    names_to = "condition",
    values_to = "avg_severity"
   ) %>%
   separate(condition, into = c("symptom_type", "day_type"), sep = "_") %>%
   mutate(symptom_type = "Non-Motor")
 ) %>%
  mutate(
   day_type = case_when(
    day_type == "menstruating" ~ "Menstruating",
    day_type == "nonmenstruating" ~ "Non-Menstruating"
   ),
   day_type = factor(day_type, levels = c("Menstruating", "Non-Menstruating")),
   symptom_type = factor(symptom_type, levels = c("Motor", "Non-Motor"))
  )

summary_stats_motor_nonmotor_separate <- plot_data_motor_nonmotor_separate %>%
  group_by(symptom_type, day_type) %>%
  summarise(
   n = n(),
   mean_severity = mean(avg_severity, na.rm = TRUE),
   median_severity = median(avg_severity, na.rm = TRUE),
   .groups = 'drop'
  )

motor_nonmotor_comparison_plot_separate <- ggplot(plot_data_motor_nonmotor_separate,
                                  aes(x = day_type, y = avg_severity, fill = day_type)) +
  facet_wrap(~ symptom_type, scales = "free_y") +
  geom_boxplot(alpha = 0.7, outlier.shape = NA, width = 0.6) +
  geom_jitter(width = 0.2, height = 0, alpha = 0.6, size = 2, color = "black") +
  geom_point(
   data = summary_stats_motor_nonmotor_separate,
   aes(x = day_type, y = mean_severity),
   shape = 18, size = 5, color = "black"
  ) +
  geom_text(
   data = summary_stats_motor_nonmotor_separate,
   aes(x = day_type, y = max(plot_data_motor_nonmotor_separate$avg_severity) * 0.95,
      label = paste("n =", n)),
   size = 4, fontface = "bold"
  ) +
  scale_fill_manual(values = c("Menstruating" = "#FFB3B3", "Non-Menstruating" = "#ADD8E6")) +
  labs(
   title = "Motor vs Non-Motor Symptoms: Menstruating vs Non-Menstruating Days",
   subtitle = "Period-tracking cohort: Motor vs Non-Motor symptoms on bleeding vs non-bleeding days",
```

```
   x = "Day Type",
   y = "Average Daily Symptom Burden",
   caption = paste0("Motor: p = ", round(motor_paired_wilcox_separate$p.value, 3), " (n = ",
length(motor_men_clean_separate), ")",
                    " | Non-Motor: p = ", round(nonmotor_paired_wilcox_separate$p.value, 3), " (n = ",
length(nonmotor_men_clean_separate), ")",
                    "\nBlack diamonds = group means")
  ) +
  theme_minimal() +
  theme(
   plot.title = element_text(face = "bold", size = 12, hjust = 0.5),
   plot.subtitle = element_text(size = 11, hjust = 0.5),
   legend.position = "none",
   axis.text.x = element_text(size = 10),
   axis.title = element_text(size = 12),
   strip.text = element_text(size = 11, face = "bold"),
   plot.caption = element_text(size = 10, hjust = 0.5)
  )

print(motor_nonmotor_comparison_plot_separate)

motor_nonmotor_menstruation_results_separate <- list(
  motor_data = motor_averages_wide_separate,
  nonmotor_data = nonmotor_averages_wide_separate,
  plot_data = plot_data_motor_nonmotor_separate,
  plot = motor_nonmotor_comparison_plot_separate,
  motor_t_test = motor_paired_t_separate,
  motor_wilcox_test = motor_paired_wilcox_separate,
  nonmotor_t_test = nonmotor_paired_t_separate,
  nonmotor_wilcox_test = nonmotor_paired_wilcox_separate,
  motor_cohens_d = motor_cohens_d_separate,
  nonmotor_cohens_d = nonmotor_cohens_d_separate,
  summary_stats = summary_stats_motor_nonmotor_separate,
  sample_sizes = list(
   motor_n = length(motor_men_clean_separate),
   nonmotor_n = length(nonmotor_men_clean_separate),
   total_unique = length(union(eligible_participants_motor, eligible_participants_nonmotor))
  )
)

cat("\nStep 4: Motor/non-motor comparison visualization complete\n")
cat("Results stored in 'motor_nonmotor_menstruation_results_separate'\n")
```

Script 9

```
rm(list = ls(pattern = "^(distance_.*|days_from_.*|last_menstruation_.*|cycle_distance_)"))
rm(list = ls(pattern = "^(temporal_.*|cycle_.*_analysis|menstruation_.*_distance|time_.*_analysis)$"))
rm(list = ls(pattern = "^(cycle_day_.*|day_.*_calculation|menstrual_.*_timing|period_.*_distance)$"))
rm(list = ls(pattern = "^(calculate_.*_distance|analyze_.*_timing|compute_cycle_day)$"))
rm(list = ls(pattern = "^(distance_.*_plot|timing_.*_chart|cycle_.*_visualization)$"))
required_vars <- c("multi_cycle_merged_data", "single_cycle_merged_data", "symptom_categories")
missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 stop("Required variables from previous scripts missing: ", paste(missing_vars, collapse = ", "), "\n",
    "Please run Scripts 1-8.3 first.")
}

if(!exists("multi_cycle_participants") || !exists("single_cycle_participants")) {
 stop("Participant classification variables missing. Please run Scripts 1 and 2 first.")
}

required_packages <- c("ggplot2", "dplyr", "tidyr", "lubridate")
missing_packages <- c()

for(pkg in required_packages) {
 if(!require(pkg, character.only = TRUE, quietly = TRUE)) {
  missing_packages <- c(missing_packages, pkg)
 }
}

if(length(missing_packages) > 0) {
 stop("Required packages for distance analysis missing: ", paste(missing_packages, collapse = ", "))
}

multi_period_data <- multi_cycle_merged_data[multi_cycle_merged_data$Flow != "No Entry", ]
single_period_data <- single_cycle_merged_data[single_cycle_merged_data$Flow != "No Entry", ]

total_period_entries <- nrow(multi_period_data) + nrow(single_period_data)

if(total_period_entries == 0) {
 stop("No menstrual period data found. Cannot calculate distance from last menstruation day.")
}

multi_dated_entries <- sum(!is.na(multi_cycle_merged_data$dates))
single_dated_entries <- sum(!is.na(single_cycle_merged_data$dates))

total_dated_entries <- multi_dated_entries + single_dated_entries
```

```
if(total_dated_entries == 0) {
 stop("No dated entries found. Cannot calculate temporal distances.")
}

participants_with_periods <- unique(c(
 multi_period_data$research_id,
 single_period_data$research_id
))

if(length(participants_with_periods) == 0) {
 stop("No participants with menstrual period data found.")
}

participants_with_multiple_dates <- c()
for(participant in participants_with_periods) {
 participant_dates <- unique(c(
  multi_cycle_merged_data$dates[multi_cycle_merged_data$research_id == participant],
  single_cycle_merged_data$dates[single_cycle_merged_data$research_id == participant]
 ))
 participant_dates <- participant_dates[!is.na(participant_dates)]

 if(length(participant_dates) > 1) {
  participants_with_multiple_dates <- c(participants_with_multiple_dates, participant)
 }
}

if(length(participants_with_multiple_dates) == 0) {
 stop("No participants with multiple dated entries found. Cannot calculate meaningful distances.")
}
menstruating_participants <- c(
 "UCC040405", "UCC041330", "UCC075P63", "UCC0772NS",
 "UCC0774NJ", "UCC0783SU", "UCC0786JF", "UCC0787HP",
 "UCC0793JL", "UCC176088", "UCC180701", "UCC2502K0", "UCC626142",
 "UCC6473P8", "UCC650148", "UCC651779", "UCC657012"
)

cat("Restricting analysis to 17 selected menstruating participants\n")
cat("Participants:", paste(menstruating_participants, collapse = ", "), "\n\n")

post_menstrual_all_existing_phases <- bind_rows(
 multi_cycle_lengths %>%
  filter(research_id %in% menstruating_participants) %>%
  select(research_id, phase_id, start_date, end_date) %>%
  mutate(participant_type = "multi_cycle"),
 single_cycle_length %>%
  filter(research_id %in% menstruating_participants) %>%
  select(research_id, phase_id, start_date, end_date) %>%
  mutate(participant_type = "single_cycle")
```

```
)
post_menstrual_combined_data <- bind_rows(
  multi_cycle_merged_data %>%
    filter(research_id %in% menstruating_participants) %>%
    mutate(participant_type = "multi_cycle"),
  single_cycle_merged_data %>%
    filter(research_id %in% menstruating_participants) %>%
    mutate(participant_type = "single_cycle")
)

post_menstrual_bleeding_days_per_phase <- post_menstrual_combined_data %>%
  filter(Flow %in% c("1", "2", "3")) %>%
  select(research_id, phase_id, dates) %>%
  distinct() %>%
  group_by(research_id, phase_id) %>%
  summarise(bleeding_days_count = n(), .groups = 'drop')

post_menstrual_phase_validation <- post_menstrual_all_existing_phases %>%
  left_join(post_menstrual_bleeding_days_per_phase, by = c("research_id", "phase_id")) %>%
  mutate(
    bleeding_days_count = ifelse(is.na(bleeding_days_count), 0, bleeding_days_count),
    duration_days = as.numeric(end_date - start_date) + 1,
    meets_criteria = (bleeding_days_count >= 3)
  )

post_menstrual_valid_periods <- post_menstrual_phase_validation %>%
  filter(meets_criteria) %>%
  select(research_id, phase_id, participant_type, start_date, end_date, bleeding_days_count)

cat("Validation results:\n")
cat("- Phases that meet ≥3 consecutive bleeding days:", nrow(post_menstrual_valid_periods), "\n")
cat("- Phases that don't meet criteria:", nrow(post_menstrual_phase_validation) -
nrow(post_menstrual_valid_periods), "\n")
cat("- Participants with valid periods:", n_distinct(post_menstrual_valid_periods$research_id), "\n\n")

cat("Sample of valid menstruation periods:\n")
print(head(post_menstrual_valid_periods, 10))

cat("\nExamples that didn't meet criteria:\n")
post_menstrual_invalid_examples <- post_menstrual_phase_validation %>%
  filter(!meets_criteria) %>%
  select(research_id, phase_id, bleeding_days_count, duration_days, meets_criteria) %>%
  head(5)
print(post_menstrual_invalid_examples)

cat("=== PREPARING SYMPTOM DATA FOR POST-MENSTRUAL ANALYSIS ===\n")

post_menstrual_symptom_data <- post_menstrual_combined_data %>%
  filter(Symptom != "No Entry" & Severity != "No Entry") %>%
```

```
 mutate(Severity = as.numeric(Severity)) %>%
 filter(!is.na(Severity)) %>%
 select(research_id, dates, Symptom, Severity, participant_type) %>%
 arrange(research_id, dates)

cat("Symptom data prepared (19 participants only):\n")
cat("- Total symptom entries:", nrow(post_menstrual_symptom_data), "\n")
cat("- Participants with symptoms:", n_distinct(post_menstrual_symptom_data$research_id), "\n")
cat("- Severity range:", min(post_menstrual_symptom_data$Severity), "to",
max(post_menstrual_symptom_data$Severity), "\n\n")

post_menstrual_daily_burden <- post_menstrual_symptom_data %>%
 group_by(research_id, dates) %>%
 summarise(
  daily_burden = sum(Severity, na.rm = TRUE),
  symptom_count = n(),
  .groups = 'drop'
 ) %>%
 arrange(research_id, dates)

cat("Daily symptom burden calculated:\n")
cat("- Total participant-days with symptoms:", nrow(post_menstrual_daily_burden), "\n")
cat("- Daily burden range:", min(post_menstrual_daily_burden$daily_burden), "to",
max(post_menstrual_daily_burden$daily_burden), "\n")
cat("- Average daily burden:", round(mean(post_menstrual_daily_burden$daily_burden), 2), "\n\n")

cat("Sample of daily symptom burden:\n")
print(head(post_menstrual_daily_burden, 10))

cat("=== CREATING POST-MENSTRUAL ANALYSIS DATASET ===\n")

post_menstrual_analysis_dataset <- post_menstrual_valid_periods %>%
 rowwise() %>%
 mutate(
  follow_up_dates = list(seq(from = end_date + 1, length.out = 12, by = "day"))
 ) %>%
 ungroup() %>%
 unnest(follow_up_dates) %>%
 mutate(
  days_since_menstruation_end = as.numeric(follow_up_dates - end_date)
 ) %>%
 left_join(
  post_menstrual_daily_burden,
  by = c("research_id" = "research_id", "follow_up_dates" = "dates")
 ) %>%
 select(research_id, phase_id, participant_type,
     menstruation_start = start_date, menstruation_end = end_date,
     follow_up_date = follow_up_dates, days_since_menstruation_end,
     daily_burden, symptom_count) %>%
```

```
 arrange(research_id, phase_id, days_since_menstruation_end)

post_menstrual_symptom_availability <- post_menstrual_analysis_dataset %>%
 summarise(
  total_observations = n(),
  days_with_symptom_data = sum(!is.na(daily_burden)),
  days_missing_data = sum(is.na(daily_burden)),
  percent_with_data = round(days_with_symptom_data / total_observations * 100, 1)
 )

cat("Symptom data availability:\n")
print(post_menstrual_symptom_availability)
cat("\n")

cat("Sample of post-menstrual analysis dataset:\n")
print(head(post_menstrual_analysis_dataset, 15))

cat("=== CHECKING FOR OVERLAPPING PERIODS (FIXED) ===\n")

post_menstrual_period_overlap_check <- post_menstrual_valid_periods %>%
 group_by(research_id) %>%
 arrange(start_date) %>%
 mutate(
  next_menstruation_start = lead(start_date),
  days_to_next_period = as.numeric(next_menstruation_start - end_date),
  has_overlap = !is.na(days_to_next_period) & days_to_next_period <= 12
 ) %>%
 ungroup()

post_menstrual_overlap_summary <- post_menstrual_period_overlap_check %>%
 summarise(
  total_periods = n(),
  periods_with_overlap = sum(has_overlap, na.rm = TRUE),
  periods_without_overlap = sum(!has_overlap | is.na(has_overlap))
 )

cat("Overlap analysis results:\n")
print(post_menstrual_overlap_summary)

cat("\nExamples of periods with overlaps:\n")
post_menstrual_overlapping_examples <- post_menstrual_period_overlap_check %>%
 filter(has_overlap == TRUE) %>%
 select(research_id, phase_id, end_date, next_menstruation_start, days_to_next_period) %>%
 head(10)
print(post_menstrual_overlapping_examples)

post_menstrual_non_overlapping_periods <- post_menstrual_period_overlap_check %>%
 filter(!has_overlap | is.na(has_overlap)) %>%
 select(research_id, phase_id)
```

```
cat("\nPeriods without overlaps:", nrow(post_menstrual_non_overlapping_periods), "\n")

post_menstrual_non_overlapping_analysis <- post_menstrual_analysis_dataset %>%
 semi_join(post_menstrual_non_overlapping_periods, by = c("research_id", "phase_id"))

cat("=== APPLYING DATA QUALITY FILTER ===\n")

post_menstrual_data_quality_check <- post_menstrual_non_overlapping_analysis %>%
 group_by(research_id, phase_id) %>%
 summarise(
  total_follow_up_days = n(),
  days_with_data = sum(!is.na(daily_burden)),
  days_missing_data = sum(is.na(daily_burden)),
  data_coverage_percent = round(days_with_data / total_follow_up_days * 100, 1),
  meets_quality_threshold = days_with_data >= 6,
  .groups = 'drop'
 )

cat("Data quality summary:\n")
post_menstrual_quality_summary <- post_menstrual_data_quality_check %>%
 summarise(
  total_periods = n(),
  periods_meeting_threshold = sum(meets_quality_threshold),
  periods_below_threshold = sum(!meets_quality_threshold),
  avg_coverage_percent = round(mean(data_coverage_percent), 1)
 )
print(post_menstrual_quality_summary)

cat("\nDistribution of data coverage:\n")
post_menstrual_coverage_distribution <- post_menstrual_data_quality_check %>%
 count(days_with_data, name = "periods_count") %>%
 arrange(days_with_data)
print(post_menstrual_coverage_distribution)

cat("\nExamples of periods meeting threshold (≥6 days):\n")
post_menstrual_meeting_threshold <- post_menstrual_data_quality_check %>%
 filter(meets_quality_threshold == TRUE) %>%
 head(5)
print(post_menstrual_meeting_threshold)

cat("\nExamples of periods below threshold (<6 days):\n")
post_menstrual_below_threshold <- post_menstrual_data_quality_check %>%
 filter(meets_quality_threshold == FALSE) %>%
 head(5)
print(post_menstrual_below_threshold)

post_menstrual_high_quality_periods <- post_menstrual_data_quality_check %>%
 filter(meets_quality_threshold) %>%
```

```
 select(research_id, phase_id)

cat("\nHigh-quality periods identified:", nrow(post_menstrual_high_quality_periods), "\n")

post_menstrual_final_analysis_data <- post_menstrual_non_overlapping_analysis %>%
 semi_join(post_menstrual_high_quality_periods, by = c("research_id", "phase_id")) %>%
 filter(!is.na(daily_burden))

cat("\nFinal analysis dataset:\n")
cat("- Total observations:", nrow(post_menstrual_final_analysis_data), "\n")
cat("- Periods included:", n_distinct(paste(post_menstrual_final_analysis_data$research_id,
post_menstrual_final_analysis_data$phase_id)), "\n")
cat("- Participants included:", n_distinct(post_menstrual_final_analysis_data$research_id), "\n")
cat("- Days since menstruation range:",
min(post_menstrual_final_analysis_data$days_since_menstruation_end), "to",
max(post_menstrual_final_analysis_data$days_since_menstruation_end), "\n")
cat("- Daily burden range:", min(post_menstrual_final_analysis_data$daily_burden), "to",
max(post_menstrual_final_analysis_data$daily_burden), "\n\n")

cat("=== POST-MENSTRUAL SYMPTOM BURDEN ANALYSIS ===\n")

cat("5A. SUMMARY STATISTICS BY DAY SINCE MENSTRUATION END:\n")

post_menstrual_analysis_summary <- post_menstrual_final_analysis_data %>%
 group_by(days_since_menstruation_end) %>%
 summarise(
  n_observations = n(),
  n_participants = n_distinct(research_id),
  n_periods = n_distinct(paste(research_id, phase_id)),
  mean_burden = round(mean(daily_burden, na.rm = TRUE), 2),
  median_burden = median(daily_burden, na.rm = TRUE),
  sd_burden = round(sd(daily_burden, na.rm = TRUE), 2),
  min_burden = min(daily_burden, na.rm = TRUE),
  max_burden = max(daily_burden, na.rm = TRUE),
  .groups = 'drop'
 ) %>%
 arrange(days_since_menstruation_end)

print(post_menstrual_analysis_summary)

cat("\n5B. CORRELATION ANALYSIS:\n")

post_menstrual_correlation_test <-
cor.test(post_menstrual_final_analysis_data$days_since_menstruation_end,
                        post_menstrual_final_analysis_data$daily_burden,
                        method = "spearman")

cat("Spearman correlation between days since menstruation and symptom burden:\n")
cat("- Correlation coefficient (rho):", round(post_menstrual_correlation_test$estimate, 3), "\n")
```

```
cat("- p-value:", format(post_menstrual_correlation_test$p.value, digits = 3), "\n")
cat("- 95% CI: [", round(as.numeric(post_menstrual_correlation_test$conf.int[1]), 3), ", ",
round(as.numeric(post_menstrual_correlation_test$conf.int[2]), 3), "]\n", sep="")

if(post_menstrual_correlation_test$p.value < 0.05) {
 direction <- ifelse(post_menstrual_correlation_test$estimate > 0, "positive", "negative")
 cat("- Result: Statistically significant", direction, "correlation\n")
} else {
 cat("- Result: No statistically significant correlation\n")
}

cat("\n5C. LINEAR REGRESSION MODEL:\n")

post_menstrual_burden_model <- lm(daily_burden ~ days_since_menstruation_end, data =
post_menstrual_final_analysis_data)
post_menstrual_model_summary <- summary(post_menstrual_burden_model)

cat("Linear model: daily_burden ~ days_since_menstruation_end\n")
cat("- Intercept:", round(post_menstrual_model_summary$coefficients[1,1], 3), "\n")
cat("- Slope:", round(post_menstrual_model_summary$coefficients[2,1], 3), "\n")
cat("- R-squared:", round(post_menstrual_model_summary$r.squared, 3), "\n")
cat("- p-value for slope:", format(post_menstrual_model_summary$coefficients[2,4], digits = 3), "\n")

cat("\n5D. CREATING MAIN VISUALIZATION:\n")

library(ggplot2)

post_menstrual_burden_trend_plot <- ggplot(post_menstrual_final_analysis_data,
                                           aes(x = days_since_menstruation_end, y = daily_burden)) +
 geom_jitter(alpha = 0.6, width = 0.2, height = 0, size = 2, color = "darkblue") +
 geom_smooth(method = "loess", se = TRUE, color = "blue", fill = "lightblue", alpha = 0.3) +
 geom_smooth(method = "lm", se = FALSE, color = "red", linetype = "dashed", linewidth = 1) +
 scale_x_continuous(breaks = 1:12, limits = c(0.5, 12.5)) +
 scale_y_continuous(limits = c(0, max(post_menstrual_final_analysis_data$daily_burden) * 1.05)) +
 labs(
  title = "Symptom Burden in 12 Days Following End of Menstruation",
  subtitle = paste("Based on", n_distinct(post_menstrual_final_analysis_data$research_id),
                   "participants and",
                   n_distinct(paste(post_menstrual_final_analysis_data$research_id,
post_menstrual_final_analysis_data$phase_id)),
                   "menstruation periods"),
  x = "Days Since End of Menstruation",
  y = "Daily Symptom Burden",
  caption = paste("Blue line: LOESS trend | Red dashed: Linear trend | Correlation rho =",
                  round(post_menstrual_correlation_test$estimate, 3))
 ) +
 theme_minimal() +
 theme(
  plot.title = element_text(face = "bold", size = 14),
```

```
   plot.subtitle = element_text(size = 11),
   panel.grid.minor = element_blank()
  )

print(post_menstrual_burden_trend_plot)

cat("\n5E. CREATING PARTICIPANT-LEVEL VISUALIZATION:\n")

post_menstrual_participant_trend_plot <- ggplot(post_menstrual_final_analysis_data,
                                 aes(x = days_since_menstruation_end, y = daily_burden)) +
  geom_line(aes(group = paste(research_id, phase_id)), alpha = 0.3, color = "gray") +
  geom_smooth(method = "loess", se = TRUE, color = "blue", linewidth = 1.5) +
  {if(length(unique(post_menstrual_final_analysis_data$participant_type)) > 1)
   facet_wrap(~participant_type, labeller = label_both)} +
  scale_x_continuous(breaks = c(1, 6, 12)) +
  labs(
   title = "Individual Period Trends in Post-Menstrual Symptom Burden",
   x = "Days Since End of Menstruation",
   y = "Daily Symptom Burden"
  ) +
  theme_minimal()

print(post_menstrual_participant_trend_plot)

cat("=== PARTICIPANT TYPE BREAKDOWN FOR FINAL ANALYSIS ===\n")

post_menstrual_participant_breakdown <- post_menstrual_final_analysis_data %>%
  group_by(participant_type) %>%
  summarise(
   n_participants = n_distinct(research_id),
   n_periods = n_distinct(paste(research_id, phase_id)),
   n_observations = n(),
   .groups = 'drop'
  )

cat("Participant type summary:\n")
print(post_menstrual_participant_breakdown)

cat("\nSpecific participants included in final analysis:\n")

post_menstrual_single_cycle_participants <- post_menstrual_final_analysis_data %>%
  filter(participant_type == "single_cycle") %>%
  distinct(research_id) %>%
  pull(research_id)

post_menstrual_multi_cycle_participants <- post_menstrual_final_analysis_data %>%
  filter(participant_type == "multi_cycle") %>%
  distinct(research_id) %>%
  pull(research_id)
```

```
cat("Single-cycle participants (", length(post_menstrual_single_cycle_participants), "):\n", sep="")
if(length(post_menstrual_single_cycle_participants) > 0) {
 cat(paste(post_menstrual_single_cycle_participants, collapse=", "), "\n")
} else {
 cat("None\n")
}

cat("\nMulti-cycle participants (", length(post_menstrual_multi_cycle_participants), "):\n", sep="")
if(length(post_menstrual_multi_cycle_participants) > 0) {
 cat(paste(post_menstrual_multi_cycle_participants, collapse=", "), "\n")
} else {
 cat("None\n")
}

cat("\nPeriods per participant in final analysis:\n")
post_menstrual_periods_per_participant <- post_menstrual_final_analysis_data %>%
 group_by(research_id, participant_type) %>%
 summarise(
  n_periods = n_distinct(paste(research_id, phase_id)),
  .groups = 'drop'
 ) %>%
 arrange(participant_type, research_id)

print(post_menstrual_periods_per_participant)

cat("\nAnalysis complete!\n")

cat("\n=== EXTENDED ANALYSIS WITH 3-7 DAY MENSTRUATION FILTER ===\n")

cat("Step 1: Filtering for typical menstruation periods (3-7 days)...\n")

filtered_menstruation_periods <- post_menstrual_high_quality_periods %>%
 left_join(
  post_menstrual_valid_periods %>% select(research_id, phase_id, start_date, end_date,
participant_type),
  by = c("research_id", "phase_id")
 ) %>%
 mutate(
  period_duration = as.numeric(end_date - start_date) + 1
 ) %>%
 filter(period_duration >= 3 & period_duration <= 7)

cat("Period duration filtering results:\n")
cat("- Original high-quality periods:", nrow(post_menstrual_high_quality_periods), "\n")
cat("- Periods with 3-7 day duration:", nrow(filtered_menstruation_periods), "\n")
cat("- Periods removed:", nrow(post_menstrual_high_quality_periods) -
nrow(filtered_menstruation_periods), "\n")
cat("- Remaining participants:", n_distinct(filtered_menstruation_periods$research_id), "\n")
```

```
duration_distribution <- filtered_menstruation_periods %>%
 count(period_duration, name = "count") %>%
 arrange(period_duration)

cat("\nDuration distribution of filtered periods:\n")
print(duration_distribution)

cat("\nSample of filtered menstruation periods:\n")
print(head(filtered_menstruation_periods %>% select(research_id, phase_id, start_date, end_date,
period_duration), 10))

cat("\nStep 2: Separating motor and non-motor symptoms (using 3-7 day periods only)...\n")

extended_menstruation_symptoms <- filtered_menstruation_periods %>%
 rowwise() %>%
 mutate(
  menstruation_dates = list(seq(from = start_date, to = end_date, by = "day"))
 ) %>%
 ungroup() %>%
 unnest(menstruation_dates) %>%
 left_join(
  post_menstrual_symptom_data,
  by = c("research_id" = "research_id", "menstruation_dates" = "dates")
 ) %>%
 filter(!is.na(Symptom)) %>%
 group_by(research_id, phase_id) %>%
 arrange(research_id, phase_id, menstruation_dates) %>%
 mutate(
  days_from_menstruation_end = as.numeric(menstruation_dates - end_date)
 ) %>%
 ungroup() %>%
 left_join(symptom_categories, by = c("Symptom" = "metric_name_abbr")) %>%
 filter(symptom_type %in% c("Motor", "Non-Motor")) %>%
 select(research_id, phase_id, menstruation_dates,
     days_from_menstruation_end, Symptom, Severity, symptom_type)

extended_menstruation_motor_nonmotor <- extended_menstruation_symptoms %>%
 group_by(research_id, phase_id, menstruation_dates, days_from_menstruation_end,
symptom_type) %>%
 summarise(
  daily_symptom_burden = sum(Severity, na.rm = TRUE),
  symptom_count = n(),
  .groups = 'drop'
 ) %>%
 complete(symptom_type = c("Motor", "Non-Motor"),
      fill = list(daily_symptom_burden = 0, symptom_count = 0))

extended_post_menstruation_symptoms <- post_menstrual_final_analysis_data %>%
```

```
 semi_join(filtered_menstruation_periods, by = c("research_id", "phase_id")) %>%
 left_join(
  post_menstrual_symptom_data,
  by = c("research_id" = "research_id", "follow_up_date" = "dates")
 ) %>%
 filter(!is.na(Symptom)) %>%
 left_join(symptom_categories, by = c("Symptom" = "metric_name_abbr")) %>%
 filter(symptom_type %in% c("Motor", "Non-Motor")) %>%
 select(research_id, phase_id, follow_up_date, days_since_menstruation_end,
     Symptom, Severity, symptom_type)

extended_post_menstruation_motor_nonmotor <- extended_post_menstruation_symptoms %>%
 group_by(research_id, phase_id, follow_up_date, days_since_menstruation_end, symptom_type) %>%
 summarise(
  daily_symptom_burden = sum(Severity, na.rm = TRUE),
  symptom_count = n(),
  .groups = 'drop'
 ) %>%
 complete(symptom_type = c("Motor", "Non-Motor"),
       fill = list(daily_symptom_burden = 0, symptom_count = 0))

cat("Motor and non-motor separation complete (FILTERED DATA):\n")
cat("- Filtered menstruation days with motor/non-motor data:",
nrow(extended_menstruation_motor_nonmotor), "\n")
cat("- Filtered post-menstruation days with motor/non-motor data:",
nrow(extended_post_menstruation_motor_nonmotor), "\n")

menstruation_summary <- extended_menstruation_motor_nonmotor %>%
 group_by(symptom_type) %>%
 summarise(
  observations = n(),
  days_with_data = sum(daily_symptom_burden > 0),
  mean_burden = round(mean(daily_symptom_burden[daily_symptom_burden > 0], na.rm = TRUE), 2),
  .groups = 'drop'
 )

post_menstruation_summary <- extended_post_menstruation_motor_nonmotor %>%
 group_by(symptom_type) %>%
 summarise(
  observations = n(),
  days_with_data = sum(daily_symptom_burden > 0),
  mean_burden = round(mean(daily_symptom_burden[daily_symptom_burden > 0], na.rm = TRUE), 2),
  .groups = 'drop'
 )

cat("\nMenstruation days summary by symptom type:\n")
print(menstruation_summary)

cat("\nPost-menstruation days summary by symptom type:\n")
```

```
print(post_menstruation_summary)

cat("\nStep 3: Combining into final extended dataset...\n")

extended_menstruation_final <- extended_menstruation_motor_nonmotor %>%
 rename(
  symptom_date = menstruation_dates,
  days_from_end = days_from_menstruation_end
 ) %>%
 mutate(period_phase = "Menstruation") %>%
 select(research_id, phase_id, symptom_date, days_from_end, period_phase,
      symptom_type, daily_symptom_burden, symptom_count)

extended_post_menstruation_final <- extended_post_menstruation_motor_nonmotor %>%
 rename(
  symptom_date = follow_up_date,
  days_from_end = days_since_menstruation_end
 ) %>%
 mutate(period_phase = "Post-Menstruation") %>%
 select(research_id, phase_id, symptom_date, days_from_end, period_phase,
      symptom_type, daily_symptom_burden, symptom_count)

extended_combined_data <- bind_rows(
 extended_menstruation_final,
 extended_post_menstruation_final
) %>%
 filter(days_from_end != 0) %>%
 arrange(research_id, phase_id, days_from_end, symptom_type)

cat("Extended combined dataset created:\n")
cat("- Total observations:", nrow(extended_combined_data), "\n")
cat("- Day range:", min(extended_combined_data$days_from_end), "to",
max(extended_combined_data$days_from_end), "\n")
cat("- Periods included:", n_distinct(paste(extended_combined_data$research_id,
extended_combined_data$phase_id)), "\n")
cat("- Participants included:", n_distinct(extended_combined_data$research_id), "\n")

day_sequence_check <- extended_combined_data %>%
 distinct(days_from_end) %>%
 arrange(days_from_end) %>%
 pull(days_from_end)

cat("- Day sequence:", paste(head(day_sequence_check, 10), collapse = ", "), "...",
paste(tail(day_sequence_check, 5), collapse = ", "), "\n")

combined_summary <- extended_combined_data %>%
 group_by(period_phase, symptom_type) %>%
 summarise(
  observations = n(),
```

```
  unique_days = n_distinct(days_from_end),
  mean_burden = round(mean(daily_symptom_burden, na.rm = TRUE), 2),
  median_burden = median(daily_symptom_burden, na.rm = TRUE),
  .groups = 'drop'
 )

cat("\nSummary by period phase and symptom type:\n")
print(combined_summary)

cat("\nSample of combined extended dataset:\n")
print(head(extended_combined_data, 15))

cat("\nData quality checks:\n")
cat("- Any missing days between -1 and 1?", !(-1 %in% day_sequence_check && 1 %in%
day_sequence_check && !0 %in% day_sequence_check), "\n")
cat("- Both motor and non-motor present?", all(c("Motor", "Non-Motor") %in%
extended_combined_data$symptom_type), "\n")
cat("- Both period phases present?", all(c("Menstruation", "Post-Menstruation") %in%
extended_combined_data$period_phase), "\n")

cat("\nStep 4: Creating extended visualization...\n")

library(ggplot2)

extended_symptom_plot <- ggplot(extended_combined_data, aes(x = days_from_end, y =
daily_symptom_burden, color = symptom_type)) +
 annotate("rect", xmin = -7, xmax = -0.5, ymin = -Inf, ymax = Inf,
       fill = "#FFB3B3", alpha = 0.3) +
 geom_jitter(alpha = 0.6, width = 0.3, height = 0, size = 1.5) +
 geom_smooth(method = "loess", se = TRUE, linewidth = 1.2, alpha = 0.2) +
 geom_vline(xintercept = 0, linetype = "dashed", color = "black", alpha = 0.5) +
 scale_color_manual(
  values = c("Motor" = "#1f77b4", "Non-Motor" = "#ff7f0e"),
  name = "Symptom Type"
 ) +
 scale_fill_manual(
  values = c("Motor" = "#1f77b4", "Non-Motor" = "#ff7f0e"),
  name = "Symptom Type"
 ) +
 scale_x_continuous(
  breaks = c(-6, -5, -3, -1, 1, 5, 10, 12),
  labels = c("-6", "-5", "-3", "-1", "1", "5", "10", "12"),
  limits = c(-7, 12)
 ) +
 scale_y_continuous(limits = c(0, max(extended_combined_data$daily_symptom_burden) * 1.05)) +
 labs(
  title = "Motor and Non-Motor Symptom Burden During and After Menstruation",
  subtitle = paste("Extended analysis (3-7 day periods):",
n_distinct(extended_combined_data$research_id),
```

```
            "participants,", n_distinct(paste(extended_combined_data$research_id,
extended_combined_data$phase_id)),
            "menstruation periods"),
   x = "Days Relative to End of Menstruation",
   y = "Daily Symptom Burden",
   caption = "Pink background = Menstruation period | Dashed line = End of menstruation | Trend lines
with 95% CI"
  ) +
  theme_minimal() +
  theme(
   plot.title = element_text(face = "bold", size = 14, hjust = 0.5),
   plot.subtitle = element_text(size = 12, hjust = 0.5),
   plot.caption = element_text(size = 10, hjust = 0.5),
   legend.position = "top",
   legend.title = element_text(face = "bold"),
   axis.title = element_text(size = 12),
   axis.text = element_text(size = 10),
   panel.grid.minor.x = element_blank(),
   panel.grid.major.x = element_line(color = "grey", linewidth = 0.3)
  )

print(extended_symptom_plot)

cat("\nPlot summary statistics:\n")

plot_summary <- extended_combined_data %>%
  group_by(symptom_type, period_phase) %>%
  summarise(
   observations = n(),
   mean_burden = round(mean(daily_symptom_burden, na.rm = TRUE), 2),
   sd_burden = round(sd(daily_symptom_burden, na.rm = TRUE), 2),
   se_mean_burden = round(sd(daily_symptom_burden, na.rm = TRUE) / sqrt(n()), 2),
   median_burden = median(daily_symptom_burden, na.rm = TRUE),
   min_burden = min(daily_symptom_burden, na.rm = TRUE),
   max_burden = max(daily_symptom_burden, na.rm = TRUE),
   .groups = 'drop'
  )

print(plot_summary)

cat("\n=== EXTENDED ANALYSIS COMPLETE ===\n")
cat("Results stored in 'extended_analysis_results' for future reference\n")
```

Script 10

```
rm(list = ls(pattern = "^(selected_.*_timelines|timeline_.*|participant_.*_timeline|id_.*_analysis)"))
rm(list = ls(pattern =
"^(selected_participants|chosen_.*_ids|timeline_participants|participant_.*_selection)$"))
rm(list = ls(pattern = "^(timeline_.*_plot|participant_.*_chart|id_.*_visualization|gantt_.*_chart)$"))
rm(list = ls(pattern = "^(create_.*_timeline|generate_.*_timeline|plot_participant_timeline)$"))
rm(list = ls(pattern = "^(timeline_data|processed_.*_timeline|participant_.*_data|id_.*_processed)$"))
required_vars <- c("multi_cycle_merged_data", "single_cycle_merged_data", "no_period_final",
"symptom_categories")
missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 stop("Required variables from previous scripts missing: ", paste(missing_vars, collapse = ", "), "\n",
    "Please run Scripts 1-9 first.")
}

if(!exists("multi_cycle_participants") || !exists("single_cycle_participants")
|| !exists("no_period_participants")) {
 stop("Participant classification variables missing. Please run Scripts 1 and 2 first.")
}

required_packages <- c("ggplot2", "dplyr", "tidyr", "lubridate")
missing_packages <- c()

for(pkg in required_packages) {
 if(!require(pkg, character.only = TRUE, quietly = TRUE)) {
  missing_packages <- c(missing_packages, pkg)
 }
}

if(length(missing_packages) > 0) {
 stop("Required packages for timeline visualization missing: ", paste(missing_packages, collapse = ", "))
}

all_available_participants <- unique(c(
 multi_cycle_participants$research_id,
 single_cycle_participants$research_id,
 no_period_participants
))

if(length(all_available_participants) == 0) {
 stop("No participants found across any datasets.")
}
```

```
participants_with_timeline_data <- c()
for(participant in all_available_participants) {
 multi_dates <- multi_cycle_merged_data$dates[multi_cycle_merged_data$research_id == participant]
 multi_dates <- multi_dates[!is.na(multi_dates)]

 single_dates <- single_cycle_merged_data$dates[single_cycle_merged_data$research_id ==
participant]
 single_dates <- single_dates[!is.na(single_dates)]

 no_period_dates <- c()
 if(participant %in% no_period_final$research_id) {
  no_period_dates <- no_period_final$dates[no_period_final$research_id == participant]
  no_period_dates <- no_period_dates[!is.na(no_period_dates)]
 }

 total_dates <- length(c(multi_dates, single_dates, no_period_dates))
 if(total_dates > 0) {
  participants_with_timeline_data <- c(participants_with_timeline_data, participant)
 }
}

if(length(participants_with_timeline_data) == 0) {
 stop("No participants with dated entries found. Cannot create timelines.")
}

participants_with_rich_data <- c()
for(participant in participants_with_timeline_data) {
 multi_entries <- nrow(multi_cycle_merged_data[multi_cycle_merged_data$research_id ==
participant, ])
 single_entries <- nrow(single_cycle_merged_data[single_cycle_merged_data$research_id ==
participant, ])
 no_period_entries <- nrow(no_period_final[no_period_final$research_id == participant, ])

 total_entries <- multi_entries + single_entries + no_period_entries

 if(total_entries >= 10) {
  participants_with_rich_data <- c(participants_with_rich_data, participant)
 }
}

cat("🧹 Environment cleaned - Script 10 variables cleared\n")
cat("👥 Total participants across all datasets:", length(all_available_participants), "\n")
cat("👥 Participants with dated entries for timelines:", length(participants_with_timeline_data), "\n")
cat("👥 Participants with rich data for detailed timelines:", length(participants_with_rich_data), "\n")
cat("📊 Multi-cycle participants:", length(multi_cycle_participants$research_id), "\n")
cat("📊 Single-cycle participants:", length(single_cycle_participants$research_id), "\n")
cat("📊 No-period participants:", length(no_period_participants), "\n")
cat("🔄 Ready for participant timeline visualization\n\n")
```

```
prepare_symptom_data_for_timeline <- function(research_id, start_date = NULL, end_date = NULL) {
 cat("=== Preparing symptom data for", research_id, "===\n")

 symptom_data <- multi_cycle_merged_data %>%
  filter(
   research_id == !!research_id,
   Symptom != "No Entry"
  ) %>%
  select(research_id, dates, Symptom, Severity) %>%
  mutate(Severity = as.numeric(Severity))

 cat("Total symptom entries found:", nrow(symptom_data), "\n")

 if(!is.null(start_date) && !is.null(end_date)) {
  original_count <- nrow(symptom_data)
  symptom_data <- symptom_data %>%
   filter(dates >= as.Date(start_date) & dates <= as.Date(end_date))
  cat("After date filtering (", as.character(start_date), "to", as.character(end_date), "):",
nrow(symptom_data), "entries\n")
 }

 symptom_data_classified <- symptom_data %>%
  left_join(
   symptom_categories %>% select(metric_name_abbr, symptom_type),
   by = c("Symptom" = "metric_name_abbr")
  ) %>%
  filter(!is.na(symptom_type)) %>%
  filter(symptom_type %in% c("Motor", "Non-Motor"))

 cat("Classified symptom entries:", nrow(symptom_data_classified), "\n")
 cat("- Motor symptoms:", sum(symptom_data_classified$symptom_type == "Motor"), "\n")
 cat("- Non-Motor symptoms:", sum(symptom_data_classified$symptom_type == "Non-Motor"), "\n")

 if(nrow(symptom_data_classified) > 0) {
  date_range <- range(symptom_data_classified$dates)
  cat("Date range:", as.character(date_range[1]), "to", as.character(date_range[2]), "\n")

  cat("\nSample of classified data:\n")
  print(head(symptom_data_classified, 8))
 } else {
  cat("No classified symptom data found\n")
 }

 motor_max_severity <- if(any(symptom_data_classified$symptom_type == "Motor")) {
  max(symptom_data_classified$Severity[symptom_data_classified$symptom_type == "Motor"], na.rm
= TRUE)
 } else {
  NA
```

```
 }

 nonmotor_max_severity <- if(any(symptom_data_classified$symptom_type == "Non-Motor")) {
  max(symptom_data_classified$Severity[symptom_data_classified$symptom_type == "Non-Motor"],
na.rm = TRUE)
 } else {
  NA
 }

 severity_limits <- list(
  motor_max = motor_max_severity,
  nonmotor_max = nonmotor_max_severity,
  motor_cap = if(!is.na(motor_max_severity)) motor_max_severity + 0.2 else NA,
  nonmotor_cap = if(!is.na(nonmotor_max_severity)) nonmotor_max_severity + 0.2 else NA
 )

 cat("\n=== Severity Limits Calculated ===\n")
 if(!is.na(severity_limits$motor_max)) {
  cat("Motor symptoms: Max =", severity_limits$motor_max, "| Cap =", severity_limits$motor_cap, "\n")
 } else {
  cat("Motor symptoms: None found\n")
 }

 if(!is.na(severity_limits$nonmotor_max)) {
  cat("Non-Motor symptoms: Max =", severity_limits$nonmotor_max, "| Cap =",
severity_limits$nonmotor_cap, "\n")
 } else {
  cat("Non-Motor symptoms: None found\n")
 }

 return(list(
  data = symptom_data_classified,
  severity_limits = severity_limits
 ))
}

test_result <- prepare_symptom_data_for_timeline("UCC2502K0", "2024-08-07", "2024-09-09")

test_data_filtered <- prepare_symptom_data_for_timeline(
 "UCC033567",
 start_date = "2024-06-20",
 end_date = "2024-07-31"
)

test_data_unfiltered <- prepare_symptom_data_for_timeline("UCC033567")

get_menstruation_phase_dates <- function(research_id, start_date = NULL, end_date = NULL) {
 cat("=== Getting menstruation phase dates for", research_id, "===\n")
```

```
 menstrual_periods <- multi_cycle_lengths %>%
  filter(research_id == !!research_id) %>%
  select(research_id, phase_id, start_date, end_date)

 cat("Menstruation periods found:", nrow(menstrual_periods), "\n")

 if(nrow(menstrual_periods) > 0) {
  cat("Menstruation periods:\n")
  print(menstrual_periods)

  menstrual_dates <- menstrual_periods %>%
   rowwise() %>%
   mutate(
    dates = list(seq(start_date, end_date, by = "day"))
   ) %>%
   unnest(dates) %>%
   select(research_id, phase_id, dates) %>%
   mutate(period_type = "menstruation")

  cat("\nTotal menstruation dates:", nrow(menstrual_dates), "\n")

  if(!is.null(start_date) && !is.null(end_date)) {
   original_count <- nrow(menstrual_dates)
   menstrual_dates <- menstrual_dates %>%
    filter(dates >= as.Date(start_date) & dates <= as.Date(end_date))
   cat("After date filtering (", as.character(start_date), "to", as.character(end_date), "):",
nrow(menstrual_dates), "menstruation dates\n")
  }

  cat("Sample menstruation dates:\n")
  print(head(menstrual_dates, 8))

 } else {
  cat("No menstruation periods found\n")
  menstrual_dates <- data.frame(
   research_id = character(0),
   phase_id = numeric(0),
   dates = as.Date(character(0)),
   period_type = character(0)
  )
 }

 return(menstrual_dates)
}

test_menstrual_dates_filtered <- get_menstruation_phase_dates(
 "UCC040405",
 start_date = "2024-07-29",
 end_date = "2024-09-29"
```

```
)

test_menstrual_dates_unfiltered <- get_menstruation_phase_dates("UCC040405")

prepare_flow_and_gaps <- function(research_id, menstrual_dates, start_date = NULL, end_date = NULL)
{
 cat("=== Preparing flow data and gaps for", research_id, "===\n")

 flow_data <- multi_cycle_merged_data %>%
  filter(
   research_id == !!research_id,
   Flow %in% c("1", "2", "3")
  ) %>%
  select(research_id, dates, Flow) %>%
  arrange(dates)

 cat("Flow entries found:", nrow(flow_data), "\n")

 if(!is.null(start_date) && !is.null(end_date)) {
  original_count <- nrow(flow_data)
  flow_data <- flow_data %>%
   filter(dates >= as.Date(start_date) & dates <= as.Date(end_date))
  cat("After date filtering (", as.character(start_date), "to", as.character(end_date), "):", nrow(flow_data),
"flow entries\n")
 }

 if(nrow(flow_data) > 0) {
  flow_summary <- flow_data %>%
   count(Flow) %>%
   arrange(Flow)
  cat("Flow level distribution:\n")
  print(flow_summary)

  cat("Sample flow data:\n")
  print(head(flow_data, 6))
 }

 if(nrow(menstrual_dates) > 0) {
  flow_dates <- if(nrow(flow_data) > 0) flow_data$dates else as.Date(character(0))

  gap_dates <- menstrual_dates %>%
   filter(!dates %in% flow_dates) %>%
   mutate(gap_type = "menstruation_no_flow")

  cat("\nGap analysis:\n")
  cat("Total menstruation phase dates:", nrow(menstrual_dates), "\n")
  cat("Dates with flow logged:", length(flow_dates), "\n")
  cat("Gap dates (menstruation phase but no flow):", nrow(gap_dates), "\n")
```

```
  if(nrow(gap_dates) > 0) {
    cat("Sample gap dates:\n")
    print(head(gap_dates, 6))
  }
 } else {
  gap_dates <- data.frame(
    research_id = character(0),
    phase_id = numeric(0),
    dates = as.Date(character(0)),
    period_type = character(0),
    gap_type = character(0)
  )
 }

 return(list(
  flow_data = flow_data,
  gap_dates = gap_dates
 ))
}

test_flow_and_gaps_filtered <- prepare_flow_and_gaps(
 "UCC040405",
 test_menstrual_dates_filtered,
 start_date = "2024-07-29",
 end_date = "2024-09-29"
)

test_flow_and_gaps_unfiltered <- prepare_flow_and_gaps("UCC040405",
test_menstrual_dates_unfiltered)

create_complete_timeline_plot <- function(research_id, symptom_result, flow_data, gap_dates,
start_date = NULL, end_date = NULL) {
 cat("=== Creating complete timeline plot for", research_id, "===\n")

 library(ggplot2)

 symptom_data <- symptom_result$data
 severity_limits <- symptom_result$severity_limits

 flow_colors <- c(
  "1" = "#FFB3BA",
  "2" = "#FF9999",
  "3" = "#FF6B6B"
 )
 gap_color <- "#E6E6FA"

 symptom_colors <- c(
  "Motor" = "#1f77b4",
  "Non-Motor" = "#2ca02c"
```

```
 )

 if(!is.null(start_date) && !is.null(end_date)) {
  date_range <- c(as.Date(start_date) - 1, as.Date(end_date) + 1)
  cat("Using specified date range with buffer:", as.character(date_range[1]), "to",
as.character(date_range[2]), "\n")
  cat("Original range was:", as.character(as.Date(start_date)), "to", as.character(as.Date(end_date)),
"\n")

  total_days <- as.numeric(as.Date(end_date) - as.Date(start_date)) + 1
  cat("Total tracking duration:", total_days, "days\n")

  if(total_days <= 14) {
   break_interval <- ifelse(total_days <= 7, "2 days", "3 days")
   cat("Short duration - using breaks every", break_interval, "\n")
   date_breaks <- break_interval
  } else if(total_days <= 42) {
   cat("Medium duration - using weekly breaks\n")
   date_breaks <- "1 week"
  } else {
   cat("Long duration - using bi-weekly breaks\n")
   date_breaks <- "2 weeks"
  }

  if(total_days <= 14) {
   custom_breaks <- seq(from = as.Date(start_date), to = as.Date(end_date), by = break_interval)
  } else if(total_days <= 42) {
   custom_breaks <- seq(from = as.Date(start_date), to = as.Date(end_date), by = "1 week")
  } else {
   custom_breaks <- seq(from = as.Date(start_date), to = as.Date(end_date), by = "2 weeks")
  }

  custom_breaks <- unique(c(as.Date(start_date), custom_breaks, as.Date(end_date)))
  custom_breaks <- custom_breaks[custom_breaks >= as.Date(start_date) & custom_breaks <=
as.Date(end_date)]
  custom_breaks <- sort(custom_breaks)

  cat("Custom breaks calculated:", as.character(custom_breaks), "\n")

 } else {
  date_range <- range(symptom_data$dates)
  cat("Using symptom data date range:", as.character(date_range[1]), "to", as.character(date_range[2]),
"\n")

  total_days <- as.numeric(diff(date_range)) + 1
  cat("Data span duration:", total_days, "days\n")

  if(total_days <= 14) {
```

```
    custom_breaks <- seq(from = date_range[1], to = date_range[2], by = ifelse(total_days <= 7, "2 days",
"3 days"))
   } else if(total_days <= 42) {
    custom_breaks <- seq(from = date_range[1], to = date_range[2], by = "1 week")
   } else {
    custom_breaks <- seq(from = date_range[1], to = date_range[2], by = "2 weeks")
   }

   custom_breaks <- unique(c(date_range[1], custom_breaks, date_range[2]))
   custom_breaks <- custom_breaks[custom_breaks >= date_range[1] & custom_breaks <=
date_range[2]]
   custom_breaks <- sort(custom_breaks)
  }

  y_limits <- c(-0.2, 3.2)

  has_motor <- any(symptom_data$symptom_type == "Motor")
  has_nonmotor <- any(symptom_data$symptom_type == "Non-Motor")

  legend_colors <- c()
  legend_labels <- c()

  if(has_motor) {
   legend_colors["Motor"] <- symptom_colors["Motor"]
   legend_labels <- c(legend_labels, "Motor")
  }

  if(has_nonmotor) {
   legend_colors["Non-Motor"] <- symptom_colors["Non-Motor"]
   legend_labels <- c(legend_labels, "Non-Motor")
  }

  cat("Available symptom types:", paste(legend_labels, collapse = ", "), "\n")

  create_constrained_smooth <- function(data, symptom_type, severity_cap) {
   type_data <- data %>% filter(symptom_type == !!symptom_type)

   if(nrow(type_data) < 3) {
    cat("Not enough", symptom_type, "data points for trend line (need 3+, have", nrow(type_data), ")\n")
    return(NULL)
   }

   unique_dates <- length(unique(type_data$dates))
   date_span <- as.numeric(diff(range(type_data$dates))) + 1
   data_density <- unique_dates / date_span

   if(data_density < 0.2) {
    cat("Data too sparse for", symptom_type, "smoothing (", unique_dates, "dates over", date_span,
"days) - using simple linear trend\n")
```

```
  tryCatch({
   lm_fit <- lm(Severity ~ as.numeric(dates), data = type_data)
   date_seq <- seq(min(type_data$dates), max(type_data$dates), by = "day")
   predicted_values <- predict(lm_fit, newdata = data.frame(dates = as.numeric(date_seq)))

   predicted_values <- pmax(0, pmin(predicted_values, severity_cap))

   smooth_data <- data.frame(
    dates = date_seq,
    Severity = predicted_values,
    symptom_type = symptom_type
   )

   return(geom_line(data = smooth_data,
                    aes(x = dates, y = Severity, color = symptom_type),
                    linewidth = 1.2, alpha = 0.8))
  }, error = function(e) {
   cat("Error creating", symptom_type, "linear trend:", e$message, "\n")
   return(NULL)
  })
 } else {
  cat("Creating constrained trend line for", symptom_type, "with cap at", severity_cap, "\n")

  tryCatch({
   date_seq <- seq(min(type_data$dates), max(type_data$dates), by = "day")

   loess_fit <- loess(Severity ~ as.numeric(dates), data = type_data, span = 0.9)
   predicted_values <- predict(loess_fit, newdata = as.numeric(date_seq))

   predicted_values <- pmax(0, pmin(predicted_values, severity_cap))

   smooth_data <- data.frame(
    dates = date_seq,
    Severity = predicted_values,
    symptom_type = symptom_type
   )

   smooth_data <- smooth_data[!is.na(smooth_data$Severity), ]

   return(geom_line(data = smooth_data,
                    aes(x = dates, y = Severity, color = symptom_type),
                    linewidth = 1.2, alpha = 0.8))
  }, error = function(e) {
   cat("Error creating", symptom_type, "trend line:", e$message, "\n")
   return(NULL)
  })
 }
}
```

```
create_flow_background <- function() {
 if(nrow(flow_data) == 0) {
  cat("No flow data to display\n")
  return(list())
 }

 cat("Creating flow background with", nrow(flow_data), "flow entries\n")

 if(!is.null(start_date) && !is.null(end_date)) {
  normal_flow <- flow_data %>%
   filter(dates >= as.Date(start_date) & dates <= as.Date(end_date))
  boundary_flow <- flow_data %>%
   filter(dates < as.Date(start_date) | dates > as.Date(end_date))
 } else {
  normal_flow <- flow_data
  boundary_flow <- data.frame()
 }

 cat("Normal flow entries:", nrow(normal_flow), ", Boundary entries:", nrow(boundary_flow), "\n")

 return(list(
  if(nrow(normal_flow) > 0) {
   lapply(unique(normal_flow$Flow), function(flow_level) {
    flow_subset <- normal_flow %>% filter(Flow == flow_level)
    geom_rect(data = flow_subset,
              aes(xmin = dates, xmax = dates + 0.99,
                  ymin = y_limits[1], ymax = y_limits[2]),
              fill = flow_colors[flow_level], alpha = 0.5)
   })
  },
  if(nrow(boundary_flow) > 0) {
   lapply(unique(boundary_flow$Flow), function(flow_level) {
    flow_subset <- boundary_flow %>% filter(Flow == flow_level)
    geom_vline(data = flow_subset,
               aes(xintercept = as.numeric(dates)),
               color = flow_colors[flow_level], linewidth = 8, alpha = 0.6)
   })
  }
 ))
}

p <- tryCatch({
 ggplot() +
  create_flow_background() +
  {if(nrow(gap_dates) > 0 && "dates" %in% names(gap_dates)) {
   geom_vline(xintercept = as.numeric(gap_dates$dates),
              color = gap_color, linewidth = 3, alpha = 0.7)
  }} +
```

```
    {if(has_motor) create_constrained_smooth(symptom_data, "Motor", severity_limits$motor_cap)} +
    {if(has_nonmotor) create_constrained_smooth(symptom_data, "Non-Motor",
severity_limits$nonmotor_cap)} +
    geom_point(data = symptom_data,
           aes(x = dates, y = Severity, color = symptom_type),
           position = position_jitter(width = 0.3, height = 0.2),
           alpha = 0.7, size = 2.5) +
    scale_x_date(limits = date_range,
            breaks = custom_breaks,
            labels = scales::date_format("%b %d"),
            expand = c(0.02, 0.02)) +
    scale_y_continuous(limits = y_limits, breaks = c(0, 1, 2, 3)) +
    scale_color_manual(values = legend_colors, name = "Symptom Type", labels = legend_labels) +
    labs(title = paste("Menstrual Timeline and Symptoms:", research_id),
       x = "Date", y = "Symptom Severity") +
    theme_minimal() +
    theme(
     axis.text.x = element_text(angle = 45, hjust = 1, size = 10),
     plot.title = element_text(face = "bold", size = 14, hjust = 0.5)
    )
 }, error = function(e) {
  cat("Error creating plot:", e$message, "\n")
  ggplot(symptom_data, aes(x = dates, y = Severity, color = symptom_type)) +
   geom_point(alpha = 0.7, size = 2.5) +
   scale_x_date(limits = date_range, expand = c(0.02, 0.02)) +
   scale_y_continuous(limits = y_limits, breaks = c(0, 1, 2, 3)) +
   scale_color_manual(values = legend_colors, name = "Symptom Type", labels = legend_labels) +
   labs(title = paste("Menstrual Timeline and Symptoms:", research_id),
       x = "Date", y = "Symptom Severity") +
   theme_minimal()
 })

 cat("Created complete plot with adaptive date breaks and Y-axis limits:", y_limits[1], "to", y_limits[2],
"for severity range 0-3\n")
 print(p)
 return(p)
}

symptom_result <- prepare_symptom_data_for_timeline("UCC2502K0", "2024-08-07", "2024-09-09")
menstrual_dates <- get_menstruation_phase_dates("UCC2502K0", "2024-08-07", "2024-09-09")
flow_and_gaps <- prepare_flow_and_gaps("UCC2502K0", menstrual_dates, "2024-08-07", "2024-09-09")

test_plot <- create_complete_timeline_plot("UCC2502K0", symptom_result, flow_and_gaps$flow_data,
flow_and_gaps$gap_dates, "2024-08-07", "2024-09-09")

test_plot_filtered <- create_complete_timeline_plot(
 "UCC040405",
 test_data_filtered,
 test_flow_and_gaps_filtered$flow_data,
```

```
 test_flow_and_gaps_filtered$gap_dates,
 start_date = "2024-07-29",
 end_date = "2024-09-29"
)

test_plot_unfiltered <- create_complete_timeline_plot(
 "UCC040405",
 test_data_unfiltered,
 test_flow_and_gaps_unfiltered$flow_data,
 test_flow_and_gaps_unfiltered$gap_dates
)

create_all_timeline_plots <- function(research_ids_with_dates, save_plots = FALSE, output_dir = NULL,
pause_between = TRUE) {
 cat("=== CREATING TIMELINE PLOTS FOR ALL RESEARCH IDs WITH DATE RANGES ===\n")

 research_ids <- unique(research_ids_with_dates$research_id)
 cat("Total research IDs to process:", length(research_ids), "\n")
 cat("Processing order:", paste(head(research_ids, 5), collapse = ", "),
   if(length(research_ids) > 5) "..." else "", "\n")

 if(pause_between) {
  cat("Pause mode enabled - press ENTER after each plot to continue\n")
 }
 cat("\n")

 all_plots <- list()

 successful_plots <- 0
 failed_plots <- 0
 no_data_plots <- 0

 for(i in seq_along(research_ids)) {
  research_id <- research_ids[i]

  date_info <- research_ids_with_dates %>%
   filter(research_id == !!research_id) %>%
   slice(1)

  start_date <- date_info$start_date
  end_date <- date_info$end_date

  cat("Processing", i, "of", length(research_ids), ":", research_id,
     "(", as.character(start_date), "to", as.character(end_date), ")\n")

  tryCatch({
   symptom_result <- prepare_symptom_data_for_timeline(research_id, start_date, end_date)

   if(nrow(symptom_result$data) == 0) {
```

```
      cat("No symptom data found for", research_id, "in specified date range - skipping\n")
      no_data_plots <- no_data_plots + 1

      if(pause_between && i < length(research_ids)) {
        cat("Press ENTER to continue to next participant...")
        invisible(readline())
      }
      cat("\n")
      next
    }

    menstrual_dates <- get_menstruation_phase_dates(research_id, start_date, end_date)

    flow_and_gaps <- prepare_flow_and_gaps(research_id, menstrual_dates, start_date, end_date)

    timeline_plot <- create_complete_timeline_plot(
      research_id,
      symptom_result,
      flow_and_gaps$flow_data,
      flow_and_gaps$gap_dates,
      start_date,
      end_date
    )

    all_plots[[research_id]] <- timeline_plot

    if(save_plots && !is.null(output_dir)) {
      if(!dir.exists(output_dir)) {
        dir.create(output_dir, recursive = TRUE)
      }

      filename <- file.path(output_dir, paste0("timeline_", research_id, "_", start_date, "_to_", end_date,
".png"))
      ggsave(filename, timeline_plot, width = 12, height = 8, dpi = 300)
      cat("Saved plot to:", filename, "\n")
    }

    successful_plots <- successful_plots + 1
    cat("✓ Successfully created plot for", research_id, "\n")

    if(pause_between && i < length(research_ids)) {
      cat("Press ENTER to continue to next participant...")
      invisible(readline())
    }
    cat("\n")

  }, error = function(e) {
    cat("✗ Error processing", research_id, ":", e$message, "\n")
    failed_plots <- failed_plots + 1
```

```
    if(pause_between && i < length(research_ids)) {
      cat("Press ENTER to continue to next participant...")
      invisible(readline())
    }
    cat("\n")
  })
 }

 cat("=== PROCESSING COMPLETE ===\n")
 cat("Total research IDs:", length(research_ids), "\n")
 cat("Successful plots:", successful_plots, "\n")
 cat("No data (skipped):", no_data_plots, "\n")
 cat("Failed plots:", failed_plots, "\n")

 return(list(
  plots = all_plots,
  summary = list(
   total = length(research_ids),
   successful = successful_plots,
   no_data = no_data_plots,
   failed = failed_plots
  )
 ))
}

test_research_ids_with_dates <- data.frame(
 research_id = c("UCC2502K0", "UCC040405"),
 start_date = as.Date(c("2024-08-07", "2024-07-29")),
 end_date = as.Date(c("2024-09-09", "2024-09-29"))
)

test_results <- create_all_timeline_plots(test_research_ids_with_dates, pause_between = FALSE)

research_ids_with_dates <- data.frame(
 research_id = c(
  "UCC040405", "UCC041330", "UCC075P63", "UCC0772NS",
  "UCC0774NJ", "UCC0783SU", "UCC0786JF", "UCC0787HP",
  "UCC0793JL", "UCC176088", "UCC180701", "UCC2502K0", "UCC626142",
  "UCC6473P8", "UCC650148", "UCC651779", "UCC657012"
 ),
 start_date = as.Date(c(
  "2024-07-29", "2024-10-24", "2025-03-22", "2024-03-08",
  "2024-03-11", "2024-12-29", "2024-05-23", "2024-05-28",
  "2025-01-14", "2024-01-29", "2024-10-03", "2024-08-07", "2024-06-25",
  "2024-03-04", "2024-01-29", "2024-01-29", "2024-02-18"
 )),
 end_date = as.Date(c(
  "2024-09-29", "2025-03-13", "2025-04-28", "2024-05-30",
```

```
   "2024-08-22", "2025-02-06", "2024-08-19", "2024-07-29",
   "2025-02-11", "2024-07-04", "2024-11-10", "2024-09-09", "2024-09-16",
   "2024-04-03", "2024-03-16", "2024-03-28", "2024-03-20"
  ))
)

all_timeline_results <- create_all_timeline_plots(research_ids_with_dates, pause_between = TRUE)
```

Script 11

```
rm(list = ls(pattern = "^(menstrual_.*_boxplot|phase_.*_boxplot|premenstrual_.*|postmenstrual_.*)"))
rm(list = ls(pattern =
"^(phase_.*_analysis|menstrual_.*_analysis|cycle_.*_phases|phase_boundaries)$"))
rm(list = ls(pattern =
"^(.*_phase_boundaries|define_.*_boundaries|phase_.*_definition|.*_phase_data)$"))
rm(list = ls(pattern =
"^(selected_participants|selected_timelines|participant_.*_cycles|.*_timeline_duration)$"))
rm(list = ls(pattern = "^(.*_burden_.*|phase_.*_severity|menstrual_.*_burden|daily_.*_phase)$"))
rm(list = ls(pattern = "^(phase_.*_comparison|menstrual_.*_stats|pre_.*_post|wilcoxon_.*_phase)$"))
rm(list = ls(pattern =
"^(define_.*_boundaries|calculate_.*_phase|analyze_.*_menstrual|create_.*_phase)"))
rm(list = ls(pattern =
"^(phase_.*_plot|menstrual_.*_chart|pre_.*_plot|post_.*_plot|phase_.*_visualization)$"))
rm(list = ls(pattern =
"^(complete_.*_participants|.*_complete_case|participant_.*_complete|complete_.*_analysis)$"))
rm(list = ls(pattern = "^(global_.*_burden|all_.*_burden|burden_.*_global|.*_global_burden)$"))
required_vars <- c("multi_cycle_merged_data", "single_cycle_merged_data", "no_period_final",
"symptom_categories")

missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 stop("Required variables from previous scripts missing: ", paste(missing_vars, collapse = ", "), "\n",
    "Please run Scripts 1-10 first.")
}

if(!exists("multi_cycle_participants") || !exists("single_cycle_participants")
|| !exists("no_period_participants")) {
 stop("Participant classification variables missing. Please run Scripts 1 and 2 first.")
}

script_10_vars <- c("research_ids_with_dates", "all_available_participants")

missing_script_10_vars <- c()
for(var in script_10_vars) {
 if(!exists(var)) {
  missing_script_10_vars <- c(missing_script_10_vars, var)
 }
}

if(length(missing_script_10_vars) > 0) {
 stop("Required variables from Script 10 missing: ", paste(missing_script_10_vars, collapse = ", "), "\n",
    "Please run Script 10 (Selected ID Timelines) first.")
```

```
}

required_packages <- c("ggplot2", "dplyr", "tidyr", "lubridate")
missing_packages <- c()

for(pkg in required_packages) {
  if(!require(pkg, character.only = TRUE, quietly = TRUE)) {
    missing_packages <- c(missing_packages, pkg)
  }
}

if(length(missing_packages) > 0) {
  stop("Required packages for menstrual phase boxplot analysis missing: ", paste(missing_packages,
collapse = ", "))
}

if(!"symptom_type" %in% colnames(symptom_categories)) {
  stop("Symptom categories missing symptom_type column. Cannot classify Motor vs Non-Motor
symptoms for phase analysis.")
}

motor_symptoms_available <- sum(symptom_categories$symptom_type == "Motor", na.rm = TRUE)
nonmotor_symptoms_available <- sum(symptom_categories$symptom_type == "Non-Motor", na.rm =
TRUE)

if(motor_symptoms_available == 0 || nonmotor_symptoms_available == 0) {
  stop("Insufficient motor or non-motor symptoms in categories. Need both types for menstrual phase
analysis.")
}

multi_flow_count <- sum(multi_cycle_merged_data$Flow != "No Entry", na.rm = TRUE)
single_flow_count <- sum(single_cycle_merged_data$Flow != "No Entry", na.rm = TRUE)

total_flow_entries <- multi_flow_count + single_flow_count

if(total_flow_entries == 0) {
  stop("No menstrual flow data found. Menstrual phase analysis requires period tracking data.")
}

multi_symptom_count <- sum(multi_cycle_merged_data$Symptom != "No Entry", na.rm = TRUE)
single_symptom_count <- sum(single_cycle_merged_data$Symptom != "No Entry", na.rm = TRUE)
no_period_symptom_count <- sum(no_period_final$Symptom != "No Entry", na.rm = TRUE)

total_symptom_entries <- multi_symptom_count + single_symptom_count + no_period_symptom_count

if(total_symptom_entries == 0) {
  stop("No symptom data found across any datasets. Cannot perform menstrual phase analysis.")
}
```

```
if(!all(c("research_id", "start_date", "end_date") %in% colnames(research_ids_with_dates))) {
  stop("research_ids_with_dates from Script 10 missing required columns (research_id, start_date,
end_date).")
}

script_10_participant_count <- length(unique(research_ids_with_dates$research_id))

if(script_10_participant_count < 5) {
  warning("Only ", script_10_participant_count, " participants found in research_ids_with_dates.
Menstrual phase analysis may be limited.")
}

multi_cycle_participant_count <- length(unique(multi_cycle_merged_data$research_id))

if(multi_cycle_participant_count == 0) {
  stop("No multi-cycle participants found. Menstrual phase analysis requires participants with multiple
menstrual cycles.")
}

participants_with_periods <-
unique(multi_cycle_merged_data$research_id[multi_cycle_merged_data$Flow != "No Entry"])
participants_with_symptoms <-
unique(multi_cycle_merged_data$research_id[multi_cycle_merged_data$Symptom != "No Entry"])
participants_with_both <- intersect(participants_with_periods, participants_with_symptoms)

if(length(participants_with_both) == 0) {
  stop("No participants found with both period tracking AND symptom data. Cannot perform menstrual
phase analysis.")
}

cat(" 🧹 Environment cleaned - Script 11 variables cleared, previous script data preserved\n")
cat(" 📊 Script 1-10 variables verified and found\n")
cat(" 👥 Total participants available for phase analysis:", multi_cycle_participant_count, "\n")
cat(" 👥 Script 10 participants with date ranges:", script_10_participant_count, "\n")
cat(" 👥 Participants with both periods and symptoms:", length(participants_with_both), "\n")
cat(" 🔬 Motor symptoms available:", motor_symptoms_available, "\n")
cat(" 🔬 Non-motor symptoms available:", nonmotor_symptoms_available, "\n")
cat(" 📈 Total symptom entries across all datasets:", total_symptom_entries, "\n")
cat(" 🩸 Total menstrual flow entries:", total_flow_entries, "\n")
cat(" 🔄 Ready for menstrual phase boxplot analysis\n\n")

cat("=== STEP 1: PARTICIPANT SETUP AND TIMELINE EXTRACTION ===\n")

selected_participants <- c(
  "UCC040405", "UCC041330", "UCC075P63", "UCC0772NS",
  "UCC0774NJ", "UCC0783SU", "UCC0786JF", "UCC0787HP",
  "UCC0793JL", "UCC176088", "UCC180701", "UCC2502K0", "UCC626142",
  "UCC6473P8", "UCC650148", "UCC651779", "UCC657012"
```

```
)

cat("Selected participants for menstrual phase analysis:\n")
cat(paste(selected_participants, collapse = ", "), "\n")
cat("Total participants:", length(selected_participants), "\n\n")

selected_timelines <- research_ids_with_dates %>%
 filter(research_id %in% selected_participants) %>%
 arrange(research_id)

cat("=== TIMELINE EXTRACTION RESULTS ===\n")
cat("Participants found in research_ids_with_dates:", nrow(selected_timelines), "\n")

found_participants <- selected_timelines$research_id
missing_participants <- setdiff(selected_participants, found_participants)

if(length(missing_participants) > 0) {
 cat("WARNING: Missing participants not found in research_ids_with_dates:\n")
 cat(paste(missing_participants, collapse = ", "), "\n")
} else {
 cat("✓ All selected participants found in research_ids_with_dates\n")
}

cat("\n=== SELECTED TIMELINES SUMMARY ===\n")
print(selected_timelines)

selected_timelines_with_duration <- selected_timelines %>%
 mutate(
  duration_days = as.numeric(end_date - start_date) + 1,
  start_date_char = as.character(start_date),
  end_date_char = as.character(end_date)
 ) %>%
 select(research_id, start_date_char, end_date_char, duration_days)

cat("\n=== TIMELINE DURATIONS ===\n")
print(selected_timelines_with_duration)

cat("\nTimeline duration summary:\n")
cat("Mean duration:", round(mean(selected_timelines_with_duration$duration_days), 1), "days\n")
cat("Range:", min(selected_timelines_with_duration$duration_days), "to",
  max(selected_timelines_with_duration$duration_days), "days\n")

cat("\n=== STEP 1 COMPLETE ===\n")
cat("Ready to proceed with cycle identification within these timelines\n")

cat("=== STEP 2: CYCLE IDENTIFICATION WITHIN TIMELINES ===\n")

identify_cycles_for_participant <- function(research_id, start_date, end_date) {
```

```
 cat("\nProcessing cycles for", research_id, "(", as.character(start_date), "to", as.character(end_date),
")...\n")

 participant_periods <- mini_period_data %>%
  filter(research_id == !!research_id) %>%
  filter(`Period Entry Date` >= start_date & `Period Entry Date` <= end_date) %>%
  arrange(`Period Entry Date`)

 cat("  Period entries within timeline:", nrow(participant_periods), "\n")

 if(nrow(participant_periods) < 3) {
  cat("  ⚠ Insufficient period data (need ≥3 consecutive days)\n")
  return(data.frame(
    research_id = character(0),
    cycle_id = integer(0),
    cycle_start = as.Date(character(0)),
    cycle_end = as.Date(character(0)),
    bleeding_days = integer(0)
  ))
 }

 period_dates <- sort(unique(participant_periods$`Period Entry Date`))
 cat("  Unique period dates:", length(period_dates), "\n")
 cat("  Date range:", as.character(min(period_dates)), "to", as.character(max(period_dates)), "\n")

 cycles <- list()
 current_cycle <- c(period_dates[1])
 cycle_count <- 0

 for(i in 2:length(period_dates)) {
  date_diff <- as.numeric(period_dates[i] - period_dates[i-1])

  if(date_diff <= 1) {
    current_cycle <- c(current_cycle, period_dates[i])
  } else {
    if(length(current_cycle) >= 3) {
      cycle_count <- cycle_count + 1
      cycles[[cycle_count]] <- current_cycle
    }
    current_cycle <- c(period_dates[i])
  }
 }

 if(length(current_cycle) >= 3) {
  cycle_count <- cycle_count + 1
  cycles[[cycle_count]] <- current_cycle
 }

 cat("  Valid cycles found (≥3 consecutive days):", length(cycles), "\n")
```

```
 if(length(cycles) == 0) {
  cat("  ⚠ No valid cycles found\n")
  return(data.frame(
   research_id = character(0),
   cycle_id = integer(0),
   cycle_start = as.Date(character(0)),
   cycle_end = as.Date(character(0)),
   bleeding_days = integer(0)
  ))
 }

 cycle_summary <- data.frame(
  research_id = research_id,
  cycle_id = 1:length(cycles),
  cycle_start = sapply(cycles, min),
  cycle_end = sapply(cycles, max),
  bleeding_days = sapply(cycles, length)
 ) %>%
  mutate(
   cycle_start = as.Date(cycle_start, origin = "1970-01-01"),
   cycle_end = as.Date(cycle_end, origin = "1970-01-01")
  )

 cat("  Cycle summary:\n")
 for(i in 1:nrow(cycle_summary)) {
  cat("   Cycle", i, ":", as.character(cycle_summary$cycle_start[i]),
      "to", as.character(cycle_summary$cycle_end[i]),
      "(", cycle_summary$bleeding_days[i], "days )\n")
 }

 return(cycle_summary)
}

all_cycles <- data.frame()

for(i in 1:nrow(selected_timelines)) {
 participant_cycles <- identify_cycles_for_participant(
  selected_timelines$research_id[i],
  selected_timelines$start_date[i],
  selected_timelines$end_date[i]
 )

 if(nrow(participant_cycles) > 0) {
  all_cycles <- rbind(all_cycles, participant_cycles)
 }
}

cat("\n=== OVERALL CYCLE IDENTIFICATION RESULTS ===\n")
```

```
cat("Total participants processed:", nrow(selected_timelines), "\n")
cat("Participants with valid cycles:", length(unique(all_cycles$research_id)), "\n")
cat("Total valid cycles found:", nrow(all_cycles), "\n")

if(nrow(all_cycles) > 0) {
 cat("\n=== ALL CYCLES SUMMARY ===\n")
 print(all_cycles)

 cycles_per_participant <- all_cycles %>%
  group_by(research_id) %>%
  summarise(n_cycles = n(), .groups = 'drop') %>%
  arrange(desc(n_cycles))

 cat("\n=== CYCLES PER PARTICIPANT ===\n")
 print(cycles_per_participant)

 cat("\nCycle count distribution:\n")
 cat("Mean cycles per participant:", round(mean(cycles_per_participant$n_cycles), 1), "\n")
 cat("Range:", min(cycles_per_participant$n_cycles), "to", max(cycles_per_participant$n_cycles),
"cycles\n")

 cat("\nBleeding days per cycle:\n")
 cat("Mean:", round(mean(all_cycles$bleeding_days), 1), "days\n")
 cat("Range:", min(all_cycles$bleeding_days), "to", max(all_cycles$bleeding_days), "days\n")

} else {
 cat("⚠ No valid cycles found for any participants\n")
}

participants_with_cycles <- unique(all_cycles$research_id)
participants_without_cycles <- setdiff(selected_participants, participants_with_cycles)

if(length(participants_without_cycles) > 0) {
 cat("\n ⚠️  PARTICIPANTS WITHOUT VALID CYCLES:\n")
 cat(paste(participants_without_cycles, collapse = ", "), "\n")
}

cat("\n=== STEP 2 COMPLETE ===\n")
cat("Cycle identification finished. Ready for phase boundary definition.\n")

cat("=== STEP 3: DEFINE PHASE BOUNDARIES FOR EACH CYCLE ===\n")

define_phase_boundaries_for_participant <- function(research_id, participant_cycles, timeline_start,
timeline_end) {
 cat("\nDefining phases for", research_id, "(", nrow(participant_cycles), "cycles )...\n")

 participant_cycles <- participant_cycles %>%
  arrange(cycle_start)
```

```
phase_boundaries <- data.frame()

for(i in 1:nrow(participant_cycles)) {
 cycle <- participant_cycles[i, ]
 cat("  Cycle", i, "(", as.character(cycle$cycle_start), "to", as.character(cycle$cycle_end), "):\n")

 premenstrual_end <- cycle$cycle_start - 1
 premenstrual_start <- cycle$cycle_start - 5

 if(premenstrual_start < timeline_start) {
  premenstrual_start <- timeline_start
  cat("    ⚠️  Premenstrual phase truncated by timeline start\n")
 }

 if(premenstrual_start <= premenstrual_end && premenstrual_end >= timeline_start) {
  premenstrual_days <- as.numeric(premenstrual_end - premenstrual_start) + 1
  cat("    Premenstrual:", as.character(premenstrual_start), "to",
      as.character(premenstrual_end), "(", premenstrual_days, "days )\n")

  phase_boundaries <- rbind(phase_boundaries, data.frame(
   research_id = research_id,
   cycle_id = cycle$cycle_id,
   phase = "Premenstrual",
   phase_start = premenstrual_start,
   phase_end = premenstrual_end,
   phase_days = premenstrual_days
  ))
 } else {
  cat("    ⚠ No premenstrual phase (insufficient days)\n")
 }

 menstrual_days <- cycle$bleeding_days
 cat("    Menstrual:", as.character(cycle$cycle_start), "to",
     as.character(cycle$cycle_end), "(", menstrual_days, "days )\n")

 phase_boundaries <- rbind(phase_boundaries, data.frame(
  research_id = research_id,
  cycle_id = cycle$cycle_id,
  phase = "Menstrual",
  phase_start = cycle$cycle_start,
  phase_end = cycle$cycle_end,
  phase_days = menstrual_days
 ))

 postmenstrual_start <- cycle$cycle_end + 1

 if(i < nrow(participant_cycles)) {
  next_cycle <- participant_cycles[i + 1, ]
  postmenstrual_end <- next_cycle$cycle_start - 6
```

```
  } else {
    postmenstrual_end <- timeline_end
    cat("    ⚠️ Last cycle: post-menstrual goes to timeline end\n")
  }

  if(postmenstrual_end > timeline_end) {
    postmenstrual_end <- timeline_end
    cat("    ⚠️ Post-menstrual phase truncated by timeline end\n")
  }

  if(postmenstrual_start <= postmenstrual_end && postmenstrual_start <= timeline_end) {
    postmenstrual_days <- as.numeric(postmenstrual_end - postmenstrual_start) + 1
    cat("    Post-menstrual:", as.character(postmenstrual_start), "to",
        as.character(postmenstrual_end), "(", postmenstrual_days, "days )\n")

    phase_boundaries <- rbind(phase_boundaries, data.frame(
      research_id = research_id,
      cycle_id = cycle$cycle_id,
      phase = "Post-menstrual",
      phase_start = postmenstrual_start,
      phase_end = postmenstrual_end,
      phase_days = postmenstrual_days
    ))
  } else {
    cat("    ⚠ No post-menstrual phase (insufficient days)\n")
  }
 }

 return(phase_boundaries)
}

all_phase_boundaries <- data.frame()

for(research_id in unique(all_cycles$research_id)) {
 participant_cycles <- all_cycles %>%
  filter(research_id == !!research_id)

 timeline_info <- selected_timelines %>%
  filter(research_id == !!research_id)

 participant_phases <- define_phase_boundaries_for_participant(
  research_id,
  participant_cycles,
  timeline_info$start_date,
  timeline_info$end_date
 )

 if(nrow(participant_phases) > 0) {
```

```
   all_phase_boundaries <- rbind(all_phase_boundaries, participant_phases)
  }
}

cat("\n=== OVERALL PHASE BOUNDARY RESULTS ===\n")
cat("Total participants with phases:", length(unique(all_phase_boundaries$research_id)), "\n")
cat("Total phase periods defined:", nrow(all_phase_boundaries), "\n")

if(nrow(all_phase_boundaries) > 0) {
 all_phase_boundaries <- all_phase_boundaries %>%
  mutate(
   phase_start = as.Date(phase_start, origin = "1970-01-01"),
   phase_end = as.Date(phase_end, origin = "1970-01-01")
  )

 cat("\n=== ALL PHASE BOUNDARIES SUMMARY ===\n")
 print(all_phase_boundaries)

 phase_distribution <- all_phase_boundaries %>%
  count(phase, name = "count") %>%
  arrange(desc(count))

 cat("\n=== PHASE TYPE DISTRIBUTION ===\n")
 print(phase_distribution)

 phase_duration_stats <- all_phase_boundaries %>%
  group_by(phase) %>%
  summarise(
   n_phases = n(),
   mean_days = round(mean(phase_days), 1),
   median_days = median(phase_days),
   min_days = min(phase_days),
   max_days = max(phase_days),
   .groups = 'drop'
  )

 cat("\n=== PHASE DURATION STATISTICS ===\n")
 print(phase_duration_stats)

 phases_per_participant <- all_phase_boundaries %>%
  group_by(research_id, phase) %>%
  summarise(n_phase_periods = n(), .groups = 'drop') %>%
  pivot_wider(names_from = phase, values_from = n_phase_periods, values_fill = 0) %>%
  arrange(research_id)

 cat("\n=== PHASES PER PARTICIPANT ===\n")
 print(phases_per_participant)

} else {
```

```
 cat("⚠ No phase boundaries could be defined\n")
}

cat("\n=== STEP 3 COMPLETE ===\n")
cat("Phase boundary definition finished. Ready for symptom data collection.\n")

cat("=== STEP 4: COLLECT SYMPTOM DATA FOR EACH PHASE ===\n")

collect_phase_symptom_data <- function(research_id, cycle_id, phase, phase_start, phase_end) {
 participant_symptoms <- bind_rows(
  multi_cycle_merged_data %>%
   filter(research_id == !!research_id, Symptom != "No Entry") %>%
   mutate(source = "multi_cycle"),
  single_cycle_merged_data %>%
   filter(research_id == !!research_id, Symptom != "No Entry") %>%
   mutate(source = "single_cycle")
 )

 if(nrow(participant_symptoms) == 0) {
  return(data.frame())
 }

 phase_symptoms <- participant_symptoms %>%
  filter(dates >= phase_start & dates <= phase_end) %>%
  filter(Severity != "No Entry") %>%
  mutate(Severity = as.numeric(Severity)) %>%
  filter(!is.na(Severity))

 if(nrow(phase_symptoms) == 0) {
  return(data.frame())
 }

 phase_symptoms_classified <- phase_symptoms %>%
  left_join(
   symptom_categories %>% select(metric_name_abbr, symptom_type),
   by = c("Symptom" = "metric_name_abbr")
  ) %>%
  filter(!is.na(symptom_type)) %>%
  filter(symptom_type %in% c("Motor", "Non-Motor"))

 if(nrow(phase_symptoms_classified) == 0) {
  return(data.frame())
 }

 phase_symptoms_final <- phase_symptoms_classified %>%
  mutate(
   phase_cycle_id = cycle_id,
   phase_name = phase
  ) %>%
```

```
   select(research_id, phase_cycle_id, phase_name, dates, Symptom, Severity, symptom_type)

  return(phase_symptoms_final)
}

all_phase_symptom_data <- data.frame()
phase_summary_stats <- data.frame()

cat("\nProcessing symptom data for all phases...\n")

for(i in 1:nrow(all_phase_boundaries)) {
  phase_info <- all_phase_boundaries[i, ]

  cat("  Processing", phase_info$research_id, "Cycle", phase_info$cycle_id,
      phase_info$phase, "(", as.character(phase_info$phase_start), "to",
      as.character(phase_info$phase_end), ")...")

  phase_data <- collect_phase_symptom_data(
    phase_info$research_id,
    phase_info$cycle_id,
    phase_info$phase,
    phase_info$phase_start,
    phase_info$phase_end
  )

  motor_count <- sum(phase_data$symptom_type == "Motor", na.rm = TRUE)
  nonmotor_count <- sum(phase_data$symptom_type == "Non-Motor", na.rm = TRUE)
  total_count <- nrow(phase_data)

  cat(" Found", total_count, "symptoms (", motor_count, "Motor,", nonmotor_count, "Non-Motor)\n")

  phase_summary_stats <- rbind(phase_summary_stats, data.frame(
    research_id = phase_info$research_id,
    cycle_id = phase_info$cycle_id,
    phase = phase_info$phase,
    phase_days = phase_info$phase_days,
    total_symptoms = total_count,
    motor_symptoms = motor_count,
    nonmotor_symptoms = nonmotor_count
  ))

  if(nrow(phase_data) > 0) {
    all_phase_symptom_data <- rbind(all_phase_symptom_data, phase_data)
  }
}

cat("\n=== SYMPTOM DATA COLLECTION RESULTS ===\n")
cat("Total phase periods processed:", nrow(all_phase_boundaries), "\n")
```

```
cat("Phase periods with symptom data:",
nrow(phase_summary_stats[phase_summary_stats$total_symptoms > 0, ]), "\n")
cat("Total symptom entries collected:", nrow(all_phase_symptom_data), "\n")

if(nrow(all_phase_symptom_data) > 0) {
 overall_symptom_distribution <- all_phase_symptom_data %>%
  count(symptom_type, name = "count") %>%
  arrange(desc(count))

 cat("\n=== OVERALL SYMPTOM TYPE DISTRIBUTION ===\n")
 print(overall_symptom_distribution)

 phase_symptom_distribution <- all_phase_symptom_data %>%
  group_by(phase_name, symptom_type) %>%
  summarise(n_symptoms = n(), .groups = 'drop') %>%
  pivot_wider(names_from = symptom_type, values_from = n_symptoms, values_fill = 0) %>%
  mutate(Total = Motor + `Non-Motor`) %>%
  arrange(desc(Total))

 cat("\n=== SYMPTOM DATA BY PHASE TYPE ===\n")
 print(phase_symptom_distribution)

 participant_symptom_distribution <- all_phase_symptom_data %>%
  group_by(research_id, symptom_type) %>%
  summarise(n_symptoms = n(), .groups = 'drop') %>%
  pivot_wider(names_from = symptom_type, values_from = n_symptoms, values_fill = 0) %>%
  mutate(Total = Motor + `Non-Motor`) %>%
  arrange(desc(Total))

 cat("\n=== SYMPTOM DATA BY PARTICIPANT ===\n")
 print(participant_symptom_distribution)

 phase_coverage_stats <- phase_summary_stats %>%
  group_by(phase) %>%
  summarise(
   n_phases = n(),
   phases_with_data = sum(total_symptoms > 0),
   coverage_percent = round(phases_with_data / n_phases * 100, 1),
   mean_symptoms_per_phase = round(mean(total_symptoms), 1),
   total_symptoms = sum(total_symptoms),
   .groups = 'drop'
  ) %>%
  arrange(desc(total_symptoms))

 cat("\n=== PHASE COVERAGE STATISTICS ===\n")
 print(phase_coverage_stats)

 severity_distribution <- all_phase_symptom_data %>%
  count(Severity, name = "count") %>%
```

```
  mutate(percentage = round(count / sum(count) * 100, 1)) %>%
  arrange(Severity)

 cat("\n=== SEVERITY DISTRIBUTION ===\n")
 print(severity_distribution)

 participant_phase_data_check <- all_phase_symptom_data %>%
  group_by(research_id, phase_name, symptom_type) %>%
  summarise(n_symptoms = n(), .groups = 'drop') %>%
  pivot_wider(names_from = c(phase_name, symptom_type),
          values_from = n_symptoms, values_fill = 0)

 cat("\n=== PARTICIPANT DATA AVAILABILITY MATRIX ===\n")
 print(participant_phase_data_check)

} else {
 cat("⚠ No symptom data found for any phase periods\n")
}

cat("\n=== STEP 4 COMPLETE ===\n")
cat("Symptom data collection finished. Ready for individual boxplot creation.\n")

cat("=== STEP 5: CREATE INDIVIDUAL BOXPLOTS FOR EACH PARTICIPANT ===\n")

create_individual_phase_boxplot <- function(research_id, symptom_type_filter) {
 participant_data <- all_phase_symptom_data %>%
  filter(research_id == !!research_id, symptom_type == symptom_type_filter)

 if(nrow(participant_data) == 0) {
  cat(" ⚠ No", symptom_type_filter, "data for", research_id, "\n")
  return(NULL)
 }

 plot_data <- participant_data %>%
  mutate(
   phase_name = factor(phase_name, levels = c("Premenstrual", "Menstrual", "Post-menstrual"))
  )

 data_counts <- plot_data %>%
  count(phase_name, name = "n_points") %>%
  mutate(label = paste0("n=", n_points))

 total_points <- nrow(plot_data)
 severity_range <- paste0(min(plot_data$Severity), "-", max(plot_data$Severity))

 p <- ggplot(plot_data, aes(x = phase_name, y = Severity)) +
  geom_boxplot(
   alpha = 0.7,
   outlier.shape = NA,
```

```
  width = 0.6,
  fill = case_when(
    symptom_type_filter == "Motor" ~ "#1f77b4",
    symptom_type_filter == "Non-Motor" ~ "#ff7f0e"
  ),
  color = "black",
  linewidth = 0.8,
  fatten = 3,
  colour = "black"
) +

geom_jitter(
  width = 0.2,
  height = 0.05,
  alpha = 0.6,
  size = 2,
  color = "black"
) +

geom_text(
  data = data_counts,
  aes(x = phase_name, y = 3.3, label = label),
  size = 3.5,
  fontface = "bold"
) +

scale_y_continuous(
  breaks = c(0, 1, 2, 3),
  limits = c(-0.1, 3.5),
  labels = c("0", "1 (Mild)", "2 (Moderate)", "3 (Severe)")
) +

labs(
  title = paste0(research_id, ": ", symptom_type_filter, " Symptoms Across Menstrual Phases"),
  subtitle = paste0("Total data points: ", total_points, " | Severity range: ", severity_range),
  x = "Menstrual Phase",
  y = "Symptom Severity",
  caption = "Each dot represents one individual symptom entry"
) +

theme_minimal() +
theme(
  plot.title = element_text(face = "bold", size = 12, hjust = 0.5),
  plot.subtitle = element_text(size = 10, hjust = 0.5),
  plot.caption = element_text(size = 9, hjust = 0.5),
  axis.text.x = element_text(size = 11),
  axis.text.y = element_text(size = 10),
  axis.title = element_text(size = 11, face = "bold"),
  panel.grid.minor = element_blank()
```

```
  )

 return(p)
}

create_participant_boxplots <- function(research_id) {
 cat("\nCreating boxplots for", research_id, "...\n")

 participant_data <- all_phase_symptom_data %>%
  filter(research_id == !!research_id)

 if(nrow(participant_data) == 0) {
  cat("  ⚠ No data found for", research_id, "\n")
  return(list(motor_plot = NULL, nonmotor_plot = NULL))
 }

 data_summary <- participant_data %>%
  group_by(symptom_type, phase_name) %>%
  summarise(n_points = n(), .groups = 'drop') %>%
  pivot_wider(names_from = phase_name, values_from = n_points, values_fill = 0)

 cat("  Data summary:\n")
 print(data_summary)

 motor_plot <- create_individual_phase_boxplot(research_id, "Motor")
 if(!is.null(motor_plot)) {
  cat("  ✓ Motor symptoms plot created\n")
 }

 nonmotor_plot <- create_individual_phase_boxplot(research_id, "Non-Motor")
 if(!is.null(nonmotor_plot)) {
  cat("  ✓ Non-Motor symptoms plot created\n")
 }

 return(list(motor_plot = motor_plot, nonmotor_plot = nonmotor_plot))
}

all_participant_plots <- list()
participants_with_data <- unique(all_phase_symptom_data$research_id)

cat("\nCreating individual boxplots for", length(participants_with_data), "participants...\n")

for(research_id in participants_with_data) {
 participant_plots <- create_participant_boxplots(research_id)
 all_participant_plots[[research_id]] <- participant_plots

 if(!is.null(participant_plots$motor_plot)) {
  print(participant_plots$motor_plot)
 }
```

```
 if(!is.null(participant_plots$nonmotor_plot)) {
  print(participant_plots$nonmotor_plot)
 }
}

cat("\n=== INDIVIDUAL BOXPLOT CREATION SUMMARY ===\n")

motor_plots_created <- 0
nonmotor_plots_created <- 0
total_plots_created <- 0

for(research_id in names(all_participant_plots)) {
 plots <- all_participant_plots[[research_id]]

 has_motor <- !is.null(plots$motor_plot)
 has_nonmotor <- !is.null(plots$nonmotor_plot)

 if(has_motor) motor_plots_created <- motor_plots_created + 1
 if(has_nonmotor) nonmotor_plots_created <- nonmotor_plots_created + 1

 total_plots_created <- total_plots_created + sum(has_motor, has_nonmotor)

 cat(research_id, ": Motor =", ifelse(has_motor, "✓", "⚠"),
    "| Non-Motor =", ifelse(has_nonmotor, "✓", "⚠"), "\n")
}

cat("\nPlot creation summary:\n")
cat("- Total participants processed:", length(participants_with_data), "\n")
cat("- Motor symptom plots created:", motor_plots_created, "\n")
cat("- Non-Motor symptom plots created:", nonmotor_plots_created, "\n")
cat("- Total individual plots created:", total_plots_created, "\n")

participants_motor_only <- c()
participants_nonmotor_only <- c()
participants_both <- c()

for(research_id in names(all_participant_plots)) {
 plots <- all_participant_plots[[research_id]]
 has_motor <- !is.null(plots$motor_plot)
 has_nonmotor <- !is.null(plots$nonmotor_plot)

 if(has_motor && has_nonmotor) {
  participants_both <- c(participants_both, research_id)
 } else if(has_motor && !has_nonmotor) {
  participants_motor_only <- c(participants_motor_only, research_id)
 } else if(!has_motor && has_nonmotor) {
  participants_nonmotor_only <- c(participants_nonmotor_only, research_id)
 }
```

```
}

cat("\nParticipant plot coverage:\n")
cat("- Both Motor & Non-Motor plots:", length(participants_both),
  ifelse(length(participants_both) > 0, paste0(" (", paste(participants_both, collapse = ", "), ")"), ""), "\n")
cat("- Motor plots only:", length(participants_motor_only),
  ifelse(length(participants_motor_only) > 0, paste0(" (", paste(participants_motor_only, collapse = ", "),
")"), ""), "\n")
cat("- Non-Motor plots only:", length(participants_nonmotor_only),
  ifelse(length(participants_nonmotor_only) > 0, paste0(" (", paste(participants_nonmotor_only, collapse
= ", "), ")"), ""), "\n")

cat("\n=== STEP 5 COMPLETE ===\n")
cat("Individual boxplot creation finished!\n")
cat("All plots are stored in 'all_participant_plots' list for future reference.\n")
```

Script 11.1

```
rm(list = ls(pattern =
"^(all_phase_global_burden_data|global_burden_.*|phase_.*_burden|.*_global_burden)$"))
rm(list = ls(pattern = "^(collect_.*_burden|phase_burden_.*|burden_.*_data|.*_burden_stats)$"))
rm(list = ls(pattern =
"^(burden_.*_summary|global_.*_summary|.*_burden_distribution|burden_.*_validation)$"))
rm(list = ls(pattern =
"^(.*_global_burden_plots|global_burden_.*_plot|burden_.*_boxplot|.*_burden_chart)$"))
rm(list = ls(pattern =
"^(complete_.*_burden|.*_complete_case|participant_.*_complete|complete_.*_global)$"))
rm(list = ls(pattern =
"^(daily_.*_burden|phase_.*_daily|.*_daily_burden|daily_global_.*|burden_.*_daily)$"))
rm(list = ls(pattern =
"^(participant_.*_burden|.*_participant_burden|individual_.*_burden|burden_.*_participant)$"))
rm(list = ls(pattern =
"^(create_.*_global_burden|calculate_.*_burden|display_.*_burden|identify_.*_burden)"))
rm(list = ls(pattern = "^(test_.*_burden|burden_.*_test|validation_.*_burden|burden_.*_validation)$"))
rm(list = ls(pattern =
"^(wilcoxon_.*_burden|paired_.*_burden|burden_.*_comparison|statistical_.*_burden)$"))

required_script_11_vars <- c("selected_participants", "selected_timelines", "all_phase_boundaries",
"symptom_categories")

missing_script_11_vars <- c()
for(var in required_script_11_vars) {
 if(!exists(var)) {
  missing_script_11_vars <- c(missing_script_11_vars, var)
 }
}

if(length(missing_script_11_vars) > 0) {
 stop("Required variables from Script 11 missing: ", paste(missing_script_11_vars, collapse = ", "), "\n",
    "Please run Script 11 (Premenstrual, Menstrual, Post-menstrual boxplots) first.")
}

required_vars <- c("multi_cycle_merged_data", "single_cycle_merged_data", "no_period_final",
"symptom_categories")

missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 stop("Required variables from previous scripts missing: ", paste(missing_vars, collapse = ", "), "\n",
    "Please run Scripts 1-11 first.")
}
```

```
if(!exists("multi_cycle_participants") || !exists("single_cycle_participants")
|| !exists("no_period_participants")) {
  stop("Participant classification variables missing. Please run Scripts 1 and 2 first.")
}

required_packages <- c("ggplot2", "dplyr", "tidyr", "lubridate")
missing_packages <- c()

for(pkg in required_packages) {
  if(!require(pkg, character.only = TRUE, quietly = TRUE)) {
    missing_packages <- c(missing_packages, pkg)
  }
}

if(length(missing_packages) > 0) {
  stop("Required packages for menstrual phase global burden analysis missing: ",
paste(missing_packages, collapse = ", "))
}

if(!all(c("research_id", "cycle_id", "phase", "phase_start", "phase_end", "phase_days") %in%
colnames(all_phase_boundaries))) {
  stop("all_phase_boundaries from Script 11 missing required columns. Cannot perform global burden
analysis.")
}

phase_boundary_count <- nrow(all_phase_boundaries)
participants_with_phases <- length(unique(all_phase_boundaries$research_id))

if(phase_boundary_count == 0) {
  stop("No phase boundaries found from Script 11. Cannot calculate global burden.")
}

if(participants_with_phases == 0) {
  stop("No participants with phase boundaries found. Cannot perform global burden analysis.")
}

available_phases <- unique(all_phase_boundaries$phase)
required_phases <- c("Premenstrual", "Menstrual", "Post-menstrual")
missing_phases <- setdiff(required_phases, available_phases)

if(length(missing_phases) > 0) {
  stop("Missing required phase types from Script 11: ", paste(missing_phases, collapse = ", "),
       "\nGlobal burden analysis requires all three menstrual phases.")
}

selected_participant_count <- length(selected_participants)
phase_participants <- unique(all_phase_boundaries$research_id)
participants_match <- all(selected_participants %in% phase_participants)
```

```
if(!participants_match) {
  missing_in_phases <- setdiff(selected_participants, phase_participants)
  warning("Some selected participants not found in phase boundaries: ",
          paste(missing_in_phases, collapse = ", "))
}

if(!"symptom_type" %in% colnames(symptom_categories)) {
  stop("Symptom categories missing symptom_type column. Cannot classify Motor vs Non-Motor
symptoms for global burden analysis.")
}

motor_symptoms_available <- sum(symptom_categories$symptom_type == "Motor", na.rm = TRUE)
nonmotor_symptoms_available <- sum(symptom_categories$symptom_type == "Non-Motor", na.rm =
TRUE)

if(motor_symptoms_available == 0 || nonmotor_symptoms_available == 0) {
  stop("Insufficient motor or non-motor symptoms in categories. Need both types for global burden
analysis.")
}

multi_symptom_count <- sum(multi_cycle_merged_data$Symptom != "No Entry", na.rm = TRUE)
single_symptom_count <- sum(single_cycle_merged_data$Symptom != "No Entry", na.rm = TRUE)
no_period_symptom_count <- sum(no_period_final$Symptom != "No Entry", na.rm = TRUE)

total_symptom_entries <- multi_symptom_count + single_symptom_count + no_period_symptom_count

if(total_symptom_entries == 0) {
  stop("No symptom data found across any datasets. Cannot calculate global burden.")
}

participants_with_symptoms <- unique(c(
  multi_cycle_merged_data$research_id[multi_cycle_merged_data$Symptom != "No Entry"],
  single_cycle_merged_data$research_id[single_cycle_merged_data$Symptom != "No Entry"],
  no_period_final$research_id[no_period_final$Symptom != "No Entry"]
))

participants_with_both <- intersect(phase_participants, participants_with_symptoms)

if(length(participants_with_both) == 0) {
  stop("No participants found with both phase boundaries AND symptom data. Cannot calculate global
burden.")
}

required_variables <- c("selected_participants", "selected_timelines", "all_phase_boundaries",
"symptom_categories")
missing_variables <- c()

for(var in required_variables) {
```

```
 if(!exists(var)) {
  missing_variables <- c(missing_variables, var)
 }
}

if(length(missing_variables) > 0) {
 stop("Missing required variables from Script 11. Please run Script 11 first. Missing:",
paste(missing_variables, collapse = ", "))
}

print(head(all_phase_boundaries, 6))

collect_phase_global_burden_data <- function(research_id, cycle_id, phase, phase_start, phase_end) {

 participant_symptoms <- bind_rows(
  multi_cycle_merged_data %>%
   filter(research_id == !!research_id, Symptom != "No Entry") %>%
   mutate(source = "multi_cycle"),
  single_cycle_merged_data %>%
   filter(research_id == !!research_id, Symptom != "No Entry") %>%
   mutate(source = "single_cycle")
 )

 if(nrow(participant_symptoms) == 0) {
  return(data.frame())
 }

 phase_symptoms <- participant_symptoms %>%
  filter(dates >= phase_start & dates <= phase_end) %>%
  filter(Severity != "No Entry") %>%
  mutate(Severity = as.numeric(Severity)) %>%
  filter(!is.na(Severity))

 if(nrow(phase_symptoms) == 0) {
  return(data.frame())
 }

 phase_symptoms_classified <- phase_symptoms %>%
  left_join(
   symptom_categories %>% select(metric_name_abbr, symptom_type),
   by = c("Symptom" = "metric_name_abbr")
  ) %>%
  filter(!is.na(symptom_type)) %>%
  filter(symptom_type %in% c("Motor", "Non-Motor"))

 if(nrow(phase_symptoms_classified) == 0) {
  return(data.frame())
 }
```

```
 daily_burden_by_type <- phase_symptoms_classified %>%
  group_by(dates, symptom_type) %>%
  summarise(
   daily_severity = sum(Severity, na.rm = TRUE),
   symptom_count = n(),
   .groups = 'drop'
  )

 global_burden_by_type <- daily_burden_by_type %>%
  mutate(
   research_id = research_id,
   phase_cycle_id = cycle_id,
   phase_name = phase,
   days_with_data = 1,
   total_severity = daily_severity,
   global_burden_score = daily_severity
  ) %>%
  select(research_id, phase_cycle_id, phase_name, symptom_type,
      days_with_data, total_severity, global_burden_score)

 return(global_burden_by_type)
}

test_phase <- all_phase_boundaries[1, ]

test_result <- collect_phase_global_burden_data(
 test_phase$research_id,
 test_phase$cycle_id,
 test_phase$phase,
 test_phase$phase_start,
 test_phase$phase_end
)

if(nrow(test_result) > 0) {
 print(test_result)
}

all_phase_global_burden_data <- data.frame()
global_burden_summary_stats <- data.frame()

for(i in 1:nrow(all_phase_boundaries)) {
 phase_info <- all_phase_boundaries[i, ]

 phase_burden_data <- collect_phase_global_burden_data(
  phase_info$research_id,
  phase_info$cycle_id,
  phase_info$phase,
  phase_info$phase_start,
  phase_info$phase_end
```

```
  )

  motor_found <- sum(phase_burden_data$symptom_type == "Motor", na.rm = TRUE)
  nonmotor_found <- sum(phase_burden_data$symptom_type == "Non-Motor", na.rm = TRUE)
  total_found <- nrow(phase_burden_data)

  global_burden_summary_stats <- rbind(global_burden_summary_stats, data.frame(
    research_id = phase_info$research_id,
    cycle_id = phase_info$cycle_id,
    phase = phase_info$phase,
    phase_days = phase_info$phase_days,
    total_records = total_found,
    motor_records = motor_found,
    nonmotor_records = nonmotor_found,
    has_motor = motor_found > 0,
    has_nonmotor = nonmotor_found > 0,
    has_both = (motor_found > 0) & (nonmotor_found > 0)
  ))

  if(nrow(phase_burden_data) > 0) {
    all_phase_global_burden_data <- rbind(all_phase_global_burden_data, phase_burden_data)
  }
}

if(nrow(all_phase_global_burden_data) > 0) {

  overall_burden_distribution <- all_phase_global_burden_data %>%
    count(symptom_type, name = "count") %>%
    arrange(desc(count))

  print(overall_burden_distribution)

  phase_burden_distribution <- all_phase_global_burden_data %>%
    group_by(phase_name, symptom_type) %>%
    summarise(n_records = n(), .groups = 'drop') %>%
    pivot_wider(names_from = symptom_type, values_from = n_records, values_fill = 0) %>%
    mutate(Total = Motor + `Non-Motor`) %>%
    arrange(desc(Total))

  print(phase_burden_distribution)
}

print(head(all_phase_global_burden_data, 10))

missing_check <- sapply(all_phase_global_burden_data, function(x) sum(is.na(x)))
print(missing_check)

burden_stats <- all_phase_global_burden_data %>%
  group_by(symptom_type) %>%
```

```
 summarise(
  n_records = n(),
  min_burden = round(min(global_burden_score, na.rm = TRUE), 3),
  max_burden = round(max(global_burden_score, na.rm = TRUE), 3),
  mean_burden = round(mean(global_burden_score, na.rm = TRUE), 3),
  median_burden = round(median(global_burden_score, na.rm = TRUE), 3),
  sd_burden = round(sd(global_burden_score, na.rm = TRUE), 3),
  .groups = 'drop'
 )

print(burden_stats)

phase_burden_stats <- all_phase_global_burden_data %>%
 group_by(phase_name, symptom_type) %>%
 summarise(
  n_records = n(),
  mean_burden = round(mean(global_burden_score, na.rm = TRUE), 3),
  median_burden = round(median(global_burden_score, na.rm = TRUE), 3),
  min_burden = round(min(global_burden_score, na.rm = TRUE), 3),
  max_burden = round(max(global_burden_score, na.rm = TRUE), 3),
  .groups = 'drop'
 ) %>%
 arrange(phase_name, symptom_type)

print(phase_burden_stats)

participant_summary <- all_phase_global_burden_data %>%
 group_by(research_id) %>%
 summarise(
  total_records = n(),
  motor_records = sum(symptom_type == "Motor"),
  nonmotor_records = sum(symptom_type == "Non-Motor"),
  phases_represented = n_distinct(phase_name),
  cycles_represented = n_distinct(phase_cycle_id),
  .groups = 'drop'
 ) %>%
 arrange(desc(total_records))

print(participant_summary)

days_stats <- all_phase_global_burden_data %>%
 group_by(symptom_type) %>%
 summarise(
  min_days = min(days_with_data, na.rm = TRUE),
  max_days = max(days_with_data, na.rm = TRUE),
  mean_days = round(mean(days_with_data, na.rm = TRUE), 1),
  median_days = median(days_with_data, na.rm = TRUE),
  .groups = 'drop'
 )
```

```
print(days_stats)

calculation_check <- all_phase_global_burden_data %>%
 mutate(
  calculated_burden = total_severity / days_with_data,
  difference = abs(global_burden_score - calculated_burden),
  consistent = difference < 0.001
 )

if(sum(!calculation_check$consistent) > 0) {
 print(calculation_check[!calculation_check$consistent, ])
}

create_individual_global_burden_boxplot <- function(research_id, symptom_type_filter) {

 participant_data <- all_phase_global_burden_data %>%
  filter(research_id == !!research_id, symptom_type == symptom_type_filter)

 if(nrow(participant_data) == 0) {
  return(NULL)
 }

 participant_data <- participant_data %>%
  mutate(phase_name = factor(phase_name, levels = c("Premenstrual", "Menstrual", "Post-menstrual")))

 total_points <- nrow(participant_data)
 burden_range <- paste0(round(min(participant_data$global_burden_score), 1),
                " - ", round(max(participant_data$global_burden_score), 1))

 if(symptom_type_filter == "Motor") {
  fill_color <- "#1f77b4"
  title_text <- "Motor Symptom Daily Burden"
 } else {
  fill_color <- "#ff7f0e"
  title_text <- "Non-Motor Symptom Daily Burden"
 }

 p <- ggplot(participant_data, aes(x = phase_name, y = global_burden_score)) +

  geom_boxplot(
   fill = fill_color,
   alpha = 0.7,
   outlier.shape = NA,
   width = 0.6
  ) +

  geom_point(
   color = fill_color,
```

```
    alpha = 0.8,
    size = 2.5,
    position = position_jitter(width = 0.2, height = 0)
   ) +

   scale_y_continuous(
    limits = c(0, max(participant_data$global_burden_score) * 1.1),
    expand = c(0.02, 0)
   ) +

   labs(
    title = paste0(research_id, ": ", title_text, " Across Menstrual Phases"),
    subtitle = paste0("Total daily observations: ", total_points, " | Daily burden range: ", burden_range),
    x = "Menstrual Phase",
    y = "Daily Burden Score (Total Symptom Severity per Day)",
    caption = "Each dot represents one day's total symptom burden"
   ) +

   theme_minimal() +
   theme(
    plot.title = element_text(face = "bold", size = 12, hjust = 0.5),
    plot.subtitle = element_text(size = 10, hjust = 0.5),
    plot.caption = element_text(size = 9, hjust = 0.5),
    axis.text.x = element_text(size = 11),
    axis.text.y = element_text(size = 10),
    axis.title = element_text(size = 11, face = "bold"),
    panel.grid.minor = element_blank()
   )

  return(p)
}

create_participant_global_burden_boxplots <- function(research_id) {

  participant_data <- all_phase_global_burden_data %>%
   filter(research_id == !!research_id)

  if(nrow(participant_data) == 0) {
   return(list(motor_plot = NULL, nonmotor_plot = NULL))
  }

  data_summary <- participant_data %>%
   group_by(symptom_type, phase_name) %>%
   summarise(n_daily_records = n(), .groups = 'drop') %>%
   pivot_wider(names_from = phase_name, values_from = n_daily_records, values_fill = 0)

  print(data_summary)

  motor_plot <- create_individual_global_burden_boxplot(research_id, "Motor")
```

```
 nonmotor_plot <- create_individual_global_burden_boxplot(research_id, "Non-Motor")

 return(list(motor_plot = motor_plot, nonmotor_plot = nonmotor_plot))
}

all_participant_global_burden_plots <- list()
participants_with_burden_data <- unique(all_phase_global_burden_data$research_id)

for(research_id in participants_with_burden_data) {
 participant_plots <- create_participant_global_burden_boxplots(research_id)
 all_participant_global_burden_plots[[research_id]] <- participant_plots
}

motor_plots_created <- 0
nonmotor_plots_created <- 0

for(research_id in names(all_participant_global_burden_plots)) {
 plots <- all_participant_global_burden_plots[[research_id]]
 if(!is.null(plots$motor_plot)) motor_plots_created <- motor_plots_created + 1
 if(!is.null(plots$nonmotor_plot)) nonmotor_plots_created <- nonmotor_plots_created + 1
}

display_participant_global_burden_plots <- function(research_id) {
 if(research_id %in% names(all_participant_global_burden_plots)) {
  plots <- all_participant_global_burden_plots[[research_id]]

  if(!is.null(plots$motor_plot)) {
   print(plots$motor_plot)
  }

  if(!is.null(plots$nonmotor_plot)) {
   print(plots$nonmotor_plot)
  }

 }
}

display_participant_global_burden_plots(participants_with_burden_data[1])

identify_complete_case_participants <- function() {

 participant_completion <- all_phase_global_burden_data %>%
  group_by(research_id, symptom_type) %>%
  summarise(
   has_premenstrual = any(phase_name == "Premenstrual"),
   has_menstrual = any(phase_name == "Menstrual"),
   has_postmenstrual = any(phase_name == "Post-menstrual"),
   phases_present = paste(sort(unique(phase_name)), collapse = ", "),
   .groups = 'drop'
```

```
  ) %>%
  mutate(
    complete_case = has_premenstrual & has_menstrual & has_postmenstrual
  )

 motor_complete <- participant_completion %>%
  filter(symptom_type == "Motor", complete_case == TRUE) %>%
  pull(research_id)

 nonmotor_complete <- participant_completion %>%
  filter(symptom_type == "Non-Motor", complete_case == TRUE) %>%
  pull(research_id)

 complete_case_participants <- intersect(motor_complete, nonmotor_complete)

 all_participants <- unique(all_phase_global_burden_data$research_id)
 excluded_participants <- setdiff(all_participants, complete_case_participants)

 if(length(excluded_participants) > 0) {
  for(participant in excluded_participants) {
    participant_phases <- participant_completion %>%
      filter(research_id == participant)

    for(i in 1:nrow(participant_phases)) {
      phase_info <- participant_phases[i, ]
    }
  }
 }

 return(complete_case_participants)
}

complete_case_participants <- identify_complete_case_participants()

calculate_participant_phase_medians_for_visualization_complete <- function() {

 participant_phase_medians <- all_phase_global_burden_data %>%
  filter(research_id %in% complete_case_participants) %>%
  group_by(research_id, symptom_type, phase_name) %>%
  summarise(
    n_days = n(),
    median_daily_burden = median(global_burden_score, na.rm = TRUE),
    mean_daily_burden = mean(global_burden_score, na.rm = TRUE),
    min_daily_burden = min(global_burden_score, na.rm = TRUE),
    max_daily_burden = max(global_burden_score, na.rm = TRUE),
    .groups = 'drop'
  ) %>%
  arrange(research_id, symptom_type, phase_name)
```

```
 return(participant_phase_medians)
}

participant_phase_medians_step7_complete <-
calculate_participant_phase_medians_for_visualization_complete()

print(head(participant_phase_medians_step7_complete, 10))

display_phase_count_summaries_complete <- function() {

 for(symptom_type in c("Motor", "Non-Motor")) {

  symptom_data <- participant_phase_medians_step7_complete %>%
   filter(symptom_type == symptom_type) %>%
   select(research_id, phase_name, median_daily_burden) %>%
   pivot_wider(names_from = phase_name, values_from = median_daily_burden, values_fn = list)

  pre_men <- sum(!is.na(symptom_data$Premenstrual) & !is.na(symptom_data$Menstrual))
  pre_post <- sum(!is.na(symptom_data$Premenstrual) & !is.na(symptom_data$`Post-menstrual`))
  men_post <- sum(!is.na(symptom_data$Menstrual) & !is.na(symptom_data$`Post-menstrual`))
  all_three <- sum(!is.na(symptom_data$Premenstrual) &
                   !is.na(symptom_data$Menstrual) &
                   !is.na(symptom_data$`Post-menstrual`))
 }
}

create_statistical_caption_complete <- function() {

 if(!exists("paired_wilcoxon_results_step8_complete")) {
  return("Box plots show median, quartiles, and range. Statistical tests pending - run Step 8 for
significance testing")
 }

 significant_results <- paired_wilcoxon_results_step8_complete %>%
  filter(significant == TRUE) %>%
  arrange(p_value)

 base_caption <- "Box plots show median, quartiles, and range. "

 if(nrow(significant_results) == 0) {
  return(paste0(base_caption, "No significant differences found between menstrual phases (all p >
0.05)"))
 }

 sig_text <- ""
 for(i in 1:nrow(significant_results)) {
  result <- significant_results[i, ]
  sig_indicator <- ifelse(result$p_value < 0.001, "***",
                          ifelse(result$p_value < 0.01, "**",
```

```
                      ifelse(result$p_value < 0.05, "*", "")))
   direction <- ifelse(result$median_difference > 0, "higher", "lower")

   comparison_text <- paste0(result$symptom_type, ": ", result$phase1, " burden ",
                      direction, " than ", result$phase2, " (p = ",
                      round(result$p_value, 3), sig_indicator, ", r = ",
                      round(result$effect_size_r, 2), ")")

   if(i == 1) {
    sig_text <- comparison_text
   } else {
    sig_text <- paste(sig_text, comparison_text, sep = "; ")
   }
  }

  return(paste0(base_caption, sig_text))
}

calculate_sample_sizes_step7_complete <- function() {

  sample_sizes <- participant_phase_medians_step7_complete %>%
   group_by(symptom_type, phase_name) %>%
   summarise(n = n(), .groups = 'drop') %>%
   arrange(symptom_type, phase_name)

  return(sample_sizes)
}

create_phase_comparison_boxplots_complete <- function() {

  participant_phase_medians_step7_complete$phase_name <- factor(
   participant_phase_medians_step7_complete$phase_name,
   levels = c("Premenstrual", "Menstrual", "Post-menstrual"),
   labels = c("Premenstrual", "Menstrual", "Post-menstrual")
  )

  sample_sizes <- calculate_sample_sizes_step7_complete()

  y_max <- max(participant_phase_medians_step7_complete$median_daily_burden, na.rm = TRUE)
  y_limits <- c(0, y_max * 1.15)

  statistical_caption <- create_statistical_caption_complete()

  annotation_data <- sample_sizes %>%
   mutate(
    y_position = y_limits[2] * 0.95,
    label = paste0("n=", n)
   )
```

```
phase_comparison_plot <- ggplot(participant_phase_medians_step7_complete,
                    aes(x = phase_name, y = median_daily_burden)) +

 geom_boxplot(aes(fill = phase_name),
           alpha = 0.7,
           outlier.shape = NA,
           width = 0.6) +

 geom_jitter(aes(color = phase_name),
          width = 0.2,
          size = 2.5,
          alpha = 0.8) +

 geom_text(data = annotation_data,
         aes(x = phase_name, y = y_position, label = label),
         size = 3.5,
         fontface = "bold",
         color = "black",
         inherit.aes = FALSE) +

 facet_wrap(~ symptom_type, ncol = 2) +

 scale_fill_manual(values = c("Premenstrual" = "#FF69B4",
                     "Menstrual" = "#DC143C",
                     "Post-menstrual" = "#4169E1")) +
 scale_color_manual(values = c("Premenstrual" = "#E1477A",
                      "Menstrual" = "#B91032",
                      "Post-menstrual" = "#2952CC")) +

 coord_cartesian(ylim = y_limits) +

 labs(
  title = "Global Burden Across Menstrual Phases (Complete Case Analysis)",
  subtitle = paste("Each point represents one participant's median daily burden for that phase (n=",
              length(complete_case_participants), " participants)", sep=""),
  x = "Menstrual Phase",
  y = "Median Daily Symptom Burden",
  caption = statistical_caption
 ) +

 theme_minimal() +
 theme(
  plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
  plot.subtitle = element_text(hjust = 0.5, size = 12),
  plot.caption = element_text(hjust = 0.5, size = 10, face = "italic"),
  strip.text = element_text(size = 12, face = "bold"),
  axis.text.x = element_text(angle = 45, hjust = 1),
  legend.position = "none",
  panel.grid.minor = element_blank(),
```

```
    panel.grid.major.x = element_blank()
  )

 return(phase_comparison_plot)
}

main_phase_plot_complete <- create_phase_comparison_boxplots_complete()

print(main_phase_plot_complete)

if (!require(boot, quietly = TRUE)) {
 install.packages("boot")
 library(boot)
}

calculate_median_for_boot <- function(data, indices) {
 return(median(data[indices], na.rm = TRUE))
}

descriptive_stats_step7_complete <- participant_phase_medians_step7_complete %>%
 group_by(symptom_type, phase_name) %>%
 summarise(
  n = n(),
  mean_of_medians = round(mean(median_daily_burden, na.rm = TRUE), 2),
  median_value = round(median(median_daily_burden, na.rm = TRUE), 2),
  sd = round(sd(median_daily_burden, na.rm = TRUE), 2),
  se_of_mean = round(sd(median_daily_burden, na.rm = TRUE) / sqrt(n()), 2),
  se_of_median = {
   if(n() >= 3) {
    tryCatch({
     boot_result <- boot(median_daily_burden, calculate_median_for_boot, R = 1000)
     round(sd(boot_result$t), 2)
    }, error = function(e) NA)
   } else {
    NA
   }
  },
  q1 = round(quantile(median_daily_burden, 0.25, na.rm = TRUE), 2),
  q3 = round(quantile(median_daily_burden, 0.75, na.rm = TRUE), 2),
  iqr = round(q3 - q1, 2),
  min_val = round(min(median_daily_burden, na.rm = TRUE), 2),
  max_val = round(max(median_daily_burden, na.rm = TRUE), 2),
  .groups = 'drop'
 ) %>%
 arrange(symptom_type, phase_name)

print(descriptive_stats_step7_complete)

se_comparison <- descriptive_stats_step7_complete %>%
```

```
 select(symptom_type, phase_name, se_of_mean, se_of_median) %>%
 mutate(
  phase_label = paste(symptom_type, phase_name),
  difference = abs(se_of_mean - se_of_median)
 )

print(se_comparison)

display_phase_count_summaries_complete()

print(main_phase_plot_complete)

calculate_participant_phase_medians_step8_complete <- function() {

 participant_phase_medians <- all_phase_global_burden_data %>%
  filter(research_id %in% complete_case_participants) %>%
  group_by(research_id, symptom_type, phase_name) %>%
  summarise(
   n_days = n(),
   median_daily_burden = median(global_burden_score, na.rm = TRUE),
   mean_daily_burden = mean(global_burden_score, na.rm = TRUE),
   min_daily_burden = min(global_burden_score, na.rm = TRUE),
   max_daily_burden = max(global_burden_score, na.rm = TRUE),
   .groups = 'drop'
  ) %>%
  arrange(research_id, symptom_type, phase_name)

 return(participant_phase_medians)
}

perform_paired_wilcoxon_tests_complete <- function(phase_medians) {

 comparisons <- list(
  c("Premenstrual", "Menstrual"),
  c("Premenstrual", "Post-menstrual"),
  c("Menstrual", "Post-menstrual")
 )

 paired_test_results <- data.frame()

 for(symptom in c("Motor", "Non-Motor")) {

  symptom_data <- phase_medians %>%
   filter(symptom_type == symptom)

  for(comparison in comparisons) {
   phase1 <- comparison[1]
   phase2 <- comparison[2]
```

```
phase1_data <- symptom_data %>%
 filter(phase_name == phase1) %>%
 select(research_id, median_daily_burden)

phase2_data <- symptom_data %>%
 filter(phase_name == phase2) %>%
 select(research_id, median_daily_burden)

paired_data <- phase1_data %>%
 inner_join(phase2_data, by = "research_id", suffix = c("_phase1", "_phase2"))

if(nrow(paired_data) >= 3) {

 tryCatch({
  wilcox_result <- wilcox.test(
   paired_data$median_daily_burden_phase1,
   paired_data$median_daily_burden_phase2,
   paired = TRUE
  )

  n_pairs <- nrow(paired_data)
  z_score <- qnorm(wilcox_result$p.value/2, lower.tail = FALSE)
  effect_size <- z_score / sqrt(n_pairs)

  median_phase1 <- median(paired_data$median_daily_burden_phase1)
  median_phase2 <- median(paired_data$median_daily_burden_phase2)
  sd_phase1 <- sd(paired_data$median_daily_burden_phase1)
  sd_phase2 <- sd(paired_data$median_daily_burden_phase2)
  se_phase1 <- sd_phase1 / sqrt(n_pairs)
  se_phase2 <- sd_phase2 / sqrt(n_pairs)

  paired_test_results <- rbind(paired_test_results, data.frame(
   symptom_type = symptom,
   comparison = paste(phase1, "vs", phase2),
   phase1 = phase1,
   phase2 = phase2,
   n_participants = n_pairs,
   median_phase1 = median_phase1,
   median_phase2 = median_phase2,
   median_difference = median_phase1 - median_phase2,
   sd_phase1 = sd_phase1,
   sd_phase2 = sd_phase2,
   se_phase1 = se_phase1,
   se_phase2 = se_phase2,
   p_value = wilcox_result$p.value,
   v_statistic = wilcox_result$statistic,
   effect_size_r = effect_size,
   significant = wilcox_result$p.value < 0.05,
   participants = paste(paired_data$research_id, collapse = ", "),
```

```
        stringsAsFactors = FALSE
      ))

    }, error = function(e) {
    })

  }
 }
}

 return(paired_test_results)
}

participant_phase_medians_step8_complete <-
calculate_participant_phase_medians_step8_complete()

print(head(participant_phase_medians_step8_complete, 12))

paired_wilcoxon_results_step8_complete <-
perform_paired_wilcoxon_tests_complete(participant_phase_medians_step8_complete)

if(nrow(paired_wilcoxon_results_step8_complete) > 0) {

 print(paired_wilcoxon_results_step8_complete)

 significant_paired_step8_complete <- paired_wilcoxon_results_step8_complete %>%
  filter(significant == TRUE) %>%
  arrange(p_value)

 if(nrow(significant_paired_step8_complete) > 0) {
  print(significant_paired_step8_complete)

  for(i in 1:nrow(significant_paired_step8_complete)) {
   result <- significant_paired_step8_complete[i, ]
   direction <- ifelse(result$median_difference > 0, "higher", "lower")
  }

 }

}

sample_size_check <- paired_wilcoxon_results_step8_complete %>%
 select(symptom_type, comparison, n_participants) %>%
 arrange(symptom_type, comparison)

print(sample_size_check)

unique_sample_sizes <- unique(sample_size_check$n_participants)
```

```
if(nrow(paired_wilcoxon_results_step8_complete) > 0) {
  print(paired_wilcoxon_results_step8_complete)

  significant_paired_step8_complete <- paired_wilcoxon_results_step8_complete %>%
    filter(significant == TRUE) %>%
    arrange(p_value)

  if(nrow(significant_paired_step8_complete) > 0) {
    print(significant_paired_step8_complete)
  }
}

if(sum(paired_wilcoxon_results_step8_complete$significant, na.rm = TRUE) > 0) {
  sig_results <-
paired_wilcoxon_results_step8_complete[paired_wilcoxon_results_step8_complete$significant, ]
  for(i in 1:nrow(sig_results)) {
  }
}
```

Script 6

```
rm(list = ls(pattern = "^(symptom_counts|num_symptoms|symptoms_per_|daily_symptom_)"))
rm(list = ls(pattern = "^(frequency_.*|count_.*_analysis|symptom_.*_summary)"))
rm(list = ls(pattern = "^(daily_.*_data|weekly_.*|monthly_.*|temporal_.*_analysis)$"))
rm(list = ls(pattern = "^(count_.*_symptoms|analyze_symptom_frequency|calculate_.*_counts)$"))
rm(list = ls(pattern = "^(count_.*_plot|frequency_.*_plot|symptom_.*_chart)$"))
required_vars <- c("multi_cycle_merged_data", "single_cycle_merged_data", "no_period_final",
"symptom_categories")
missing_vars <- c()
for(var in required_vars) {
 if(!exists(var)) {
  missing_vars <- c(missing_vars, var)
 }
}

if(length(missing_vars) > 0) {
 stop("Required variables from previous scripts missing: ", paste(missing_vars, collapse = ", "), "\n",
    "Please run Scripts 1-5 first.")
}

if(!exists("multi_cycle_participants") || !exists("single_cycle_participants")
|| !exists("no_period_participants")) {
 stop("Participant classification variables missing. Please run Scripts 1 and 2 first.")
}

required_script11_vars <- c("selected_participants", "all_phase_boundaries")

missing_script11_vars <- c()
for(var in required_script11_vars) {
 if(!exists(var)) {
  missing_script11_vars <- c(missing_script11_vars, var)
 }
}

if(length(missing_script11_vars) > 0) {
 stop("Required variables from Script 11 missing: ", paste(missing_script11_vars, collapse = ", "), "\n",
    "Please run Script 11 (Premenstrual,Menstrual,Post menstrual boxplots) first.")
}

if(!all(c("research_id", "phase", "phase_start", "phase_end") %in% colnames(all_phase_boundaries))) {
 stop("all_phase_boundaries from Script 11 missing required columns (research_id, phase, phase_start,
phase_end).")
}

available_phases_script11 <- unique(all_phase_boundaries$phase)
expected_phases <- c("Premenstrual", "Menstrual", "Post-menstrual")
missing_phases <- setdiff(expected_phases, available_phases_script11)

if(length(missing_phases) > 0) {
```

```
 stop("Missing required phase types from Script 11: ", paste(missing_phases, collapse = ", "),
     "\nScript 6 requires all three menstrual phases: Premenstrual, Menstrual, Post-menstrual")
}

required_packages <- c("ggplot2", "dplyr", "tidyr")
missing_packages <- c()

for(pkg in required_packages) {
 if(!require(pkg, character.only = TRUE, quietly = TRUE)) {
  missing_packages <- c(missing_packages, pkg)
 }
}

if(length(missing_packages) > 0) {
 stop("Required packages for symptom counting missing: ", paste(missing_packages, collapse = ", "))
}

multi_symptom_count <- sum(multi_cycle_merged_data$Symptom != "No Entry", na.rm = TRUE)
single_symptom_count <- sum(single_cycle_merged_data$Symptom != "No Entry", na.rm = TRUE)
no_period_symptom_count <- sum(no_period_final$Symptom != "No Entry", na.rm = TRUE)

total_symptom_entries <- multi_symptom_count + single_symptom_count + no_period_symptom_count

if(total_symptom_entries == 0) {
 stop("No symptom data found across any datasets. Please check Scripts 1 and 2.")
}

script11_participant_count <- length(selected_participants)
phase_boundary_participants <- length(unique(all_phase_boundaries$research_id))

if(script11_participant_count == 0) {
 stop("No participants found in Script 11's selected_participants.")
}

participants_with_phases <- intersect(selected_participants,
unique(all_phase_boundaries$research_id))
if(length(participants_with_phases) != script11_participant_count) {
 missing_phase_participants <- setdiff(selected_participants,
unique(all_phase_boundaries$research_id))
 warning("Some Script 11 participants missing phase boundaries: ",
       paste(missing_phase_participants, collapse = ", "))
}

cat(" 🧹 Environment cleaned - Script 6 variables cleared\n")
cat(" 📊 Total symptom entries across all datasets:", total_symptom_entries, "\n")
cat(" 📊 Multi-cycle symptom entries:", multi_symptom_count, "\n")
cat(" 📊 Single-cycle symptom entries:", single_symptom_count, "\n")
cat(" 📊 No-period symptom entries:", no_period_symptom_count, "\n")
```

```
cat("👥 Script 11 selected participants:", script11_participant_count, "\n")
cat("👥 Participants with phase boundaries:", length(participants_with_phases), "\n")
cat("📋 Available phases from Script 11:", paste(available_phases_script11, collapse = ", "), "\n")
cat("🔄 Ready for symptom counting analysis using Script 11's 3-phase system\n\n")

count_symptom_entries <- function(research_id, data = all_phase_boundaries) {
 if (!research_id %in% unique(data$research_id)) {
  cat("Warning: Participant", research_id, "not found in dataset\n")
  return(NULL)
 }

 participant_data <- data %>%
  filter(research_id == !!research_id)

 participant_symptoms <- bind_rows(
  multi_cycle_merged_data %>% filter(research_id == !!research_id, Symptom != "No Entry"),
  single_cycle_merged_data %>% filter(research_id == !!research_id, Symptom != "No Entry")
 )

 if(nrow(participant_symptoms) == 0) {
  return(NULL)
 }

 participant_symptoms$dates <- as.Date(participant_symptoms$dates)
 participant_data$phase_start <- as.Date(participant_data$phase_start)
 participant_data$phase_end <- as.Date(participant_data$phase_end)

 phase_counts <- data.frame(
  phase = character(),
  total = numeric(),
  participant = character(),
  stringsAsFactors = FALSE
 )

 for(i in 1:nrow(participant_data)) {
  phase_name <- participant_data$phase[i]
  phase_start <- participant_data$phase_start[i]
  phase_end <- participant_data$phase_end[i]

  symptoms_in_phase <- participant_symptoms %>%
   filter(dates >= phase_start & dates <= phase_end) %>%
   nrow()

  phase_counts <- rbind(phase_counts, data.frame(
   phase = phase_name,
   total = symptoms_in_phase,
   participant = research_id,
   stringsAsFactors = FALSE
```

```
  ))
 }

 final_counts <- phase_counts %>%
  group_by(phase, participant) %>%
  summarise(total = sum(total), .groups = 'drop')

 return(final_counts)
}

plot_participant_entries <- function(symptom_counts, research_id) {
 if(is.null(symptom_counts) || nrow(symptom_counts) == 0) {
  return(NULL)
 }

 symptom_counts$phase <- factor(symptom_counts$phase,
                levels = c("Premenstrual", "Menstrual", "Post-menstrual"))

 phase_colors <- c("Premenstrual" = "#FFB3B3", "Menstrual" = "#FFCCCC", "Post-menstrual" =
"#ADD8E6")
 present_phases <- as.character(symptom_counts$phase)
 fill_colors <- phase_colors[present_phases]

 p <- ggplot(symptom_counts, aes(x = phase, y = total)) +
  geom_bar(stat = "identity", aes(fill = phase)) +
  scale_fill_manual(values = phase_colors) +
  geom_text(aes(label = total), vjust = -0.5, size = 5, fontface = "bold") +
  labs(
   title = paste("Total Symptom Entries by Phase for", research_id),
   x = "Menstrual Phase",
   y = "Number of Symptom Entries"
  ) +
  theme_minimal() +
  theme(
   plot.title = element_text(hjust = 0.5, size = 16),
   axis.title = element_text(size = 12),
   axis.text = element_text(size = 10),
   legend.position = "none"
  )

 return(p)
}

test_participant_entries <- function(research_id) {
 symptom_counts <- count_symptom_entries(research_id)

 if(is.null(symptom_counts) || nrow(symptom_counts) == 0) {
  cat("No data available for statistical testing for participant:", research_id, "\n")
  return(NULL)
```

```
}

participant_symptoms <- bind_rows(
 multi_cycle_merged_data %>% filter(research_id == !!research_id, Symptom != "No Entry"),
 single_cycle_merged_data %>% filter(research_id == !!research_id, Symptom != "No Entry")
)

if(nrow(participant_symptoms) == 0) {
 cat("No symptom data available for participant:", research_id, "\n")
 return(NULL)
}

participant_phases <- all_phase_boundaries %>%
 filter(research_id == !!research_id)

participant_symptoms$dates <- as.Date(participant_symptoms$dates)
participant_phases$phase_start <- as.Date(participant_phases$phase_start)
participant_phases$phase_end <- as.Date(participant_phases$phase_end)

participant_data <- participant_symptoms %>%
 mutate(phase = NA_character_)

for(i in 1:nrow(participant_phases)) {
 phase_name <- participant_phases$phase[i]
 phase_start <- participant_phases$phase_start[i]
 phase_end <- participant_phases$phase_end[i]

 participant_data$phase[participant_data$dates >= phase_start &
                        participant_data$dates <= phase_end] <- phase_name
}

participant_data <- participant_data %>%
 filter(!is.na(phase))

if(nrow(participant_data) == 0) {
 cat("No symptoms could be matched to phases for participant:", research_id, "\n")
 return(NULL)
}

phase_table <- table(participant_data$phase)

if(length(phase_table) >= 2) {
 expected <- rep(sum(phase_table)/length(phase_table), length(phase_table))
 chi_result <- chisq.test(phase_table, p = expected/sum(expected))

 cat("Chi-square test for", research_id, "- symptom distribution across phases:\n")
 print(chi_result)

 if(chi_result$p.value < 0.05) {
```

```
    cat("\nThe chi-square test indicates significant differences in symptom entry distribution across
phases (p =",
        format(chi_result$p.value, digits=3), ")\n")
   } else {
    cat("\nThe chi-square test does not indicate significant differences in symptom entry distribution
across phases (p =",
        format(chi_result$p.value, digits=3), ")\n")
   }
  } else {
   chi_result <- NULL
   cat("Insufficient phase diversity for chi-square test\n")
  }

  daily_data <- participant_data %>%
   group_by(phase, dates) %>%
   summarise(entries = n(), .groups = "drop")

  phase_counts <- table(daily_data$phase)
  sufficient_daily_data <- all(phase_counts >= 5)

  if(sufficient_daily_data) {
   cat("\nKruskal-Wallis test for differences in daily entries across phases:\n")
   kw_result <- kruskal.test(entries ~ phase, data = daily_data)
   print(kw_result)

   if(kw_result$p.value < 0.05) {
    cat("\nThe Kruskal-Wallis test indicates significant differences in daily symptom entries across phases
(p =",
        format(kw_result$p.value, digits=3), ")\n")
   } else {
    cat("\nThe Kruskal-Wallis test does not indicate significant differences in daily symptom entries
across phases (p =",
        format(kw_result$p.value, digits=3), ")\n")
   }
  } else {
   kw_result <- NULL
   cat("\nInsufficient daily data for Kruskal-Wallis test (need at least 5 days per phase)\n")
  }

  return(list(
   symptom_counts = symptom_counts,
   chi_result = chi_result,
   kw_result = if(sufficient_daily_data) kw_result else NULL,
   daily_data = daily_data
  ))
}

analyze_participant <- function(research_id) {
  cat("\n=====================================\n")
```

```
  cat("Analyzing participant:", research_id, "\n")
  cat("=====================================\n")

  symptom_counts <- count_symptom_entries(research_id)

  if(is.null(symptom_counts) || nrow(symptom_counts) == 0) {
    cat("No valid data for participant:", research_id, "\n")
    return(NULL)
  }

  plot <- plot_participant_entries(symptom_counts, research_id)
  print(plot)

  stats_result <- test_participant_entries(research_id)

  return(list(
    symptom_counts = symptom_counts,
    plot = plot,
    stats = stats_result
  ))
}

analyze_all_multicycle_participants <- function(data = multi_cycle_phases, pause = TRUE) {
  all_participants <- unique(data$research_id)

  cat("Found", length(all_participants), "participants to analyze\n")

  all_results <- list()

  for(i in 1:length(all_participants)) {
    id <- all_participants[i]

    cat("\nProcessing participant", i, "of", length(all_participants), ":", id, "\n")

    results <- analyze_participant(id)

    all_results[[id]] <- results

    if(pause && i < length(all_participants)) {
      cat("\nPress [Enter] to continue to next participant...\n")
      invisible(readline())
    }
  }

  return(all_results)
}

create_aggregate_plots <- function() {
  all_participants <- participants_with_phases
```

```
all_counts <- data.frame()

for(id in all_participants) {
 counts <- count_symptom_entries(id)

 if(is.null(counts) || nrow(counts) == 0) next

 all_counts <- rbind(all_counts, counts)
}

aggregate_stats <- all_counts %>%
 group_by(phase) %>%
 summarise(
  total_entries = sum(total),
  n_participants = n_distinct(participant),
  mean_per_participant = mean(total),
  median_per_participant = median(total),
  sd = sd(total),
  se = sd / sqrt(n_participants)
 )

aggregate_stats$phase <- factor(aggregate_stats$phase,
                levels = c("Premenstrual", "Menstrual", "Post-menstrual"))

aggregate_stats <- aggregate_stats %>% arrange(phase)

cat("\nAggregate Statistics (in correct phase order):\n")
print(aggregate_stats)

p1 <- ggplot(aggregate_stats, aes(x = phase, y = total_entries)) +
 geom_bar(stat = "identity", aes(fill = phase)) +
 scale_fill_manual(values = c("Premenstrual" = "#FFB3B3",
                "Menstrual" = "#FFCCCC",
                "Post-menstrual" = "#ADD8E6")) +
 geom_text(aes(label = total_entries), vjust = -0.5, size = 5, fontface = "bold") +
 geom_text(aes(label = sprintf("n = %d", n_participants)), vjust = -2.5, size = 4) +
 labs(
  title = "Total Symptom Entries by Menstrual Phase",
  subtitle = "Aggregated across Script 11 participants",
  x = "Menstrual Phase",
  y = "Total Number of Symptom Entries"
 ) +
 theme_minimal() +
 theme(
  plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
  plot.subtitle = element_text(hjust = 0.5, size = 12),
  legend.position = "none"
 )
```

```
 p2 <- ggplot(aggregate_stats, aes(x = phase, y = mean_per_participant)) +
  geom_bar(stat = "identity", aes(fill = phase)) +
  scale_fill_manual(values = c("Premenstrual" = "#FFB3B3",
                               "Menstrual" = "#FFCCCC",
                               "Post-menstrual" = "#ADD8E6")) +
  geom_errorbar(aes(ymin = mean_per_participant - se,
                    ymax = mean_per_participant + se),
                width = 0.2) +
  geom_text(aes(label = sprintf("%.1f", mean_per_participant)),
            vjust = -0.5, size = 5, fontface = "bold") +
  geom_text(aes(label = sprintf("n = %d", n_participants)),
            vjust = -2.5, size = 4) +
  labs(
    title = "Average Symptom Entries per Participant by Menstrual Phase",
    subtitle = "Error bars represent standard error",
    x = "Menstrual Phase",
    y = "Average Number of Symptom Entries per Participant (± SE)"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5, size = 12),
    legend.position = "none"
  )

 print(p1)
 cat("\n\n")
 print(p2)

 observed <- aggregate_stats$total_entries
 names(observed) <- aggregate_stats$phase
 expected <- rep(sum(observed)/length(observed), length(observed))

 chi_result <- chisq.test(observed, p = expected/sum(expected))

 cat("\nChi-square test for differences in total entry distribution across phases:\n")
 print(chi_result)

 return(list(
  aggregate_stats = aggregate_stats,
  total_plot = p1,
  average_plot = p2,
  chi_result = chi_result
 ))
}

analyze_complete_phase_participants <- function() {
 all_participants <- participants_with_phases
```

```
cat("Total number of participants:", length(all_participants), "\n")

phase_tracking <- list()

for(id in all_participants) {
 participant_counts <- count_symptom_entries(id)

 if(is.null(participant_counts) || nrow(participant_counts) == 0) {
  phase_tracking[[id]] <- character(0)
  next
 }

 participant_phases <- unique(participant_counts$phase)
 phase_tracking[[id]] <- participant_phases
}

complete_participants <- names(phase_tracking)[sapply(phase_tracking, function(phases) {
 all(c("Premenstrual", "Menstrual", "Post-menstrual") %in% phases)
})]

cat("Number of participants with data in all three phases:", length(complete_participants), "\n")
cat("These participants are:", paste(complete_participants, collapse=", "), "\n")

all_counts <- data.frame()

for(id in complete_participants) {
 counts <- count_symptom_entries(id)

 all_counts <- rbind(all_counts, counts)
}

aggregate_stats <- all_counts %>%
 group_by(phase) %>%
 summarise(
  total_entries = sum(total),
  n_participants = n_distinct(participant),
  mean_per_participant = mean(total),
  median_per_participant = median(total),
  sd = sd(total),
  se = sd / sqrt(n_participants)
 )

aggregate_stats$phase <- factor(aggregate_stats$phase,
                levels = c("Premenstrual", "Menstrual", "Post-menstrual"))

aggregate_stats <- aggregate_stats %>% arrange(phase)

cat("\nAggregate Statistics for Participants with Complete Phase Data:\n")
print(aggregate_stats)
```

```
p1 <- ggplot(aggregate_stats, aes(x = phase, y = total_entries)) +
 geom_bar(stat = "identity", aes(fill = phase)) +
 scale_fill_manual(values = c("Premenstrual" = "#FFB3B3",
                              "Menstrual" = "#FFCCCC",
                              "Post-menstrual" = "#ADD8E6")) +
 geom_text(aes(label = total_entries), vjust = -0.5, size = 5, fontface = "bold") +
 geom_text(aes(label = sprintf("n = %d", n_participants)), vjust = -2.5, size = 4) +
 labs(
  title = "Total Symptom Entries by Menstrual Phase",
  subtitle = paste("Participants with data in all three phases (n =", length(complete_participants), ")"),
  x = "Menstrual Phase",
  y = "Total Number of Symptom Entries"
 ) +
 theme_minimal() +
 theme(
  plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
  plot.subtitle = element_text(hjust = 0.5, size = 12),
  legend.position = "none"
 )

p2 <- ggplot(aggregate_stats, aes(x = phase, y = mean_per_participant)) +
 geom_bar(stat = "identity", aes(fill = phase)) +
 scale_fill_manual(values = c("Premenstrual" = "#FFB3B3",
                              "Menstrual" = "#FFCCCC",
                              "Post-menstrual" = "#ADD8E6")) +
 geom_errorbar(aes(ymin = mean_per_participant - se,
                   ymax = mean_per_participant + se),
               width = 0.2) +
 geom_text(aes(label = sprintf("%.1f", mean_per_participant)),
           vjust = -0.5, size = 5, fontface = "bold") +
 geom_text(aes(label = sprintf("n = %d", n_participants)),
           vjust = -2.5, size = 4) +
 labs(
  title = "Average Symptom Entries per Participant by Menstrual Phase",
  subtitle = paste("Participants with data in all three phases (n =", length(complete_participants), ")"),
  x = "Menstrual Phase",
  y = "Average Number of Symptom Entries per Participant (± SE)"
 ) +
 theme_minimal() +
 theme(
  plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
  plot.subtitle = element_text(hjust = 0.5, size = 12),
  legend.position = "none"
 )

print(p1)
cat("\n\n")
print(p2)
```

```
 cat("\nPerforming Friedman test (non-parametric test for repeated measures):\n")

 friedman_data <- all_counts %>%
  spread(key = phase, value = total) %>%
  select(Premenstrual, Menstrual, `Post-menstrual`)

 if(nrow(friedman_data) >= 3) {
  friedman_result <- friedman.test(as.matrix(friedman_data))
  print(friedman_result)

  if(friedman_result$p.value < 0.05) {
   cat("\nFriedman test is significant (p <", format(friedman_result$p.value, digits=3), ")\n")
   cat("Performing post-hoc pairwise Wilcoxon signed-rank tests:\n")

   phases <- c("Premenstrual", "Menstrual", "Post-menstrual")
   for(i in 1:(length(phases)-1)) {
    for(j in (i+1):length(phases)) {
     phase1 <- phases[i]
     phase2 <- phases[j]

     col1 <- if(phase1 == "Post-menstrual") "Post-menstrual" else phase1
     col2 <- if(phase2 == "Post-menstrual") "Post-menstrual" else phase2

     wilcox_result <- wilcox.test(friedman_data[[col1]],
                  friedman_data[[col2]],
                  paired = TRUE)

     p_adjusted <- wilcox_result$p.value * 3
     p_adjusted <- min(p_adjusted, 1.0)

     cat(phase1, "vs", phase2, "- p-value:", format(wilcox_result$p.value, digits=3),
       "(adjusted p =", format(p_adjusted, digits=3), ")\n")
    }
   }
  } else {
   cat("\nFriedman test is not significant (p =", format(friedman_result$p.value, digits=3), ")\n")
  }
 } else {
  cat("Insufficient data for Friedman test (need at least 3 participants)\n")
 }
  return(list(
  complete_participants = complete_participants,
  aggregate_stats = aggregate_stats,
  total_plot = p1,
  average_plot = p2
 ))
}
```

```
complete_phase_results <- analyze_complete_phase_participants()

analyze_symptoms_per_day_by_phase <- function() {
 cat("=== SYMPTOMS PER DAY ANALYSIS ===\n")
 cat("Calculating symptom intensity (symptoms/day) normalized for phase duration\n\n")

 library(tibble)
 all_participants <- participants_with_phases

 per_day_data <- data.frame(
  research_id = character(),
  phase = character(),
  phase_start = as.Date(character()),
  phase_end = as.Date(character()),
  phase_days = numeric(),
  total_symptoms = numeric(),
  symptoms_per_day = numeric(),
  stringsAsFactors = FALSE
 )

 for(id in all_participants) {
  cat("Processing participant:", id, "\n")

  participant_phases <- all_phase_boundaries %>%
   filter(research_id == id)

  participant_symptoms <- bind_rows(
   multi_cycle_merged_data %>% filter(research_id == id, Symptom != "No Entry"),
   single_cycle_merged_data %>% filter(research_id == id, Symptom != "No Entry")
  )

  if(nrow(participant_symptoms) == 0 || nrow(participant_phases) == 0) {
   cat("  No data available for", id, "\n")
   next
  }

  participant_symptoms$dates <- as.Date(participant_symptoms$dates)
  participant_phases$phase_start <- as.Date(participant_phases$phase_start)
  participant_phases$phase_end <- as.Date(participant_phases$phase_end)

  for(i in 1:nrow(participant_phases)) {
   phase_name <- participant_phases$phase[i]
   phase_start <- participant_phases$phase_start[i]
   phase_end <- participant_phases$phase_end[i]

   phase_days <- as.numeric(phase_end - phase_start) + 1

   symptoms_in_phase <- participant_symptoms %>%
    filter(dates >= phase_start & dates <= phase_end) %>%
```

```
    nrow()

   symptoms_per_day <- if(phase_days > 0) symptoms_in_phase / phase_days else 0

   per_day_data <- rbind(per_day_data, data.frame(
    research_id = id,
    phase = phase_name,
    phase_start = phase_start,
    phase_end = phase_end,
    phase_days = phase_days,
    total_symptoms = symptoms_in_phase,
    symptoms_per_day = symptoms_per_day,
    stringsAsFactors = FALSE
   ))
 }
}

aggregated_data <- per_day_data %>%
 group_by(research_id, phase) %>%
 summarise(
  symptoms_per_day = mean(symptoms_per_day, na.rm = TRUE),
  .groups = 'drop'
 )

participant_level_means <- aggregated_data %>%
 rename(participant_mean_symptoms_per_day = symptoms_per_day)

complete_participants <- aggregated_data %>%
 group_by(research_id) %>%
 summarise(phase_count = n(), .groups = 'drop') %>%
 filter(phase_count == 3) %>%
 pull(research_id)

cat("Participants with complete data across all 3 phases:", length(complete_participants), "\n")

aggregated_data <- aggregated_data %>%
 filter(research_id %in% complete_participants)

participant_level_means <- participant_level_means %>%
 filter(research_id %in% complete_participants)

aggregated_data$phase <- factor(aggregated_data$phase,
                levels = c("Premenstrual", "Menstrual", "Post-menstrual"))

library(tibble)
friedman_data <- aggregated_data %>%
 select(research_id, phase, symptoms_per_day) %>%
 spread(key = phase, value = symptoms_per_day) %>%
 column_to_rownames("research_id")
```

```
cat("Friedman test data structure:\n")
print(head(friedman_data))
cat("Dimensions:", nrow(friedman_data), "participants ×", ncol(friedman_data), "phases\n")

phase_summary <- aggregated_data %>%
 group_by(phase) %>%
 summarise(
  n_participants = n(),

  mean_symptoms_per_day = mean(symptoms_per_day, na.rm = TRUE),
  median_symptoms_per_day = median(symptoms_per_day, na.rm = TRUE),
  sd_symptoms_per_day = sd(symptoms_per_day, na.rm = TRUE),
  se_symptoms_per_day = sd(symptoms_per_day, na.rm = TRUE) / sqrt(n()),

  .groups = 'drop'
 )

phase_summary$phase <- factor(phase_summary$phase,
                levels = c("Premenstrual", "Menstrual", "Post-menstrual"))
phase_summary <- phase_summary %>% arrange(phase)

cat("\n=== SYMPTOMS PER DAY SUMMARY (KEY RESULTS) ===\n")
print(phase_summary)
cat("\nFull column view:\n")
glimpse(phase_summary)

p1 <- ggplot(phase_summary, aes(x = phase, y = mean_symptoms_per_day)) +
 geom_bar(stat = "identity", aes(fill = phase)) +
 scale_fill_manual(values = c("Premenstrual" = "#FFB3B3",
                "Menstrual" = "#FFCCCC",
                "Post-menstrual" = "#ADD8E6")) +
 geom_errorbar(aes(ymin = mean_symptoms_per_day - se_symptoms_per_day,
          ymax = mean_symptoms_per_day + se_symptoms_per_day),
         width = 0.2) +
 geom_text(aes(label = sprintf("%.2f", mean_symptoms_per_day)),
       vjust = -0.5, size = 5, fontface = "bold") +
 labs(
  title = "Daily Symptom Reports Across Menstrual Phases (Repeated Measures)",
  subtitle = paste("Participant-level means | n =", length(complete_participants), "participants with
complete data"),
  x = "Menstrual Phase",
  y = "Average No. of Symptoms Logged Per Day"
 ) +
 theme_minimal() +
 theme(
  plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
  plot.subtitle = element_text(hjust = 0.5, size = 12),
  legend.position = "none"
```

```
  )

  print(p1)

  cat("\n=== FRIEDMAN TEST: REPEATED MEASURES ACROSS PHASES ===\n")
  cat("Testing", nrow(friedman_data), "participants across 3 phases\n")

  friedman_result <- friedman.test(as.matrix(friedman_data))
  print(friedman_result)

  if(friedman_result$p.value < 0.05) {
   cat("\nFriedman test is significant (p =",
      format(friedman_result$p.value, digits=3), ")\n")

   cat("\n=== POST-HOC PAIRWISE COMPARISONS (Paired Wilcoxon Tests) ===\n")
   phases <- c("Premenstrual", "Menstrual", "Post-menstrual")

   for(i in 1:(length(phases)-1)) {
    for(j in (i+1):length(phases)) {
      phase1 <- phases[i]
      phase2 <- phases[j]

      wilcox_result <- wilcox.test(friedman_data[[phase1]],
                          friedman_data[[phase2]],
                          paired = TRUE)

      p_adjusted <- wilcox_result$p.value * 3
      p_adjusted <- min(p_adjusted, 1.0)

      cat(phase1, "vs", phase2, "- p-value:", format(wilcox_result$p.value, digits=3),
        "(adjusted p =", format(p_adjusted, digits=3), ")\n")
    }
   }
  } else {
   cat("\nFriedman test is not significant (p =",
      format(friedman_result$p.value, digits=3), ")\n")
  }

  return(list(
   per_day_data = per_day_data,
   aggregated_data = aggregated_data,
   complete_participants = complete_participants,
   friedman_data = friedman_data,
   phase_summary = phase_summary,
   symptoms_per_day_plot = p1,
   friedman_result = friedman_result
  ))
}
```

Script 4.2

```
participants_with_phases <- intersect(selected_participants,
unique(all_phase_boundaries$research_id))

combined_symptom_data <- bind_rows(
 multi_cycle_merged_data %>%
  filter(research_id %in% participants_with_phases, Symptom != "No Entry") %>%
  mutate(data_source = "multi_cycle"),

 single_cycle_merged_data %>%
  filter(research_id %in% participants_with_phases, Symptom != "No Entry") %>%
  mutate(data_source = "single_cycle")
)

phase_boundaries_filtered <- all_phase_boundaries %>%
 filter(research_id %in% participants_with_phases) %>%
 mutate(
  phase_start = as.Date(phase_start),
  phase_end = as.Date(phase_end)
 )

symptom_data_for_matching <- combined_symptom_data %>%
 mutate(dates = as.Date(dates)) %>%
 select(research_id, dates, Symptom, Severity, data_source)

three_phase_data <- symptom_data_for_matching %>%
 mutate(phase = NA_character_,
      cycle_id = NA_character_,
      phase_start = as.Date(NA),
      phase_end = as.Date(NA))

for(i in 1:nrow(phase_boundaries_filtered)) {
 participant_id <- phase_boundaries_filtered$research_id[i]
 cycle_id <- phase_boundaries_filtered$cycle_id[i]
 phase_name <- phase_boundaries_filtered$phase[i]
 phase_start <- phase_boundaries_filtered$phase_start[i]
 phase_end <- phase_boundaries_filtered$phase_end[i]

 matching_indices <- which(
  three_phase_data$research_id == participant_id &
   three_phase_data$dates >= phase_start &
   three_phase_data$dates <= phase_end &
   is.na(three_phase_data$phase)
 )

 if(length(matching_indices) > 0) {
  three_phase_data$phase[matching_indices] <- phase_name
  three_phase_data$cycle_id[matching_indices] <- cycle_id
  three_phase_data$phase_start[matching_indices] <- phase_start
```

```
   three_phase_data$phase_end[matching_indices] <- phase_end
  }
}

three_phase_data <- three_phase_data %>%
  filter(!is.na(phase)) %>%
  mutate(
    phase = factor(phase, levels = c("Premenstrual", "Menstrual", "Post-menstrual")),
    Severity = as.numeric(Severity)
  ) %>%
  filter(!is.na(Severity))

cat("three_phase_data created successfully:\n")
cat("  Total rows:", nrow(three_phase_data), "\n")
cat("  Participants:", length(unique(three_phase_data$research_id)), "\n")
cat("  Unique symptoms:", length(unique(three_phase_data$Symptom)), "\n")

check_symptom_representation_modified <- function(data, symptom_name) {
  cat("Checking representation for:", symptom_name, "\n")

  symptom_data <- data %>%
    filter(!is.na(Symptom), Symptom == symptom_name)

  if(nrow(symptom_data) == 0) {
    cat("  No data found for this symptom\n")
    return(list(
      meets_criteria = FALSE,
      total_participants = 0,
      participants_by_phase_count = 0,
      phase_counts = data.frame(
        phase = factor(c("Premenstrual", "Menstrual", "Post-menstrual"),
                       levels = c("Premenstrual", "Menstrual", "Post-menstrual")),
        distinct_participants = c(0, 0, 0)
      ),
      criteria_met = "No data for this symptom",
      qualifying_participants = character(0)
    ))
  }

  cat("  Found", nrow(symptom_data), "symptom entries\n")

  participant_phase_counts <- symptom_data %>%
    group_by(research_id) %>%
    summarise(
      phases_with_symptom = n_distinct(phase),
      phases_list = paste(sort(unique(phase)), collapse = ", "),
      .groups = 'drop'
    )
```

```
 cat("  Total participants who ever logged this symptom:", nrow(participant_phase_counts), "\n")

 phase_breakdown <- table(participant_phase_counts$phases_with_symptom)
 cat("  Breakdown by phases per participant:\n")
 for(i in names(phase_breakdown)) {
  cat("    ", i, "phases:", phase_breakdown[i], "participants\n")
 }

 qualifying_participants <- participant_phase_counts %>%
  filter(phases_with_symptom >= 1) %>%
  pull(research_id)

 cat("  Participants qualifying (≥1 phase):", length(qualifying_participants), "\n")
 if(length(qualifying_participants) > 0) {
  cat("  Qualifying participant IDs:", paste(qualifying_participants, collapse = ", "), "\n")
 }

 phase_counts <- symptom_data %>%
  filter(research_id %in% qualifying_participants) %>%
  group_by(phase) %>%
  summarise(distinct_participants = n_distinct(research_id), .groups = 'drop') %>%
  complete(phase = factor(c("Premenstrual", "Menstrual", "Post-menstrual"),
                          levels = c("Premenstrual", "Menstrual", "Post-menstrual")),
           fill = list(distinct_participants = 0))

 cat("  Phase counts (from qualifying participants only):\n")
 for(i in 1:nrow(phase_counts)) {
  cat("    ", phase_counts$phase[i], ":", phase_counts$distinct_participants[i], "participants\n")
 }

 total_qualifying_participants <- length(qualifying_participants)
 meets_criteria <- total_qualifying_participants >= 3

 criteria_met <- if(meets_criteria) {
  paste0("✅ ", total_qualifying_participants, " participants with symptom in ≥1 phase (≥3 required)")
 } else {
  paste0("❌ Only ", total_qualifying_participants, " participants with symptom in ≥1 phase (≥3
required)")
 }

 cat("  Result:", criteria_met, "\n")

 return(list(
  meets_criteria = meets_criteria,
  total_participants = total_qualifying_participants,
  participants_by_phase_count = nrow(participant_phase_counts),
  phase_counts = phase_counts,
  criteria_met = criteria_met,
  qualifying_participants = qualifying_participants,
```

```
  participant_phase_details = participant_phase_counts
 ))
}

test_result <- check_symptom_representation_modified(three_phase_data, "Dyskinesia")

library(lme4)
library(lmerTest)
library(emmeans)

analyze_symptom_modified <- function(data, symptom_name, verbose = TRUE) {

 if(verbose) cat("\n=== ANALYZING SYMPTOM:", symptom_name, "(NEW CRITERIA) ===\n")

 representation_check <- check_symptom_representation_modified(data, symptom_name)

 if(!representation_check$meets_criteria) {
  if(verbose) {
   cat("❌ Symptom", symptom_name, "does not meet NEW representation criteria.\n")
   cat("  ", representation_check$criteria_met, "\n")
  }
  return(NULL)
 }

 qualifying_participants <- representation_check$qualifying_participants

 if(verbose) {
  cat("✅ Symptom meets NEW criteria:", representation_check$criteria_met, "\n")
  cat("Qualifying participants:", paste(qualifying_participants, collapse = ", "), "\n")
  cat("Will analyze", length(qualifying_participants), "participants\n")
 }

 symptom_data <- data %>%
  filter(!is.na(Symptom),
       Symptom == symptom_name,
       research_id %in% qualifying_participants)

 if(verbose) cat("Using", nrow(symptom_data), "symptom entries for analysis\n")

 participant_averages <- symptom_data %>%
  group_by(research_id, phase) %>%
  summarise(
   avg_severity = mean(as.numeric(Severity), na.rm = TRUE),
   n_recordings = n(),
   .groups = 'drop'
  ) %>%
  mutate(phase = factor(phase, levels = c("Premenstrual", "Menstrual", "Post-menstrual")))

 if(verbose) {
```

```
 cat("Calculated averages for", nrow(participant_averages), "participant-phase combinations\n")
}

phase_summary <- participant_averages %>%
 group_by(phase) %>%
 summarise(
  n_participants = n_distinct(research_id),
  n_recordings = sum(n_recordings),
  median = median(avg_severity, na.rm = TRUE),
  mean = mean(avg_severity, na.rm = TRUE),
  sd = sd(avg_severity, na.rm = TRUE),
  q1 = quantile(avg_severity, 0.25, na.rm = TRUE),
  q3 = quantile(avg_severity, 0.75, na.rm = TRUE),
  .groups = 'drop'
 )

if(verbose) {
 cat("\nPhase Summary (NEW criteria - potentially unequal n across phases):\n")
 print(phase_summary)
}

statistical_results <- list()

if(verbose) cat("\n1. Statistical Testing (Linear Mixed Effects with F-test):\n")

n_participants <- n_distinct(participant_averages$research_id)
n_phases_per_participant <- participant_averages %>%
 group_by(research_id) %>%
 summarise(n_phases = n(), .groups = 'drop')

if(verbose) {
 cat("  Participants in analysis:", n_participants, "\n")
 cat("  Phases per participant range:", min(n_phases_per_participant$n_phases),
    "to", max(n_phases_per_participant$n_phases), "\n")
}

tryCatch({
 lmm_model <- lmer(avg_severity ~ phase + (1|research_id), data = participant_averages)

 if(verbose) cat("  ✅ Linear Mixed Effects model fitted successfully\n")

 anova_result <- anova(lmm_model)
 overall_p <- anova_result[1, "Pr(>F)"]
 f_statistic <- anova_result[1, "F value"]

 statistical_results$lmm_model <- lmm_model
 statistical_results$anova_result <- anova_result
 statistical_results$overall_f_test <- list(
  f_statistic = f_statistic,
```

```
  p_value = overall_p,
  significant = overall_p < 0.05
 )

 if(verbose) {
  cat("  Overall F-test: F =", round(f_statistic, 3),
      ", p =", format(overall_p, digits = 3))
  if(overall_p < 0.001) cat(" ***")
  else if(overall_p < 0.01) cat(" **")
  else if(overall_p < 0.05) cat(" *")
  cat("\n")
 }

 if(overall_p < 0.05) {
  if(verbose) cat("  ✅ Overall test significant - performing pairwise t-tests...\n")

  emm_results <- emmeans(lmm_model, ~ phase)
  pairwise_results <- pairs(emm_results, adjust = "bonferroni")
  pairwise_summary <- summary(pairwise_results)

  statistical_results$pairwise_t_tests <- list()

  for(i in 1:nrow(pairwise_summary)) {
   contrast_name <- as.character(pairwise_summary$contrast[i])
   p_val <- pairwise_summary$p.value[i]
   t_val <- pairwise_summary$t.ratio[i]

   statistical_results$pairwise_t_tests[[contrast_name]] <- list(
    t_statistic = t_val,
    p_value = p_val,
    bonferroni_p_value = p_val,
    significant = p_val < 0.05
   )

   if(verbose) {
    cat("    ", contrast_name, ": t =", round(t_val, 3),
        ", p =", format(p_val, digits = 3))
    if(p_val < 0.001) cat(" ***")
    else if(p_val < 0.01) cat(" **")
    else if(p_val < 0.05) cat(" *")
    cat("\n")
   }
  }

 } else {
  if(verbose) cat("  Pairwise tests skipped (overall test not significant)\n")
  statistical_results$pairwise_t_tests <- "Skipped - overall test not significant"
 }
```

```
 }, error = function(e) {
  if(verbose) cat("  ❌ Statistical testing failed:", e$message, "\n")
  statistical_results$error <- e$message
 })

 results <- list(
  symptom = symptom_name,
  meets_criteria = TRUE,
  representation_check = representation_check,
  participant_averages = participant_averages,
  phase_summary = phase_summary,
  statistical_results = statistical_results,
  qualifying_participants = qualifying_participants,
  total_participants = length(qualifying_participants)
 )

 if(verbose) cat("\n ✅ Analysis complete for", symptom_name, "(NEW CRITERIA)\n")

 return(results)
}

dyskinesia_analysis <- analyze_symptom_modified(three_phase_data, "Dyskinesia")

create_3phase_boxplot_modified <- function(data, symptom_name, analysis_results = NULL,
show_stats = TRUE) {

 if(is.null(analysis_results)) {
  analysis_results <- analyze_symptom_modified(data, symptom_name, verbose = FALSE)
 }

 if(is.null(analysis_results)) {
  cat("❌ Cannot create plot for", symptom_name, "- analysis failed\n")
  return(NULL)
 }

 participant_averages <- analysis_results$participant_averages
 phase_summary <- analysis_results$phase_summary
 n_participants <- analysis_results$total_participants

 p <- ggplot(participant_averages, aes(x = phase, y = avg_severity)) +
  geom_boxplot(aes(fill = phase), alpha = 0.7, outlier.shape = 21,
           outlier.fill = "white", outlier.size = 2) +
  geom_jitter(width = 0.2, size = 2, alpha = 0.8, color = "black") +
  scale_fill_manual(
   values = c("Premenstrual" = "#FFB3B3",
           "Menstrual" = "#FFCCCC",
           "Post-menstrual" = "#ADD8E6"),
   name = "Phase"
```

```
  ) +
  scale_y_continuous(
    limits = c(0.5, 3.5),
    breaks = seq(1, 3, 0.5),
    labels = c("1", "1.5", "2", "2.5", "3")
  ) +
  labs(
    title = paste("Symptom Severity Across Menstrual Cycle Phases:", symptom_name),
    x = "Menstrual Cycle Phase",
    y = "Average Symptom Severity",
    caption = paste("NEW CRITERIA: ≥3 participants with symptom in ≥1 phase")
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold", hjust = 0.5),
    axis.title = element_text(size = 12, face = "bold"),
    axis.text = element_text(size = 11),
    legend.position = "none",
    panel.grid.minor = element_blank(),
    plot.caption = element_text(size = 10, face = "italic")
  )

for(i in 1:nrow(phase_summary)) {
  phase_name <- phase_summary$phase[i]
  n_phase <- phase_summary$n_participants[i]

  x_pos <- which(levels(participant_averages$phase) == phase_name)

  p <- p + annotate("text", x = x_pos, y = 0.7,
                    label = paste0("n=", n_phase),
                    size = 3.5, fontface = "bold", color = "black")
}

if(show_stats && !is.null(analysis_results$statistical_results)) {

  stats <- analysis_results$statistical_results

  if(!is.null(stats$overall_f_test)) {

    f_stat <- stats$overall_f_test$f_statistic
    f_p <- stats$overall_f_test$p_value

    if(f_p < 0.001) {
      legend_text <- paste0("Linear Mixed Effects: F = ", round(f_stat, 2),
                            ", p < 0.001*** (Total n=", n_participants, ")")
    } else if(f_p < 0.01) {
      legend_text <- paste0("Linear Mixed Effects: F = ", round(f_stat, 2),
                            ", p = ", format(f_p, digits = 3), "** (Total n=", n_participants, ")")
    } else if(f_p < 0.05) {
```

```
 legend_text <- paste0("Linear Mixed Effects: F = ", round(f_stat, 2),
                       ", p = ", format(f_p, digits = 3), "* (Total n=", n_participants, ")")
} else {
 legend_text <- paste0("No significant differences (p > 0.05) (Total n=", n_participants, ")")
}

p <- p + labs(subtitle = legend_text)

if(f_p < 0.05 && !is.null(stats$pairwise_t_tests) &&
  is.list(stats$pairwise_t_tests)) {

 pairwise_results <- stats$pairwise_t_tests
 significant_pairs <- c()
 sig_info <- list()

 for(comparison in names(pairwise_results)) {
  if(pairwise_results[[comparison]]$significant) {
   significant_pairs <- c(significant_pairs, comparison)
   sig_info[[comparison]] <- pairwise_results[[comparison]]
  }
 }

 if(length(significant_pairs) > 0) {
  y_max <- max(participant_averages$avg_severity, na.rm = TRUE)
  bracket_height <- 0.15

  for(i in 1:length(significant_pairs)) {
   comparison <- significant_pairs[i]

   phases <- strsplit(comparison, " - ")[[1]]
   phase1 <- trimws(phases[1])
   phase2 <- trimws(phases[2])

   x1 <- which(levels(participant_averages$phase) == phase1)
   x2 <- which(levels(participant_averages$phase) == phase2)
   y_bracket <- y_max + (i * bracket_height)

   bonf_p <- sig_info[[comparison]]$bonferroni_p_value

   if(bonf_p < 0.001) {
    sig_symbol <- "***"
   } else if(bonf_p < 0.01) {
    sig_symbol <- "**"
   } else if(bonf_p < 0.05) {
    sig_symbol <- "*"
   } else {
    sig_symbol <- ""
   }
```

```
        if(sig_symbol != "") {
          p <- p +
            annotate("segment", x = x1, xend = x2, y = y_bracket, yend = y_bracket,
                     linewidth = 0.6, color = "black") +
            annotate("segment", x = x1, xend = x1, y = y_bracket - 0.03, yend = y_bracket,
                     linewidth = 0.6, color = "black") +
            annotate("segment", x = x2, xend = x2, y = y_bracket - 0.03, yend = y_bracket,
                     linewidth = 0.6, color = "black") +
            annotate("text", x = (x1 + x2) / 2, y = y_bracket + 0.05, label = sig_symbol,
                     size = 4, fontface = "bold", color = "black")
        }
       }

       if(length(significant_pairs) > 0) {
        new_y_max <- y_max + (length(significant_pairs) * bracket_height) + 0.3
        p <- p + scale_y_continuous(
          limits = c(0.5, new_y_max),
          breaks = seq(1, ceiling(new_y_max), 0.5)
        )
       }
      }
     }
    } else {
     legend_text <- paste0("Statistical tests not available (Total n=", n_participants, ")")
     p <- p + labs(subtitle = legend_text)
    }
  }

  return(p)
}

dyskinesia_plot <- create_3phase_boxplot_modified(three_phase_data, "Dyskinesia",
dyskinesia_analysis)
print(dyskinesia_plot)
```

Script 2.6

```
rm(list = ls(pattern =
"^(multi_cycle_medication_table|single_cycle_medication_table|no_period_medication_table)$"))
rm(list = ls(pattern = "^(drug_classification_lookup|drug_file_data|medication_lookup)$"))
rm(list = ls(pattern = "^(.*_scheduled_.*|.*_adhoc_.*|.*_classified)$"))
rm(list = ls(pattern =
"^(apply_drug_classification|get_participant_medications|view_participant_medications)$"))

required_script2_vars <- c("multi_cycle_participants", "single_cycle_participants",
"no_period_participants",
                 "mini_scheduled_medication_data", "mini_adhoc_medication_data")

missing_script2_vars <- c()
for(var in required_script2_vars) {
 if(!exists(var)) {
  missing_script2_vars <- c(missing_script2_vars, var)
 }
}

if(length(missing_script2_vars) > 0) {
 if(!exists("medication_data")) {
  if(file.exists("Medication_Adherence_200525.xlsx")) {
   library(readxl)
   medication_data <- read_excel("Medication_Adherence_200525.xlsx", col_names = TRUE, skip = 1)
   medication_data <- medication_data %>%
    mutate(research_id = gsub("\\s+", "", toupper(research_id)))
  } else {
   stop("Required variables from Script 2 missing and cannot recreate from raw data: ",
      paste(missing_script2_vars, collapse = ", "))
  }
 }

 if("mini_adhoc_medication_data" %in% missing_script2_vars) {
  mini_adhoc_medication_data <- medication_data %>%
   mutate(
    research_id = gsub("\\s+", "", toupper(research_id)),
    `Adhoc Date` = as.Date(medication_due_at_date),
    Dosage = ifelse(is.na(adhoc_drug_name), NA, medication_dosage),
    `No. of Tablets` = ifelse(is.na(adhoc_drug_name), NA, medication_no_of_tablets)
   ) %>%
   select(research_id, adhoc_drug_name, Dosage, `No. of Tablets`, `Adhoc Date`) %>%
   distinct() %>%
   rename(`Adhoc med` = adhoc_drug_name)
 }

 if("mini_scheduled_medication_data" %in% missing_script2_vars) {
  mini_scheduled_medication_data <- medication_data %>%
   mutate(research_id = gsub("\\s+", "", toupper(research_id))) %>%
   filter(!is.na(scheduled_drug_name)) %>%
```

```
    select(research_id, scheduled_drug_name, medication_dosage, medication_no_of_tablets,
medication_due_at_time) %>%
    distinct() %>%
    rename(
      `Scheduled Med` = scheduled_drug_name,
      Dosage = medication_dosage,
      `No. of Tablets` = medication_no_of_tablets,
      Time = medication_due_at_time
    )
 }
}

if(!file.exists("MMM Latest Drug File 2025.xlsx")) {
  stop("Required file missing: MMM Latest Drug File 2025.xlsx\n",
     "Please ensure this file is in the working directory.")
}

library(readxl)
library(dplyr)

drug_file_data <- read_excel("MMM Latest Drug File 2025.xlsx",
                     sheet = "List",
                     col_names = FALSE,
                     skip = 0)

category_names <- as.character(drug_file_data[2, ])
category_names <- category_names[-1]
category_names[is.na(category_names)] <- ""

drug_classification_lookup <- drug_file_data %>%
  slice(4:n()) %>%
  as.matrix() %>%
  as.data.frame() %>%
  setNames(c("drug_name", paste0("cat_", 1:(ncol(.) - 1)))) %>%
  filter(!is.na(drug_name) & drug_name != "") %>%
  rowwise() %>%
  mutate(
   category = {
    drug_row <- c_across(-drug_name)
    true_positions <- which(drug_row == TRUE | drug_row == "TRUE")

    if(length(true_positions) > 0) {
     category_names[true_positions[1]]
    } else {
     "Uncategorized"
    }
   }
  ) %>%
  ungroup() %>%
```

```
  select(drug_name, category) %>%
  mutate(category = ifelse(category == "" | is.na(category), "Uncategorized", category))

print(head(drug_classification_lookup, 10))

category_summary <- drug_classification_lookup %>%
  group_by(category) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
print(category_summary)

uncategorized_count <- sum(drug_classification_lookup$category == "Uncategorized")

if(uncategorized_count > 0) {
  uncategorized_drugs <- drug_classification_lookup %>%
    filter(category == "Uncategorized") %>%
    pull(drug_name)
  print(uncategorized_drugs)
}

if(!exists("multi_cycle_participants")) {
  stop("ERROR: multi_cycle_participants not found. Please run Script 1/2 first.")
}
if(!exists("single_cycle_participants")) {
  stop("ERROR: single_cycle_participants not found. Please run Script 1/2 first.")
}
if(!exists("no_period_participants")) {
  stop("ERROR: no_period_participants not found. Please run Script 1/2 first.")
}

if(!exists("mini_scheduled_medication_data")) {
  stop("ERROR: mini_scheduled_medication_data not found. Please run Script 1/2 first.")
}
if(!exists("mini_adhoc_medication_data")) {
  stop("ERROR: mini_adhoc_medication_data not found. Please run Script 1/2 first.")
}

multi_cycle_ids <- multi_cycle_participants$research_id
single_cycle_ids <- single_cycle_participants$research_id
no_period_ids <- no_period_participants

multi_cycle_scheduled_meds <- mini_scheduled_medication_data %>%
  filter(research_id %in% multi_cycle_ids) %>%
  select(research_id, `Scheduled Med`, Dosage, `No. of Tablets`, Time)

single_cycle_scheduled_meds <- mini_scheduled_medication_data %>%
  filter(research_id %in% single_cycle_ids) %>%
  select(research_id, `Scheduled Med`, Dosage, `No. of Tablets`, Time)
```

```
no_period_scheduled_meds <- mini_scheduled_medication_data %>%
 filter(research_id %in% no_period_ids) %>%
 select(research_id, `Scheduled Med`, Dosage, `No. of Tablets`, Time)

multi_cycle_adhoc_meds <- mini_adhoc_medication_data %>%
 filter(research_id %in% multi_cycle_ids) %>%
 select(research_id, `Adhoc med`, Dosage, `No. of Tablets`, `Adhoc Date`)

single_cycle_adhoc_meds <- mini_adhoc_medication_data %>%
 filter(research_id %in% single_cycle_ids) %>%
 select(research_id, `Adhoc med`, Dosage, `No. of Tablets`, `Adhoc Date`)

no_period_adhoc_meds <- mini_adhoc_medication_data %>%
 filter(research_id %in% no_period_ids) %>%
 select(research_id, `Adhoc med`, Dosage, `No. of Tablets`, `Adhoc Date`)

if(nrow(multi_cycle_scheduled_meds) > 0) {
 print(head(multi_cycle_scheduled_meds, 5))
}

if(nrow(multi_cycle_adhoc_meds) > 0) {
 print(head(multi_cycle_adhoc_meds, 5))
}

summary_stats <- data.frame(
 Group = c("Multi-cycle", "Single-cycle", "No-period"),
 Participants = c(length(multi_cycle_ids), length(single_cycle_ids), length(no_period_ids)),
 Scheduled_Meds = c(nrow(multi_cycle_scheduled_meds), nrow(single_cycle_scheduled_meds),
nrow(no_period_scheduled_meds)),
 Adhoc_Meds = c(nrow(multi_cycle_adhoc_meds), nrow(single_cycle_adhoc_meds),
nrow(no_period_adhoc_meds))
)

print(summary_stats)

multi_cycle_with_scheduled <- n_distinct(multi_cycle_scheduled_meds$research_id)
multi_cycle_with_adhoc <- n_distinct(multi_cycle_adhoc_meds$research_id)

single_cycle_with_scheduled <- n_distinct(single_cycle_scheduled_meds$research_id)
single_cycle_with_adhoc <- n_distinct(single_cycle_adhoc_meds$research_id)

no_period_with_scheduled <- n_distinct(no_period_scheduled_meds$research_id)
no_period_with_adhoc <- n_distinct(no_period_adhoc_meds$research_id)

multi_cycle_scheduled_standardized <- multi_cycle_scheduled_meds %>%
 filter(!is.na(`Scheduled Med`) & `Scheduled Med` != "") %>%
 mutate(
  medication_name = `Scheduled Med`,
  dosage = Dosage,
```

```
   no_of_tablets = `No. of Tablets`,
   medication_type = "Scheduled",
   time_info = Time,
   date_taken = NA_character_
  ) %>%
  select(research_id, medication_name, dosage, no_of_tablets, medication_type, time_info,
date_taken) %>%
  distinct(research_id, medication_name, dosage, .keep_all = TRUE)

multi_cycle_adhoc_standardized <- multi_cycle_adhoc_meds %>%
  filter(!is.na(`Adhoc med`) & `Adhoc med` != "") %>%
  mutate(
   medication_name = `Adhoc med`,
   dosage = Dosage,
   no_of_tablets = `No. of Tablets`,
   medication_type = "Adhoc",
   time_info = NA_character_,
   date_taken = as.character(`Adhoc Date`)
  ) %>%
  select(research_id, medication_name, dosage, no_of_tablets, medication_type, time_info, date_taken)

single_cycle_scheduled_standardized <- single_cycle_scheduled_meds %>%
  filter(!is.na(`Scheduled Med`) & `Scheduled Med` != "") %>%
  mutate(
   medication_name = `Scheduled Med`,
   dosage = Dosage,
   no_of_tablets = `No. of Tablets`,
   medication_type = "Scheduled",
   time_info = Time,
   date_taken = NA_character_
  ) %>%
  select(research_id, medication_name, dosage, no_of_tablets, medication_type, time_info,
date_taken) %>%
  distinct(research_id, medication_name, dosage, .keep_all = TRUE)

single_cycle_adhoc_standardized <- single_cycle_adhoc_meds %>%
  filter(!is.na(`Adhoc med`) & `Adhoc med` != "") %>%
  mutate(
   medication_name = `Adhoc med`,
   dosage = Dosage,
   no_of_tablets = `No. of Tablets`,
   medication_type = "Adhoc",
   time_info = NA_character_,
   date_taken = as.character(`Adhoc Date`)
  ) %>%
  select(research_id, medication_name, dosage, no_of_tablets, medication_type, time_info, date_taken)

no_period_scheduled_standardized <- no_period_scheduled_meds %>%
  filter(!is.na(`Scheduled Med`) & `Scheduled Med` != "") %>%
```

```
  mutate(
   medication_name = `Scheduled Med`,
   dosage = Dosage,
   no_of_tablets = `No. of Tablets`,
   medication_type = "Scheduled",
   time_info = Time,
   date_taken = NA_character_
  ) %>%
  select(research_id, medication_name, dosage, no_of_tablets, medication_type, time_info,
date_taken) %>%
  distinct(research_id, medication_name, dosage, .keep_all = TRUE)

no_period_adhoc_standardized <- no_period_adhoc_meds %>%
  filter(!is.na(`Adhoc med`) & `Adhoc med` != "") %>%
  mutate(
   medication_name = `Adhoc med`,
   dosage = Dosage,
   no_of_tablets = `No. of Tablets`,
   medication_type = "Adhoc",
   time_info = NA_character_,
   date_taken = as.character(`Adhoc Date`)
  ) %>%
  select(research_id, medication_name, dosage, no_of_tablets, medication_type, time_info, date_taken)

standardization_summary <- data.frame(
  Group = c("Multi-cycle", "Multi-cycle", "Single-cycle", "Single-cycle", "No-period", "No-period"),
  Type = c("Scheduled", "Adhoc", "Scheduled", "Adhoc", "Scheduled", "Adhoc"),
  Original_Entries = c(nrow(multi_cycle_scheduled_meds), nrow(multi_cycle_adhoc_meds),
                nrow(single_cycle_scheduled_meds), nrow(single_cycle_adhoc_meds),
                nrow(no_period_scheduled_meds), nrow(no_period_adhoc_meds)),
  Standardized_Entries = c(nrow(multi_cycle_scheduled_standardized),
nrow(multi_cycle_adhoc_standardized),
                   nrow(single_cycle_scheduled_standardized), nrow(single_cycle_adhoc_standardized),
                   nrow(no_period_scheduled_standardized), nrow(no_period_adhoc_standardized)),
  Filtered_Out = c(nrow(multi_cycle_scheduled_meds) - nrow(multi_cycle_scheduled_standardized),
             nrow(multi_cycle_adhoc_meds) - nrow(multi_cycle_adhoc_standardized),
             nrow(single_cycle_scheduled_meds) - nrow(single_cycle_scheduled_standardized),
             nrow(single_cycle_adhoc_meds) - nrow(single_cycle_adhoc_standardized),
             nrow(no_period_scheduled_meds) - nrow(no_period_scheduled_standardized),
             nrow(no_period_adhoc_meds) - nrow(no_period_adhoc_standardized))
)

print(standardization_summary)

scheduled_dedup_summary <- data.frame(
  Group = c("Multi-cycle", "Single-cycle", "No-period"),
  Original_Scheduled = c(nrow(multi_cycle_scheduled_meds), nrow(single_cycle_scheduled_meds),
nrow(no_period_scheduled_meds)),
```

```
 Deduplicated_Scheduled = c(nrow(multi_cycle_scheduled_standardized),
nrow(single_cycle_scheduled_standardized), nrow(no_period_scheduled_standardized)),
 Duplicates_Removed = c(nrow(multi_cycle_scheduled_meds) -
nrow(multi_cycle_scheduled_standardized),
                nrow(single_cycle_scheduled_meds) - nrow(single_cycle_scheduled_standardized),
                nrow(no_period_scheduled_meds) - nrow(no_period_scheduled_standardized))
)

print(scheduled_dedup_summary)

if(nrow(multi_cycle_scheduled_standardized) > 0) {
 print(head(multi_cycle_scheduled_standardized, 3))
}

if(nrow(multi_cycle_adhoc_standardized) > 0) {
 print(head(multi_cycle_adhoc_standardized, 3))
}

if(nrow(no_period_scheduled_standardized) > 0) {
 print(head(no_period_scheduled_standardized, 3))
}

datasets <- list(
 multi_cycle_scheduled_standardized, multi_cycle_adhoc_standardized,
 single_cycle_scheduled_standardized, single_cycle_adhoc_standardized,
 no_period_scheduled_standardized, no_period_adhoc_standardized
)

all_same_structure <- all(sapply(datasets, function(x) identical(names(x), names(datasets[[1]]))))

ucc033567_check <- multi_cycle_scheduled_standardized %>%
 filter(research_id == "UCC033567", medication_name == "Amantadine")

ucc040405_check <- multi_cycle_scheduled_standardized %>%
 filter(research_id == "UCC040405", medication_name == "Madopar")

apply_drug_classification <- function(medication_data, dataset_name) {

 classified_data <- medication_data %>%
  left_join(drug_classification_lookup, by = c("medication_name" = "drug_name")) %>%
  mutate(category = ifelse(is.na(category), "UNMATCHED", category))

 return(classified_data)
}

multi_cycle_scheduled_classified <- apply_drug_classification(
 multi_cycle_scheduled_standardized, "Multi-cycle Scheduled"
)
```

```
multi_cycle_adhoc_classified <- apply_drug_classification(
 multi_cycle_adhoc_standardized, "Multi-cycle Adhoc"
)

single_cycle_scheduled_classified <- apply_drug_classification(
 single_cycle_scheduled_standardized, "Single-cycle Scheduled"
)

single_cycle_adhoc_classified <- apply_drug_classification(
 single_cycle_adhoc_standardized, "Single-cycle Adhoc"
)

no_period_scheduled_classified <- apply_drug_classification(
 no_period_scheduled_standardized, "No-period Scheduled"
)

no_period_adhoc_classified <- apply_drug_classification(
 no_period_adhoc_standardized, "No-period Adhoc"
)

all_classified_datasets <- list(
 "Multi-cycle Scheduled" = multi_cycle_scheduled_classified,
 "Multi-cycle Adhoc" = multi_cycle_adhoc_classified,
 "Single-cycle Scheduled" = single_cycle_scheduled_classified,
 "Single-cycle Adhoc" = single_cycle_adhoc_classified,
 "No-period Scheduled" = no_period_scheduled_classified,
 "No-period Adhoc" = no_period_adhoc_classified
)

unmatched_drugs_list <- data.frame()

for(dataset_name in names(all_classified_datasets)) {
 dataset <- all_classified_datasets[[dataset_name]]

 unmatched_in_dataset <- dataset %>%
  filter(category == "UNMATCHED") %>%
  select(medication_name) %>%
  distinct() %>%
  mutate(found_in_dataset = dataset_name)

 if(nrow(unmatched_in_dataset) > 0) {
  unmatched_drugs_list <- bind_rows(unmatched_drugs_list, unmatched_in_dataset)
 }
}

if(nrow(unmatched_drugs_list) > 0) {
 unique_unmatched_drugs <- unmatched_drugs_list %>%
  group_by(medication_name) %>%
  summarise(
```

```
    found_in_datasets = paste(found_in_dataset, collapse = ", "),
    .groups = 'drop'
   )

 for(i in 1:nrow(unique_unmatched_drugs)) {
 }

} else {
 unique_unmatched_drugs <- data.frame()
}

for(dataset_name in names(all_classified_datasets)) {
 dataset <- all_classified_datasets[[dataset_name]]

 category_dist <- dataset %>%
  group_by(category) %>%
  summarise(count = n(), .groups = 'drop') %>%
  arrange(desc(count))
}

if(nrow(multi_cycle_scheduled_classified) > 0) {
 print(head(multi_cycle_scheduled_classified %>% select(research_id, medication_name, category,
medication_type), 3))
}

if(nrow(no_period_adhoc_classified) > 0) {
 print(head(no_period_adhoc_classified %>% select(research_id, medication_name, category,
medication_type), 3))
}

total_entries <- sum(sapply(all_classified_datasets, nrow))
total_unmatched <- sum(sapply(all_classified_datasets, function(x) sum(x$category ==
"UNMATCHED")))
total_matched <- total_entries - total_unmatched

multi_cycle_medication_table <- bind_rows(
 multi_cycle_scheduled_classified,
 multi_cycle_adhoc_classified
) %>%
 arrange(research_id, medication_type, category, medication_name) %>%
 select(research_id, medication_name, category, dosage, no_of_tablets,
      medication_type, time_info, date_taken)

single_cycle_medication_table <- bind_rows(
 single_cycle_scheduled_classified,
 single_cycle_adhoc_classified
) %>%
 arrange(research_id, medication_type, category, medication_name) %>%
 select(research_id, medication_name, category, dosage, no_of_tablets,
```

```
      medication_type, time_info, date_taken)

no_period_medication_table <- bind_rows(
 no_period_scheduled_classified,
 no_period_adhoc_classified
) %>%
 arrange(research_id, medication_type, category, medication_name) %>%
 select(research_id, medication_name, category, dosage, no_of_tablets,
      medication_type, time_info, date_taken)

final_summary <- data.frame(
 Table = c("Multi-cycle", "Single-cycle", "No-period"),
 Participants = c(
  n_distinct(multi_cycle_medication_table$research_id),
  n_distinct(single_cycle_medication_table$research_id),
  n_distinct(no_period_medication_table$research_id)
 ),
 Total_Entries = c(
  nrow(multi_cycle_medication_table),
  nrow(single_cycle_medication_table),
  nrow(no_period_medication_table)
 ),
 Scheduled_Entries = c(
  sum(multi_cycle_medication_table$medication_type == "Scheduled"),
  sum(single_cycle_medication_table$medication_type == "Scheduled"),
  sum(no_period_medication_table$medication_type == "Scheduled")
 ),
 Adhoc_Entries = c(
  sum(multi_cycle_medication_table$medication_type == "Adhoc"),
  sum(single_cycle_medication_table$medication_type == "Adhoc"),
  sum(no_period_medication_table$medication_type == "Adhoc")
 )
)

print(final_summary)

tables_list <- list(
 "Multi-cycle" = multi_cycle_medication_table,
 "Single-cycle" = single_cycle_medication_table,
 "No-period" = no_period_medication_table
)

for(table_name in names(tables_list)) {
 table_data <- tables_list[[table_name]]

 category_dist <- table_data %>%
  group_by(category) %>%
  summarise(count = n(), .groups = 'drop') %>%
  arrange(desc(count)) %>%
```

```
  head(5)
}

if(nrow(multi_cycle_medication_table) > 0) {
 sample_multi <- multi_cycle_medication_table %>%
  head(5) %>%
  select(research_id, medication_name, category, medication_type, time_info, date_taken)
 print(sample_multi)
}

if(nrow(single_cycle_medication_table) > 0) {
 sample_single <- single_cycle_medication_table %>%
  head(5) %>%
  select(research_id, medication_name, category, medication_type, time_info, date_taken)
 print(sample_single)
}

if(nrow(no_period_medication_table) > 0) {
 sample_no_period <- no_period_medication_table %>%
  head(5) %>%
  select(research_id, medication_name, category, medication_type, time_info, date_taken)
 print(sample_no_period)
}

expected_columns <- c("research_id", "medication_name", "category", "dosage",
                      "no_of_tablets", "medication_type", "time_info", "date_taken")

tables_to_check <- list(
 "multi_cycle_medication_table" = multi_cycle_medication_table,
 "single_cycle_medication_table" = single_cycle_medication_table,
 "no_period_medication_table" = no_period_medication_table
)

all_correct_structure <- TRUE

for(table_name in names(tables_to_check)) {
 table_data <- tables_to_check[[table_name]]
 table_columns <- names(table_data)

 structure_correct <- identical(table_columns, expected_columns)

 if(!structure_correct) {
  all_correct_structure <- FALSE
 }
}

total_final_entries <- nrow(multi_cycle_medication_table) +
 nrow(single_cycle_medication_table) +
 nrow(no_period_medication_table)
```

```
get_participant_medications <- function(research_id) {
 research_id <- gsub("\\s+", "", toupper(research_id))

 participant_group <- NA
 participant_data <- NULL

 if(research_id %in% multi_cycle_medication_table$research_id) {
  participant_group <- "Multi-cycle"
  participant_data <- multi_cycle_medication_table %>%
   filter(research_id == !!research_id)
 }
 else if(research_id %in% single_cycle_medication_table$research_id) {
  participant_group <- "Single-cycle"
  participant_data <- single_cycle_medication_table %>%
   filter(research_id == !!research_id)
 }
 else if(research_id %in% no_period_medication_table$research_id) {
  participant_group <- "No-period"
  participant_data <- no_period_medication_table %>%
   filter(research_id == !!research_id)
 }

 if(is.null(participant_data) || nrow(participant_data) == 0) {
  return(NULL)
 }

 result <- list(
  research_id = research_id,
  participant_group = participant_group,
  total_medications = nrow(participant_data),
  scheduled_medications = sum(participant_data$medication_type == "Scheduled"),
  adhoc_medications = sum(participant_data$medication_type == "Adhoc"),
  unique_categories = n_distinct(participant_data$category),
  medication_data = participant_data
 )

 return(result)
}

print_participant_medication_summary <- function(participant_info) {
 if(is.null(participant_info)) {
  return(invisible())
 }

 category_summary <- participant_info$medication_data %>%
  group_by(category, medication_type) %>%
  summarise(count = n(), .groups = 'drop') %>%
  arrange(category, medication_type)
```

```
  scheduled_meds <- participant_info$medication_data %>%
    filter(medication_type == "Scheduled") %>%
    select(medication_name, category, dosage, no_of_tablets, time_info) %>%
    distinct()

  adhoc_count <- participant_info$medication_data %>%
    filter(medication_type == "Adhoc") %>%
    group_by(medication_name, category) %>%
    summarise(total_taken = n(), .groups = 'drop') %>%
    arrange(desc(total_taken))
}

view_participant_medications <- function(research_id) {
  participant_info <- get_participant_medications(research_id)
  print_participant_medication_summary(participant_info)
  return(participant_info)
}

sample_multi_cycle <- multi_cycle_medication_table$research_id[1]
sample_single_cycle <- single_cycle_medication_table$research_id[1]
sample_no_period <- no_period_medication_table$research_id[1]

multi_test <- get_participant_medications(sample_multi_cycle)

single_test <- get_participant_medications(sample_single_cycle)

no_period_test <- get_participant_medications(sample_no_period)

invalid_test <- get_participant_medications("INVALID123")

participant_meds <- get_participant_medications('UCC033567')

view_participant_medications('UCC033567')

participant_meds$medication_data
participant_meds$participant_group
participant_meds$total_medications
```

Script 15

```
essential_variables <- c("categories_data", "new_menstruationstatus_table",
"mini_scheduled_medication_data", "medication_data")

for(var in essential_variables) {
 if(exists(var)) {
 } else {
  if(var == "categories_data") {
  } else if(var == "new_menstruationstatus_table") {
  } else if(var == "mini_scheduled_medication_data") {
  } else if(var == "medication_data") {
  }
 }
}

if(exists("categories_data")) {
 if("L-DOPA" %in% names(categories_data)) {
  ldopa_medication_list_from_mmm <- categories_data$`L-DOPA`[!is.na(categories_data$`L-DOPA`)
& categories_data$`L-DOPA` != ""]
 }
}

if(exists("new_menstruationstatus_table")) {
 print(table(new_menstruationstatus_table$Description))
}

missing_count <- sum(!sapply(essential_variables, exists))

unique_scheduled_meds <- unique(mini_scheduled_medication_data$`Scheduled Med`)

potential_ldopa_keywords <- c("sinemet", "madopar", "stalevo", "parcopa", "rytary", "ldopa", "l-dopa",
"duodopa", "lecado")
potential_matches <- c()
for(keyword in potential_ldopa_keywords) {
 matches <- unique_scheduled_meds[grepl(keyword, unique_scheduled_meds, ignore.case = TRUE)]
 if(length(matches) > 0) {
  potential_matches <- c(potential_matches, matches)
 }
}

if(length(potential_matches) > 0) {
 print(unique(potential_matches))
}

ldopa_scheduled_medications <- mini_scheduled_medication_data %>%
 filter(`Scheduled Med` %in% ldopa_medication_list_from_mmm)

if(nrow(ldopa_scheduled_medications) > 0) {
 ldopa_meds_in_data <- unique(ldopa_scheduled_medications$`Scheduled Med`)
```

```
 print(ldopa_meds_in_data)

 print(head(ldopa_scheduled_medications, 5))
}

print(summary(ldopa_scheduled_medications$Dosage))

print(summary(ldopa_scheduled_medications$`No. of Tablets`))

extreme_dosages <- ldopa_scheduled_medications %>%
 filter(Dosage > 1000 | Dosage < 1) %>%
 arrange(desc(Dosage))

if(nrow(extreme_dosages) > 0) {
 print(extreme_dosages)

 scientific_pattern <- ldopa_scheduled_medications %>%
  filter(Dosage %in% c(100, 250, 2500, 25100)) %>%
  arrange(desc(Dosage))

 if(nrow(scientific_pattern) > 0) {
  print(scientific_pattern)
 }
}

extreme_tablets <- ldopa_scheduled_medications %>%
 filter(`No. of Tablets` > 20 | `No. of Tablets` < 0.25) %>%
 arrange(desc(`No. of Tablets`))

if(nrow(extreme_tablets) > 0) {
 print(extreme_tablets)
}

ldopa_scheduled_clean <- ldopa_scheduled_medications %>%
 mutate(
  dosage_outlier_flag = case_when(
   Dosage > 1000 ~ "Extremely High Dosage (>1000mg)",
   Dosage < 1 ~ "Extremely Low Dosage (<1mg)",
   Dosage %in% c(25100, 50200) ~ "Potential Concatenation Error",
   Dosage > 500 ~ "High Dosage (>500mg)",
   TRUE ~ "Normal Range"
  ),

  tablet_outlier_flag = case_when(
   `No. of Tablets` > 20 ~ "Extremely High Tablet Count",
   `No. of Tablets` < 0.25 ~ "Fractional Tablet Count",
   `No. of Tablets` > 10 ~ "High Tablet Count",
   TRUE ~ "Normal Range"
  ),
```

```
   outlier_status = case_when(
    dosage_outlier_flag != "Normal Range" | tablet_outlier_flag != "Normal Range" ~ "FLAGGED",
    TRUE ~ "OK"
   )
 )

outlier_summary <- ldopa_scheduled_clean %>%
 group_by(outlier_status) %>%
 summarise(
  count = n(),
  participants = n_distinct(research_id),
  .groups = 'drop'
 )
print(outlier_summary)

dosage_flags <- table(ldopa_scheduled_clean$dosage_outlier_flag)
print(dosage_flags)

if(sum(ldopa_scheduled_clean$outlier_status == "FLAGGED") > 0) {
 flagged_entries <- ldopa_scheduled_clean %>%
  filter(outlier_status == "FLAGGED") %>%
  select(research_id, `Scheduled Med`, Dosage, `No. of Tablets`, dosage_outlier_flag,
tablet_outlier_flag)
 print(flagged_entries)
}

if(!file.exists("ldopa_component_lookup_template_name_parsed.csv")) {
 stop("ERROR: Please ensure the completed lookup table
'ldopa_component_lookup_template_name_parsed.csv' exists in your working directory")
}

ldopa_lookup_completed <- read.csv("ldopa_component_lookup_template_name_parsed.csv",
stringsAsFactors = FALSE)

completed_entries <- sum(!is.na(ldopa_lookup_completed$L_DOPA_Position) &
                ldopa_lookup_completed$L_DOPA_Position != "" &
                !is.na(ldopa_lookup_completed$L_DOPA_Value_Check) &
                ldopa_lookup_completed$L_DOPA_Value_Check != "")

if(completed_entries == 0) {
 stop("ERROR: Lookup table appears to be empty. Please complete the L_DOPA_Position and
L_DOPA_Value_Check columns")
}

extract_correct_ldopa_dose <- function(medication_name, logged_dosage) {
 medication_name <- trimws(as.character(medication_name))
 logged_dosage <- as.numeric(logged_dosage)
```

```
 lookup_row <- ldopa_lookup_completed[ldopa_lookup_completed$Medication_Name ==
medication_name, ]

 if(nrow(lookup_row) == 0) {
  return(list(
   ldopa_dose = logged_dosage,
   method = "fallback_logged_dosage",
   details = paste("Medication", medication_name, "not found in lookup table")
  ))
 }

 ldopa_position <- lookup_row$L_DOPA_Position[1]
 ldopa_value_check <- lookup_row$L_DOPA_Value_Check[1]
 pattern_type <- lookup_row$Pattern_Type[1]

 if(is.na(ldopa_position) || ldopa_position == "" ||
   is.na(ldopa_value_check) || ldopa_value_check == "") {
  return(list(
   ldopa_dose = logged_dosage,
   method = "fallback_logged_dosage",
   details = paste("Incomplete lookup data for", medication_name)
  ))
 }

 ldopa_position <- as.numeric(ldopa_position)
 ldopa_value_check <- as.numeric(ldopa_value_check)

 if(ldopa_position == 0) {
  return(list(
   ldopa_dose = logged_dosage,
   method = "logged_dosage_unparseable",
   details = paste("Used logged dosage for unparseable medication", medication_name)
  ))
 }

 if(ldopa_position %in% c(1, 2, 3)) {
  return(list(
   ldopa_dose = ldopa_value_check,
   method = "lookup_table_parsed",
   details = paste("Used component", ldopa_position, "from parsed medication", medication_name)
  ))
 }

 return(list(
  ldopa_dose = logged_dosage,
  method = "fallback_invalid_position",
  details = paste("Invalid L_DOPA_Position", ldopa_position, "for", medication_name)
 ))
}
```

```
ldopa_final_doses <- ldopa_scheduled_clean %>%
 rowwise() %>%
 mutate(
  extraction_result = list(extract_correct_ldopa_dose(`Scheduled Med`, Dosage)),
  final_dose = extraction_result$ldopa_dose,
  dose_source = extraction_result$method,
  dose_pattern = extraction_result$details,

  was_corrected = dose_source %in% c("lookup_table_parsed"),
  dosage_field_flag = case_when(
   dose_source == "fallback_logged_dosage" ~ "used_logged_dosage",
   dose_source == "logged_dosage_unparseable" ~ "unparseable_medication",
   dose_source == "lookup_table_parsed" ~ "parsed_from_name",
   TRUE ~ "other"
  )
 ) %>%
 ungroup() %>%
 select(-extraction_result)

extraction_summary <- ldopa_final_doses %>%
 group_by(dose_source) %>%
 summarise(
  count = n(),
  medications = n_distinct(`Scheduled Med`),
  participants = n_distinct(research_id),
  avg_dose = round(mean(final_dose, na.rm = TRUE), 2),
  min_dose = min(final_dose, na.rm = TRUE),
  max_dose = max(final_dose, na.rm = TRUE),
  .groups = 'drop'
 )

print(extraction_summary)

for(method in unique(ldopa_final_doses$dose_source)) {
 examples <- ldopa_final_doses %>%
  filter(dose_source == method) %>%
  select(research_id, `Scheduled Med`, original_dosage = Dosage, final_dose, dose_pattern) %>%
  head(3)
 print(examples)
}

problematic_entries <- ldopa_final_doses %>%
 filter(final_dose > 1000 | final_dose < 1 | is.na(final_dose)) %>%
 select(research_id, `Scheduled Med`, original_dosage = Dosage, final_dose, dose_source,
dose_pattern) %>%
 arrange(desc(final_dose))

if(nrow(problematic_entries) > 0) {
```

```
  print(problematic_entries)
}

participant_medication_doses <- ldopa_final_doses %>%
  group_by(research_id, `Scheduled Med`) %>%
  summarise(
    daily_dose_this_med = sum(final_dose * `No. of Tablets`, na.rm = TRUE),
    doses_per_day = n_distinct(Time),
    total_administrations = n(),
    avg_dose_per_admin = mean(final_dose, na.rm = TRUE),
    avg_tablets_per_admin = mean(`No. of Tablets`, na.rm = TRUE),
    dose_source = first(dose_source),
    had_correction = any(was_corrected),
    used_lookup_table = first(dose_source) == "lookup_table_parsed",
    .groups = 'drop'
  )

participant_daily_ldopa_doses <- participant_medication_doses %>%
  group_by(research_id) %>%
  summarise(
    total_ldopa_medications = n(),
    total_daily_ldopa_dose = sum(daily_dose_this_med, na.rm = TRUE),
    max_single_med_dose = max(daily_dose_this_med, na.rm = TRUE),
    total_daily_administrations = sum(total_administrations),
    had_any_corrections = any(had_correction),
    used_any_lookup_table = any(used_lookup_table),
    used_lookup_table_count = sum(dose_source == "lookup_table_parsed"),
    used_logged_dosage_count = sum(dose_source %in% c("fallback_logged_dosage",
"logged_dosage_unparseable")),
    medication_list = paste(`Scheduled Med`, collapse = "; "),
    .groups = 'drop'
  ) %>%
  mutate(
    dose_flag = case_when(
      total_daily_ldopa_dose > 1500 ~ "Extremely High (>1500mg/day)",
      total_daily_ldopa_dose > 800 ~ "High (>800mg/day)",
      total_daily_ldopa_dose < 50 ~ "Very Low (<50mg/day)",
      TRUE ~ "Normal Range"
    )
  )

print(summary(participant_daily_ldopa_doses$total_daily_ldopa_dose))

dose_flags <- table(participant_daily_ldopa_doses$dose_flag)
print(dose_flags)

final_summary <- participant_daily_ldopa_doses %>%
  summarise(
    total_participants = n(),
```

```
  had_corrections = sum(had_any_corrections),
  used_lookup_table = sum(used_any_lookup_table),
  used_only_lookup_table = sum(used_logged_dosage_count == 0),
  used_only_logged_dosage = sum(used_lookup_table_count == 0),
  used_mixed_sources = sum(used_lookup_table_count > 0 & used_logged_dosage_count > 0),
  avg_daily_dose = round(mean(total_daily_ldopa_dose), 2),
  median_daily_dose = round(median(total_daily_ldopa_dose), 2)
 )

print(final_summary)

comparison_sample <- ldopa_final_doses %>%
 filter(dose_source == "lookup_table_parsed") %>%
 select(research_id, `Scheduled Med`, original_dosage = Dosage, corrected_ldopa_dose = final_dose,
dose_pattern) %>%
 head(5)

if(nrow(comparison_sample) > 0) {
 print(comparison_sample)
}

write.csv(ldopa_final_doses, "ldopa_corrected_individual_entries_step4.csv", row.names = FALSE)

write.csv(participant_daily_ldopa_doses, "ldopa_corrected_daily_doses_step4.csv", row.names =
FALSE)

ldopa_with_status <- participant_daily_ldopa_doses %>%
 left_join(
  new_menstruationstatus_table %>% select(UniqueID, Description),
  by = c("research_id" = "UniqueID")
 ) %>%
 rename(menstruation_status = Description)

status_distribution <- table(ldopa_with_status$menstruation_status, useNA = "ifany")
print(status_distribution)

ldopa_final_dataset <- ldopa_with_status %>%
 filter(menstruation_status %in% c("Menstruating", "Postmenopausal")) %>%
 mutate(
  menstruation_group = factor(menstruation_status, levels = c("Menstruating", "Postmenopausal"))
 )

dose_comparison <- ldopa_final_dataset %>%
 group_by(menstruation_status) %>%
 summarise(
  n = n(),
  mean_dose = round(mean(total_daily_ldopa_dose, na.rm = TRUE), 1),
  median_dose = round(median(total_daily_ldopa_dose, na.rm = TRUE), 1),
  sd_dose = round(sd(total_daily_ldopa_dose, na.rm = TRUE), 1),
```

```
   min_dose = min(total_daily_ldopa_dose, na.rm = TRUE),
   max_dose = max(total_daily_ldopa_dose, na.rm = TRUE),
   q25 = round(quantile(total_daily_ldopa_dose, 0.25, na.rm = TRUE), 1),
   q75 = round(quantile(total_daily_ldopa_dose, 0.75, na.rm = TRUE), 1),
   .groups = 'drop'
  )

print(dose_comparison)

dose_categories <- ldopa_final_dataset %>%
  group_by(menstruation_status, dose_flag) %>%
  summarise(count = n(), .groups = 'drop') %>%
  pivot_wider(names_from = dose_flag, values_from = count, values_fill = 0) %>%
  arrange(menstruation_status)

print(dose_categories)

excluded_participants <- ldopa_with_status %>%
  filter(!menstruation_status %in% c("Menstruating", "Postmenopausal") | is.na(menstruation_status))

if(nrow(excluded_participants) > 0) {
  excluded_summary <- excluded_participants %>%
    mutate(exclusion_reason = ifelse(is.na(menstruation_status), "Missing Status",
as.character(menstruation_status))) %>%
    group_by(exclusion_reason) %>%
    summarise(count = n(), .groups = 'drop')
  print(excluded_summary)

  print(excluded_participants %>%
        select(research_id, total_daily_ldopa_dose, menstruation_status) %>%
        head(5))
}

if(nrow(ldopa_final_dataset) >= 10) {
  menstruating_doses <- ldopa_final_dataset %>%
    filter(menstruation_status == "Menstruating") %>%
    pull(total_daily_ldopa_dose)

  postmenopausal_doses <- ldopa_final_dataset %>%
    filter(menstruation_status == "Postmenopausal") %>%
    pull(total_daily_ldopa_dose)

  if(length(menstruating_doses) >= 3 && length(postmenopausal_doses) >= 3) {
    wilcox_test <- wilcox.test(menstruating_doses, postmenopausal_doses)
  }
}

library(ggplot2)
library(dplyr)
```

```
menstruating_doses <- ldopa_final_dataset %>%
 filter(menstruation_status == "Menstruating") %>%
 pull(total_daily_ldopa_dose)

postmenopausal_doses <- ldopa_final_dataset %>%
 filter(menstruation_status == "Postmenopausal") %>%
 pull(total_daily_ldopa_dose)

desc_stats <- ldopa_final_dataset %>%
 group_by(menstruation_status) %>%
 summarise(
  n = n(),
  mean = round(mean(total_daily_ldopa_dose), 1),
  median = round(median(total_daily_ldopa_dose), 1),
  sd = round(sd(total_daily_ldopa_dose), 1),
  min = min(total_daily_ldopa_dose),
  max = max(total_daily_ldopa_dose),
  q25 = round(quantile(total_daily_ldopa_dose, 0.25), 1),
  q75 = round(quantile(total_daily_ldopa_dose, 0.75), 1),
  iqr = round(q75 - q25, 1),
  .groups = 'drop'
 )

print(desc_stats)

wilcox_result <- wilcox.test(menstruating_doses, postmenopausal_doses,
                  alternative = "two.sided", exact = FALSE)

n1 <- length(menstruating_doses)
n2 <- length(postmenopausal_doses)
U_statistic <- wilcox_result$statistic
effect_size <- abs(2 * U_statistic / (n1 * n2) - 1)

median_diff <- median(menstruating_doses) - median(postmenopausal_doses)

if(wilcox_result$p.value < 0.001) {
 interpretation <- "Highly significant difference (p < 0.001)"
} else if(wilcox_result$p.value < 0.01) {
 interpretation <- "Significant difference (p < 0.01)"
} else if(wilcox_result$p.value < 0.05) {
 interpretation <- "Significant difference (p < 0.05)"
} else {
 interpretation <- "No significant difference (p >= 0.05)"
}

Q1 <- quantile(ldopa_final_dataset$total_daily_ldopa_dose, 0.25)
Q3 <- quantile(ldopa_final_dataset$total_daily_ldopa_dose, 0.75)
IQR <- Q3 - Q1
```

```
outlier_threshold <- Q3 + 2 * IQR

extreme_outliers <- ldopa_final_dataset %>%
 filter(total_daily_ldopa_dose > outlier_threshold) %>%
 arrange(desc(total_daily_ldopa_dose))

if(nrow(extreme_outliers) > 0) {
 print(extreme_outliers %>%
       select(research_id, menstruation_status, total_daily_ldopa_dose) %>%
       head(5))
}

plot_max <- min(5000, max(ldopa_final_dataset$total_daily_ldopa_dose))

clean_boxplot <- ggplot(ldopa_final_dataset, aes(x = menstruation_status, y = total_daily_ldopa_dose)) +

 geom_boxplot(aes(fill = menstruation_status),
          alpha = 0.7,
          outlier.shape = 16,
          outlier.size = 1.5,
          outlier.alpha = 0.6,
          outlier.color = "red") +

 geom_jitter(aes(color = menstruation_status),
         width = 0.25,
         alpha = 0.7,
         size = 2,
         stroke = 0.5) +

 stat_summary(fun = median, geom = "text",
          aes(label = paste("Median:", round(..y.., 0), "mg")),
          vjust = -0.5, size = 3.5, fontface = "bold") +

 scale_fill_manual(values = c("Menstruating" = "#E74C3C", "Postmenopausal" = "#3498DB"),
             name = "Group") +
 scale_color_manual(values = c("Menstruating" = "#C0392B", "Postmenopausal" = "#2980B9"),
              guide = "none") +

 coord_cartesian(ylim = c(0, plot_max)) +

 labs(
  title = "Daily L-DOPA Dose Comparison: Menstruating vs Postmenopausal",
  subtitle = paste0("Mann-Whitney U = ", wilcox_result$statistic,
               ", p = ", format(wilcox_result$p.value, digits = 3),
               " (", interpretation, ")"),
  x = "Menstruation Status",
  y = "Total Daily L-DOPA Dose (mg)",
  caption = if(nrow(extreme_outliers) > 0) {
   paste0("Note: ", nrow(extreme_outliers), " extreme outliers (>",
```

```
        round(outlier_threshold, 0), "mg) not shown for clarity")
  } else {
    "All data points shown"
  }
 ) +

 theme_minimal(base_size = 12) +
 theme(
  plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
  plot.subtitle = element_text(hjust = 0.5, size = 11,
                               color = if(wilcox_result$p.value < 0.05) "red" else "darkgreen"),
  plot.caption = element_text(size = 9, color = "gray60"),
  legend.position = "bottom",
  panel.grid.major.x = element_blank(),
  panel.grid.minor = element_blank(),
  axis.title.x = element_text(margin = margin(t = 10)),
  axis.title.y = element_text(margin = margin(r = 10))
 ) +

 scale_y_continuous(
  breaks = seq(0, plot_max, 500),
  labels = function(x) paste0(x, "mg"),
  expand = expansion(mult = c(0, 0.1))
 ) +

 annotate("text", x = 1, y = plot_max * 0.95,
       label = paste("n =", n1), size = 4, fontface = "bold") +
 annotate("text", x = 2, y = plot_max * 0.95,
       label = paste("n =", n2), size = 4, fontface = "bold")

print(clean_boxplot)

dose_category_analysis <- ldopa_final_dataset %>%
 group_by(menstruation_status, dose_flag) %>%
 summarise(count = n(), .groups = 'drop') %>%
 group_by(menstruation_status) %>%
 mutate(
  total = sum(count),
  percentage = round(count / total * 100, 1)
 ) %>%
 ungroup()

print(dose_category_analysis)

dose_contingency <- ldopa_final_dataset %>%
 select(menstruation_status, dose_flag) %>%
 table()

print(dose_contingency)
```

```
expected_frequencies <- chisq.test(dose_contingency)$expected
min_expected <- min(expected_frequencies)

if(min_expected >= 5) {
  chisq_result <- chisq.test(dose_contingency)
  print(chisq_result)
}

menstruating_sd <- desc_stats %>% filter(menstruation_status == "Menstruating") %>% pull(sd)
postmenopausal_sd <- desc_stats %>% filter(menstruation_status == "Postmenopausal") %>% pull(sd)
menstruating_se <- round(menstruating_sd / sqrt(n1), 1)
postmenopausal_se <- round(postmenopausal_sd / sqrt(n2), 1)

menstruating_normal <- sum(ldopa_final_dataset$menstruation_status == "Menstruating" &
                           ldopa_final_dataset$dose_flag == "Normal Range")
postmenopausal_normal <- sum(ldopa_final_dataset$menstruation_status == "Postmenopausal" &
                             ldopa_final_dataset$dose_flag == "Normal Range")
```

Follow-up Survey Scripts

Script 102.

```
graphics.off()

library(dplyr)
library(readxl)
library(writexl)

followup_data <- read.csv('MMMFollowUpSurvey_DATA_2025-06-26_0959.csv', colClasses =
"character")

eligible_research_ids <- read_excel('Research ID and final survey - Tracking.xlsx')

print(colnames(eligible_research_ids))

eligible_research_ids$standardized_id <- toupper(gsub("\\s+", "", eligible_research_ids$`Research ID
Original`))

if(is.null(eligible_research_ids$standardized_id) || all(is.na(eligible_research_ids$standardized_id))) {
  eligible_research_ids$standardized_id <- toupper(gsub("\\s+", "", eligible_research_ids[[2]]))
}

completed_followup_surveys <- followup_data %>%
  filter(my_moves_matter_followup_survey_timestamp != "" &
       my_moves_matter_followup_survey_timestamp != "[not completed]")

followup_ids <- completed_followup_surveys

valid_rows <- which(!is.na(followup_ids$p2) & !is.na(followup_ids$a2) &
                 followup_ids$p2 != "" & followup_ids$a2 != "")

followup_ids$StandardisedID <- NA
for (i in valid_rows) {
  id <- paste0("UCC", followup_ids$p2[i], followup_ids$a2[i])

  id <- gsub("\\s+", "", id)
  id <- toupper(id)

  followup_ids$StandardisedID[i] <- id
}

test_ids <- c("UCC089WV4", "UCC123456", "UCC123ABC")
followup_ids <- followup_ids[!followup_ids$StandardisedID %in% test_ids, ]

followup_ids$survey_timestamp <-
as.POSIXct(followup_ids$my_moves_matter_followup_survey_timestamp,
                          format="%Y-%m-%d %H:%M:%S")
```

```
duplicate_ids <- followup_ids$StandardisedID[duplicated(followup_ids$StandardisedID)]

if (length(duplicate_ids) > 0) {
  deduped_followup <- followup_ids %>%
    group_by(StandardisedID) %>%
    arrange(desc(survey_timestamp)) %>%
    summarise(across(everything(), ~first(na.omit(.))), .groups = "drop") %>%
    ungroup()
} else {
  deduped_followup <- followup_ids
}

unique_followup_ids <- unique(deduped_followup$StandardisedID)
unique_followup_ids <- unique_followup_ids[!is.na(unique_followup_ids)]

total_unique_followup_participants <- length(unique_followup_ids)

print(unique_followup_ids)

id_summary <- data.frame(
  Research_ID = unique_followup_ids,
  ID_Length = nchar(unique_followup_ids)
)
print(id_summary)

completed_followup_survey <- eligible_research_ids$standardized_id %in% unique_followup_ids

participants_completed_followup <- eligible_research_ids[completed_followup_survey, ]
participants_not_completed_followup <- eligible_research_ids[!completed_followup_survey, ]

completed_followup_count <- nrow(participants_completed_followup)
not_completed_followup_count <- nrow(participants_not_completed_followup)

print(participants_completed_followup$standardized_id)

print(participants_not_completed_followup$standardized_id)

not_completed_followup_ids <- data.frame(
  Research_ID = participants_not_completed_followup$standardized_id
)

completed_followup_ids <- data.frame(
  Research_ID = participants_completed_followup$standardized_id
)

write_xlsx(not_completed_followup_ids, "participants_who_did_not_complete_followup_survey.xlsx")
write_xlsx(completed_followup_ids, "participants_who_completed_followup_survey.xlsx")

tracking_file <- read_excel("Research ID and final survey - Tracking.xlsx")
```

```
tracking_file$Updated_Completion_Status <- tracking_file$`Research ID Original` %in%
participants_completed_followup$standardized_id

write_xlsx(tracking_file, "Research ID and final survey Tracking - Updated.xlsx")

sc_participants_df <- read_excel("single_cycle_participants05.03.25..xlsx")

followup_tracking_df <- read_excel("Research ID and final survey - Tracking.xlsx")

sc_participants_df$sc_standardized_id <- toupper(gsub("\\s+", "", sc_participants_df$research_id))

followup_tracking_df$tracking_standardized_id <- toupper(gsub("\\s+", "",
followup_tracking_df$`Research ID Original`))

sc_not_in_tracking_df <- sc_participants_df %>%
  filter(!(sc_standardized_id %in% followup_tracking_df$tracking_standardized_id)) %>%
  select(research_id, sc_standardized_id)

total_sc_participants <- nrow(sc_participants_df)
total_sc_not_in_tracking <- nrow(sc_not_in_tracking_df)
total_sc_in_tracking <- total_sc_participants - total_sc_not_in_tracking

write_xlsx(sc_not_in_tracking_df, "single_cycle_participants_not_in_tracking.xlsx")
```

Script 103.

```
library(Hmisc)
library(ggplot2)
library(dplyr)
library(tidyr)
library(lubridate)

data <- read.csv('MMMFollowUpSurvey_DATA_2025-06-26_0959.csv', colClasses = "character")

data <- data %>%
  filter(my_moves_matter_followup_survey_timestamp != "" &
       my_moves_matter_followup_survey_timestamp != "[not completed]")

fu_data_UID <- data
fu_data_UID$p2 <- as.character(fu_data_UID$p2)
fu_data_UID$a2 <- as.character(fu_data_UID$a2)

fu_data_UID$original_p2 <- fu_data_UID$p2
fu_data_UID$original_a2 <- fu_data_UID$a2

fu_data_UID$p2 <- gsub(" ", "", fu_data_UID$p2)
fu_data_UID$p2 <- trimws(fu_data_UID$p2)

fu_data_UID$a2 <- gsub(" ", "", fu_data_UID$a2)
fu_data_UID$a2 <- toupper(fu_data_UID$a2)
fu_data_UID$a2 <- trimws(fu_data_UID$a2)

fu_data_UID$UniqueID <- paste0("UCC", fu_data_UID$p2, fu_data_UID$a2)

test_ids <- c("UCC089WV4", "UCC123456", "UCC123ABC")
fu_data_UID <- fu_data_UID[!fu_data_UID$UniqueID %in% test_ids, ]

fu_data_UID$survey_timestamp <-
as.POSIXct(fu_data_UID$my_moves_matter_followup_survey_timestamp,
                        format="%Y-%m-%d %H:%M:%S")

duplicate_ids <- fu_data_UID$UniqueID[duplicated(fu_data_UID$UniqueID)]

if (length(duplicate_ids) > 0) {
  fu_data_UID <- fu_data_UID %>%
    group_by(UniqueID) %>%
    arrange(desc(survey_timestamp)) %>%
    summarise(across(everything(), ~first(na.omit(.))), .groups = "drop") %>%
    ungroup()
}

fu_data_UID <- fu_data_UID[, !colnames(fu_data_UID) %in% c('redcap_repeat_instrument',
                                'redcap_repeat_instance',
```

```
                                    'redcap_survey_identifier',
                                    'p2', 'a2', 'survey_timestamp',
                                    'original_p2', 'original_a2')]

fu_data_UID <- fu_data_UID[, c("record_id", "UniqueID",
                    setdiff(colnames(fu_data_UID),
                        c("record_id", "UniqueID")))]

legitimate_participants <- unique(c(
 data_UID$UniqueID,
 app_users,
 eligible_research_ids$standardized_id
))

before_filter_count <- nrow(fu_data_UID)
fu_data_UID <- fu_data_UID[fu_data_UID$UniqueID %in% legitimate_participants, ]
after_filter_count <- nrow(fu_data_UID)

create_demographic_table <- function(data, var_name, levels, labels, table_name = NULL) {
 result_table <- data.frame(
  RecordID = data$record_id,
  UniqueID = data$UniqueID,
  Code = data[[var_name]],
  Description = factor(data[[var_name]], levels = levels, labels = labels)
 )

 summary_table <- as.data.frame(table(result_table$Description, useNA = "ifany"))
 names(summary_table) <- c("Description", "Count")

 summary_table <- summary_table[!is.na(summary_table$Description) &
                    summary_table$Description != "Prefer not to say", ]

 summary_table$Percentage <- (summary_table$Count / sum(summary_table$Count)) * 100

 if (!is.null(table_name)) {
  assign(paste0("fu_", table_name, "_table"), result_table, envir = parent.frame())
  assign(paste0("fu_", table_name, "_summary_table"), summary_table, envir = parent.frame())
 }

 return(list(table = result_table, summary = summary_table))
}

create_checkbox_table <- function(data, column_pattern, labels, table_name = NULL) {
 cols <- grep(column_pattern, colnames(data), value = TRUE)

 if (length(cols) != length(labels)) {
  warning(paste("Mismatch in", column_pattern, "- Found", length(cols),
          "columns but have", length(labels), "labels"))
  if (length(cols) < length(labels)) {
```

```
    labels <- labels[1:length(cols)]
   } else {
    labels <- c(labels, paste("Column", (length(labels)+1):length(cols)))
   }
  }

  result_table <- data[, cols]

  result_table <- lapply(result_table, function(x) as.numeric(as.character(x)))
  result_table <- as.data.frame(result_table)

  colnames(result_table) <- labels

  result_table <- cbind(data$record_id, data$UniqueID, result_table)
  colnames(result_table)[1:2] <- c("RecordID", "UniqueID")

  if (!is.null(table_name)) {
   assign(paste0("fu_", table_name, "_table"), result_table, envir = parent.frame())
  }

  return(result_table)
}

if("s02_mental_health" %in% colnames(fu_data_UID)) {
  fu_mental_health_result <- create_demographic_table(
   fu_data_UID, "s02_mental_health",
   levels = c(1, 2, 3, 4, 5),
   labels = c("Very good", "Good", "Fair", "Poor", "Very poor"),
   table_name = "mental_health"
  )
}

if("s02_mental_health_effect" %in% colnames(fu_data_UID)) {
  fu_mental_health_effect_result <- create_demographic_table(
   fu_data_UID, "s02_mental_health_effect",
   levels = c(1, 2, 3),
   labels = c("Yes, it has had a positive impact",
          "No impact",
          "Yes, it has had a negative impact"),
   table_name = "mental_health_effect"
  )
}

if("s01_surgeries_ea988a" %in% colnames(fu_data_UID)) {
  fu_procedures_result <- create_demographic_table(
   fu_data_UID, "s01_surgeries_ea988a",
   levels = c(1, 2, 3),
   labels = c("Yes", "No", "Prefer not to say"),
   table_name = "procedures"
```

```
 )
}

motor_symptoms_labels <- c("Tremors or shaking",
                           "Stiffness or rigidity",
                           "Slowness of movement",
                           "Balance and coordination issues",
                           "Postural instability",
                           "Changes in speech and writing",
                           "Difficulty with swallowing",
                           "Other",
                           "Prefer not to say")

selected_motor_cols <- c(1:6, 8:10)
motor_cols <- paste0("s01_motor_symptoms_a38ff2___", selected_motor_cols)

motor_data <- fu_data_UID[, motor_cols]
motor_data <- lapply(motor_data, function(x) as.numeric(as.character(x)))
motor_data <- as.data.frame(motor_data)

colnames(motor_data) <- motor_symptoms_labels

fu_motorsymptoms_table <- cbind(fu_data_UID$record_id, fu_data_UID$UniqueID, motor_data)
colnames(fu_motorsymptoms_table)[1:2] <- c("RecordID", "UniqueID")

motorsymptoms2_labels <- c(
 "Dyskinesias",
 "Muscle Cramps",
 "Restless Leg Syndrome",
 "Other",
 "Prefer not to say"
)

if(any(grepl("s02_ms_2___", colnames(fu_data_UID)))) {
 fu_motorsymptoms2_table <- create_checkbox_table(
  fu_data_UID,
  "s02_ms_2___",
  motorsymptoms2_labels,
  "motorsymptoms2"
 )
}

nonmotor_labels <- c(
 "Depression",
 "Anxiety",
 "Sleep disturbances, including insomnia and restless leg syndrome",
 "Loss of sense of smell",
 "Constipation",
 "Cognitive changes, including memory problems, confusion, and dementia",
```

```
  "Difficulty Concentrating",
  "Fatigue and lack of energy",
  "Excessive sweating",
  "Sexual dysfunction",
  "Orthostatic hypotension (low blood pressure upon standing)",
  "Visual disturbances, including blurry vision or double vision",
  "Speech and swallowing difficulties",
  "Pain and discomfort",
  "Mood swings and emotional changes",
  "Skin problems, including dry skin and rashes",
  "Weight loss or gain",
  "Urinary Problems",
  "Other",
  "Prefer not to say"
)

fu_nonmotor_table <- create_checkbox_table(
  fu_data_UID,
  "s01_nonmotor_4b9bb3___",
  nonmotor_labels,
  "nonmotor"
)

nonmotor_support_labels <- c(
  "Depression and anxiety",
  "Sleep disturbances",
  "Loss of sense of smell",
  "Constipation",
  "Cognitive changes",
  "Difficulty Concentrating",
  "Fatigue",
  "Excessive sweating",
  "Sexual difficulties",
  "Low blood pressure",
  "Visual disturbances",
  "Speech and swallowing difficulties",
  "Pain",
  "Mood swings",
  "Skin problems",
  "Weight changes",
  "Urinary Problems",
  "Other",
  "None of the above",
  "Prefer not to say"
)

if(any(grepl("s02_nonmotor_support___", colnames(fu_data_UID)))) {
  fu_nonmotor_support_table <- create_checkbox_table(
    fu_data_UID,
```

```
  "s02_nonmotor_support___",
  nonmotor_support_labels,
  "nonmotor_support"
 )
}

pastmonth_labels <- c(
 "Dribbling of saliva during the daytime",
 "Loss or change in your ability to taste or smell",
 "Difficulty swallowing food or drink or problems with choking",
 "Vomiting or feelings of sickness (nausea)",
 "Constipation (less than three bowel movements a week) or having to strain to pass a stool",
 "Bowel (faecal) incontinence",
 "Feeling that your bowel emptying is incomplete after having been to the toilet",
 "A sense of urgency to pass urine makes you rush to the toilet",
 "Getting up regularly at night to pass urine",
 "Unexplained pains (not due to known conditions such as arthritis)",
 "Unexplained change in weight (not due to change in diet)",
 "Problems remembering things that have happened recently or forgetting to do things",
 "Loss of interest in what is happening around you or in doing things",
 "Seeing or hearing things that you know or are told are not there",
 "Difficulty concentrating or staying focused",
 "Feeling sad, 'low' or 'blue'",
 "Feeling anxious, frightened or panicky",
 "Feeling less interested in sex or more interested in sex",
 "Finding it difficult to have sex when you try",
 "Feeling light-headed, dizzy or weak standing from sitting or lying",
 "Falling",
 "Finding it difficult to stay awake during activities such as working, driving or eating",
 "Difficulty getting to sleep at night or staying asleep at night",
 "Intense, vivid or frightening dreams",
 "Talking or moving about in your sleep, as if you are 'acting out' a dream",
 "Unpleasant sensations in your legs at night or while resting, and a feeling that you need to move",
 "Swelling of the legs",
 "Excessive sweating",
 "Double vision",
 "Believing things are happening to you that other people say are not",
 "I have not experienced any of these in the past month"
)

pastmonth_cols <- grep("s02_symptomspastmonth_2aca72___", colnames(fu_data_UID), value = TRUE)
if(length(pastmonth_cols) > 0) {
 pastmonth_indices <- as.numeric(gsub("s02_symptomspastmonth_2aca72___", "", pastmonth_cols))

 pastmonth_mapping <- data.frame(
  ColumnName = pastmonth_cols,
  Index = pastmonth_indices,
  stringsAsFactors = FALSE
 )
```

```
 pastmonth_mapping <- pastmonth_mapping[order(pastmonth_mapping$Index),]

 ordered_pastmonth_labels <- rep(NA, length(pastmonth_mapping$Index))
 for (i in 1:length(pastmonth_mapping$Index)) {
  if (pastmonth_mapping$Index[i] <= length(pastmonth_labels)) {
   ordered_pastmonth_labels[i] <- pastmonth_labels[pastmonth_mapping$Index[i]]
  } else {
   ordered_pastmonth_labels[i] <- paste("Symptom", pastmonth_mapping$Index[i])
  }
 }

 fu_pastmonth_table <- fu_data_UID[, pastmonth_mapping$ColumnName]
 fu_pastmonth_table <- lapply(fu_pastmonth_table, function(x) as.numeric(as.character(x)))
 fu_pastmonth_table <- as.data.frame(fu_pastmonth_table)
 colnames(fu_pastmonth_table) <- ordered_pastmonth_labels
 fu_pastmonth_table <- cbind(fu_data_UID$record_id, fu_data_UID$UniqueID, fu_pastmonth_table)
 colnames(fu_pastmonth_table)[1:2] <- c("RecordID", "UniqueID")
}

pdmeds_labels <- c(
 "Adartel", "Adodespan PR", "Amantadine", "Apogo", "Apokyn",
 "Apomorphine", "Artane", "Azilect", "Benztropine", "Bromocriptine",
 "Caramet CR", "Clonazepam", "Co-Beneldopa", "Co-Careldopa", "Comtan",
 "Comtess", "Cogentin", "Dacepton", "Duodopa", "Duopa",
 "Eldepryl", "Gocovri", "Inbrija", "Ipinnia", "Kemadrin",
 "Kynmobi", "Lecado", "Madopar", "Mirapex", "Mirapexin",
 "Neupro", "Nourianz", "Oprymea", "Ongentys", "Osmolex",
 "Opicapone", "Parlodel", "Parcopa", "Pipexus", "Pramipexole",
 "RalneaXL", "Rasagiline", "Requip", "Raponer", "Rivotril",
 "Ropilynz", "Ropinirole", "Rotigotine", "Rytary", "Sastravi",
 "Selegiline", "Slepark", "Sinemet", "Sinemet Plus", "Spirico",
 "Stalevo", "Stanek", "Symmetrel", "Tasmar", "Tolcapone",
 "Trihexyphenidryl/Benzhexol", "Xadago", "Zantiva", "Zelapar",
 "Other", "None of the above"
)

pdmeds_cols <- grep("^s01t_", colnames(fu_data_UID), value = TRUE)

fu_pdmeds_table <- data.frame(RecordID = fu_data_UID$record_id, UniqueID = fu_data_UID$UniqueID)

for (label in pdmeds_labels) {
 fu_pdmeds_table[[label]] <- 0
}

for (col in pdmeds_cols) {
 for (label in pdmeds_labels) {
  simple_label <- tolower(gsub("[^a-zA-Z0-9]", "", label))
```

```
   if (grepl(paste0("s01t_", tolower(simple_label)), tolower(col))) {
    med_values <- as.numeric(as.character(fu_data_UID[[col]]))
    fu_pdmeds_table[[label]] <- pmax(fu_pdmeds_table[[label]], med_values, na.rm = TRUE)
   }
  }
}

fu_pdmeds_summary_table <- data.frame(
 Description = pdmeds_labels,
 Count = colSums(fu_pdmeds_table[, -(1:2)], na.rm = TRUE)
)

total_responses <- nrow(fu_data_UID) - sum(is.na(fu_pdmeds_table$UniqueID))
fu_pdmeds_summary_table$Percentage <- (fu_pdmeds_summary_table$Count / total_responses) * 100

fu_pdmeds_summary_table <- fu_pdmeds_summary_table[fu_pdmeds_summary_table$Count > 0, ]

othermeds_labels <- c(
 "Thyroid medication for underactive thyroid",
 "Thyroid medication for overactive thyroid",
 "HRT",
 "Other",
 "None of the above"
)

fu_othermeds_table <- create_checkbox_table(
 fu_data_UID,
 "s01_othermeds_3203ed___",
 othermeds_labels,
 "othermeds"
)

contraceptive_labels <- c(
 "an implant",
 "an intrauterine device (IUD)",
 "injections",
 "oral contraception",
 "vaginal rings",
 "skin patches",
 "I am not currently taking any of these medications"
)

fu_contraceptive_table <- create_checkbox_table(
 fu_data_UID,
 "s02_othermedshor_2___",
 contraceptive_labels,
 "contraceptive"
)
```

```
therapy_labels <- c(
 "Physical therapy",
 "Speech therapy",
 "Occupational therapy",
 "Nutrition therapy",
 "None of the above",
 "Prefer not to say",
 "Other"
)

fu_therapy_table <- create_checkbox_table(
 fu_data_UID,
 "s01_current_treatments_9d0fc9___",
 therapy_labels,
 "therapy"
)

supports_labels <- c(
 "Family",
 "Friends",
 "Healthcare providers",
 "Support groups",
 "Mental health professional",
 "Other",
 "I don't have any supports"
)

if(any(grepl("s02_supports___", colnames(fu_data_UID)))) {
 fu_supports_table <- create_checkbox_table(
  fu_data_UID,
  "s02_supports___",
  supports_labels,
  "supports"
 )
}

fu_menstruationstatus_table <- data.frame(
 RecordID = fu_data_UID$record_id,
 UniqueID = fu_data_UID$UniqueID,
 Code = fu_data_UID$s01_menstruationstatus_ad677f,
 Description = factor(fu_data_UID$s01_menstruationstatus_ad677f,
              levels = c(1, 2, 3, 4, 5),
              labels = c("Actively menstruating",
                   "Post menstruation (no active period for 1 year)",
                   "Pregnant",
                   "Other",
                   "Prefer not to say"))
)
```

```
if("s01_menstruationstatus_other_8c4835" %in% colnames(fu_data_UID)) {
 fu_menstruationstatus_table$Specify <- ifelse(
  fu_menstruationstatus_table$Code == 4,
  fu_data_UID$s01_menstruationstatus_other_8c4835,
  NA
 )
}

fu_menstruationstatus_summary_table <-
as.data.frame(table(fu_menstruationstatus_table$Description, useNA = "no"))
names(fu_menstruationstatus_summary_table) <- c("Description", "Count")
fu_menstruationstatus_summary_table$Percentage <- (fu_menstruationstatus_summary_table$Count /
sum(fu_menstruationstatus_summary_table$Count)) * 100

fu_filtered_menstruation_table <- fu_menstruationstatus_table[
 fu_menstruationstatus_table$Description %in% c("Actively menstruating", "Post menstruation (no
active period for 1 year)"),
]

fu_filtered_menstruation_table$DisplayDescription <- factor(
 fu_filtered_menstruation_table$Description,
 levels = c("Actively menstruating", "Post menstruation (no active period for 1 year)"),
 labels = c("Menstruating", "Post menstruation")
)

fu_filtered_summary_table <- as.data.frame(table(fu_filtered_menstruation_table$DisplayDescription))
names(fu_filtered_summary_table) <- c("Description", "Count")
fu_filtered_summary_table$Percentage <- (fu_filtered_summary_table$Count /
sum(fu_filtered_summary_table$Count)) * 100

fu_new_menstruationstatus_table <- fu_menstruationstatus_table
fu_new_menstruationstatus_summary_table <- fu_menstruationstatus_summary_table

print(fu_new_menstruationstatus_summary_table)

fu_motorsymptoms_simple_table <- fu_motorsymptoms_table
colnames(fu_motorsymptoms_simple_table) <- c("RecordID", "UniqueID", "Tremors", "Rigidity",
"Slowness", "Imbalance",
                             "Postural instability", "Speech/Writing",
                             "Swallowing",
                             "Other", "Prefer not to say")

fu_nonmotorsymptoms_simple_table <- fu_nonmotor_table
colnames(fu_nonmotorsymptoms_simple_table) <- c("RecordID", "UniqueID", "Depression", "Anxiety",
"Insomnia", "Hyposmia",
                                "Constipation", "Cognitive Changes", "Concentration", "Fatigue",
                                "Sweating", "Sexual dysfunction", "Hypotension", "Vision",
                                "Speech", "Pain", "Mood swings", "Skin", "Weight",
                                "Urinary", "Other", "Prefer not to say")
```

```
fu_motorsymptoms_table <- fu_motorsymptoms_simple_table
fu_nonmotorsymptoms_table <- fu_nonmotorsymptoms_simple_table

fu_hormonal_table <- data.frame(
 RecordID = fu_contraceptive_table$RecordID,
 UniqueID = fu_contraceptive_table$UniqueID,
 Yes = ifelse(rowSums(fu_contraceptive_table[, 3:(ncol(fu_contraceptive_table)-1)], na.rm = TRUE) > 0, 1,
0),
 No = ifelse(fu_contraceptive_table$`I am not currently taking any of these medications` == 1, 1, 0)
)

fu_med_category_table <- data.frame(
 RecordID = fu_pdmeds_table$RecordID,
 UniqueID = fu_pdmeds_table$UniqueID
)

ldopa_meds <- c("Adodespan PR", "Caramet CR", "Co-Beneldopa", "Co-Careldopa",
          "Duodopa", "Duopa", "Inbrija", "Lecado", "Madopar",
          "Parcopa", "Rytary", "Sinemet", "Sinemet Plus")
da_agonist_meds <- c("Adartel", "Amantadine", "Apogo", "Apokyn",
               "Apomorphine", "Bromocriptine", "Dacepton",
               "Ipinnia", "Kynmobi", "Mirapex", "Mirapexin",
               "Neupro", "Oprymea", "Parlodel", "Pipexus",
               "Pramipexole", "RalneaXL", "Requip", "Raponer",
               "Ropilynz", "Ropinirole", "Rotigotine", "Spirico", "Zantiva")
maob_inhibitor_meds <- c("Azilect", "Eldepryl", "Rasagiline", "Selegiline", "Xadago", "Zelapar")
anticholinergic_meds <- c("Artane", "Benztropine", "Cogentin", "Kemadrin",
"Trihexyphenidryl/Benzhexol")
comt_inhibitor_meds <- c("Comtan", "Comtess", "Ongentys", "Opicapone", "Sastravi", "Stalevo",
"Stanek", "Tasmar", "Tolcapone")
benzodiazepine_meds <- c("Clonazepam", "Rivotril")
adenosine_antagonist_meds <- c("Nourianz")

fu_med_category_table$L_DOPA <- as.numeric(rowSums(fu_pdmeds_table[, intersect(ldopa_meds,
names(fu_pdmeds_table)), drop = FALSE], na.rm = TRUE) > 0)
fu_med_category_table$DA_Agonist <- as.numeric(rowSums(fu_pdmeds_table[,
intersect(da_agonist_meds, names(fu_pdmeds_table)), drop = FALSE], na.rm = TRUE) > 0)
fu_med_category_table$MAOB_Inhibitors <- as.numeric(rowSums(fu_pdmeds_table[,
intersect(maob_inhibitor_meds, names(fu_pdmeds_table)), drop = FALSE], na.rm = TRUE) > 0)
fu_med_category_table$Anticholinergics <- as.numeric(rowSums(fu_pdmeds_table[,
intersect(anticholinergic_meds, names(fu_pdmeds_table)), drop = FALSE], na.rm = TRUE) > 0)
fu_med_category_table$COMT_Inhibitors <- as.numeric(rowSums(fu_pdmeds_table[,
intersect(comt_inhibitor_meds, names(fu_pdmeds_table)), drop = FALSE], na.rm = TRUE) > 0)
fu_med_category_table$Benzodiazepines <- as.numeric(rowSums(fu_pdmeds_table[,
intersect(benzodiazepine_meds, names(fu_pdmeds_table)), drop = FALSE], na.rm = TRUE) > 0)
fu_med_category_table$Adenosine_Antagonist <- as.numeric(rowSums(fu_pdmeds_table[,
intersect(adenosine_antagonist_meds, names(fu_pdmeds_table)), drop = FALSE], na.rm = TRUE) > 0)
```

```
label(data$record_id) <- "Record ID"
label(data$my_moves_matter_followup_survey_timestamp) <- "Follow-up Survey Timestamp"

if ("s02_mental_health" %in% colnames(data)) {
 label(data$s02_mental_health) <- "How would you describe your mental health?"
}

if ("s02_mental_health_effect" %in% colnames(data)) {
 label(data$s02_mental_health_effect) <- "Has Parkinson's disease had an impact on your mental
health?"
}

if ("s01_surgeries_ea988a" %in% colnames(data)) {
 label(data$s01_surgeries_ea988a) <- "Have you ever had any procedures/surgeries for Parkinsons
disease?"
}

if ("s01_menstruationstatus_ad677f" %in% colnames(data)) {
 label(data$s01_menstruationstatus_ad677f) <- "What is your menstruation status?"
}

for (i in 1:10) {
 field_name <- paste0("s01_motor_symptoms_a38ff2___", i)
 if (field_name %in% colnames(data)) {
  if (i == 9) {
   label(data[[field_name]]) <- paste0("What are the main motor symptoms you experience due to
Parkinsons disease?  (choice=", motor_symptoms_labels[i], " {s01_motor_other})")
  } else {
   label(data[[field_name]]) <- paste0("What are the main motor symptoms you experience due to
Parkinsons disease?  (choice=", motor_symptoms_labels[i], ")")
  }
 }
}

for (i in 1:20) {
 field_name <- paste0("s01_nonmotor_4b9bb3___", i)
 if (field_name %in% colnames(data)) {
  if (i == 19) {
   label(data[[field_name]]) <- paste0("Do you experience any non-motor symptoms associated with
Parkinsons disease? (choice=", nonmotor_labels[i], "{s01_nonmotor_other})")
  } else {
   label(data[[field_name]]) <- paste0("Do you experience any non-motor symptoms associated with
Parkinsons disease? (choice=", nonmotor_labels[i], ")")
  }
 }
}

if("s02_mental_health" %in% colnames(data)) {
 data$s02_mental_health.factor = factor(data$s02_mental_health, levels=c("1","2","3","4","5"))
```

```
}

if("s02_mental_health_effect" %in% colnames(data)) {
 data$s02_mental_health_effect.factor = factor(data$s02_mental_health_effect, levels=c("1","2","3"))
}

if("s01_surgeries_ea988a" %in% colnames(data)) {
 data$s01_surgeries_ea988a.factor = factor(data$s01_surgeries_ea988a, levels=c("1","2","3"))
}

if("s01_menstruationstatus_ad677f" %in% colnames(data)) {
 data$s01_menstruationstatus_ad677f.factor = factor(data$s01_menstruationstatus_ad677f,
levels=c("1","2","3","4","5"))
}

key_tables <- c("fu_mental_health_table", "fu_mental_health_effect_table", "fu_procedures_table",
          "fu_motorsymptoms_table", "fu_nonmotor_table", "fu_pastmonth_table",
          "fu_pdmeds_table", "fu_othermeds_table", "fu_contraceptive_table",
          "fu_therapy_table", "fu_supports_table", "fu_hormonal_table",
          "fu_med_category_table", "fu_new_menstruationstatus_table")

for (table_name in key_tables) {
 if (exists(table_name)) {
  table_obj <- get(table_name)
 }
}

print(table(fu_new_menstruationstatus_table$Description))

save.image(file = "fu_master_data_preparation_environment.RData")

fu_master_symptom_orders <- list(
 motor_symptoms = c("Tremors", "Rigidity", "Slowness", "Imbalance",
            "Postural instability", "Speech/Writing", "Swallowing"),

 nonmotor_symptoms = c("Depression/Anxiety", "Sleep", "Hyposmia", "Constipation",
              "Cognitive Changes", "Concentration", "Fatigue", "Sweating",
              "Sexual Dysfunction", "Hypotension", "Vision", "Speech",
              "Pain", "Mood Swings", "Skin", "Weight", "Urinary"),

 medication_categories = c("L_DOPA", "DA_Agonist", "MAOB_Inhibitors",
                "Anticholinergics", "COMT_Inhibitors")
)

fu_apply_standard_symptom_order <- function(results_df, symptom_type) {
 if (!symptom_type %in% names(fu_master_symptom_orders)) {
  warning("Unknown symptom type: ", symptom_type)
  return(results_df)
 }
```

```
  standard_order <- fu_master_symptom_orders[[symptom_type]]

  results_df$SymptomName <- factor(results_df$SymptomName, levels = standard_order)
  results_df <- results_df[order(results_df$SymptomName), ]
  results_df$SymptomName <- as.character(results_df$SymptomName)

  return(results_df)
}

fu_ensure_complete_symptom_coverage <- function(results_df, symptom_type) {
  if (!symptom_type %in% names(fu_master_symptom_orders)) {
    return(results_df)
  }

  standard_symptoms <- fu_master_symptom_orders[[symptom_type]]
  missing_symptoms <- setdiff(standard_symptoms, results_df$SymptomName)

  if (length(missing_symptoms) > 0) {
    missing_rows <- data.frame(
      SymptomName = missing_symptoms,
      stringsAsFactors = FALSE
    )

    other_cols <- setdiff(names(results_df), "SymptomName")
    for (col in other_cols) {
      if (is.numeric(results_df[[col]])) {
        missing_rows[[col]] <- NA_real_
      } else {
        missing_rows[[col]] <- NA_character_
      }
    }

    results_df <- rbind(results_df, missing_rows)
  }

  return(results_df)
}
```

Script 105.

```
library(dplyr)
library(tidyr)
library(knitr)
library(kableExtra)

fu_motorsymptoms_table <- fu_motorsymptoms_table[rowSums(fu_motorsymptoms_table[,-(1:2)]) >
0, ]
fu_nonmotorsymptoms_table <-
fu_nonmotorsymptoms_table[rowSums(fu_nonmotorsymptoms_table[,-(1:2)]) > 0, ]
fu_med_category_table <- fu_med_category_table[rowSums(fu_med_category_table[,-(1:2)]) > 0, ]

prepare_analysis_data <- function(df, col1, col2, status_filter = c("Actively menstruating", "Post
menstruation (no active period for 1 year)")) {
 df <- df[df$Description %in% status_filter, ]

 df$DisplayDescription <- factor(
  df$Description,
  levels = status_filter,
  labels = c("Menstruating", "Post menstruation")
 )

 df[[col1]] <- as.character(df[[col1]])
 df[[col2]] <- as.character(df[[col2]])
 return(df)
}

perform_statistical_test <- function(contingency_table) {
 if (nrow(contingency_table) < 2 || ncol(contingency_table) < 2) {
  return(list(p.value = NA, method = "Insufficient data"))
 }

 if (any(contingency_table == 0)) {
  return(fisher.test(contingency_table))
 }

 expected_counts <- chisq.test(contingency_table, simulate.p.value = TRUE)$expected

 if (any(expected_counts < 5)) {
  return(fisher.test(contingency_table))
 } else {
  return(chisq.test(contingency_table, correct = TRUE))
 }
}

display_analysis_results <- function(variable_name, contingency_table, test_result) {
 col_names <- colnames(contingency_table)
 n_values <- colSums(contingency_table)
```

```
  percentage_table <- prop.table(contingency_table, margin = 2) * 100

  print(contingency_table)

  colnames(percentage_table) <- paste0(col_names, " (N=", n_values, ") (%)")
  print(round(percentage_table, 2))

  test_name <- if(class(test_result)[1] == "htest") test_result$method else "Fisher's Exact Test"

  return(list(
    contingency_table = contingency_table,
    percentage_table = percentage_table,
    p_value = test_result$p.value,
    test = test_name
  ))
}

analyze_variable <- function(data_table_name, variable, variable_display_name) {
  if (!exists(data_table_name)) {
    return(NULL)
  }

  df <- get(data_table_name)

  if (!variable %in% colnames(df)) {
    return(NULL)
  }

  analysis_data <- merge(fu_new_menstruationstatus_table, df, by = c("RecordID", "UniqueID"))
  analysis_data <- prepare_analysis_data(analysis_data, "Description", variable)

  if (nrow(analysis_data) == 0) {
    return(NULL)
  }

  contingency_table <- table(analysis_data[[variable]], analysis_data$DisplayDescription)

  test_result <- perform_statistical_test(contingency_table)

  return(display_analysis_results(variable_display_name, contingency_table, test_result))
}

analyze_coded_variable <- function(data_table_name, variable_display_name) {
  if (!exists(data_table_name)) {
    return(NULL)
  }

  df <- get(data_table_name)
```

```
 filtered_menstruation_ids <- fu_new_menstruationstatus_table$UniqueID[
  fu_new_menstruationstatus_table$Description %in% c("Actively menstruating", "Post menstruation
(no active period for 1 year)")
 ]

 valid_data <- df$UniqueID %in% filtered_menstruation_ids & !is.na(df$Code)

 if (sum(valid_data) == 0) {
  return(NULL)
 }

 df_filtered <- df[valid_data, ]
 df_filtered$CodeValue <- as.character(df_filtered$Code)

 filtered_menstruation <- fu_new_menstruationstatus_table[
  fu_new_menstruationstatus_table$Description %in% c("Actively menstruating", "Post menstruation
(no active period for 1 year)"),
  c("RecordID", "UniqueID", "Description")
 ]

 filtered_menstruation$DisplayDescription <- factor(
  filtered_menstruation$Description,
  levels = c("Actively menstruating", "Post menstruation (no active period for 1 year)"),
  labels = c("Menstruating", "Post menstruation")
 )

 analysis_data <- merge(filtered_menstruation, df_filtered, by = c("RecordID", "UniqueID"))

 contingency_table <- table(analysis_data$CodeValue, analysis_data$DisplayDescription)

 test_result <- perform_statistical_test(contingency_table)

 return(display_analysis_results(variable_display_name, contingency_table, test_result))
}

if (!exists("fu_new_menstruationstatus_table")) {
 if (file.exists("fu_master_data_preparation_environment.RData")) {
  load("fu_master_data_preparation_environment.RData")
 } else {
  stop("Please run Script 103.0 first to generate the required follow-up data")
 }
}

print(table(fu_new_menstruationstatus_table$Description))

analysis_participants <- fu_new_menstruationstatus_table[
 fu_new_menstruationstatus_table$Description %in% c("Actively menstruating", "Post menstruation (no
active period for 1 year)"),
]
```

```
analysis_summary <- table(analysis_participants$Description)
names(analysis_summary) <- c("Menstruating", "Post menstruation")
print(analysis_summary)

if(!exists("fu_master_symptom_orders")) {
 stop("Master symptom ordering system not found. Please run Script 103.0 first.")
}

motor_symptoms <- fu_master_symptom_orders$motor_symptoms

fu_motor_results <- list()

for (symptom in motor_symptoms) {
 result <- analyze_variable("fu_motorsymptoms_table", symptom, symptom)
 if (!is.null(result)) {
  fu_motor_results[[symptom]] <- result
 }
}

nonmotor_symptoms <- fu_master_symptom_orders$nonmotor_symptoms

fu_nonmotor_results <- list()

for (symptom in nonmotor_symptoms) {
 result <- analyze_variable("fu_nonmotorsymptoms_table", symptom, symptom)
 if (!is.null(result)) {
  fu_nonmotor_results[[symptom]] <- result
 }
}

med_categories <- fu_master_symptom_orders$medication_categories

fu_med_results <- list()

for (category in med_categories) {
 result <- analyze_variable("fu_med_category_table", category, category)
 if (!is.null(result)) {
  fu_med_results[[category]] <- result
 }
}

fu_mental_health_results <- list()

if (exists("fu_mental_health_table")) {
 result <- analyze_coded_variable("fu_mental_health_table", "Mental Health")
 if (!is.null(result)) {
  fu_mental_health_results[["Mental Health"]] <- result
 }
```

```
}

if (exists("fu_mental_health_effect_table")) {
 result <- analyze_coded_variable("fu_mental_health_effect_table", "Mental Health Effect")
 if (!is.null(result)) {
  fu_mental_health_results[["Mental Health Effect"]] <- result
 }
}

support_types <- c("Family", "Friends", "Healthcare providers",
            "Support groups", "Mental health professional")

fu_support_results <- list()

for (support in support_types) {
 result <- analyze_variable("fu_supports_table", support, support)
 if (!is.null(result)) {
  fu_support_results[[support]] <- result
 }
}

save(fu_motor_results, fu_nonmotor_results, fu_med_results,
    fu_mental_health_results, fu_support_results,
    file = "followup_symptom_analysis.RData")

extract_significant_findings <- function(results_list, category_name) {
 findings_data <- data.frame(
  Category = character(),
  Variable = character(),
  P_Value = numeric(),
  Test = character(),
  Menstruating_Percent = numeric(),
  Post_menstruation_Percent = numeric(),
  stringsAsFactors = FALSE
 )

 for (var_name in names(results_list)) {
  result <- results_list[[var_name]]
  if (!is.na(result$p_value) && result$p_value < 0.05) {
   ct <- result$contingency_table

   if ("1" %in% rownames(ct) && ncol(ct) >= 2) {
    menstru_count <- ct["1", "Menstruating"]
    postmeno_count <- ct["1", "Post menstruation"]

    total_menstruating <- sum(ct[, "Menstruating"])
    total_post_menstruation <- sum(ct[, "Post menstruation"])

    menstru_percent <- (menstru_count / total_menstruating) * 100
```

```
      postmeno_percent <- (postmeno_count / total_post_menstruation) * 100

      findings_data <- rbind(findings_data, data.frame(
       Category = category_name,
       Variable = var_name,
       P_Value = result$p_value,
       Test = result$test,
       Menstruating_Percent = menstru_percent,
       Post_menstruation_Percent = postmeno_percent,
       stringsAsFactors = FALSE
      ))
     } else if (nrow(ct) > 1 && ncol(ct) >= 2) {
      findings_data <- rbind(findings_data, data.frame(
       Category = category_name,
       Variable = var_name,
       P_Value = result$p_value,
       Test = result$test,
       Menstruating_Percent = NA,
       Post_menstruation_Percent = NA,
       stringsAsFactors = FALSE
      ))
     }
    }
   }

  return(findings_data)
}

all_significant_findings <- rbind(
  extract_significant_findings(fu_motor_results, "Motor Symptoms"),
  extract_significant_findings(fu_nonmotor_results, "Non-Motor Symptoms"),
  extract_significant_findings(fu_med_results, "Medication Categories"),
  extract_significant_findings(fu_mental_health_results, "Mental Health"),
  extract_significant_findings(fu_support_results, "Support Mechanisms")
)

if (nrow(all_significant_findings) > 0) {
  all_significant_findings <- all_significant_findings[order(all_significant_findings$P_Value), ]

  display_findings <- all_significant_findings
  display_findings$P_Value <- sprintf("%.4f", display_findings$P_Value)

  for (i in 1:nrow(display_findings)) {
    if (!is.na(all_significant_findings$Menstruating_Percent[i])) {
      display_findings$Menstruating_Percent[i] <- sprintf("%.1f%%",
all_significant_findings$Menstruating_Percent[i])
      display_findings$Post_menstruation_Percent[i] <- sprintf("%.1f%%",
all_significant_findings$Post_menstruation_Percent[i])
    } else {
```

```
    display_findings$Menstruating_Percent[i] <- "See table"
    display_findings$Post_menstruation_Percent[i] <- "See table"
   }
  }

  colnames(display_findings) <- c(
   "Category", "Variable", "P-Value", "Statistical Test",
   "Menstruating %", "Post menstruation %"
  )

  enhanced_table <- kable(display_findings,
                  format = "html",
                  row.names = FALSE,
                  escape = FALSE,
                  caption = "Significant Associations between Menstruation Status and Follow-up
Variables") %>%
   kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
            full_width = FALSE, position = "center") %>%
   row_spec(0, background = "#f8f8f8", bold = TRUE) %>%
   column_spec(1:2, bold = TRUE) %>%
   collapse_rows(columns = 1, valign = "middle")

  print(enhanced_table)

  write.csv(display_findings, "followup_significant_findings.csv", row.names = FALSE)

  for (i in 1:nrow(all_significant_findings)) {
   finding <- all_significant_findings[i, ]
  }
}

save.image(file = "followup_symptom_analysis_complete.RData")
```

Script 103.2.

```
rm(list = ls(pattern = "^(app_.*_table|app_.*_ids|.*_participants_ids)$"))
rm(list = ls(pattern = "^(.*_life_events.*|.*_tracking.*|.*_usability.*)$"))
rm(list = ls(pattern = "^(response_rates|summary_.*|total_.*_participants)$"))
rm(list = ls(pattern = "^(temp_.*|extracted_.*|filtered_.*)$"))
library(dplyr)
library(readxl)
if (!file.exists("fu_master_data_preparation_environment.RData")) {
  stop("Please run Script FU_03.0 first to generate required follow-up data")
}

required_vars <- c("participants_completed_followup", "eligible_research_ids")
missing_vars <- c()

for(var in required_vars) {
  if(!exists(var)) {
    missing_vars <- c(missing_vars, var)
  }
}

if(length(missing_vars) > 0) {
  stop("Required variables missing: ", paste(missing_vars, collapse = ", "),
       "\nPlease run Follow-Up Numbers script first.")
}

load("fu_master_data_preparation_environment.RData")

if (!exists("fu_data_UID")) {
  stop("fu_data_UID not found in loaded environment")
}

if(exists("participants_completed_followup")) {
  matchable_participants <- sum(participants_completed_followup$standardized_id %in%
fu_data_UID$UniqueID)
}

expected_columns <- c("UniqueID", "s02_life_events", "s02_tracking",
                      paste0("s05_", 1:18, "c"))

missing_columns <- expected_columns[!expected_columns %in% colnames(fu_data_UID)]

target_columns_present <- expected_columns[expected_columns %in% colnames(fu_data_UID)]

usability_columns <- paste0("s05_", 1:18, "c")
usability_present <- sum(usability_columns %in% colnames(fu_data_UID))

app_invited_ids <- participants_completed_followup$standardized_id

print(app_invited_ids)
```

```
app_data_filtered <- fu_data_UID %>%
 filter(UniqueID %in% app_invited_ids)

found_ids <- app_data_filtered$UniqueID
missing_ids <- app_invited_ids[!app_invited_ids %in% found_ids]

print(sort(app_data_filtered$UniqueID))

app_life_events_table <- app_data_filtered %>%
 select(UniqueID, s02_life_events) %>%
 arrange(UniqueID)

colnames(app_life_events_table) <- c("UniqueID", "Life_Events_Response")

non_empty_responses <- sum(!is.na(app_life_events_table$Life_Events_Response) &
                  app_life_events_table$Life_Events_Response != "" &
                  trimws(app_life_events_table$Life_Events_Response) != "")

empty_responses <- nrow(app_life_events_table) - non_empty_responses

for(i in 1:nrow(app_life_events_table)) {
 response <- app_life_events_table$Life_Events_Response[i]
 if(!is.na(response) && trimws(response) != "") {
  preview <- ifelse(nchar(response) > 50,
             paste0(substr(response, 1, 50), "..."),
             response)
 }
}

app_tracking_table <- app_data_filtered %>%
 select(UniqueID, s02_tracking) %>%
 arrange(UniqueID)

colnames(app_tracking_table) <- c("UniqueID", "App_Tracking_Response")

non_empty_responses <- sum(!is.na(app_tracking_table$App_Tracking_Response) &
                  app_tracking_table$App_Tracking_Response != "" &
                  trimws(app_tracking_table$App_Tracking_Response) != "")

empty_responses <- nrow(app_tracking_table) - non_empty_responses

for(i in 1:nrow(app_tracking_table)) {
 response <- app_tracking_table$App_Tracking_Response[i]
 if(!is.na(response) && trimws(response) != "") {
  preview <- ifelse(nchar(response) > 60,
             paste0(substr(response, 1, 60), "..."),
             response)
 }
```

```
}

usability_columns <- paste0("s05_", 1:18, "c")

app_usability_table <- app_data_filtered %>%
 select(UniqueID, all_of(usability_columns)) %>%
 arrange(UniqueID)

usability_question_labels <- c(
 "App_Easy_To_Use",
 "Easy_To_Learn",
 "Navigation_Consistent",
 "Interface_All_Functions",
 "Easy_Mistake_Recovery",
 "Like_Interface",
 "Information_Well_Organized",
 "Progress_Acknowledgment",
 "Comfortable_Social_Settings",
 "Time_Fitting",
 "Would_Use_Again",
 "Overall_Satisfied",
 "Useful_Health_Wellbeing",
 "Improved_Healthcare_Access",
 "Helped_Manage_Health",
 "All_Expected_Functions",
 "Works_Poor_Internet",
 "Acceptable_Healthcare_Services"
)

colnames(app_usability_table) <- c("UniqueID", usability_question_labels)

complete_responses <- rowSums(!is.na(app_usability_table[, -1]) &
                      app_usability_table[, -1] != "" &
                      app_usability_table[, -1] != "0")

fully_completed <- sum(complete_responses == 18)
partially_completed <- sum(complete_responses > 0 & complete_responses < 18)
no_responses <- sum(complete_responses == 0)

for(i in 1:length(usability_question_labels)) {
 question_col <- usability_question_labels[i]
 responses <- as.numeric(app_usability_table[[question_col]])
 valid_responses <- responses[!is.na(responses) & responses > 0]

 if(length(valid_responses) > 0) {
  question_mean <- mean(valid_responses)
  question_sd <- sd(valid_responses)
  question_se <- question_sd / sqrt(length(valid_responses))
 }
```

```
}

for(i in 1:min(5, nrow(app_usability_table))) {
 for(j in 1:length(usability_question_labels)) {
  response <- app_usability_table[i, j+1]
  response_text <- ifelse(is.na(response) || response == "" || response == "0",
                  "[No response]", as.character(response))
 }
}

total_app_participants <- nrow(app_data_filtered)

life_events_responses <- sum(!is.na(app_life_events_table$Life_Events_Response) &
                    app_life_events_table$Life_Events_Response != "" &
                    trimws(app_life_events_table$Life_Events_Response) != "")

tracking_responses <- sum(!is.na(app_tracking_table$App_Tracking_Response) &
                  app_tracking_table$App_Tracking_Response != "" &
                  trimws(app_tracking_table$App_Tracking_Response) != "")

usability_complete_responses <- rowSums(!is.na(app_usability_table[, -1]) &
                           app_usability_table[, -1] != "" &
                           app_usability_table[, -1] != "0")

usability_fully_completed <- sum(usability_complete_responses == 18)
usability_any_responses <- sum(usability_complete_responses > 0)

summary_stats_table <- data.frame(
 Question_Category = c("Life Events", "App Tracking Experience", "Usability Survey (Full)", "Usability
Survey (Any)"),
 Total_Participants = c(total_app_participants, total_app_participants, total_app_participants,
total_app_participants),
 Participants_Responded = c(life_events_responses, tracking_responses, usability_fully_completed,
usability_any_responses),
 Response_Rate_Percent = c(
  round(life_events_responses/total_app_participants*100, 1),
  round(tracking_responses/total_app_participants*100, 1),
  round(usability_fully_completed/total_app_participants*100, 1),
  round(usability_any_responses/total_app_participants*100, 1)
 ),
 Non_Responses = c(
  total_app_participants - life_events_responses,
  total_app_participants - tracking_responses,
  total_app_participants - usability_fully_completed,
  total_app_participants - usability_any_responses
 )
)

print(summary_stats_table, row.names = FALSE)
```

```
life_responders <- app_life_events_table$UniqueID[!is.na(app_life_events_table$Life_Events_Response)
&
                                  app_life_events_table$Life_Events_Response != "" &
                                  trimws(app_life_events_table$Life_Events_Response) != ""]

tracking_responders <-
app_tracking_table$UniqueID[!is.na(app_tracking_table$App_Tracking_Response) &
                                   app_tracking_table$App_Tracking_Response != "" &
                                   trimws(app_tracking_table$App_Tracking_Response) != ""]

usability_responders <- app_usability_table$UniqueID[usability_complete_responses > 0]

all_three_responders <- intersect(intersect(life_responders, tracking_responders), usability_responders)
any_response <- unique(c(life_responders, tracking_responders, usability_responders))

print(app_life_events_table)

print(app_tracking_table)

print(app_usability_table)

print(summary_stats_table, row.names = FALSE)

excel_export_list <- list(
  "Life_Events" = app_life_events_table,
  "App_Tracking" = app_tracking_table,
  "Usability_Survey" = app_usability_table
)

excel_filename <- "App_Participants_Follow_Up_Responses.xlsx"

write_xlsx(excel_export_list, excel_filename)
```

Script 103.3.

```
library(ggplot2)
library(dplyr)
library(tidyr)
library(viridis)

prepare_usability_data <- function() {

 question_labels_short <- c(
  "App Easy To Use",
  "Easy To Learn",
  "Navigation Consistent",
  "Interface All Functions",
  "Easy Mistake Recovery",
  "Like Interface",
  "Information Well Organised",
  "Progress Acknowledgment",
  "Comfortable Social Settings",
  "Time Fitting",
  "Would Use Again",
  "Overall Satisfied",
  "Useful Health Wellbeing",
  "Improved Healthcare Access",
  "Helped Manage Health",
  "All Expected Functions",
  "Works Poor Internet",
  "Acceptable Healthcare Services"
 )

 usability_long <- app_usability_table %>%
  select(-UniqueID) %>%
  pivot_longer(cols = everything(),
           names_to = "Question",
           values_to = "Response") %>%
  mutate(Response = as.numeric(Response)) %>%
  filter(!is.na(Response) & Response > 0 & Response <= 7) %>%
  mutate(Question_Order = match(Question, colnames(app_usability_table)[-1]),
      Question_Label = question_labels_short[Question_Order],
      Question_Label = factor(Question_Label, levels = question_labels_short))

 return(usability_long)
}

calculate_summary_stats <- function(data) {
 summary_stats <- data %>%
  group_by(Question_Label) %>%
  summarise(
   n_responses = n(),
   mean_response = mean(Response),
   median_response = median(Response),
   sd_response = sd(Response),
   se_response = sd(Response) / sqrt(n()),
   mode_response = {
    freq_table <- table(Response)
    as.numeric(names(freq_table)[which.max(freq_table)])
   },
```

```
    .groups = 'drop'
  )

 return(summary_stats)
}

create_boxplot_version <- function(data, summary_stats) {

 p_box <- ggplot(data, aes(x = Question_Label, y = Response)) +

  geom_boxplot(alpha = 0.7, fill = "lightblue", outlier.shape = NA) +

  geom_jitter(width = 0.2, alpha = 0.6, size = 1, colour = "darkgrey") +

  stat_summary(fun = mean, geom = "point",
          shape = 18, size = 3, colour = "red") +

  geom_point(data = summary_stats,
         aes(x = Question_Label, y = mode_response),
         shape = 17, size = 3, colour = "darkgreen") +

  scale_y_continuous(breaks = 1:7,
             labels = c("1\nStrongly\nDisagree", "2", "3", "4", "5", "6",
                   "7\nStrongly\nAgree"),
             limits = c(0.5, 7.5)) +

  labs(title = "App Usability Survey Results - Box Plot Version",
     subtitle = "Red diamonds = Mean, Green triangles = Most common response, Grey dots = Individual
responses",
     x = "Survey Questions",
     y = "Response (1-7 Scale)",
     caption = paste("Based on responses from", length(unique(data$Question_Label)),
              "questions. Response rates vary by question.")) +

  theme_minimal() +
  theme(
   axis.text.x = element_text(angle = 45, hjust = 1, size = 10),
   plot.title = element_text(size = 14, face = "bold"),
   plot.subtitle = element_text(size = 10),
   panel.grid.minor = element_blank()
  )

 return(p_box)
}

create_violin_version <- function(data, summary_stats) {

 p_violin <- ggplot(data, aes(x = Question_Label, y = Response)) +

  geom_violin(alpha = 0.7, fill = "lightblue", scale = "width") +

  geom_boxplot(width = 0.1, alpha = 0.7, outlier.shape = NA) +

  geom_jitter(width = 0.1, alpha = 0.5, size = 0.8, colour = "darkgrey") +

  stat_summary(fun = mean, geom = "point",
          shape = 18, size = 3, colour = "red") +
```

```
  geom_point(data = summary_stats,
             aes(x = Question_Label, y = mode_response),
             shape = 17, size = 3, colour = "darkgreen") +

  scale_y_continuous(breaks = 1:7,
                     labels = c("1\nStrongly\nDisagree", "2", "3", "4", "5", "6",
                                "7\nStrongly\nAgree"),
                     limits = c(0.5, 7.5)) +

  labs(title = "App Usability Survey Results - Violin Plot Version",
       subtitle = "Red diamonds = Mean, Green triangles = Most common response, Grey dots = Individual
responses",
       x = "Survey Questions",
       y = "Response (1-7 Scale)",
       caption = paste("Based on responses from", length(unique(data$Question_Label)),
                       "questions. Violin width shows response distribution density.")) +

  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, size = 10),
    plot.title = element_text(size = 14, face = "bold"),
    plot.subtitle = element_text(size = 10),
    panel.grid.minor = element_blank()
  )

 return(p_violin)
}

usability_data <- prepare_usability_data()
summary_stats <- calculate_summary_stats(usability_data)

print(summary_stats, n = 18)

boxplot_chart <- create_boxplot_version(usability_data, summary_stats)

print(boxplot_chart)

overall_mean <- mean(usability_data$Response)
positive_questions <- sum(summary_stats$mean_response >= 5)
very_positive_questions <- sum(summary_stats$mean_response >= 6)

best_question <- summary_stats$Question_Label[which.max(summary_stats$mean_response)]
worst_question <- summary_stats$Question_Label[which.min(summary_stats$mean_response)]

create_thematic_data <- function() {

 thematic_groups <- list(
  "Core Usability" = c("App_Easy_To_Use", "Easy_To_Learn", "Navigation_Consistent",
                       "Interface_All_Functions", "Easy_Mistake_Recovery"),
  "User Experience & Satisfaction" = c("Like_Interface", "Information_Well_Organized",
                                       "Progress_Acknowledgment", "Overall_Satisfied"),
  "Practical Integration" = c("Comfortable_Social_Settings", "Time_Fitting", "Works_Poor_Internet"),
  "Future Intent" = c("Would_Use_Again"),
  "Health & Clinical Value" = c("Useful_Health_Wellbeing", "Improved_Healthcare_Access",
                                "Helped_Manage_Health", "All_Expected_Functions",
                                "Acceptable_Healthcare_Services")
```

```
  )

  participant_thematic_averages <- app_usability_table %>%
    pivot_longer(cols = -UniqueID, names_to = "Question", values_to = "Response") %>%
    mutate(Response = as.numeric(Response)) %>%
    filter(!is.na(Response) & Response > 0 & Response <= 7) %>%
    mutate(Theme = case_when(
      Question %in% thematic_groups$`Core Usability` ~ "Core Usability",
      Question %in% thematic_groups$`User Experience & Satisfaction` ~ "User Experience &
Satisfaction",
      Question %in% thematic_groups$`Practical Integration` ~ "Practical Integration",
      Question %in% thematic_groups$`Future Intent` ~ "Future Intent",
      Question %in% thematic_groups$`Health & Clinical Value` ~ "Health & Clinical Value",
      TRUE ~ "Other"
    )) %>%
    filter(Theme != "Other") %>%
    group_by(UniqueID, Theme) %>%
    summarise(Theme_Average = mean(Response, na.rm = TRUE), .groups = 'drop') %>%
    mutate(Theme = factor(Theme, levels = names(thematic_groups)))

  return(participant_thematic_averages)
}

calculate_thematic_summary <- function(thematic_data) {
  thematic_summary <- thematic_data %>%
    group_by(Theme) %>%
    summarise(
      n_participants = n(),
      mean_theme = mean(Theme_Average),
      median_theme = median(Theme_Average),
      sd_theme = sd(Theme_Average),
      se_theme = sd(Theme_Average) / sqrt(n()),
      mode_theme = {
        rounded_avg <- round(Theme_Average * 2) / 2
        freq_table <- table(rounded_avg)
        as.numeric(names(freq_table)[which.max(freq_table)])
      },
      .groups = 'drop'
    )

  return(thematic_summary)
}

create_thematic_chart <- function(thematic_data, thematic_summary) {

  theme_colours <- c(
    "Core Usability" = "#1f77b4",
    "User Experience & Satisfaction" = "#ff7f0e",
    "Practical Integration" = "#2ca02c",
    "Future Intent" = "#d62728",
    "Health & Clinical Value" = "#9467bd"
  )

  p_thematic <- ggplot(thematic_data, aes(x = Theme, y = Theme_Average, fill = Theme)) +

    geom_boxplot(alpha = 0.7, outlier.shape = NA) +
```

```
  geom_jitter(width = 0.2, alpha = 0.6, size = 1.5, colour = "darkgrey") +

  stat_summary(fun = mean, geom = "point",
           shape = 18, size = 4, colour = "red") +

  geom_point(data = thematic_summary,
          aes(x = Theme, y = mode_theme),
          shape = 17, size = 4, colour = "darkgreen", inherit.aes = FALSE) +

  scale_fill_manual(values = theme_colours) +

  scale_y_continuous(breaks = 1:7,
                labels = c("1\nStrongly\nDisagree", "2", "3", "4", "5", "6",
                      "7\nStrongly\nAgree"),
                limits = c(0.5, 7.5)) +

  labs(title = "App Usability Survey - Thematic Summary",
      subtitle = "Red diamonds = Mean, Green triangles = Most common response, Grey dots = Participant
averages",
      x = "Thematic Categories",
      y = "Average Response (1-7 Scale)",
      caption = "Each participant's average score within each thematic category") +

  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, size = 11),
    plot.title = element_text(size = 14, face = "bold"),
    plot.subtitle = element_text(size = 10),
    legend.position = "none",
    panel.grid.minor = element_blank()
  )

 return(p_thematic)
}

thematic_data <- create_thematic_data()
thematic_summary <- calculate_thematic_summary(thematic_data)

print(thematic_summary, n = 5)

thematic_chart <- create_thematic_chart(thematic_data, thematic_summary)

print(thematic_chart)

theme_ranking <- thematic_summary %>%
 arrange(desc(mean_theme)) %>%
 mutate(Rank = row_number())

for(i in 1:nrow(theme_ranking)) {
}

best_theme <- theme_ranking$Theme[1]
worst_theme <- theme_ranking$Theme[nrow(theme_ranking)]
overall_thematic_mean <- mean(thematic_data$Theme_Average)

excellent_themes <- sum(thematic_summary$mean_theme >= 6)
good_themes <- sum(thematic_summary$mean_theme >= 5 & thematic_summary$mean_theme < 6)
```

```
moderate_themes <- sum(thematic_summary$mean_theme >= 4 & thematic_summary$mean_theme <
5)
poor_themes <- sum(thematic_summary$mean_theme < 4)
```