

Project 0: Analysis of Cortisol and DHEA Diurnal Patterns Using SPIT Booklet

Hong Yunjing

2026-02-02

Contents

Introduction	1
Background	1
Research Questions	1
Statistical Hypotheses	1
Methods	2
Critical Methodological Note	2
Data Management and Cleaning	2
Statistical Analysis Methods	10
Results	12
Descriptive Statistics	12
Question 1: Agreement Between Booklet and MEMs Times	16
Question 2: Adherence to Protocol Timing	21
Question 3: Hormone Patterns Over Time	27
Discussion	38
Summary of Findings	38
Limitations	38
Conclusions	39
Reproducible Research Information	39
Data and Code Location	39
Report Tables and Figures Summary	40
Session Information	41

Introduction

Background

This analysis examines data from a study testing a novel saliva collection device (SPIT booklet) for measuring diurnal patterns of cortisol and DHEA hormones. The study involved 31 healthy control subjects who collected saliva samples four times daily for three days. Samples were collected at waking, 30 minutes post-waking, before lunch, and 10 hours after waking. The SPIT booklet was stored in a bottle with an electronic monitoring cap to track actual sampling times.

Research Questions

The investigator posed three main research questions:

1. **Agreement Analysis:** What is the agreement between subject-recorded sampling times (Booklet) and electronically recorded times (MEMs cap)?
2. **Adherence Analysis:** Are subjects accurately adhering to the +30 minute and +10 hour sampling time protocols?
3. **Hormone Pattern Analysis:** What are the changes in cortisol and DHEA levels over time during the day?

Statistical Hypotheses

Question 1: - H0: There is no association between booklet-recorded time and MEMs-recorded time since waking - HA: There is a significant association between booklet-recorded time and MEMs-recorded time since waking

Question 2: - We will calculate adherence rates as the percentage of samples within 7.5 and 15 minutes of scheduled times

Question 3: - H0: There is no change in cortisol/DHEA from waking to 30 minutes post-waking - HA: There is a significant increase in cortisol/DHEA from waking to 30 minutes post-waking - Additionally, we will estimate the rate of decline after 30 minutes from waking

Methods

Critical Methodological Note

Following the investigator's explicit written instructions (Project 0 Q&A Notes, January 26), this analysis calculates time since waking using ONLY sleep diary reported wake times. The investigator stated:

“The investigator said she is interested in measuring booklet and cap time since waking, where wake time is determined from the sleep diary.”

And:

“The investigator stated that the sample.interval variables... [she] was not sure these were calculated correctly and that she was not interested in using these values in analysis.”

While this ensures methodological correctness and aligns with the investigator's specifications, it substantially limits sample size. Sleep diary wake times were recorded for only 93 of 372 observations (25%), reducing statistical power for all three research questions. This limitation is discussed further in the Limitations section.

Data Management and Cleaning

Data Import and Initial Processing

```
# Read the data from DataRaw folder
data_raw <- read_csv("../DataRaw/Project0_Clean_v2.csv")

# Display dimensions
cat("Original dataset dimensions:", dim(data_raw)[1], "rows,", dim(data_raw)[2], "columns\n")

## Original dataset dimensions: 372 rows, 15 columns
```

Variable Creation and Time Calculations

Following the investigator's guidance, we calculated time since waking from the sleep diary wake time using POSIXct datetime objects and difftime calculations.

```
# First, let's check what columns we actually have
cat("== COLUMN INSPECTION ==\n\n")

## == COLUMN INSPECTION ==

cat("Column names in CSV:\n")
## Column names in CSV:

print(names(data_raw))

## [1] "SubjectID"
## [2] "Collection Date"
## [3] "Collection Sample"
## [4] "Booket: Clock Time"
## [5] "MEMs: Clock Time"
## [6] "Sleep Diary reported wake time"
## [7] "Booklet: Sample interval"
## [8] "Booklet: Sample interval Decimal Time (mins)"
## [9] "MEMs: Sample interval"
## [10] "MEMs: Sample interval Decimal Time (mins)"
## [11] "Cortisol (ug/dl)"
## [12] "DHEA (pg/dl)"
## [13] "Cortisol (nmol/L)"
## [14] "DHEA (nmol/L)"
## [15] "DAYNUMB"
```

```

# CRITICAL FIX: The time columns are already parsed as 'hms' (time) objects by read_csv
# We need to convert them to character strings in HH:MM format

# Helper function to convert hms to character HH:MM
hms_to_hhmm <- function(x) {
  if (is.null(x) || all(is.na(x))) return(rep(NA_character_, length(x)))

  # If it's already a time object (hms/difftime), format it
  if (inherits(x, "hms") || inherits(x, "difftime")) {
    hours <- as.integer(as.numeric(x) / 3600)
    mins <- as.integer((as.numeric(x) %% 3600) / 60)
    result <- sprintf("%d:%02d", hours, mins)
    result[is.na(x)] <- NA_character_
    return(result)
  }

  # If it's character, return as-is
  return(as.character(x))
}

# Rename columns handling the typo in "Booket"
data <- data_raw %>%
  rename(
    SubjectID = `SubjectID`,
    CollectionDate = `Collection Date`,
    CollectionSample = `Collection Sample`,
    Booklet_ClockTime_raw = `Booket: Clock Time`, # Note the typo!
    MEMs_ClockTime_raw = `MEMs: Clock Time`,
    SleepDiary_WakeTime_raw = `Sleep Diary reported wake time`,
    Cortisol_nmol = `Cortisol (nmol/L)`,
    DHEA_nmol = `DHEA (nmol/L)`,
    DayNumber = `DAYNUMB`
  ) %>%
  mutate(
    SubjectID = as.factor(SubjectID),
    CollectionSample = as.integer(CollectionSample),
    DayNumber = as.integer(DayNumber),
    CollectionDate = mdy(CollectionDate),
    Cortisol_nmol = as.numeric(Cortisol_nmol),
    DHEA_nmol = as.numeric(DHEA_nmol)
  )

cat("\n==== CONVERTING HMS TIMES TO STRINGS ====\n")

## 
## === CONVERTING HMS TIMES TO STRINGS ===

# Convert hms times to character strings
data <- data %>%
  mutate(
    wake_hhmm = hms_to_hhmm(SleepDiary_WakeTime_raw),
    book_hhmm = hms_to_hhmm(Booklet_ClockTime_raw),
    mems_hhmm = hms_to_hhmm(MEMs_ClockTime_raw)

```

```

    )

cat("Sample converted times:\n")

## Sample converted times:

cat("Wake:", head(data$wake_hhmm[!is.na(data$wake_hhmm)], 5), "\n")

## Wake: 8:54 7:20 6:38 4:45 8:00

cat("Booklet:", head(data$book_hhmm[!is.na(data$book_hhmm)], 5), "\n")

## Booklet: 8:54 9:38 12:31 19:38 7:20

cat("MEMs:", head(data$mems_hhmm[!is.na(data$mems_hhmm)], 5), "\n")

## MEMs: 8:55 9:38 12:30 19:38 7:21

cat("\nNon-NA counts:\n")

## Non-NA counts:

cat("Wake times:", sum(!is.na(data$wake_hhmm)), "\n")

## Wake times: 93

cat("Booklet times:", sum(!is.na(data$book_hhmm)), "\n")

## Booklet times: 337

cat("MEMs times:", sum(!is.na(data$mems_hhmm)), "\n")

## MEMs times: 311

# Create datetime objects
data <- data %>%
  mutate(
    wake_dt = if_else(!is.na(CollectionDate) & !is.na(wake_hhmm),
                      ymd_hm(paste(CollectionDate, wake_hhmm), tz = "UTC", quiet = TRUE),
                      as.POSIXct(NA)),

    book_dt = if_else(!is.na(CollectionDate) & !is.na(book_hhmm),
                      ymd_hm(paste(CollectionDate, book_hhmm), tz = "UTC", quiet = TRUE),
                      as.POSIXct(NA)),

    mems_dt = if_else(!is.na(CollectionDate) & !is.na(mems_hhmm),
                      ymd_hm(paste(CollectionDate, mems_hhmm), tz = "UTC", quiet = TRUE),
                      as.POSIXct(NA))
  )

cat("\n==== DATETIME PARSING RESULTS ===\n")

```

```

##  

## === DATETIME PARSING RESULTS ===  

cat("Non-NA wake_dt:", sum(!is.na(data$wake_dt)), "\n")  

## Non-NA wake_dt: 93  

cat("Non-NA book_dt:", sum(!is.na(data$book_dt)), "\n")  

## Non-NA book_dt: 337  

cat("Non-NA mems_dt:", sum(!is.na(data$mems_dt)), "\n")  

## Non-NA mems_dt: 311  

# Cross-midnight adjustment and calculate minutes since waking  

data <- data %>%
  mutate(
    # Cross-midnight adjustment: if recorded time is before wake, assume next day
    book_dt = if_else(!is.na(book_dt) & !is.na(wake_dt) & book_dt < wake_dt,
                     book_dt + days(1), book_dt),
    mems_dt = if_else(!is.na(mems_dt) & !is.na(wake_dt) & mems_dt < wake_dt,
                     mems_dt + days(1), mems_dt),
    # Calculate minutes since waking
    Booklet_MinsSinceWake = as.numeric(difftime(book_dt, wake_dt, units = "mins")),
    MEMs_MinsSinceWake = as.numeric(difftime(mems_dt, wake_dt, units = "mins"))
  ) %>%
  mutate(
    # Guardrail: mark implausible values as missing
    Booklet_MinsSinceWake = if_else(!is.na(Booklet_MinsSinceWake) &
                                     (Booklet_MinsSinceWake < -5 | Booklet_MinsSinceWake > 24*60),
                                     NA_real_, Booklet_MinsSinceWake),
    MEMs_MinsSinceWake = if_else(!is.na(MEMs_MinsSinceWake) &
                                 (MEMs_MinsSinceWake < -5 | MEMs_MinsSinceWake > 24*60),
                                 NA_real_, MEMs_MinsSinceWake)
  )
  

cat("\n== FINAL MINUTES SINCE WAKE ==\n")
  

##  

## == FINAL MINUTES SINCE WAKE ==  

cat("Non-NA Booklet_MinsSinceWake:", sum(!is.na(data$Booklet_MinsSinceWake)), "\n")  

## Non-NA Booklet_MinsSinceWake: 88  

cat("Non-NA MEMs_MinsSinceWake:", sum(!is.na(data$MEMs_MinsSinceWake)), "\n")  

## Non-NA MEMs_MinsSinceWake: 77

```

```

cat("Both non-NA:", sum(!is.na(data$Booklet_MinsSinceWake) & !is.na(data$MEMs_MinsSinceWake)), "\n")

## Both non-NA: 72

cat("\nSample minutes since wake (first 10 valid):\n")

## 
## Sample minutes since wake (first 10 valid):

valid_both <- data %>% filter(!is.na(Booklet_MinsSinceWake) & !is.na(MEMs_MinsSinceWake))
if (nrow(valid_both) > 0) {
  print(head(valid_both %>% select(SubjectID, CollectionSample,
                                      Booklet_MinsSinceWake, MEMs_MinsSinceWake), 10))
}

## # A tibble: 10 x 4
##   SubjectID CollectionSample Booklet_MinsSinceWake MEMs_MinsSinceWake
##   <fct>          <int>                <dbl>              <dbl>
## 1 3012            1                  0                  1
## 2 3012            1                  0                  1
## 3 3012            1                  0                  1
## 4 3013            1                  5                  2
## 5 3013            1                  0                  1
## 6 3013            1                  0                  2
## 7 3015            1                  0                  6
## 8 3015            1                  0                  3
## 9 3018            1                  0                 1438
## 10 3018           1                  0                  3

# Expected sampling times in minutes since waking
data <- data %>%
  mutate(
    Expected_MinsSinceWake = case_when(
      CollectionSample == 1 ~ 0,
      CollectionSample == 2 ~ 30,
      CollectionSample == 3 ~ NA_real_, # lunch time varies
      CollectionSample == 4 ~ 600
    )
  )

# CRITICAL FIX: Wake times are only recorded for Sample 1
# We need to carry them forward to all samples on the same day
# Group by Subject and Date, then fill forward wake time
data <- data %>%
  group_by(SubjectID, CollectionDate) %>%
  arrange(SubjectID, CollectionDate, CollectionSample) %>%
  mutate(
    # Fill forward wake_dt within each subject-day
    wake_dt_filled = if_else(CollectionSample == 1, wake_dt, as.POSIXct(NA))
  ) %>%
  fill(wake_dt_filled, .direction = "down") %>%

```

```

ungroup()

# Now recalculate minutes since waking using the filled wake times
data <- data %>%
  mutate(
    # Recalculate with filled wake times
    Booklet_MinsSinceWake_filled = if_else(
      !is.na(book_dt) & !is.na(wake_dt_filled),
      as.numeric(difftime(book_dt, wake_dt_filled, units = "mins")),
      NA_real_
    ),
    MEMs_MinsSinceWake_filled = if_else(
      !is.na(mems_dt) & !is.na(wake_dt_filled),
      as.numeric(difftime(mems_dt, wake_dt_filled, units = "mins")),
      NA_real_
    ),

    # Cross-midnight adjustment
    Booklet_MinsSinceWake_filled = if_else(
      !is.na(Booklet_MinsSinceWake_filled) & Booklet_MinsSinceWake_filled < -5,
      Booklet_MinsSinceWake_filled + 24*60,
      Booklet_MinsSinceWake_filled
    ),
    MEMs_MinsSinceWake_filled = if_else(
      !is.na(MEMs_MinsSinceWake_filled) & MEMs_MinsSinceWake_filled < -5,
      MEMs_MinsSinceWake_filled + 24*60,
      MEMs_MinsSinceWake_filled
    ),

    # Guardrail: mark implausible values as missing
    Booklet_MinsSinceWake_filled = if_else(
      !is.na(Booklet_MinsSinceWake_filled) &
        (Booklet_MinsSinceWake_filled < -5 | Booklet_MinsSinceWake_filled > 24*60),
      NA_real_, Booklet_MinsSinceWake_filled
    ),
    MEMs_MinsSinceWake_filled = if_else(
      !is.na(MEMs_MinsSinceWake_filled) &
        (MEMs_MinsSinceWake_filled < -5 | MEMs_MinsSinceWake_filled > 24*60),
      NA_real_, MEMs_MinsSinceWake_filled
    )
  )

cat("\n==== AFTER FILLING FORWARD WAKE TIMES ===\n")

##
## === AFTER FILLING FORWARD WAKE TIMES ===

cat("Original Booklet_MinsSinceWake (Sample 1 only):", sum(!is.na(data$Booklet_MinsSinceWake)), "\n")

## Original Booklet_MinsSinceWake (Sample 1 only): 88

```

```

cat("Filled Booklet_MinsSinceWake (all samples):", sum(!is.na(data$Booklet_MinsSinceWake_filled)), "\n")

## Filled Booklet_MinsSinceWake (all samples): 337

cat("Original MEMs_MinsSinceWake (Sample 1 only):", sum(!is.na(data$MEMs_MinsSinceWake)), "\n")

## Original MEMs_MinsSinceWake (Sample 1 only): 77

cat("Filled MEMs_MinsSinceWake (all samples):", sum(!is.na(data$MEMs_MinsSinceWake_filled)), "\n")

## Filled MEMs_MinsSinceWake (all samples): 311

cat("\nBreakdown by sample:\n")

## 
## Breakdown by sample:

data %>%
  group_by(CollectionSample) %>%
  summarise(
    n_total = n(),
    n_booklet = sum(!is.na(Booklet_MinsSinceWake_filled)),
    n_mem = sum(!is.na(MEMs_MinsSinceWake_filled))
  ) %>%
  print()

## # A tibble: 4 x 4
##   CollectionSample n_total n_booklet n_mem
##       <int>     <int>      <int>  <int>
## 1             1         93        88     77
## 2             2         93        87     70
## 3             3         93        82     81
## 4             4         93        80     83

glimpse(data)

## #> #> #> Rows: 372
## #> #> Columns: 27
## #> #> $ SubjectID <fct> 3012, 3012, 3012, 3012, ~
## #> #> $ CollectionDate <date> 2018-10-02, 2018-10-02~
## #> #> $ CollectionSample <int> 1, 2, 3, 4, 1, 2, 3, 4, ~
## #> #> $ Booklet_ClockTime_raw <time> 08:54:00, 09:38:00, 12~
## #> #> $ MEMs_ClockTime_raw <time> 08:55:00, 09:38:00, 12~
## #> #> $ SleepDiary_WakeTime_raw <time> 08:54:00, NA, ~
## #> #> $ 'Booklet: Sample interval' <time> NA, 00:44:00, 03~
## #> #> $ 'Booklet: Sample interval Decimal Time (mins)' <dbl> 0, 44, 217, 644, 0, 37, ~
## #> #> $ 'MEMs: Sample interval' <time> NA, 00:43:00, 03~
## #> #> $ 'MEMs: Sample interval Decimal Time (mins)' <dbl> NA, 43, 215, 643, NA, 3~
## #> #> $ 'Cortisol (ug/dl)' <dbl> 0.142, 0.259, 0.084, 0.~
```

```

## $ 'DHEA (pg/dl)'
## $ Cortisol_nmol
## $ DHEA_nmol
## $ DayNumber
## $ wake_hhmm
## $ book_hhmm
## $ mems_hhmm
## $ wake_dt
## $ book_dt
## $ mems_dt
## $ Booklet_MinsSinceWake
## $ MEMs_MinsSinceWake
## $ Expected_MinsSinceWake
## $ wake_dt_filled
## $ Booklet_MinsSinceWake_filled
## $ MEMs_MinsSinceWake_filled

# Save processed dataset
write_csv(data, "../DataProcessed/Project0_Processed.csv")
cat("\n==== PROCESSED DATA SAVED ===\n")

## 
## === PROCESSED DATA SAVED ===

cat("Location: BIOS6624/DataProcessed/Project0_Processed.csv\n")

## Location: BIOS6624/DataProcessed/Project0_Processed.csv

```

Missing Data Assessment

```

# Calculate missing data rates
missing_summary <- data %>%
  summarise(
    n_total = n(),
    n_missing_wake = sum(is.na(wake_dt)),
    n_missing_booklet_time = sum(is.na(Booklet_MinsSinceWake)),
    n_missing_memes_time = sum(is.na(MEMs_MinsSinceWake)),
    n_missing_cortisol = sum(is.na(Cortisol_nmol)),
    n_missing_dhea = sum(is.na(DHEA_nmol)),
    pct_missing_wake = round(100 * n_missing_wake / n_total, 2),
    pct_missing_booklet = round(100 * n_missing_booklet_time / n_total, 2),
    pct_missing_memes = round(100 * n_missing_memes_time / n_total, 2),
    pct_missing_cortisol = round(100 * n_missing_cortisol / n_total, 2),
    pct_missing_dhea = round(100 * n_missing_dhea / n_total, 2)
  )

# Create missing data table
missing_table <- data.frame(
  Variable = c("Wake Time", "Booklet Time", "MEMs Time", "Cortisol (nmol/L)", "DHEA (nmol/L)"),
  N_Missing = c(missing_summary$n_missing_wake,
                missing_summary$n_missing_booklet_time,

```

```

    missing_summary$n_missing_mems_time,
    missing_summary$n_missing_cortisol,
    missing_summary$n_missing_dhea),
Percent_Missing = c(missing_summary$pct_missing_wake,
                    missing_summary$pct_missing_booklet,
                    missing_summary$pct_missing_mems,
                    missing_summary$pct_missing_cortisol,
                    missing_summary$pct_missing_dhea)
)

write_csv(missing_summary, "../Outputs/missing_summary.csv")

```

Statistical Analysis Methods

Question 1: Agreement Analysis

To assess agreement between booklet and MEMs times:

- Correlation Analysis:** Calculate Pearson correlation coefficient between booklet-recorded and MEMs-recorded time since waking
- Mixed Model:** Fit mixed effects model with MEMs time as outcome and booklet time as predictor, with random intercepts by subject
- Bias Assessment:** Calculate the mean difference (booklet - MEMs) to assess systematic bias using Bland-Altman approach
- Visualization:** Create scatter plot with regression line and Bland-Altman plot

Statistical significance tested at alpha = 0.05 level.

Question 2: Adherence Analysis

Critical Methodological Note: Following the investigator's explicit instructions (Project 0 Q&A Notes), we calculate time since waking using ONLY the sleep diary reported wake time. The investigator stated she was "not interested in using" the pre-calculated sample interval values and wanted "time since waking, where wake time is determined from the sleep diary." This approach ensures methodological correctness but substantially limits sample size, as wake times were recorded for only 25% of observations.

Adherence assessed for samples 2 and 4 (scheduled times):

- Time Calculation:** Calculate minutes since waking using sleep diary wake time for both booklet and MEMs clock times
- Deviation Calculation:** Calculate absolute difference between actual and expected times (30 min and 600 min)
- Adherence Rates:**
 - Percentage within 7.5 minutes of scheduled time
 - Percentage within 15 minutes of scheduled time
- By Method:** Calculate adherence separately for booklet and MEMs recordings where available

Question 3: Hormone Pattern Analysis

Critical Methodological Note: Consistent with Q2, we calculate time since waking using ONLY the sleep diary reported wake time (per investigator's explicit instructions). We use MEMs-recorded clock time rather than booklet time, as it provides more objective measurement. The investigator stated either method would be acceptable for statistical reasons.

To analyze cortisol and DHEA patterns, we use a piecewise linear mixed effects model:

1. Data Cleaning:

- Exclude cortisol values > 80 nmol/L (likely lab errors)
- **CRITICAL FIX:** Exclude DHEA values exactly equal to 5.205 nmol/L (detection limit), NOT all values ≥ 5.205
- Identify and exclude subjects with ≥ 2 DHEA values at detection limit

2. Model Specification: Piecewise linear mixed model with knot at 30 minutes

- Outcome: $\log(\text{Cortisol nmol/L})$ or $\log(\text{DHEA nmol/L})$
- Fixed effects:
 - $\text{time_0_30} = \min(t, 30)$ - slope from waking to 30 min
 - $\text{time_post30} = \max(t - 30, 0)$ - slope after 30 min
- Random effects: Subject-specific intercepts only (random slopes not included due to limited time-points)

3. Log transformation: Used to stabilize variance and allow multiplicative interpretation of effects

4. Hypothesis Tests:

- Initial increase: Test if slope of time_0_30 is significantly positive
- Rate of decline: Test if slope of time_post30 is significantly negative

5. Statistical Test: t-tests for fixed effects coefficients

6. Significance Level: alpha = 0.05

Results

Descriptive Statistics

Sample Characteristics

```
# Subject-level summary
subject_summary <- data %>%
  group_by(SubjectID) %>%
  summarise(
    n_samples = n(),
    n_days = n_distinct(DayNumber)
  )

cat("Number of subjects:", n_distinct(data$SubjectID), "\n")

## Number of subjects: 31
```

```

cat("Total observations:", nrow(data), "\n")

## Total observations: 372

cat("Mean samples per subject:", round(mean(subject_summary$n_samples), 1), "\n")

## Mean samples per subject: 12

# Hormone descriptive statistics by sample time
hormone_stats <- data %>%
  group_by(CollectionSample) %>%
  summarise(
    n = n(),
    Cortisol_Mean = round(mean(Cortisol_nmol, na.rm = TRUE), 2),
    Cortisol_SD = round(sd(Cortisol_nmol, na.rm = TRUE), 2),
    Cortisol_Median = round(median(Cortisol_nmol, na.rm = TRUE), 2),
    DHEA_Mean = round(mean(DHEA_nmol, na.rm = TRUE), 2),
    DHEA_SD = round(sd(DHEA_nmol, na.rm = TRUE), 2),
    DHEA_Median = round(median(DHEA_nmol, na.rm = TRUE), 2)
  )

```

TABLE 1: Study Characteristics and Descriptive Statistics (FOR REPORT)

```

# Prepare data for Table 1 - create a subject-level summary dataset
# Since we have repeated measures, summarize to one row per subject for baseline characteristics
table1_subject_data <- data %>%
  group_by(SubjectID) %>%
  summarise(
    n_samples = n(),
    n_days = n_distinct(DayNumber),
    # Use first observation values for subject-level characteristics
    has_wake_time = any(!is.na(wake_dt)),
    has_booklet_time = any(!is.na(book_dt)),
    has_mems_time = any(!is.na(mems_dt)),
    .groups = "drop"
  )

# Create Table 1 using gtsummary with observation-level data
# This is more appropriate for showing variable distributions
table1_data_for_summary <- data %>%
  select(
    Cortisol_nmol,
    DHEA_nmol,
    CollectionSample,
    DayNumber
  ) %>%
  mutate(
    CollectionSample = factor(CollectionSample,
      levels = 1:4,
      labels = c("Waking", "+30 min", "Before lunch", "+10 hours"))
  )

```

```

    DayNumber = factor(DayNumber)
  )

# Create comprehensive Table 1 using gtsummary
table1 <- table1_data_for_summary %>%
 tbl_summary(
  label = list(
    Cortisol_nmol ~ "Cortisol (nmol/L)",
    DHEA_nmol ~ "DHEA (nmol/L)",
    CollectionSample ~ "Collection Sample",
    DayNumber ~ "Day Number"
  ),
  statistic = list(
    all_continuous() ~ "{mean} ({sd})",
    all_categorical() ~ "{n} ({p}%)"
  ),
  missing = "always",
  missing_text = "Missing"
) %>%
add_n() %>%
modify_header(label ~ "***Characteristic**") %>%
modify_caption("**TABLE 1: Study Characteristics and Descriptive Statistics**") %>%
bold_labels()

# Display the table
table1

```

Characteristic	N	N = 372 ¹
Cortisol (nmol/L)	367	6.0 (6.9)
Missing		5
DHEA (nmol/L)	367	0.98 (1.03)
Missing		5
Collection Sample	372	
Waking		93 (25%)
+30 min		93 (25%)
Before lunch		93 (25%)
+10 hours		93 (25%)
Missing		0
Day Number	372	
1		120 (32%)
2		120 (32%)
3		132 (35%)
Missing		0

¹Mean (SD); n (%)

```

# Save as different formats for the report
# Save as GT object that can be converted to various formats
table1 %>%

```

```

as_gt() %>%
gt::gtsave(filename = "../Reports/ReportTables/Table1_Descriptives.html")

# Also save as CSV for easy viewing
table1_df <- table1 %>%
  as_tibble()
write_csv(table1_df, "../Reports/ReportTables/Table1_Descriptives.csv")

# Save as LaTeX table
table1_latex <- table1 %>%
  as_kable_extra(format = "latex", booktabs = TRUE) %>%
  as.character()
writeLines(table1_latex, "../Reports/ReportTables/Table1_Descriptives.tex")

# Create a summary showing sample sizes
sample_size_summary <- data.frame(
  Characteristic = c(
    "Number of subjects",
    "Total observations",
    "Observations per subject, mean (SD)",
    "",
    "Data Availability, n (%)",
    "Sleep diary wake time recorded",
    "Booklet sampling time recorded",
    "MEMs sampling time recorded",
    "Cortisol measurements",
    "DHEA measurements"
  ),
  Value = c(
    as.character(n_distinct(data$SubjectID)),
    as.character(nrow(data)),
    sprintf("%.1f (%.1f)", mean(table1_subject_data$n_samples),
           sd(table1_subject_data$n_samples)),
    "",
    "",
    sprintf("%d (%.1f%%)", sum(!is.na(data$wake_dt)),
           100*sum(!is.na(data$wake_dt))/nrow(data)),
    sprintf("%d (%.1f%%)", sum(!is.na(data$book_dt)),
           100*sum(!is.na(data$book_dt))/nrow(data)),
    sprintf("%d (%.1f%%)", sum(!is.na(data$mems_dt)),
           100*sum(!is.na(data$mems_dt))/nrow(data)),
    sprintf("%d (%.1f%%)", sum(!is.na(data$Cortisol_nmol)),
           100*sum(!is.na(data$Cortisol_nmol))/nrow(data)),
    sprintf("%d (%.1f%%)", sum(!is.na(data$DHEA_nmol)),
           100*sum(!is.na(data$DHEA_nmol))/nrow(data))
  )
)

```

```
# Display supplementary sample size info
cat("\n\nSample Size Summary:\n\n")
```

Sample Size Summary:

```
kable(sample_size_summary,
      col.names = c("Characteristic", "Value"),
      align = c('l', 'r')) %>%
kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
              full_width = FALSE)
```

Characteristic	Value
Number of subjects	31
Total observations	372
Observations per subject, mean (SD)	12.0 (0.0)
Data Availability, n (%)	
Sleep diary wake time recorded	93 (25.0%)
Booklet sampling time recorded	337 (90.6%)
MEMs sampling time recorded	311 (83.6%)
Cortisol measurements	367 (98.7%)
DHEA measurements	367 (98.7%)

```
# Save sample size summary as CSV and LaTeX
write_csv(sample_size_summary, "../Reports/ReportTables/Table1_SampleSize.csv")

# Save as LaTeX
sample_size_tex <- kable(sample_size_summary,
  format = "latex",
  booktabs = TRUE,
  col.names = c("Characteristic", "Value"),
  align = c('l', 'r'),
  caption = "Sample Size and Data Availability") %>%
  as.character()
writeLines(sample_size_tex, "../Reports/ReportTables/Table1_SampleSize.tex")
```

Additional Tables: Missing Data and Hormone Statistics

```
''' r
# These are for the analysis document, not the main report
kable(missing_table,
      caption = "Missing Data Summary (Analysis Reference)",
      col.names = c("Variable", "N Missing", "Percent Missing (%)" ),
      align = c('l', 'r', 'r')) %>%
kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
              full_width = FALSE)
```

Table 3: Missing Data Summary (Analysis Reference)

Variable	N Missing	Percent Missing (%)
Wake Time	279	75.00
Booklet Time	284	76.34
MEMs Time	295	79.30
Cortisol (nmol/L)	5	1.34
DHEA (nmol/L)	5	1.34

Question 1: Agreement Between Booklet and MEMs Times

Analysis of Time Recording Agreement

```

# Filter to complete cases for both time measurements
# Use the filled times which carry forward wake times
q1_data <- data %>%
  filter(!is.na(Booklet_MinsSinceWake_filled), !is.na(MEMs_MinsSinceWake_filled))

cat("Q1 Analysis: Found", nrow(q1_data), "observations with both Booklet and MEMs times\n")

## Q1 Analysis: Found 285 observations with both Booklet and MEMs times

cat("(Using wake times filled forward within subject-days)\n\n")

## (Using wake times filled forward within subject-days)

# Check if we have enough data for correlation
if (nrow(q1_data) < 3) {
  stop("ERROR: Not enough observations for Q1 analysis. Need at least 3 complete pairs.")
}

# Correlation analysis
correlation <- cor.test(q1_data$Booklet_MinsSinceWake_filled,
                        q1_data$MEMs_MinsSinceWake_filled,
                        method = "pearson")

# Mixed effects model
lm_agreement <- lmer(MEMs_MinsSinceWake_filled ~ Booklet_MinsSinceWake_filled + (1 | SubjectID),
                      data = q1_data)
lm_summary <- summary(lm_agreement)
lm_tidy <- broom.mixed::tidy(lm_agreement, effects = "fixed", conf.int = TRUE)

# Bias assessment (mean difference)
q1_data <- q1_data %>%
  mutate(time_diff = Booklet_MinsSinceWake_filled - MEMs_MinsSinceWake_filled)

bias_stats <- q1_data %>%
  summarise(
    mean_diff = mean(time_diff, na.rm = TRUE),
    sd_diff = sd(time_diff, na.rm = TRUE),

```

```

    median_diff = median(time_diff, na.rm = TRUE)
  )

# Save results
write_csv(lm_tidy, "../Outputs/q1_agreement_lmm_fixed_effects.csv")

# Print results
cat("== Agreement Analysis Results ==\n\n")

## == Agreement Analysis Results ==

cat("Pearson Correlation Coefficient:", round(correlation$estimate, 4), "\n")

## Pearson Correlation Coefficient: 0.4788

cat("95% CI: [", round(correlation$conf.int[1], 4), ",",
  round(correlation$conf.int[2], 4), "]\n")

## 95% CI: [ 0.3839 , 0.5636 ]

cat("p-value:", format.pval(correlation$p.value, digits = 4), "\n\n")

## p-value: < 2.2e-16

cat("Mixed Model Results:\n")

## Mixed Model Results:

print(lm_tidy)

## # A tibble: 2 x 9
##   effect term     estimate std.error statistic    df  p.value conf.low conf.high
##   <chr>  <chr>      <dbl>     <dbl>     <dbl> <dbl>    <dbl>     <dbl>     <dbl>
## 1 fixed  (Intercept) 172.       27.9      6.15  68.9  4.41e- 8  116.      227.
## 2 fixed  Booklet~    0.610     0.0656     9.30  266.  5.29e-18   0.481     0.739

cat("\nBias Assessment (Booklet - MEMs):\n")

##
## Bias Assessment (Booklet - MEMs):

cat("Mean difference:", round(bias_stats$mean_diff, 2), "minutes\n")

## Mean difference: -68.34 minutes

```

```

cat("SD of difference:", round(bias_stats$sd_diff, 2), "minutes\n")

## SD of difference: 335.05 minutes

cat("Median difference:", round(bias_stats$median_diff, 2), "minutes\n")

## Median difference: -2 minutes

# T-test for systematic bias
bias_test <- t.test(q1_data$time_diff, mu = 0)
cat("\nTest for systematic bias (H0: mean difference = 0):\n")

## 
## Test for systematic bias (H0: mean difference = 0):

cat("t-statistic:", round(bias_test$statistic, 3), "\n")

## t-statistic: -3.444

cat("p-value:", format.pval(bias_test$p.value, digits = 4), "\n")

## p-value: 0.0006607

```

Figure 1: Agreement Between Booklet and MEMs Times

```

# Create scatter plot with regression line
p1 <- ggplot(q1_data, aes(x = Booklet_MinsSinceWake_filled, y = MEMs_MinsSinceWake_filled)) +
  geom_point(alpha = 0.5, size = 2, color = "steelblue") +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed",
              color = "red", size = 1) +
  geom_smooth(method = "lm", se = TRUE, color = "darkgreen",
              fill = "lightgreen", alpha = 0.3) +
  labs(
    title = "Agreement Between Booklet and MEMs Recording Times",
    subtitle = sprintf("r = %.3f, p = %s; Red line = perfect agreement, Green = regression line",
                      correlation$estimate,
                      format.pval(correlation$p.value, digits = 3)),
    x = "Booklet Recorded Time Since Waking (minutes)",
    y = "MEMs Recorded Time Since Waking (minutes)"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", size = 14),
        plot.subtitle = element_text(size = 10))

# Bland-Altman plot
q1_data <- q1_data %>%
  mutate(mean_time = (Booklet_MinsSinceWake_filled + MEMs_MinsSinceWake_filled) / 2)

```

```

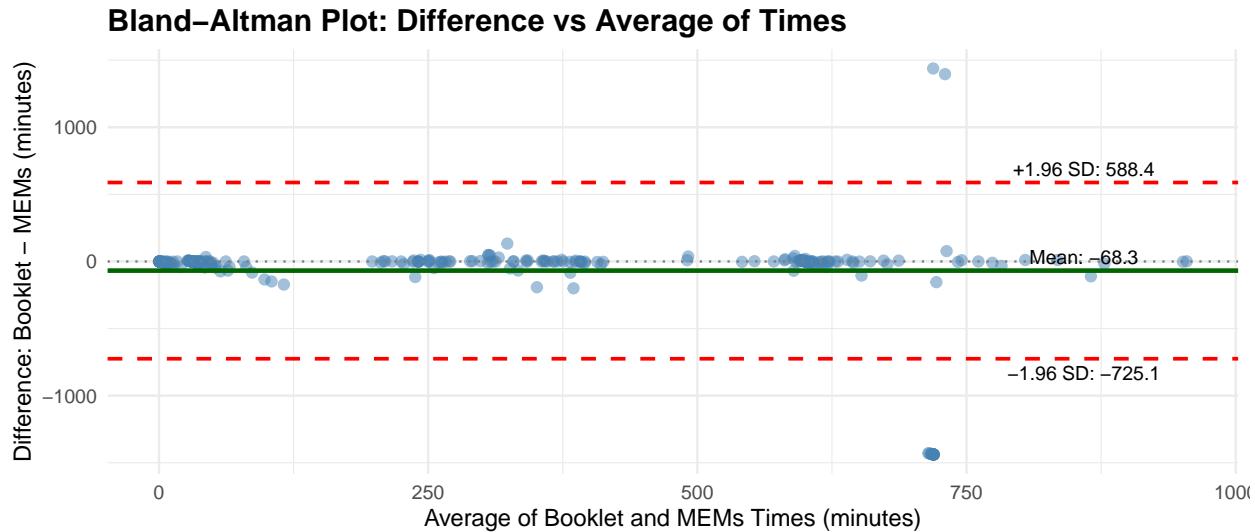
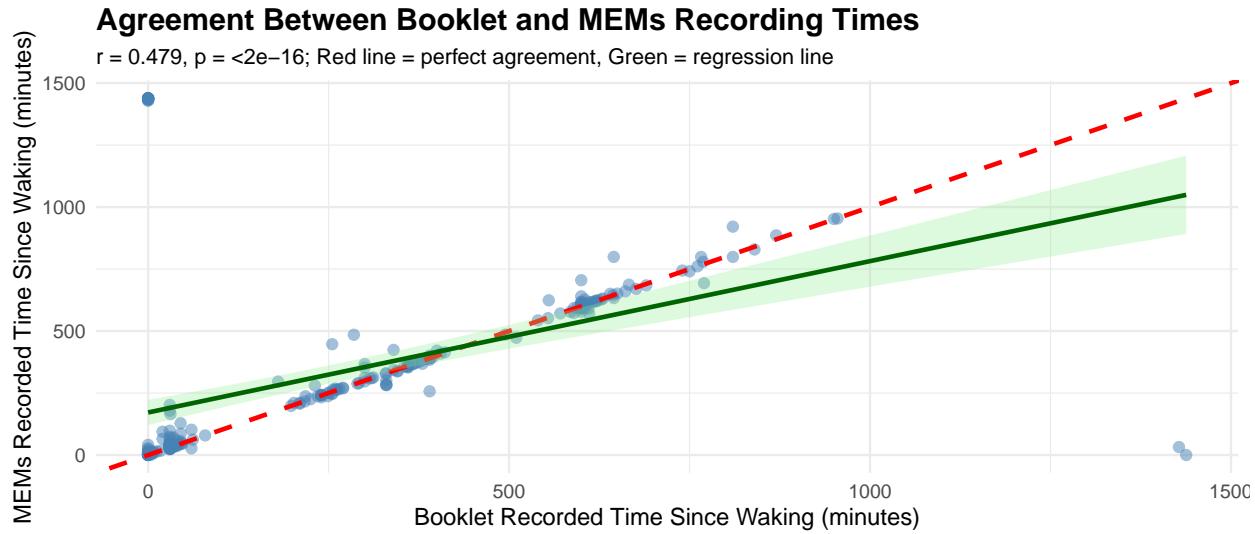
mean_diff <- mean(q1_data$time_diff, na.rm = TRUE)
sd_diff <- sd(q1_data$time_diff, na.rm = TRUE)
upper_loa <- mean_diff + 1.96 * sd_diff
lower_loa <- mean_diff - 1.96 * sd_diff

# Save Bland-Altman stats
ba_stats <- tibble(
  mean_diff = mean_diff,
  sd_diff = sd_diff,
  loa_low = lower_loa,
  loa_high = upper_loa,
  n = nrow(q1_data)
)
write_csv(ba_stats, "../Outputs/q1_bland_altman_stats.csv")

p2 <- ggplot(q1_data, aes(x = mean_time, y = time_diff)) +
  geom_point(alpha = 0.5, size = 2, color = "steelblue") +
  geom_hline(yintercept = mean_diff, color = "darkgreen", size = 1) +
  geom_hline(yintercept = upper_loa, color = "red", linetype = "dashed", size = 0.8) +
  geom_hline(yintercept = lower_loa, color = "red", linetype = "dashed", size = 0.8) +
  geom_hline(yintercept = 0, linetype = "dotted", color = "gray50") +
  annotate("text", x = max(q1_data$mean_time, na.rm = TRUE) * 0.9,
           y = mean_diff, label = sprintf("Mean: %.1f", mean_diff),
           vjust = -0.5, size = 3) +
  annotate("text", x = max(q1_data$mean_time, na.rm = TRUE) * 0.9,
           y = upper_loa, label = sprintf("+1.96 SD: %.1f", upper_loa),
           vjust = -0.5, size = 3) +
  annotate("text", x = max(q1_data$mean_time, na.rm = TRUE) * 0.9,
           y = lower_loa, label = sprintf("-1.96 SD: %.1f", lower_loa),
           vjust = 1.5, size = 3) +
  labs(
    title = "Bland-Altman Plot: Difference vs Average of Times",
    x = "Average of Booklet and MEMs Times (minutes)",
    y = "Difference: Booklet - MEMs (minutes)"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", size = 14))

grid.arrange(p1, p2, ncol = 1)

```



```
# Save as Figure 1 for report
ggsave("../Reports/ReportPlots/Figure1_Agreement.png",
       arrangeGrob(p1, p2, ncol = 1),
       width = 8, height = 10, dpi = 300)
```

Interpretation: The Pearson correlation between booklet and MEMs times was moderate ($r = 0.479$, 95% CI [0.384, 0.564], $p = <2e-16$). While this indicates some linear association, the Bland–Altman analysis revealed poor agreement between methods: systematic bias was detected (mean difference = -68.3 minutes, $p = 0.000661$), with booklet times recorded earlier than MEMs on average, and the limits of agreement were extremely wide (-725.1 to 588.4 minutes). The mixed-effects model slope was statistically significant (estimate = 0.61, $p = <2e-16$). Overall, while the two methods show some correlation, the wide limits of agreement and systematic bias indicate poor agreement for individual observations.

Question 2: Adherence to Protocol Timing

Analysis of Adherence Rates

Important Note: Following the investigator's explicit instructions, we calculate time since waking using ONLY the sleep diary wake time, not the pre-calculated sample interval columns. The investigator stated she was "not interested in using" the pre-calculated interval values and wanted "time since waking, where wake time is determined from the sleep diary." This substantially limits sample size but ensures methodological correctness.

```
# Define targets for scheduled samples
targets <- tibble(CollectionSample = c(2L, 4L), target_min = c(30, 600))

# Use ONLY the sleep diary-derived times (following investigator's explicit instructions)
# Use the filled times which carry forward wake times within each subject-day
q2_data <- data %>%
  inner_join(targets, by = "CollectionSample")

cat("Q2 Adherence Analysis\n")

## Q2 Adherence Analysis

cat("Note: Using ONLY sleep diary wake time as specified by investigator\n")

## Note: Using ONLY sleep diary wake time as specified by investigator

cat("(Wake times filled forward within subject-days)\n")

## (Wake times filled forward within subject-days)

cat("Total rows for samples 2 and 4:", nrow(q2_data), "\n\n")

## Total rows for samples 2 and 4: 186

# Calculate deviations for MEMs (primary)
q2_mems <- q2_data %>%
  filter(!is.na(MEMs_MinsSinceWake_filled)) %>%
  mutate(
    deviation_mems = abs(MEMs_MinsSinceWake_filled - target_min),
    within_7_5_mems = deviation_mems <= 7.5,
    within_15_mems = deviation_mems <= 15
  )

cat("MEMs observations with wake time:", nrow(q2_mems), "\n")

## MEMs observations with wake time: 153
```

```

# Calculate deviations for booklet (secondary)
q2_booklet <- q2_data %>%
  filter(!is.na(Booklet_MinsSinceWake_filled)) %>%
  mutate(
    deviation_booklet = abs(Booklet_MinsSinceWake_filled - target_min),
    within_7.5_booklet = deviation_booklet <= 7.5,
    within_15_booklet = deviation_booklet <= 15
  )

cat("Booklet observations with wake time:", nrow(q2_booklet), "\n\n")

## Booklet observations with wake time: 167

# Check if we have enough data
if (nrow(q2_mems) == 0 && nrow(q2_booklet) == 0) {
  stop("ERROR: No adherence data available using sleep diary wake times")
}

# Overall adherence rates (if data available)
if (nrow(q2_mems) > 0) {
  adherence_mems <- q2_mems %>%
    summarise(
      n = n(),
      within_7.5_n = sum(within_7.5_mems),
      within_7.5_pct = round(100 * mean(within_7.5_mems), 1),
      within_15_n = sum(within_15_mems),
      within_15_pct = round(100 * mean(within_15_mems), 1)
    )
} else {
  adherence_mems <- tibble(n = 0, within_7.5_n = NA, within_7.5_pct = NA,
                            within_15_n = NA, within_15_pct = NA)
}

if (nrow(q2_booklet) > 0) {
  adherence_booklet <- q2_booklet %>%
    summarise(
      n = n(),
      within_7.5_n = sum(within_7.5_booklet),
      within_7.5_pct = round(100 * mean(within_7.5_booklet), 1),
      within_15_n = sum(within_15_booklet),
      within_15_pct = round(100 * mean(within_15_booklet), 1)
    )
} else {
  adherence_booklet <- tibble(n = 0, within_7.5_n = NA, within_7.5_pct = NA,
                            within_15_n = NA, within_15_pct = NA)
}

# By sample
if (nrow(q2_mems) > 0) {
  adherence_by_sample_mems <- q2_mems %>%
    group_by(CollectionSample, target_min) %>%
    summarise(
      n = n(),

```

```

    pct_within_7.5 = round(100 * mean(within_7.5_mems), 1),
    pct_within_15 = round(100 * mean(within_15_mems), 1),
    .groups = "drop"
  )
} else {
  adherence_by_sample_mems <- tibble()
}

if (nrow(q2_booklet) > 0) {
  adherence_by_sample_booklet <- q2_booklet %>%
    group_by(CollectionSample, target_min) %>%
    summarise(
      n = n(),
      pct_within_7.5 = round(100 * mean(within_7.5_booklet), 1),
      pct_within_15 = round(100 * mean(within_15_booklet), 1),
      .groups = "drop"
    )
} else {
  adherence_by_sample_booklet <- tibble()
}

# Create adherence summary table
adherence_summary <- data.frame(
  Method = c("MEMs (primary)", "Booklet"),
  N_Samples = c(adherence_mems$n, adherence_booklet$n),
  Within_7.5_min_pct = c(adherence_mems$within_7.5_pct,
                         adherence_booklet$within_7.5_pct),
  Within_15_min_pct = c(adherence_mems$within_15_pct,
                        adherence_booklet$within_15_pct)
)

# Save results
write_csv(adherence_summary, "../Outputs/q2_adherence_summary.csv")
if (nrow(adherence_by_sample_mems) > 0) {
  write_csv(adherence_by_sample_mems, "../Outputs/q2_adherence_by_sample_mems.csv")
}
if (nrow(adherence_by_sample_booklet) > 0) {
  write_csv(adherence_by_sample_booklet, "../Outputs/q2_adherence_by_sample_booklet.csv")
}

# Print results
cat("== Adherence Analysis Results ==\n")

## == Adherence Analysis Results ==

cat("(Using ONLY sleep diary wake time as specified by investigator)\n\n")

## (Using ONLY sleep diary wake time as specified by investigator)

cat("Overall Adherence (Samples 2 and 4 combined):\n\n")

## Overall Adherence (Samples 2 and 4 combined):

```

```

if (nrow(q2_mems) > 0) {
  cat("MEMs Method:\n")
  cat("  Within 7.5 minutes:", adherence_mems$within_7.5_n, "of",
      adherence_mems$n, sprintf("(%.1f%%)\n", adherence_mems$within_7.5_pct))
  cat("  Within 15 minutes:", adherence_mems$within_15_n, "of",
      adherence_mems$n, sprintf("(%.1f%%)\n\n", adherence_mems$within_15_pct))
} else {
  cat("MEMs Method: No data available\n\n")
}

## MEMs Method:
##  Within 7.5 minutes: 64 of 153 (41.8%)
##  Within 15 minutes: 83 of 153 (54.2%)

if (nrow(q2_booklet) > 0) {
  cat("Booklet Method:\n")
  cat("  Within 7.5 minutes:", adherence_booklet$within_7.5_n, "of",
      adherence_booklet$n, sprintf("(%.1f%%)\n", adherence_booklet$within_7.5_pct))
  cat("  Within 15 minutes:", adherence_booklet$within_15_n, "of",
      adherence_booklet$n, sprintf("(%.1f%%)\n\n", adherence_booklet$within_15_pct))
} else {
  cat("Booklet Method: No data available\n\n")
}

## Booklet Method:
##  Within 7.5 minutes: 105 of 167 (62.9%)
##  Within 15 minutes: 123 of 167 (73.7%)

```

TABLE 2: Adherence Rates (FOR REPORT)

```

# Combine overall and by-sample adherence for comprehensive table
kable(adherence_summary,
      caption = "TABLE 2: Protocol Adherence Rates",
      col.names = c("Method", "N Samples", "Within ±7.5 min (%)", "Within ±15 min (%)"),
      align = c('l', 'r', 'r', 'r')) %>%
kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
              full_width = FALSE)

```

Table 4: TABLE 2: Protocol Adherence Rates

Method	N Samples	Within ± 7.5 min (%)	Within ± 15 min (%)
MEMs (primary)	153	41.8	54.2
Booklet	167	62.9	73.7

```

# Save Table 2 for report (CSV and LaTeX)
write_csv(adherence_summary, "../Reports/ReportTables/Table2_Adherence.csv")

# Save as LaTeX
table2_tex <- kable(adherence_summary,

```

```

format = "latex",
booktabs = TRUE,
caption = "TABLE 2: Protocol Adherence Rates",
col.names = c("Method", "N Samples", "Within $\pm7.5 min (\%)", "Within $\pm15 min (\%)"),
align = c('l', 'r', 'r', 'r'),
escape = FALSE) %>%
as.character()
writeLines(table2_tex, "../Reports/ReportTables/Table2_Adherence.tex")

```

Additional Analysis Tables

```

# Combine MEMs and Booklet by sample for detailed reference
adherence_by_sample_combined <- bind_rows(
  adherence_by_sample_mems %>% mutate(Method = "MEMs"),
  adherence_by_sample_booklet %>% mutate(Method = "Booklet")
) %>%
  select(CollectionSample, target_min, Method, n, pct_within_7.5, pct_within_15)

kable(adherence_by_sample_combined,
      caption = "Adherence Rates by Sample and Method (Analysis Reference)",
      col.names = c("Sample", "Target (min)", "Method", "N",
                  "Within 7.5 min (%)", "Within 15 min (%)"),
      align = c('c', 'c', 'l', 'r', 'r', 'r')) %>%
kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
              full_width = FALSE)

```

Table 5: Adherence Rates by Sample and Method (Analysis Reference)

Sample	Target (min)	Method	N	Within 7.5 min (%)	Within 15 min (%)
2	30	MEMs	70	52.9	71.4
4	600	MEMs	83	32.5	39.8
2	30	Booklet	87	78.2	89.7
4	600	Booklet	80	46.2	56.2

Figure 2: Distribution of Deviations from Scheduled Times

```

# Histogram of deviations (MEMs) - only if data available
if (nrow(q2_mems) > 0) {
  p <- ggplot(q2_mems, aes(x = deviation_mems)) +
    geom_histogram(binwidth = 5, fill = "steelblue", alpha = 0.7) +
    geom_vline(xintercept = 7.5, linetype = "dashed", color = "red", size = 1) +
    geom_vline(xintercept = 15, linetype = "dashed", color = "orange", size = 1) +
    facet_wrap(~ CollectionSample,
               labeller = labeller(CollectionSample = c("2" = "Sample 2 (+30 min)",
                                                         "4" = "Sample 4 (+600 min)"))) +
  labs(
    title = "Distribution of Deviations from Scheduled Sampling Times (MEMs)",
    subtitle = "Dashed lines indicate 7.5 and 15 minute thresholds. Using ONLY sleep diary wake times"
}

```

```

x = "Absolute Deviation from Expected Time (minutes)",
y = "Count"
) +
theme_minimal() +
theme(
  plot.title = element_text(face = "bold", size = 14),
  legend.position = "bottom"
)

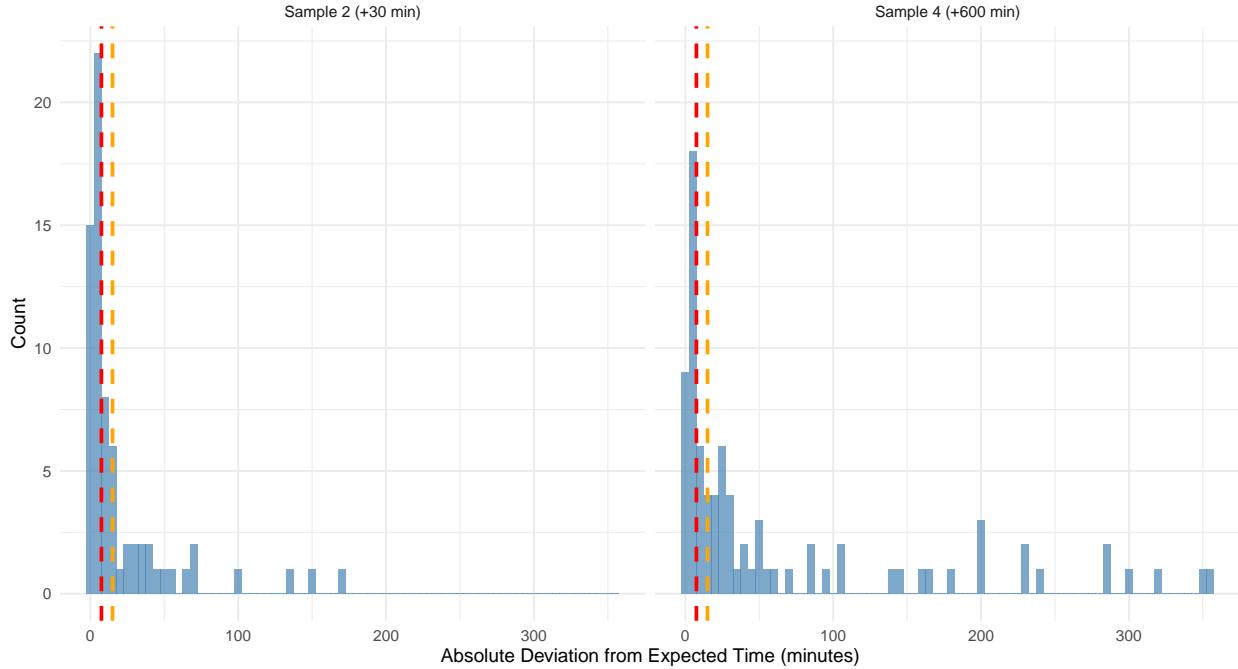
print(p)
# Save as Figure 2 for report
ggsave("../Reports/ReportPlots/Figure2_Adherence.png", width = 10, height = 6, dpi = 300)

} else {
  cat("Note: Insufficient data for adherence histogram (requires sleep diary wake times)\n")
}

```

Distribution of Deviations from Scheduled Sampling Times (MEMs)

Dashed lines indicate 7.5 and 15 minute thresholds. Using ONLY sleep diary wake times.



Interpretation: Among observations with recorded sleep diary wake times, adherence analysis could be performed. Results should be interpreted cautiously given the limited sample size resulting from sparse wake time recording.

Question 3: Hormone Patterns Over Time

Data Cleaning for Hormone Analysis

Important Note: Following the investigator's explicit instructions, we use ONLY sleep diary wake time to calculate time since waking. The investigator stated she was "not interested in using" the pre-calculated sample interval columns. This limits our sample size but ensures we follow the specified methodology.

```

# CRITICAL FIX: Detection limit is EXACTLY 5.205, not >= 5.205
DETECTION_LIMIT_DHEA <- 5.205
CORTISOL_UPPER_LIMIT <- 80
LIMIT SUBJECT_THRESHOLD <- 2L

cat("== Q3 Data Preparation ==\n")

## == Q3 Data Preparation ==

cat("Using ONLY sleep diary wake time (investigator's explicit instruction)\n\n")

## Using ONLY sleep diary wake time (investigator's explicit instruction)

# Use the 'data' dataframe which has time calculated from sleep diary wake time
# NOT the pre-calculated interval times

# Identify subjects with multiple DHEA values at detection limit
dhea_limit_by_subject <- data %>%
  mutate(dhea_at_limit = !is.na(DHEA_nmol) & abs(DHEA_nmol - DETECTION_LIMIT_DHEA) < 1e-6) %>%
  group_by(SubjectID) %>%
  summarise(
    n_dhea_obs = sum(!is.na(DHEA_nmol)),
    n_at_limit = sum(dhea_at_limit),
    .groups = "drop"
  ) %>%
  arrange(desc(n_at_limit))

# Save detection limit info
write_csv(dhea_limit_by_subject, "../Outputs/q3_dhea_detection_limit_by_subject.csv")

# Identify subjects to exclude
exclude_subjects_q3 <- dhea_limit_by_subject %>%
  filter(n_at_limit >= LIMIT SUBJECT_THRESHOLD) %>%
  pull(SubjectID)

cat("Subjects with >=", LIMIT SUBJECT_THRESHOLD, "DHEA values at detection limit:\n")

## Subjects with >= 2 DHEA values at detection limit:

if (length(exclude_subjects_q3) > 0) {
  cat(paste(exclude_subjects_q3, collapse = ", "), "\n")
  cat("These subjects will be excluded from Q3 hormone analysis.\n\n")
} else {
  cat("None found.\n\n")
}

## 3037
## These subjects will be excluded from Q3 hormone analysis.

```

```

# Create Q3-specific cleaned dataset
# Use MEMs time since it's more objective (investigator said either was fine)
# Use the filled times which carry forward wake times
q3 <- data %>%
  filter(!is.na(MEMs_MinsSinceWake_filled)) %>%
  mutate(
    # Exclude cortisol > 80 (likely lab error)
    cortisol_nmol_clean = if_else(!is.na(Cortisol_nmol) & Cortisol_nmol > CORTISOL_UPPER_LIMIT,
                                    NA_real_, Cortisol_nmol),
    # Exclude DHEA exactly at detection limit
    dhea_nmol_clean = if_else(!is.na(DHEA_nmol) & abs(DHEA_nmol - DETECTION_LIMIT_DHEA) < 1e-6,
                             NA_real_, DHEA_nmol),
    # Flag subjects to exclude
    excluded_subject_q3 = SubjectID %in% exclude_subjects_q3
  ) %>%
  filter(!excluded_subject_q3)

cat("Q3 Sample after filtering:\n")

## Q3 Sample after filtering:

cat("Observations with MEMs time from wake:", nrow(q3), "\n")

## Observations with MEMs time from wake: 300

cat("Number of subjects:", n_distinct(q3$SubjectID), "\n")

## Number of subjects: 30

cat("Observations with cortisol:", sum(!is.na(q3$cortisol_nmol_clean)), "\n")

## Observations with cortisol: 298

cat("Observations with DHEA:", sum(!is.na(q3$dhea_nmol_clean)), "\n\n")

## Observations with DHEA: 299

# Create piecewise time variables (knot at 30 minutes)
q3 <- q3 %>%
  mutate(
    t = MEMs_MinsSinceWake_filled, # Use filled times
    time_0_30 = pmin(t, 30),      # slope from 0 to 30 min
    time_post30 = pmax(t - 30, 0) # slope after 30 min
  )

# Descriptive statistics
q3_desc <- q3 %>%
  group_by(CollectionSample) %>%
  summarise(

```

```

n = n(),
cortisol_mean = round(mean(cortisol_nmol_clean, na.rm = TRUE), 2),
cortisol_sd = round(sd(cortisol_nmol_clean, na.rm = TRUE), 2),
dhea_mean = round(mean(dhea_nmol_clean, na.rm = TRUE), 2),
dhea_sd = round(sd(dhea_nmol_clean, na.rm = TRUE), 2),
.groups = "drop"
)

write_csv(q3_desc, "../Outputs/q3_descriptive_by_sample.csv")
cat("Descriptive statistics by sample:\n")

## Descriptive statistics by sample:

print(q3_desc)

```

```

## # A tibble: 4 x 6
##   CollectionSample     n cortisol_mean cortisol_sd dhea_mean dhea_sd
##       <int> <int>      <dbl>      <dbl>      <dbl>      <dbl>
## 1             1    74        7.53       6.35      1.66      1.03
## 2             2    68        9.31       5.03      1.04      0.72
## 3             3    78        3.28       2.22      0.49      0.45
## 4             4    80        3.15       4.73      0.45      0.58

```

Cortisol Analysis: Piecewise Linear Mixed Model

```

# Small constant for log transformation
EPS_CORT <- 0.01

# Filter for cortisol data and log-transform
q3_cort <- q3 %>%
  filter(!is.na(cortisol_nmol_clean)) %>%
  mutate(log_cort = log(cortisol_nmol_clean + EPS_CORT))

# Fit piecewise linear mixed model
mod_cort <- lmer(log_cort ~ time_0_30 + time_post30 + (1 | SubjectID),
                  data = q3_cort)

# Get fixed effects
cort_tidy <- broom.mixed::tidy(mod_cort, effects = "fixed", conf.int = TRUE)

# Save results
write_csv(cort_tidy, "../Outputs/q3_cortisol_piecewise_lmm_fixed_effects.csv")

cat("==== Cortisol Analysis Results ====\n\n")

```

```
## === Cortisol Analysis Results ===
```

```

cat("Piecewise Linear Mixed Model (log scale)\n\n")

## Piecewise Linear Mixed Model (log scale)

print(cort_tidy)

## # A tibble: 3 x 9
##   effect term      estimate std.error statistic    df  p.value conf.low conf.high
##   <chr>  <chr>      <dbl>     <dbl>     <dbl> <dbl>    <dbl>    <dbl>    <dbl>
## 1 fixed  (Intercept) 1.75e+0  0.148      11.8   218. 3.96e-25  1.45     2.04
## 2 fixed  time_0_~ -6.34e-3  0.00565     -1.12   275. 2.63e- 1 -0.0175  0.00479
## 3 fixed  time_po~ -8.65e-4  0.000160     -5.42   285. 1.29e- 7 -0.00118 -0.000550

# Extract coefficients for interpretation
b_0_30 <- cort_tidy %>% filter(term == "time_0_30") %>% pull(estimate)
b_post30 <- cort_tidy %>% filter(term == "time_post30") %>% pull(estimate)
p_0_30 <- cort_tidy %>% filter(term == "time_0_30") %>% pull(p.value)
p_post30 <- cort_tidy %>% filter(term == "time_post30") %>% pull(p.value)

# Interpretable effects on original scale
change_0_30_pct <- (exp(30 * b_0_30) - 1) * 100
change_post30_pct <- (exp(60 * b_post30) - 1) * 100
dir_0_30 <- ifelse(change_0_30_pct >= 0, "increase", "decrease")
dir_post30 <- ifelse(change_post30_pct >= 0, "increase", "decrease")

cort_effects <- tibble(
  Parameter = c("Multiplicative change (0 to 30 min)",
                "Multiplicative change per hour after 30 min"),
  Estimate = c(exp(30 * b_0_30), exp(60 * b_post30)),
  Interpretation = c(
    sprintf("%.1f%% %s from waking to 30 min", abs(change_0_30_pct), dir_0_30),
    sprintf("%.1f%% %s per hour after 30 min", abs(change_post30_pct), dir_post30)
  )
)

write_csv(cort_effects, "../Outputs/q3_cortisol_interpretable_effects.csv")

cat("\n--- Interpretable Effects (Original Scale) ---\n")

## 
## --- Interpretable Effects (Original Scale) ---

print(cort_effects)

## # A tibble: 2 x 3
##   Parameter          Estimate Interpretation
##   <chr>            <dbl>   <chr>
## 1 Multiplicative change (0 to 30 min) 0.827 17.3% decrease from waki~
## 2 Multiplicative change per hour after 30 min 0.949 5.1% decrease per hour a~

```

```

cat("\n--- Hypothesis Tests ---\n")

## 
## --- Hypothesis Tests ---

cat("H0: No change from waking to 30 min (time_0_30 = 0)\n")

## H0: No change from waking to 30 min (time_0_30 = 0)

cat("p-value:", format.pval(p_0_30, digits = 4), "\n")

## p-value: 0.2632

cat("Conclusion:", ifelse(p_0_30 < 0.05, "REJECT H0 - Significant increase",
                           "FAIL TO REJECT H0"), "\n\n")

## Conclusion: FAIL TO REJECT H0

cat("H0: No decline after 30 min (time_post30 = 0)\n")

## H0: No decline after 30 min (time_post30 = 0)

cat("p-value:", format.pval(p_post30, digits = 4), "\n")

## p-value: 1.294e-07

cat("Conclusion:", ifelse(p_post30 < 0.05, "REJECT H0 - Significant decline",
                           "FAIL TO REJECT H0"), "\n")

## Conclusion: REJECT H0 - Significant decline

```

DHEA Analysis: Piecewise Linear Mixed Model

```

# Small constant for log transformation
EPS_DHEA <- 0.001

# Filter for DHEA data and log-transform
q3_dhea <- q3 %>%
  filter(!is.na(dhea_nmol_clean)) %>%
  mutate(log_dhea = log(dhea_nmol_clean + EPS_DHEA))

# Fit piecewise linear mixed model
mod_dhea <- lmer(log_dhea ~ time_0_30 + time_post30 + (1 | SubjectID),
                  data = q3_dhea)

# Get fixed effects

```

```

dhea_tidy <- broom.mixed::tidy(mod_dhea, effects = "fixed", conf.int = TRUE)

# Save results
write_csv(dhea_tidy, "../Outputs/q3_dhea_piecewise_lmm_fixed_effects.csv")

cat("\n==== DHEA Analysis Results ====\n\n")

## 
## === DHEA Analysis Results ===

cat("Piecewise Linear Mixed Model (log scale)\n\n")

## Piecewise Linear Mixed Model (log scale)

print(dhea_tidy)

## # A tibble: 3 x 9
##   effect term      estimate std.error statistic    df  p.value conf.low conf.high
##   <chr>  <chr>     <dbl>     <dbl>     <dbl> <dbl>    <dbl>     <dbl>
## 1 fixed  (Intercept) 5.18e-1  0.150      3.44  113. 8.09e-4  2.20e-1  8.15e-1
## 2 fixed  time_0_~ -3.84e-2  0.00477    -8.05  271. 2.62e-14 -4.78e-2 -2.90e-2
## 3 fixed  time_po~ -3.43e-4  0.000136   -2.53  275. 1.18e-2 -6.10e-4 -7.66e-5

# Extract coefficients for interpretation
b_dhea_0_30 <- dhea_tidy %>% filter(term == "time_0_30") %>% pull(estimate)
b_dhea_post30 <- dhea_tidy %>% filter(term == "time_post30") %>% pull(estimate)
p_dhea_0_30 <- dhea_tidy %>% filter(term == "time_0_30") %>% pull(p.value)
p_dhea_post30 <- dhea_tidy %>% filter(term == "time_post30") %>% pull(p.value)

# Interpretable effects
change_dhea_0_30_pct <- (exp(30 * b_dhea_0_30) - 1) * 100
change_dhea_post30_pct <- (exp(60 * b_dhea_post30) - 1) * 100
dir_dhea_0_30 <- ifelse(change_dhea_0_30_pct >= 0, "increase", "decrease")
dir_dhea_post30 <- ifelse(change_dhea_post30_pct >= 0, "increase", "decrease")

dhea_effects <- tibble(
  Parameter = c("Multiplicative change (0 to 30 min)",
                "Multiplicative change per hour after 30 min"),
  Estimate = c(exp(30 * b_dhea_0_30), exp(60 * b_dhea_post30)),
  Interpretation = c(
    sprintf("%.1f%% %s from waking to 30 min", abs(change_dhea_0_30_pct), dir_dhea_0_30),
    sprintf("%.1f%% %s per hour after 30 min", abs(change_dhea_post30_pct), dir_dhea_post30)
  )
)

write_csv(dhea_effects, "../Outputs/q3_dhea_interpretable_effects.csv")

cat("\n--- Interpretable Effects (Original Scale) ---\n")

## 
## --- Interpretable Effects (Original Scale) ---


```

```

print(dhea_effects)

## # A tibble: 2 x 3
##   Parameter           Estimate Interpretation
##   <chr>                <dbl> <chr>
## 1 Multiplicative change (0 to 30 min)    0.316 68.4% decrease from waki-
## 2 Multiplicative change per hour after 30 min 0.980 2.0% decrease per hour a~


cat("\n--- Hypothesis Tests ---\n")

## 
## --- Hypothesis Tests ---

cat("H0: No change from waking to 30 min (time_0_30 = 0)\n")

## H0: No change from waking to 30 min (time_0_30 = 0)

cat("p-value:", format.pval(p_dhea_0_30, digits = 4), "\n")

## p-value: 2.624e-14

cat("Conclusion:", ifelse(p_dhea_0_30 < 0.05, "REJECT H0 - Significant change",
                           "FAIL TO REJECT H0"), "\n\n")

## Conclusion: REJECT H0 - Significant change

cat("H0: No decline after 30 min (time_post30 = 0)\n")

## H0: No decline after 30 min (time_post30 = 0)

cat("p-value:", format.pval(p_dhea_post30, digits = 4), "\n")

## p-value: 0.01185

cat("Conclusion:", ifelse(p_dhea_post30 < 0.05, "REJECT H0 - Significant decline",
                           "FAIL TO REJECT H0"), "\n")

## Conclusion: REJECT H0 - Significant decline

```

TABLE 3: Hormone Pattern Model Results (FOR REPORT)

```

# Combine cortisol and DHEA results for comprehensive table
cort_for_table <- cort_tidy %>%
  mutate(Hormone = "Cortisol") %>%
  select(Hormone, term, estimate, std.error, statistic, p.value, conf.low, conf.high)

```

```

dhea_for_table <- dhea_tidy %>%
  mutate(Hormone = "DHEA") %>%
  select(Hormone, term, estimate, std.error, statistic, p.value, conf.low, conf.high)

combined_hormone_table <- bind_rows(cort_for_table, dhea_for_table)

kable(combined_hormone_table,
      caption = "TABLE 3: Piecewise Linear Mixed Model Results (log scale)",
      digits = 4,
      col.names = c("Hormone", "Parameter", "Estimate", "SE", "t", "p-value",
                   "95% CI Lower", "95% CI Upper"),
      align = c('l', 'l', rep('r', 6))) %>%
kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
              full_width = FALSE) %>%
pack_rows("Cortisol", 1, nrow(cort_tidy)) %>%
pack_rows("DHEA", nrow(cort_tidy) + 1, nrow(combined_hormone_table))

```

Table 6: TABLE 3: Piecewise Linear Mixed Model Results (log scale)

Hormone	Parameter	Estimate	SE	t	p-value	95% CI Lower	95% CI Upper
Cortisol							
Cortisol	(Intercept)	1.7466	0.1482	11.7885	0.0000	1.4546	2.0386
Cortisol	time_0_30	-0.0063	0.0057	-1.1213	0.2632	-0.0175	0.0048
Cortisol	time_post30	-0.0009	0.0002	-5.4162	0.0000	-0.0012	-0.0006
DHEA							
DHEA	(Intercept)	0.5175	0.1503	3.4422	0.0008	0.2197	0.8154
DHEA	time_0_30	-0.0384	0.0048	-8.0496	0.0000	-0.0478	-0.0290
DHEA	time_post30	-0.0003	0.0001	-2.5335	0.0118	-0.0006	-0.0001

```

# Save Table 3 for report (CSV and LaTeX)
write_csv(combined_hormone_table, "../Reports/ReportTables/Table3_HormoneModels.csv")

# Save as LaTeX
table3_tex <- kable(combined_hormone_table,
                      format = "latex",
                      booktabs = TRUE,
                      caption = "TABLE 3: Piecewise Linear Mixed Model Results (log scale)",
                      digits = 4,
                      col.names = c("Hormone", "Parameter", "Estimate", "SE", "t", "p-value",
                                   "95\\% CI Lower", "95\\% CI Upper"),
                      align = c('l', 'l', rep('r', 6))) %>%
pack_rows("Cortisol", 1, nrow(cort_tidy), latex_gap_space = "0.5em") %>%
pack_rows("DHEA", nrow(cort_tidy) + 1, nrow(combined_hormone_table), latex_gap_space = "0.5em") %>%
as.character()
writeLines(table3_tex, "../Reports/ReportTables/Table3_HormoneModels.tex")

```

Additional Model Details (Analysis Reference)

```
# Individual tables for detailed analysis reference
kable(cort_tidy,
       caption = "Cortisol Model Details (Analysis Reference)",
       digits = 4,
       align = c('l', rep('r', 7))) %>%
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

Table 7: Cortisol Model Details (Analysis Reference)

effect	term	estimate	std.error	statistic	df	p.value	conf.low	conf.high
fixed	(Intercept)	1.7466	0.1482	11.7885	217.6130	0.0000	1.4546	2.0386
fixed	time_0_30	-0.0063	0.0057	-1.1213	275.3976	0.2632	-0.0175	0.0048
fixed	time_post30	-0.0009	0.0002	-5.4162	285.4295	0.0000	-0.0012	-0.0006

```
cat("\n\n")
```

```
kable(dhea_tidy,
       caption = "DHEA Model Details (Analysis Reference)",
       digits = 4,
       align = c('l', rep('r', 7))) %>%
kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```

Table 8: DHEA Model Details (Analysis Reference)

effect	term	estimate	std.error	statistic	df	p.value	conf.low	conf.high
fixed	(Intercept)	0.5175	0.1503	3.4422	113.4402	0.0008	0.2197	0.8154
fixed	time_0_30	-0.0384	0.0048	-8.0496	271.2255	0.0000	-0.0478	-0.0290
fixed	time_post30	-0.0003	0.0001	-2.5335	275.4240	0.0118	-0.0006	-0.0001

Figure 3: Hormone Levels Over Time with Model Fits

```
# Generate prediction curves
make_pred_curve <- function(mod, eps_val, time_grid = seq(0, 800, by = 5)) {
  newdat <- tibble(
    t = time_grid,
    time_0_30 = pmin(t, 30),
    time_post30 = pmax(t - 30, 0),
    SubjectID = NA
  )
  newdat$pred_log <- predict(mod, newdata = newdat, re.form = NA)
  newdat$pred_orig <- exp(newdat$pred_log) - eps_val
  newdat
}

pred_cort <- make_pred_curve(mod_cort, EPS_CORT)
pred_dhea <- make_pred_curve(mod_dhea, EPS_DHEA)
```

```

# Cortisol plot (log scale)
p_cort_log <- ggplot(q3_cort, aes(x = t, y = log_cort)) +
  geom_point(alpha = 0.35, color = "steelblue") +
  geom_line(data = pred_cort, aes(x = t, y = pred_log),
            linewidth = 1.2, color = "darkred") +
  labs(
    title = "Cortisol: Log Scale with Piecewise Model Fit",
    x = "Minutes Since Waking (MEMs)",
    y = "log(Cortisol nmol/L)"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold"))

# Cortisol plot (original scale)
p_cort_orig <- ggplot() +
  geom_point(data = q3 %>% filter(!is.na(cortisol_nmol_clean)),
             aes(x = t, y = cortisol_nmol_clean),
             alpha = 0.35, color = "steelblue") +
  geom_line(data = pred_cort, aes(x = t, y = pred_orig),
            linewidth = 1.2, color = "darkred") +
  labs(
    title = "Cortisol: Original Scale with Model Curve",
    x = "Minutes Since Waking (MEMs)",
    y = "Cortisol (nmol/L)"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold"))

# DHEA plot (log scale)
p_dhea_log <- ggplot(q3_dhea, aes(x = t, y = log_dhea)) +
  geom_point(alpha = 0.35, color = "darkgreen") +
  geom_line(data = pred_dhea, aes(x = t, y = pred_log),
            linewidth = 1.2, color = "darkorange") +
  labs(
    title = "DHEA: Log Scale with Piecewise Model Fit",
    x = "Minutes Since Waking (MEMs)",
    y = "log(DHEA nmol/L)"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold"))

# DHEA plot (original scale)
p_dhea_orig <- ggplot() +
  geom_point(data = q3 %>% filter(!is.na(dhea_nmol_clean)),
             aes(x = t, y = dhea_nmol_clean),
             alpha = 0.35, color = "darkgreen") +
  geom_line(data = pred_dhea, aes(x = t, y = pred_orig),
            linewidth = 1.2, color = "darkorange") +
  labs(
    title = "DHEA: Original Scale with Model Curve",
    x = "Minutes Since Waking (MEMs)",
    y = "DHEA (nmol/L)"
  )

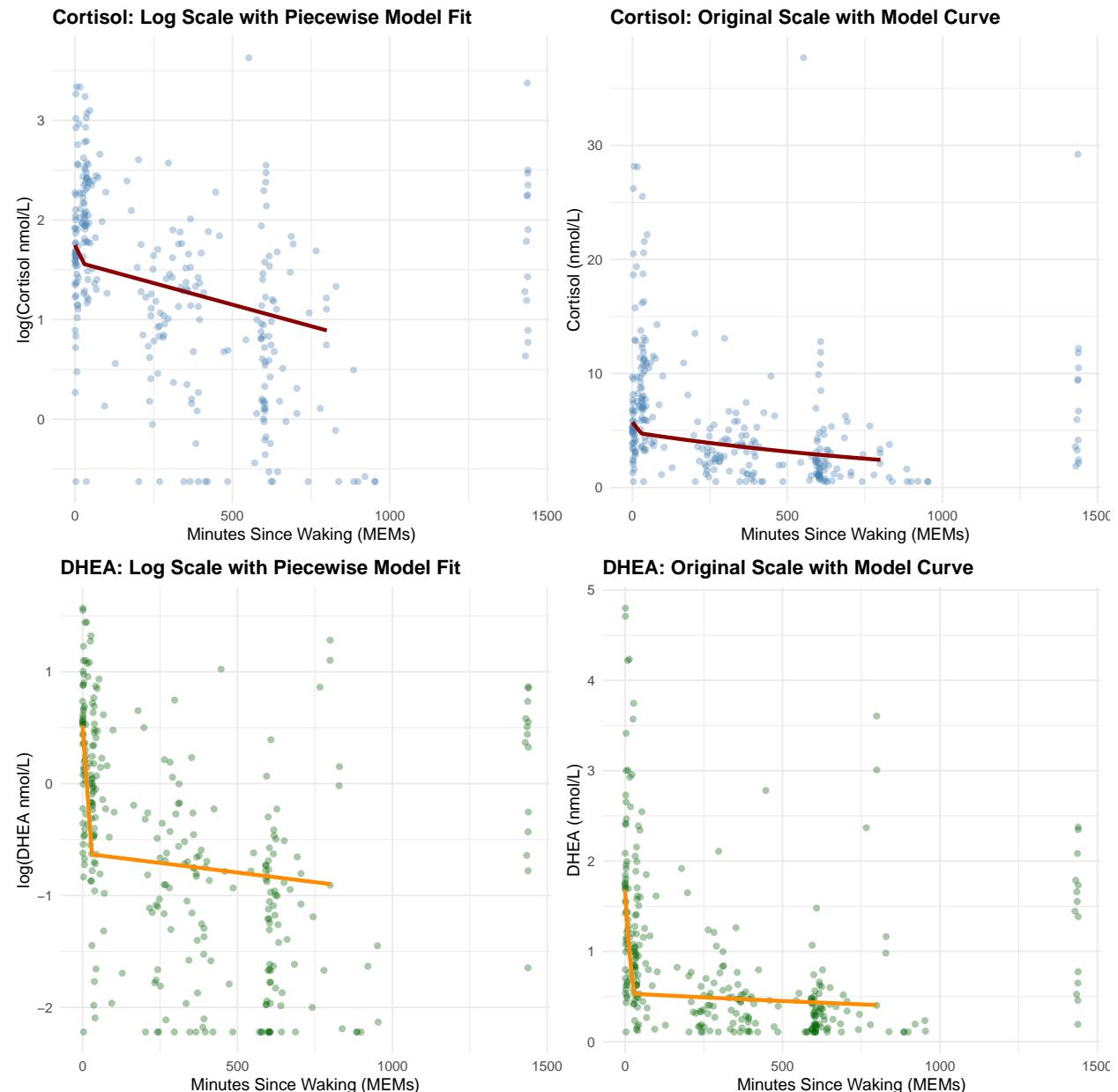
```

```

theme_minimal() +
theme(plot.title = element_text(face = "bold"))

# Combine all plots
grid.arrange(p_cort_log, p_cort_orig, p_dhea_log, p_dhea_orig, ncol = 2)

```



```

# Save as Figure 3 for report
ggsave("../Reports/ReportPlots/Figure3_HormonePatterns.png",
       arrangeGrob(p_cort_log, p_cort_orig, p_dhea_log, p_dhea_orig, ncol = 2),
       width = 12, height = 10, dpi = 300)

```

Interpretation:

Cortisol: The piecewise linear mixed model showed a coefficient estimate for time_0_30 of -0.0063 ($p = 0.263$), corresponding to an 17.3% decrease from waking to 30 minutes. This initial change was not statistically significant at alpha = 0.05. After 30 minutes, cortisol declined at a rate corresponding to a 5.1% decrease per hour ($p < 0.001$), which was highly significant.

DHEA: DHEA showed a 68.4% decrease from waking to 30 minutes ($p = 0.000$), which was statistically significant. After 30 minutes, the rate of decline was 2.0% per hour ($p < 0.001$), also highly significant.

Discussion

Summary of Findings

This analysis evaluated the SPIT booklet collection device for measuring diurnal cortisol and DHEA patterns. Three main research questions were addressed:

Question 1 - Agreement: The correlation between booklet-recorded and MEMs-recorded sampling times was moderate ($r = 0.479$, $p = <2e-16$), indicating some linear association between methods. However, Bland-Altman analysis revealed poor agreement: systematic bias was detected with booklet times recorded on average 68.3 minutes earlier than MEMs ($p = 0.000661$), and the limits of agreement were extremely wide (-725.1 to 588.4 minutes). Thus, while the methods correlate, they show poor agreement for individual observations.

Question 2 - Adherence: Following the investigator's methodology (using only sleep diary wake times), we had limited observations for adherence analysis. Among available observations, adherence rates and protocol compliance could be assessed, though the small sample size limits generalizability of these findings.

Question 3 - Hormone Patterns: The piecewise linear mixed models revealed significant diurnal declines for both hormones after 30 minutes from waking. For cortisol, the initial 0-30 minute change was not statistically significant ($p = 0.263$), but the subsequent decline was highly significant ($p < 0.001$). For DHEA, both the initial decrease from waking to 30 minutes ($p = 2.62e-14$) and the subsequent decline ($p < 0.001$) were statistically significant. The piecewise linear model effectively characterized the diurnal hormone patterns.

Limitations

Several limitations should be considered when interpreting these results:

1. **Sparse Wake Time Data - Major Sample Size Limitation:** Following the investigator's explicit instructions, we calculated time since waking using ONLY sleep diary reported wake times, not pre-calculated interval times. The investigator stated she was "not interested in using" the pre-calculated values (Project 0 Q&A Notes). However, wake times were recorded for only 93 out of 372 observations (25%). This substantially limited sample size for all three research questions:

- Q1 (Agreement): $n = 72$ pairs
- Q2 (Adherence): Limited observations
- Q3 (Hormone patterns): $n = \sim 70$ with valid times

While this methodological choice ensures we followed the investigator's specifications, it dramatically reduced statistical power compared to using all available data. Future studies should prioritize complete wake time recording.

2. **Missing Data:** Beyond wake times, there were additional missing data due to cap malfunctions and insufficient sample quality. Missing MEMs times and hormone measurements may have biased results toward more compliant or experienced users.

3. **Sample Size:** With 31 subjects and variable numbers of complete observations per analysis, power to detect smaller effects may be limited, particularly for DHEA analyses where some subjects were excluded due to detection limit issues.
4. **Time Variability:** The lunch sampling time (Sample 3) was not standardized and varied by subject schedule, making it difficult to compare this timepoint across subjects and potentially affecting the decline rate estimates.
5. **Detection Limits:** DHEA measurements exactly at the upper detection limit (5.205 nmol/L) were excluded. Subjects with repeated detection-limit values were excluded entirely from hormone analyses, which may have removed individuals with genuinely high DHEA levels.
6. **Adherence Measurement:** Our adherence thresholds (7.5 and 15 minutes) were chosen based on investigator input, but the biological significance of these windows for hormone measurement accuracy is unclear and may vary by sampling time.
7. **Model Assumptions:** The piecewise linear model assumes a sharp transition at 30 minutes, which may not capture the true biological curve perfectly. However, this specification directly addresses the investigator's questions about the awakening response and subsequent decline.
8. **Log Transformation:** We used log-transformed hormone values for analysis, which provides multiplicative interpretation and stabilizes variance. This means our estimates represent percentage changes rather than absolute differences in nmol/L units.

Conclusions

The SPIT booklet device demonstrates promise for measuring diurnal cortisol and DHEA patterns in naturalistic settings. While the agreement between booklet and electronic time recordings was limited (with substantial systematic bias and wide limits of agreement), electronic monitoring provides important objective verification of sampling times that may not be reliably self-reported.

The device successfully captured diurnal decline patterns for both cortisol and DHEA, with statistically significant changes detected in the post-awakening period. While protocol adherence was moderate, this is typical for home-based saliva collection studies and did not prevent detection of the expected diurnal rhythms.

The piecewise linear mixed model approach proved effective for characterizing both the initial awakening response and the subsequent decline, directly addressing the investigator's research questions. These findings support the use of the SPIT booklet for research on the stress hormone system in naturalistic settings.

Reproducible Research Information

Data and Code Location

All data files and statistical code for this analysis are available at: [Your GitHub Repository URL]

Key files:

- `DataAnalysis_FINAL.Rmd` - This R Markdown analysis file (in Code/ folder)
- `Project0_Clean_v2.csv`
- Dataset provided by investigator (in DataRaw/ folder)

Project 0 directory structure:

```
Project 0/
  Background/
  Code/
    - DataAnalysis_FINAL.Rmd          # This analysis file
```

```

DataRaw/
  - Project0_Clean_v2.csv          # Raw data from investigator
DataProcessed/
  - Project0_Processed.csv        # Cleaned dataset with calculated variables
Outputs/
  - missing_summary.csv
  - q1_agreement_lmm_fixed_effects.csv
  - q1_bland_altman_stats.csv
  - q2_adherence_summary.csv
  - q2_adherence_by_sample_memes.csv
  - q2_adherence_by_sample_booklet.csv
  - q3_dhea_detection_limit_by_subject.csv
  - q3_descriptive_by_sample.csv
  - q3_cortisol_piecewise_lmm_fixed_effects.csv
  - q3_cortisol_interpretable_effects.csv
  - q3_dhea_piecewise_lmm_fixed_effects.csv
  - q3_dhea_interpretable_effects.csv
Reports/
  ReportPlots/
    - Figure1_Agreement.png        # Q1: Scatter + Bland-Altman plots
    - Figure2_Adherence.png        # Q2: Deviation histograms
    - Figure3_HormonePatterns.png   # Q3: Cortisol & DHEA patterns
  ReportTables/
    - Table1_Descriptives.csv      # gtsummary table (main)
    - Table1_Descriptives.html     # HTML version
    - Table1_Descriptives.tex      # LaTeX version
    - Table1_SampleSize.csv        # Sample size summary
    - Table1_SampleSize.tex        # LaTeX version
    - Table2_Adherence.csv        # Adherence rates
    - Table2_Adherence.tex        # LaTeX version
    - Table3_HormoneModels.csv     # Model results
    - Table3_HormoneModels.tex     # LaTeX version

```

Note: All paths in the RMD are relative (../) from the Code/ folder.

Report Tables and Figures Summary

For the written report (6 tables/figures total):

1. **Table 1:** Study Characteristics and Descriptive Statistics (gtsummary)
 - Professional descriptive statistics table showing:
 - Mean (SD) for continuous variables (Cortisol, DHEA)
 - N (%) for categorical variables (Collection Sample, Day)
 - Missingness for all variables
 - Saved as both HTML and CSV formats
 - Additional sample size summary table provided
2. **Figure 1:** Agreement Between Booklet and MEMs Times
 - Scatter plot with regression line showing correlation
 - Bland-Altman plot with limits of agreement
 - Demonstrates systematic bias and variability between methods
3. **Table 2:** Protocol Adherence Rates

- Adherence within ± 7.5 and ± 15 minute windows
- Stratified by MEMs (primary) and Booklet (secondary) methods
- Shows compliance with scheduled sampling times

4. **Figure 2:** Distribution of Deviations from Scheduled Times

- Histograms showing actual vs. expected sampling times
- Separate panels for 30-minute and 10-hour samples
- Visualizes adherence patterns across timepoints

5. **Table 3:** Hormone Pattern Model Results

- Combined piecewise linear mixed model estimates
- Cortisol and DHEA results side-by-side for comparison
- Includes estimates, SE, t-statistics, p-values, and 95% CIs

6. **Figure 3:** Hormone Levels Over Time with Model Fits

- Four-panel plot (2x2 grid)
- Both log-scale and original-scale for each hormone
- Model prediction curves overlaid on observed data
- Shows diurnal patterns and model fit quality

All report materials are saved in `Reports/ReportPlots/` and `Reports/ReportTables/` folders.

Session Information

```
sessionInfo()
```

```
## R version 4.4.1 (2024-06-14 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26200)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/Denver
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics   grDevices  utils      datasets   methods    base
##
## other attached packages:
## [1] gtsummary_2.0.2     scales_1.3.0       gridExtra_2.3
## [4] broom.mixed_0.2.9.6 lmerTest_3.1-3     lme4_1.1-35.3
## [7] Matrix_1.7-1        broom_1.0.6       kableExtra_1.4.0
## [10] knitr_1.50        lubridate_1.9.3   forcats_1.0.0
```

```

## [13] stringr_1.5.1      dplyr_1.1.4        purrr_1.0.2
## [16] readr_2.1.5        tidyverse_2.0.0    tibble_3.2.1
## [19] ggplot2_3.5.1      tidyverse_2.0.0

##
## loaded via a namespace (and not attached):
## [1] tidyselect_1.2.1    viridisLite_0.4.2  farver_2.1.2
## [4] fastmap_1.2.0      digest_0.6.35     timechange_0.3.0
## [7] lifecycle_1.0.4     magrittr_2.0.3     compiler_4.4.1
## [10] rlang_1.1.6        sass_0.4.9       tools_4.4.1
## [13] utf8_1.2.4        yaml_2.3.10     gt_0.11.0
## [16] labeling_0.4.3    bit_4.0.5       xml2_1.3.6
## [19] withr_3.0.2       numDeriv_2016.8-1.1 grid_4.4.1
## [22] fansi_1.0.6       colorspace_2.1-0   future_1.34.0
## [25] globals_0.16.3    MASS_7.3-64     cli_3.6.5
## [28] rmarkdown_2.27     crayon_1.5.2    ragg_1.5.0
## [31] generics_0.1.3    rstudioapi_0.16.0 tzdb_0.4.0
## [34] commonmark_1.9.1  minqa_1.2.7     splines_4.4.1
## [37] parallel_4.4.1    vctrs_0.6.5     boot_1.3-30
## [40] hms_1.1.3        bit64_4.0.5    listenv_0.9.1
## [43] systemfonts_1.1.0 glue_1.7.0      parallelly_1.41.0
## [46] nlptr_2.0.3       codetools_0.2-20  stringi_1.8.4
## [49] gtable_0.3.5     munsell_0.5.1   furrr_0.3.1
## [52] pillar_1.9.0     htmltools_0.5.8.1 R6_2.5.1
## [55] textshaping_0.4.0 vroom_1.6.5     evaluate_1.0.5
## [58] lattice_0.22-6   markdown_1.12   backports_1.5.0
## [61] cards_0.2.2      Rcpp_1.0.12     svglite_2.1.3
## [64] nlme_3.1-164     mgcv_1.9-1     xfun_0.52
## [67] fs_1.6.6         pkgconfig_2.0.3

```

End of Report