Cortisol

Emily Kokesh

2026-01-26

Data loading and cleaning, missingness analysis

```
# load libraries

library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.5.2
```

```
## Warning: package 'ggplot2' was built under R version 4.5.2
```

```
## Warning: package 'tidyr' was built under R version 4.5.2
```

```
## Warning: package 'readr' was built under R version 4.5.2
```

```
## Warning: package 'purrr' was built under R version 4.5.2
```

```
## Warning: package 'stringr' was built under R version 4.5.2
```

```
## Warning: package 'forcats' was built under R version 4.5.2
```

```
## Warning: package 'lubridate' was built under R version 4.5.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.6
## v forcats   1.0.1     v stringr   1.6.0
## v ggplot2   4.0.1     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.2.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
```

```
library(lme4)
```

```
## Warning: package 'lme4' was built under R version 4.5.2
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```r
library(lmerTest)
```

```
## Warning: package 'lmerTest' was built under R version 4.5.2
```

```
##
## Attaching package: 'lmerTest'
##
## The following object is masked from 'package:lme4':
##
##     lmer
##
## The following object is masked from 'package:stats':
##
##     step
```

```r
library(lubridate)
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.5.2
```

```r
library(broom.mixed)
```

```
## Warning: package 'broom.mixed' was built under R version 4.5.2
```

```r
# load data

path <- "C:/Users/kokeshem/Spring 2026/Advanced Statistical Methods/Project0_Clean_v2.cs
df <- read.csv(path, na.strings = c("", "NA"))


# data cleaning, renaming, trimming


df <- df %>%
  rename(
    Booklet_Clock_Time = Booket..Clock.Time,
    MEMs_Clock_Time = MEMs..Clock.Time,
    Wake_Time = Sleep.Diary.reported.wake.time,
    Cortisol_nmol = Cortisol..nmol.L.,
    DHEA_nmol = DHEA..nmol.L.
  ) %>%
  #remove unnecessary columns for analysis
  select(
    SubjectID,
    Collection.Date,
    Collection.Sample,
    DAYNUMB,
    Booklet_Clock_Time,
    MEMs_Clock_Time,
```

```
    Wake_Time,
    Cortisol_nmol,
    DHEA_nmol
  )


#fill wake time so each sample per day has the same wake time
df <- df %>%
  group_by(SubjectID, Collection.Date) %>%
  fill(Wake_Time, .direction = "downup") %>%
  ungroup()


#helper function
to_mins <- function(time_str) {
  t <- hm(time_str, quiet = TRUE)
  return(as.numeric(t) / 60)
}


#calculate minutes since waking
df <- df %>%
  mutate(
    Wake_Min = to_mins(Wake_Time),
    Booklet_Clock_Min = to_mins(Booklet_Clock_Time),
```

```
    MEMs_Clock_Min = to_mins(MEMs_Clock_Time),


    #final analysis variables

    Booklet_Min_Since_Wake = Booklet_Clock_Min - Wake_Min,

    MEMs_Min_Since_Wake = MEMs_Clock_Min - Wake_Min

  )


#check simplified structure

str(df)
```

```
## tibble [372 x 14] (S3: tbl_df/tbl/data.frame)

##  $ SubjectID             : int [1:372] 3012 3012 3012 3012 3012 3012 3012 3012 3012 3

##  $ Collection.Date       : chr [1:372] "10/2/2018" "10/2/2018" "10/2/2018" "10/2/2018

##  $ Collection.Sample     : int [1:372] 1 2 3 4 1 2 3 4 1 2 ...

##  $ DAYNUMB               : int [1:372] 1 1 1 1 2 2 2 2 3 3 ...

##  $ Booklet_Clock_Time    : chr [1:372] "8:54" "9:38" "12:31" "19:38" ...

##  $ MEMs_Clock_Time       : chr [1:372] "8:55" "9:38" "12:30" "19:38" ...

##  $ Wake_Time             : chr [1:372] "8:54" "8:54" "8:54" "8:54" ...

##  $ Cortisol_nmol         : num [1:372] 3.918 7.146 2.318 0.579 3.835 ...

##  $ DHEA_nmol             : num [1:372] 1.561 0.53 0.328 0.153 1.844 ...

##  $ Wake_Min              : num [1:372] 534 534 534 534 440 440 440 440 398 398 ...

##  $ Booklet_Clock_Min     : num [1:372] 534 578 751 1178 440 ...

##  $ MEMs_Clock_Min        : num [1:372] 535 578 750 1178 441 ...
```

```
##  $ Booklet_Min_Since_Wake: num [1:372] 0 44 217 644 0 37 370 661 0 34 ...
##  $ MEMs_Min_Since_Wake   : num [1:372] 1 44 216 644 1 37 370 660 1 35 ...
```

```r
view(df)


#create missing data frame
missing_table <- data.frame(
  variable = c("mems cap time", "booklet time", "cortisol level", "dhea level"),
  missing_count = c(
    sum(is.na(df$MEMs_Clock_Time)),
    sum(is.na(df$Booklet_Clock_Time)),
    sum(is.na(df$Cortisol_nmol)),
    sum(is.na(df$DHEA_nmol))
  ),
  percentage = c(
    mean(is.na(df$MEMs_Clock_Time)) * 100,
    mean(is.na(df$Booklet_Clock_Time)) * 100,
    mean(is.na(df$Cortisol_nmol)) * 100,
    mean(is.na(df$DHEA_nmol)) * 100
  )
)


#fromat and print missingness table
```

```r
missing_table %>%
  kable(
    col.names = c("variable", "missing count (n)", "percentage (%)"),
    digits = 1,
    caption = "summary of missing data across primary study variables"
  )
```

Table 1: summary of missing data across primary study variables

variable	missing count (n)	percentage (%)
mems cap time	61	16.4
booklet time	35	9.4
cortisol level	5	1.3
dhea level	5	1.3

Question 1: Agreement Analysis

```r
#filter data complete pairs of Booklet and MEM times
df_q1 <- df %>%
  filter(!is.na(Booklet_Min_Since_Wake) & !is.na(MEMs_Min_Since_Wake)) %>%
  mutate(bias = Booklet_Min_Since_Wake - MEMs_Min_Since_Wake)
#check df_q1
```

```r
view(df_q1)
```

```r
#calculate correlation
correlation_result <- cor(df_q1$Booklet_Min_Since_Wake, df_q1$MEMs_Min_Since_Wake)
print(paste("correlation between booklet and cap:", round(correlation_result, 4)))
```

```
## [1] "correlation between booklet and cap: 0.9927"
```

```r
#calculate bias summary
bias_summary <- summary(df_q1$bias)
print("summary statistics for bias:")
```

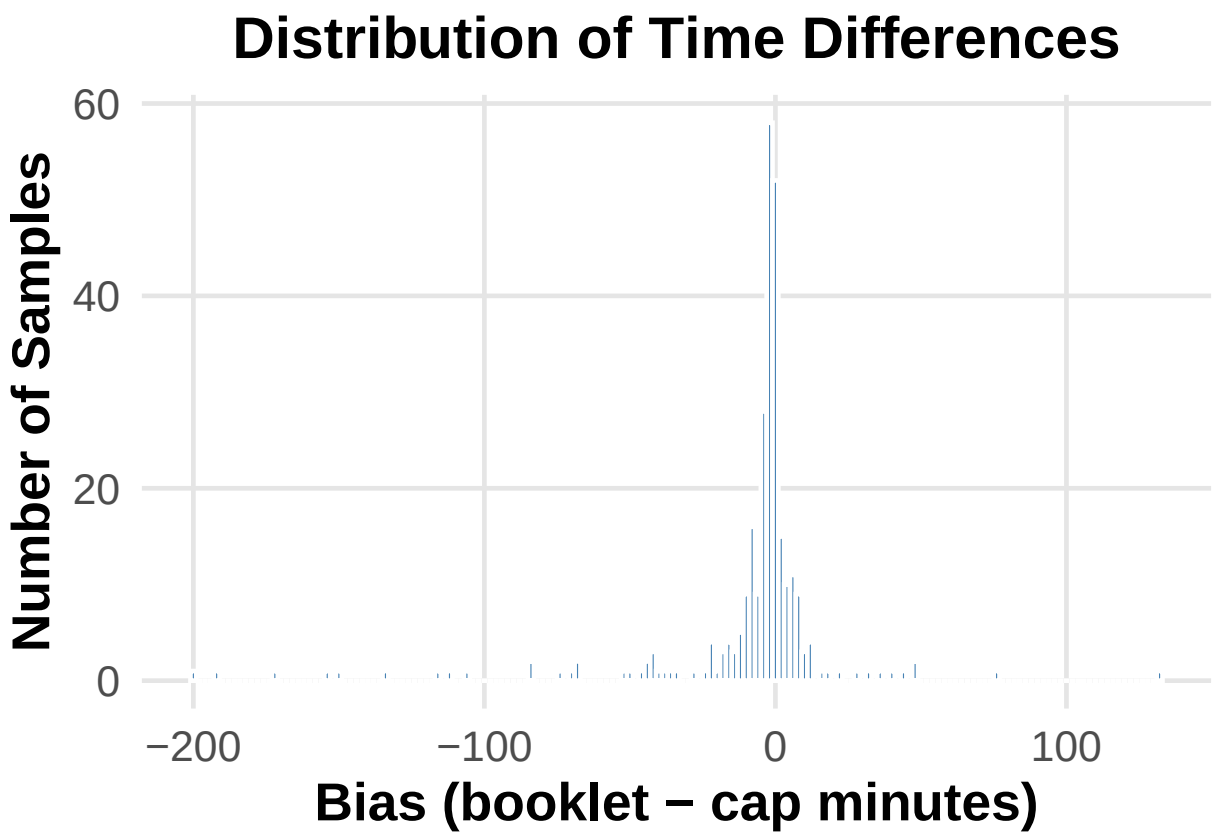
```
## [1] "summary statistics for bias:"
```

```r
print(bias_summary)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -200.000   -7.000   -1.000   -7.712    1.000  133.000
```

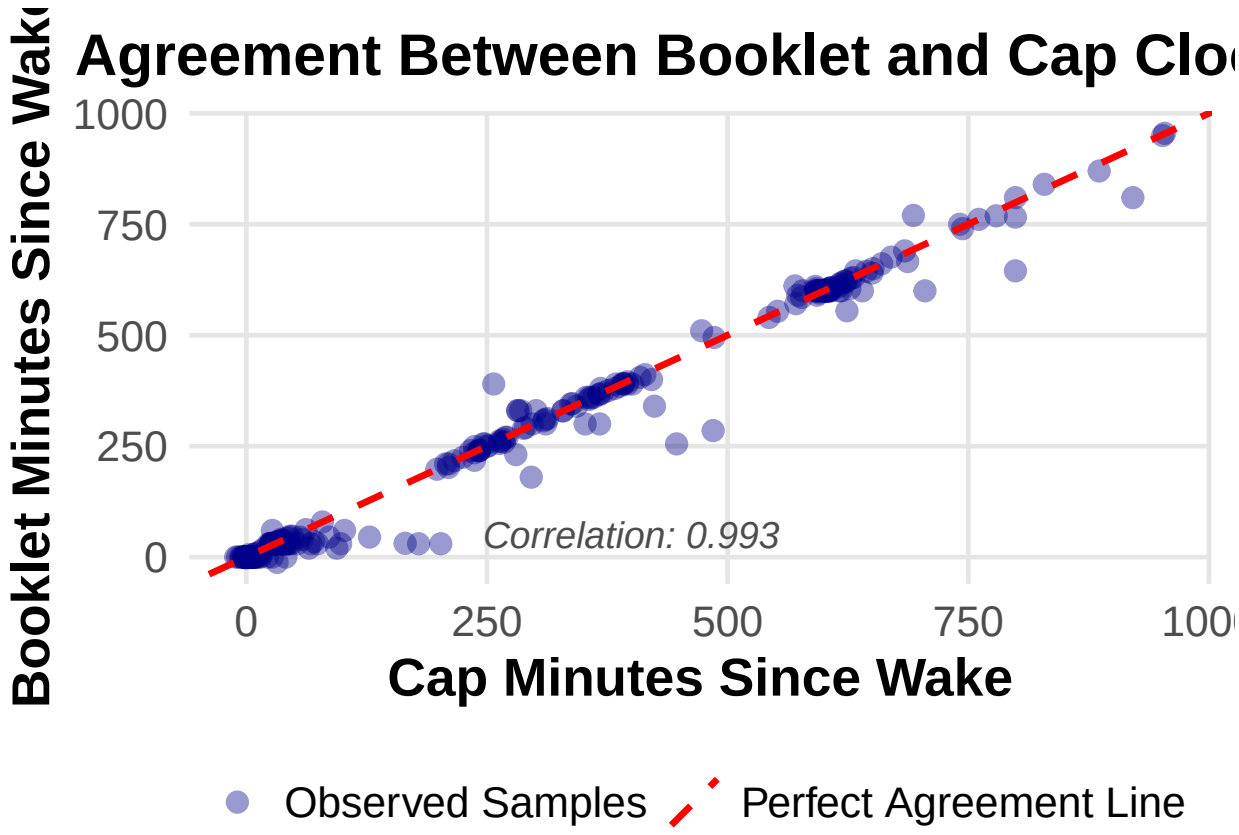
```r
#make presentation theme
presentation_theme <- theme_minimal(base_size = 18) +
  theme(
    plot.title = element_text(size = 22, face = "bold", hjust = 0.5),
    axis.title = element_text(size = 20, face = "bold"),
```

```
    axis.text = element_text(size = 16),
    legend.title = element_text(size = 18),
    legend.text = element_text(size = 16),
    panel.grid.major = element_line(color = "grey90"),
    panel.grid.minor = element_blank()
  )


#visualize bias distribution
#increased binwidth and clearer labels for the histogram
ggplot(df_q1, aes(x = bias)) +
  geom_histogram(binwidth = 2, fill = "steelblue", color = "white") +
  presentation_theme +
  labs(
    title = "Distribution of Time Differences",
    x = "Bias (booklet - cap minutes)",
    y = "Number of Samples"
  )
```




```
# prepare the correlation label

cor_label <- paste0("Correlation: ", round(correlation_result, 3))



# visualize agreement scatterplot

ggplot(df_q1) +

  geom_point(aes(x = MEMs_Min_Since_Wake, y = Booklet_Min_Since_Wake, color = "Observed

             alpha = 0.4, size = 3) +

  geom_abline(aes(intercept = 0, slope = 1, color = "Perfect Agreement Line"),

              linetype = "dashed", linewidth = 1.2) +

  annotate("text", x = 400, y = 50, label = cor_label,
```

```
          size = 5, fontface = "italic", color = "grey30") +
scale_color_manual(name = NULL,
                     values = c("Observed Samples" = "darkblue", "Perfect Agreement Line
presentation_theme +
theme(
  legend.position = "bottom",
  legend.direction = "horizontal",
  legend.box = "horizontal",
  legend.key = element_blank()
) +
labs(
  title = "Agreement Between Booklet and Cap Clocks",
  x = "Cap Minutes Since Wake",
  y = "Booklet Minutes Since Wake"
)
```




```
#linear mixed model: random intercept only

# this accounts for the fact that each subject has a different baseline bias

model_intercept <- lmer(Booklet_Min_Since_Wake ~ MEMs_Min_Since_Wake +

                          (1 | SubjectID),

                        data = df_q1)

summary(model_intercept)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [

## lmerModLmerTest]

## Formula: Booklet_Min_Since_Wake ~ MEMs_Min_Since_Wake + (1 | SubjectID)

##    Data: df_q1
```

```
##
## REML criterion at convergence: 2787.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.9420  0.0024  0.1588  0.2635  4.3866
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  SubjectID (Intercept)  44.11    6.642
##  Residual              989.60   31.458
## Number of obs: 285, groups:  SubjectID, 31
##
## Fixed effects:
##                      Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)          -6.50779    2.91608  66.29918  -2.232    0.029 *
## MEMs_Min_Since_Wake   0.99494    0.00706 262.91666 140.931   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## MEMs_Mn_S_W -0.641
```

```r
#linear mixed model: random intercept and slope
#predictor is centered to help convergence
df_q1 <- df_q1 %>%
  mutate(MEMs_centered = MEMs_Min_Since_Wake - mean(MEMs_Min_Since_Wake, na.rm = TRUE))


agreement_model_final <- lmer(Booklet_Min_Since_Wake ~ MEMs_centered +
                                (1 + MEMs_centered | SubjectID),
                              data = df_q1,
                              control = lmerControl(optimizer = "nlminbwrap",
                                                    optCtrl = list(maxfun = 1e5)))

## boundary (singular) fit: see help('isSingular')

summary(agreement_model_final)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Booklet_Min_Since_Wake ~ MEMs_centered + (1 + MEMs_centered |
##     SubjectID)
##    Data: df_q1
## Control: lmerControl(optimizer = "nlminbwrap", optCtrl = list(maxfun = 1e+05))
##
## REML criterion at convergence: 2781.8
##
```

```
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.8247 -0.0192  0.1510  0.2595  4.3738
##
## Random effects:
##  Groups    Name          Variance  Std.Dev. Corr
##  SubjectID (Intercept)   6.217e+01  7.88487
##            MEMs_centered 4.638e-04  0.02154 1.00
##  Residual                9.382e+02 30.62996
## Number of obs: 285, groups:  SubjectID, 31
##
## Fixed effects:
##                Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)   2.550e+02  2.322e+00 3.220e+01   109.8   <2e-16 ***
## MEMs_centered 9.960e-01  7.955e-03 4.755e+01   125.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## MEMs_centrd 0.311
## optimizer (nlminbwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```r
#compare to see if the random slope improves fit

anova(model_intercept, agreement_model_final)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: df_q1
## Models:
## model_intercept: Booklet_Min_Since_Wake ~ MEMs_Min_Since_Wake + (1 | SubjectID)
## agreement_model_final: Booklet_Min_Since_Wake ~ MEMs_centered + (1 + MEMs_centered |
##                        npar    AIC    BIC  logLik -2*log(L)  Chisq Df Pr(>Chisq)
## model_intercept           4 2791.0 2805.6 -1391.5    2783.0
## agreement_model_final     6 2789.3 2811.2 -1388.7    2777.3 5.6897  2    0.05814
##
## model_intercept
## agreement_model_final .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 2: Adherence Analysis

```r
#make new column with target time, calculate difference between actual (MEM) time and

df_q2 <- df %>%
  filter(Collection.Sample %in% c(2, 4)) %>%
```

```
  mutate(
    Target = ifelse(Collection.Sample == 2, 30, 600),
    Time_Diff = MEMs_Min_Since_Wake - Target,
    Abs_Diff = abs(Time_Diff)
  ) %>%
  filter(!is.na(Time_Diff))


#create adherence categories (within 7.5 or 15 minutes)
df_q2 <- df_q2 %>%
  mutate(
    In_7.5_Window = Abs_Diff <= 7.5,
    In_15_Window = Abs_Diff <= 15
  )


#proportion of adherence table
adherence_proportions <- df_q2 %>%
  group_by(Collection.Sample) %>%
  summarise(
    Total_n = n(),
    Strict_Adherence_n = sum(In_7.5_Window),
    Strict_Rate = (Strict_Adherence_n / Total_n) * 100,
    Lenient_Adherence_n = sum(In_15_Window),
```

```
    Lenient_Rate = (Lenient_Adherence_n / Total_n) * 100
  )


kable(adherence_proportions, digits = 1, caption = "Proportions of Adherent Samples")
```

Table 2: Proportions of Adherent Samples

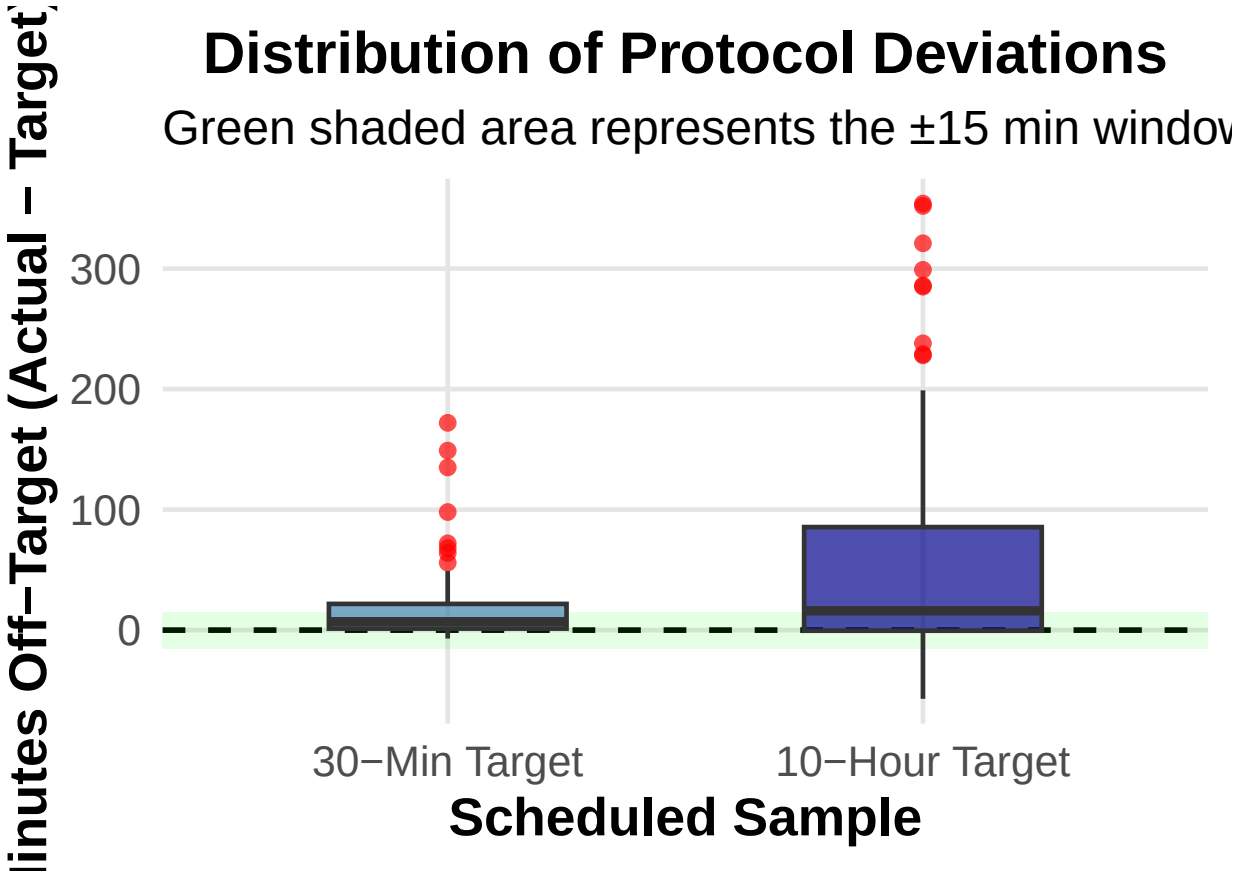
Collection.Sample	Total_n	Strict_Adherence_n	Strict_Rate	Lenient_Adherence_n	Lenient_Rate
2	70	37	52.9	50	71.4
4	83	27	32.5	33	39.8

```
#descriptive statistics table

timing_descriptives <- df_q2 %>%
  group_by(Collection.Sample) %>%
  summarise(
    Mean_Diff = mean(Time_Diff),
    Median_Diff = median(Time_Diff),
    SD_Diff = sd(Time_Diff),
    Min_Diff = min(Time_Diff),
    Max_Diff = max(Time_Diff)
  )


kable(timing_descriptives, digits = 1, caption = "Descriptive Statistics of Timing Devia
```

Table 3: Descriptive Statistics of Timing Deviations (Minutes)

Collection.Sample	Mean_Diff	Median_Diff	SD_Diff	Min_Diff	Max_Diff
2	20.1	7	35.0	-7	172
4	58.9	16	99.1	-57	354

```
#visualize the spread of timing deviations

ggplot(df_q2, aes(x = factor(Collection.Sample), y = Time_Diff, fill = factor(Collectio

  geom_hline(yintercept = 0, linetype = "dashed", color = "black", linewidth = 1) +

  annotate("rect", xmin = -Inf, xmax = Inf, ymin = -15, ymax = 15, alpha = .1, fill = "g

  geom_boxplot(width = 0.5, outlier.color = "red", alpha = 0.7) +

  scale_x_discrete(labels = c("2" = "30-Min Target", "4" = "10-Hour Target")) +

  scale_fill_manual(values = c("steelblue", "darkblue")) +

  presentation_theme +

  theme(legend.position = "none") +

  labs(

    title = "Distribution of Protocol Deviations",

    subtitle = "Green shaded area represents the ±15 min window",

    x = "Scheduled Sample",

    y = "Minutes Off-Target (Actual - Target)"

  )
```



Question 3: Changes in DHEA and Cortisol throughout the day

```
#create cleaned cortisol dataframe by excluding values over 80 nmol/L as per investiga

df_cort_clean <- df %>%

  filter(!is.na(Cortisol_nmol)) %>%

  filter(Cortisol_nmol <= 80)



#create cleaned dhea dataframe

#identify and remove subjects with multiple measures at the detection limit (5.205)

subjects_to_exclude_dhea <- df %>%
```

```r
  filter(!is.na(DHEA_nmol)) %>%
  group_by(SubjectID) %>%
  summarise(limit_count = sum(DHEA_nmol >= 5.205)) %>%
  filter(limit_count > 1) %>%
  pull(SubjectID)


#filter by removing limit values and excluded subjects
df_dhea_clean <- df %>%
  filter(!is.na(DHEA_nmol)) %>%
  filter(DHEA_nmol < 5.205) %>%
  filter(!(SubjectID %in% subjects_to_exclude_dhea))


# Print summary of cleaning actions
print(paste("Number of Samples removed for Cortisol (>80):", sum(df$Cortisol_nmol > 80,
```

```
## [1] "Number of Samples removed for Cortisol (>80): 1"
```

```r
print(paste("SubjectID excluded from DHEA df:",
            paste(subjects_to_exclude_dhea, collapse = ", ")))
```

```
## [1] "SubjectID excluded from DHEA df: 3037"
```

```r
#make piecewise time variable for cortisol data and DHEA data


#cortisol
df_cort_clean <- df_cort_clean %>%
  mutate(
    #slope 1: slope from time 0 0 to 30 minutes
    time_rise = pmin(MEMs_Min_Since_Wake, 30),


    #slope 2: starts at 0, only begins increasing after the 30-min mark.
    time_decline = pmax(0, MEMs_Min_Since_Wake - 30)
  )


#dhea
df_dhea_clean <- df_dhea_clean %>%
  mutate(
    #slope 1
    time_rise = pmin(MEMs_Min_Since_Wake, 30),
    #slope 2
    time_decline = pmax(0, MEMs_Min_Since_Wake - 30)
  )


#piecewise linear mixed models
```

```r
#cortisol- log transformations added to outcome variables
cort_model <- lmer(log(Cortisol_nmol) ~ time_rise + time_decline + (1 | SubjectID),
                   data = df_cort_clean)
summary(cort_model)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: log(Cortisol_nmol) ~ time_rise + time_decline + (1 | SubjectID)
##    Data: df_cort_clean
##
## REML criterion at convergence: 756.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.8839 -0.5453  0.0196  0.5654  3.7595
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  SubjectID (Intercept) 0.07237  0.2690
##  Residual              0.57673  0.7594
## Number of obs: 309, groups:  SubjectID, 31
##
```

```
## Fixed effects:
##                Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)   1.705e+00  1.109e-01  1.595e+02  15.365   <2e-16 ***
## time_rise     6.235e-03  4.417e-03  2.856e+02   1.412    0.159
## time_decline -2.137e-03  1.918e-04  2.838e+02 -11.138   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) tim_rs
## time_rise   -0.718
## time_declin  0.057 -0.520
```

```r
#DHEA model
dhea_model <- lmer(log(DHEA_nmol) ~ time_rise + time_decline + (1 | SubjectID),
                   data = df_dhea_clean)
summary(dhea_model)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: log(DHEA_nmol) ~ time_rise + time_decline + (1 | SubjectID)
##    Data: df_dhea_clean
##
## REML criterion at convergence: 642.3
```

```
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.3529 -0.5521 -0.0050  0.5730  2.4214
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  SubjectID (Intercept) 0.2631   0.5129
##  Residual              0.3756   0.6129
## Number of obs: 299, groups:  SubjectID, 30
##
## Fixed effects:
##                Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)   3.353e-01  1.252e-01  6.634e+01   2.679  0.00931 **
## time_rise    -2.151e-02  3.653e-03  2.707e+02  -5.889 1.15e-08 ***
## time_decline -1.527e-03  1.579e-04  2.703e+02  -9.671  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) tim_rs
## time_rise   -0.527
## time_declin  0.041 -0.522
```

```r
# tidy models, include 95% confidence intervals
cort_tidy_ci <- tidy(cort_model, effects = "fixed", conf.int = TRUE, conf.level = 0.95)
  mutate(Hormone = "Cortisol")


dhea_tidy_ci <- tidy(dhea_model, effects = "fixed", conf.int = TRUE, conf.level = 0.95)
  mutate(Hormone = "DHEA")


#prepare for display in table
hormone_final_table <- bind_rows(cort_tidy_ci, dhea_tidy_ci) %>%
  mutate(
    term = case_when(
      term == "(Intercept)" ~ "Wake Level (Intercept)",
      term == "time_rise" ~ "Morning Change (0-30 min)",
      term == "time_decline" ~ "Daily Decline (30+ min)"
    ),
    # Combine Lower and Upper into one "95% CI" column for a cleaner look
    conf_interval = paste0("[", round(conf.low, 4), ", ", round(conf.high, 4), "]"),
    estimate = round(estimate, 4),
    p.value = ifelse(p.value < 0.001, "< 0.001", round(p.value, 3))
  ) %>%
  select(Hormone, term, estimate, conf_interval, p.value)
```

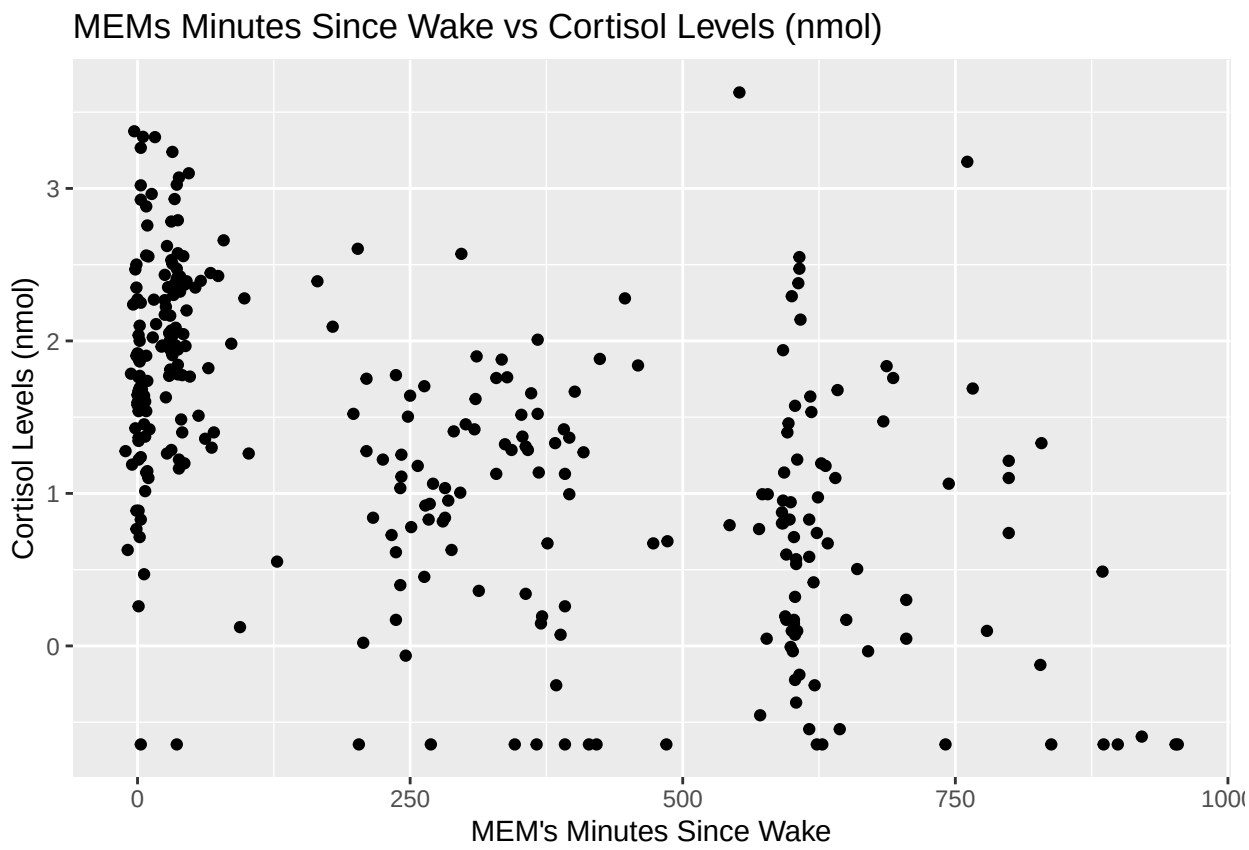
```
#make table
hormone_final_table %>%
  kable(
    col.names = c("Hormone", "Model Parameter", "Estimate (Beta)", "95% CI", "p-value"),
    caption = "Diurnal Hormone Trajectories with 95% Confidence Intervals",
    align = "llccc"
  )
```

Table 4: Diurnal Hormone Trajectories with 95% Confidence Intervals

Hormone	Model Parameter	Estimate (Beta)	95% CI	p-value
Cortisol	Wake Level (Intercept)	1.7046	[1.4855, 1.9236]	< 0.001
Cortisol	Morning Change (0-30 min)	0.0062	[-0.0025, 0.0149]	0.159
Cortisol	Daily Decline (30+ min)	-0.0021	[-0.0025, -0.0018]	< 0.001
DHEA	Wake Level (Intercept)	0.3353	[0.0854, 0.5852]	0.009
DHEA	Morning Change (0-30 min)	-0.0215	[-0.0287, -0.0143]	< 0.001
DHEA	Daily Decline (30+ min)	-0.0015	[-0.0018, -0.0012]	< 0.001

```r
#plot

ggplot(df_cort_clean, aes(x=MEMs_Min_Since_Wake, y=log(Cortisol_nmol)))+

  geom_point()+

  labs(

    title = "MEMs Minutes Since Wake vs Cortisol Levels (nmol)",

    x = "MEM's Minutes Since Wake",

    y = "Cortisol Levels (nmol)")
```

```
## Warning: Removed 57 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```
ggplot(df_dhea_clean, aes(x=MEMs_Min_Since_Wake, y=log(DHEA_nmol)))+
  geom_point()+
  labs(
    title = "MEMs Minutes Since Wake vs DHEA Levels (nmol)",
    x = "MEM's Minutes Since Wake",
    y = "DHEA Levels (nmol)")
```

```
## Warning: Removed 54 rows containing missing values or values outside the scale range
## (`geom_point()`).
```