Focus Timer Visualization Study - Subjective Outcomes Analysis

DATASCI 241 Final Project - Section 3, Group 4

Ariel Gholar, Colton Hester, Jeremy Liu, Nitya Sree Cheera

2025-12-08

Contents

1. Setup and Libraries

```
# Core data manipulation
library(jsonlite)
library(tidyverse)

# Statistical testing
library(lmtest)    # coeftest for robust SEs
library(sandwich)  # vcovHC for heteroskedasticity-consistent SEs

# Tables and visualization
library(knitr)
library(kableExtra)
library(ggplot2)

# Set theme for plots
theme_set(theme_minimal())
```

2. Data Loading

2.1 Load JSON Data

```
# Load the exported experiment data
raw_data <- fromJSON("all_data_12072025.json", flatten = TRUE)

# Display export metadata
cat("Export Date:", raw_data$export_date, "\n")
```

```
## Export Date: 2025-12-08T04:46:47.563Z
```

```
cat("Total Participants:", raw_data$total_participants, "\n")
```

```
## Total Participants: 25
```

```r
cat("Total Sessions:", raw_data$total_sessions, "\n")
```

```
## Total Sessions: 32
```

```r
cat("Total Ratings:", raw_data$total_ratings, "\n")
```

```
## Total Ratings: 26
```

```r
cat("Total Post-Treatment Surveys:", raw_data$total_post_treatment_surveys, "\n")
```

```
## Total Post-Treatment Surveys: 12
```

2.2 Extract and Clean Data Frames

```r
# Extract sessions data
sessions <- raw_data$sessions %>%
  as_tibble() %>%
  select(
    session_id = id,
    participant_id = participantId,
    condition,
    session_number = sessionNumber,
    target_duration = targetDuration,
    completed_full = completedFullSession,
    created_at = createdAt,
    # Nested participant data
    condition_sequence = participant.conditionSequence,
    cohort = participant.cohort
  ) %>%
  mutate(
    condition = factor(condition, levels = c("COUNTDOWN", "HOURGLASS")),
    first_condition = map_chr(condition_sequence, ~ .x[1])
  )

# Extract baseline survey data
baseline <- raw_data$baseline %>%
  as_tibble() %>%
  select(
    baseline_id = id,
    participant_id = participantId,
    time_anxiety_score = timeAnxietyScore,
    typical_focus_duration = typicalFocusDuration,
    classes_enrolled = classesEnrolled,
    uses_timer_currently = usesTimerCurrently,
    preferred_timer_type = preferredTimerType,
    completed_at = completedAt
  )

# Extract post-session ratings (PRIMARY OUTCOME DATA)
```

```r
ratings <- raw_data$ratings %>%
  as_tibble() %>%
  select(
    rating_id = id,
    session_id = sessionId,
    perceived_stress = perceivedStress,
    ease_of_following = easeOfFollowing,
    subjective_focus_quality = subjectiveFocusQuality,
    comments,
    completed_at = completedAt
  )

# Extract post-treatment surveys
post_treatment <- raw_data$postTreatment %>%
  as_tibble() %>%
  select(
    survey_id = id,
    participant_id = participantId,
    preferred_timer = preferredTimer,
    qualitative_feedback = qualitativeFeedback,
    would_use_again = wouldUseAgain,
    recommend_to_others = recommendToOthers,
    completed_at = completedAt
  )

cat("\nData frames extracted:\n")
```

```
##
## Data frames extracted:
```

```r
cat("- Sessions:", nrow(sessions), "rows\n")
```

```
## - Sessions: 32 rows
```

```r
cat("- Baseline surveys:", nrow(baseline), "rows\n")
```

```
## - Baseline surveys: 25 rows
```

```r
cat("- Post-session ratings:", nrow(ratings), "rows\n")
```

```
## - Post-session ratings: 26 rows
```

```r
cat("- Post-treatment surveys:", nrow(post_treatment), "rows\n")
```

```
## - Post-treatment surveys: 12 rows
```

2.3 Join Data for Analysis

```r
# Join sessions with baseline covariates
sessions_with_baseline <- sessions %>%
  left_join(baseline, by = "participant_id")

# Join with post-session ratings
analysis_df <- sessions_with_baseline %>%
  left_join(ratings, by = "session_id")

# Check join results
cat("Analysis dataframe:", nrow(analysis_df), "rows\n")
```

```
## Analysis dataframe: 32 rows
```

```r
cat("Sessions with baseline data:", sum(!is.na(analysis_df$time_anxiety_score)), "\n")
```

```
## Sessions with baseline data: 32
```

```r
cat("Sessions with ratings:", sum(!is.na(analysis_df$perceived_stress)), "\n")
```

```
## Sessions with ratings: 26
```

```r
# Filter to sessions with ratings for primary analysis
rated_sessions <- analysis_df %>%
  filter(!is.na(perceived_stress))

cat("\nAnalysis sample (sessions with ratings):", nrow(rated_sessions), "\n")
```

```
##
## Analysis sample (sessions with ratings): 26
```

3. Outcome Variable Definitions

3.1 Primary Outcomes (Likert 1-5)

Our primary outcomes are subjective measures from post-session ratings:

- **perceived_stress**: "How stressful did you find this session?" (1 = Not at all, 5 = Extremely)
- **ease_of_following**: "How easy was it to follow the timer?" (1 = Very difficult, 5 = Very easy)
- **subjective_focus_quality**: "Rate your focus quality during this session" (1 = Very poor, 5 = Excellent)

```r
# Summary of primary outcomes by condition
rated_sessions %>%
  group_by(condition) %>%
  summarise(
    n = n(),
    stress_mean = mean(perceived_stress, na.rm = TRUE),
    stress_sd = sd(perceived_stress, na.rm = TRUE),
```

```
    ease_mean = mean(ease_of_following, na.rm = TRUE),
    ease_sd = sd(ease_of_following, na.rm = TRUE),
    quality_mean = mean(subjective_focus_quality, na.rm = TRUE),
    quality_sd = sd(subjective_focus_quality, na.rm = TRUE)
  ) %>%
  kable(
    caption = "Primary Outcome Variables by Condition",
    digits = 2,
    col.names = c("Condition", "N", "Stress M", "Stress SD",
                  "Ease M", "Ease SD", "Quality M", "Quality SD")
  ) %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Table 1: Primary Outcome Variables by Condition

Condition	N	Stress M	Stress SD	Ease M	Ease SD	Quality M	Quality SD
COUNTDOWN	13	2.46	0.97	4.31	0.85	3.38	1.12
HOURGLASS	13	2.23	0.93	3.23	1.30	3.62	1.12

3.2 Secondary Outcomes (Post-Treatment)

From the post-treatment survey:

- **preferred_timer**: Which timer did participant prefer (COUNTDOWN / HOURGLASS / NO_PREFERENCE)
- **would_use_again**: Would participant use their preferred timer again (TRUE/FALSE)
- **recommend_to_others**: Would participant recommend to others (TRUE/FALSE)

4. Exploratory Data Analysis

4.1 Sample Characteristics

```
# Baseline characteristics
baseline_summary <- baseline %>%
  summarise(
    n = n(),
    time_anxiety_mean = mean(time_anxiety_score, na.rm = TRUE),
    time_anxiety_sd = sd(time_anxiety_score, na.rm = TRUE),
    typical_focus_mean = mean(typical_focus_duration, na.rm = TRUE),
    typical_focus_sd = sd(typical_focus_duration, na.rm = TRUE),
    classes_mean = mean(classes_enrolled, na.rm = TRUE),
    classes_sd = sd(classes_enrolled, na.rm = TRUE),
    uses_timer_pct = mean(uses_timer_currently, na.rm = TRUE) * 100
  )

cat("Baseline Sample Characteristics (N =", baseline_summary$n, ")\n")
```

```
## Baseline Sample Characteristics (N = 25 )
```

```r
cat("Time Anxiety Score: M =", round(baseline_summary$time_anxiety_mean, 2),
    ", SD =", round(baseline_summary$time_anxiety_sd, 2), "(scale 1-5)\n")
```

```
## Time Anxiety Score: M = 2.52 , SD = 0.92 (scale 1-5)
```

```r
cat("Typical Focus Duration: M =", round(baseline_summary$typical_focus_mean, 1),
    ", SD =", round(baseline_summary$typical_focus_sd, 1), "minutes\n")
```

```
## Typical Focus Duration: M = 51.8 , SD = 53.7 minutes
```

```r
cat("Classes Enrolled: M =", round(baseline_summary$classes_mean, 2),
    ", SD =", round(baseline_summary$classes_sd, 2), "\n")
```

```
## Classes Enrolled: M = 1.6 , SD = 0.63
```

```r
cat("Currently Uses Timer:", round(baseline_summary$uses_timer_pct, 1), "%\n")
```

```
## Currently Uses Timer: 100 %
```

4.2 Distribution of Likert Responses

```r
# Reshape data for faceted plot
likert_long <- rated_sessions %>%
  select(session_id, condition, perceived_stress, ease_of_following, subjective_focus_quality) %>%
  pivot_longer(
    cols = c(perceived_stress, ease_of_following, subjective_focus_quality),
    names_to = "outcome",
    values_to = "rating"
  ) %>%
  mutate(
    outcome = factor(outcome,
                     levels = c("perceived_stress", "ease_of_following", "subjective_focus_quality"),
                     labels = c("Perceived Stress", "Ease of Following", "Subjective Focus Quality"))
  )

# Bar chart of response distributions
ggplot(likert_long, aes(x = factor(rating), fill = condition)) +
  geom_bar(position = "dodge", alpha = 0.8) +
  facet_wrap(~outcome, ncol = 1, scales = "free_y") +
  scale_fill_manual(values = c("COUNTDOWN" = "#1f77b4", "HOURGLASS" = "#ff7f0e")) +
  labs(
    title = "Distribution of Likert Responses by Condition",
    subtitle = "1 = lowest, 5 = highest",
    x = "Rating",
    y = "Count",
    fill = "Condition"
  ) +
  theme(legend.position = "bottom")
```

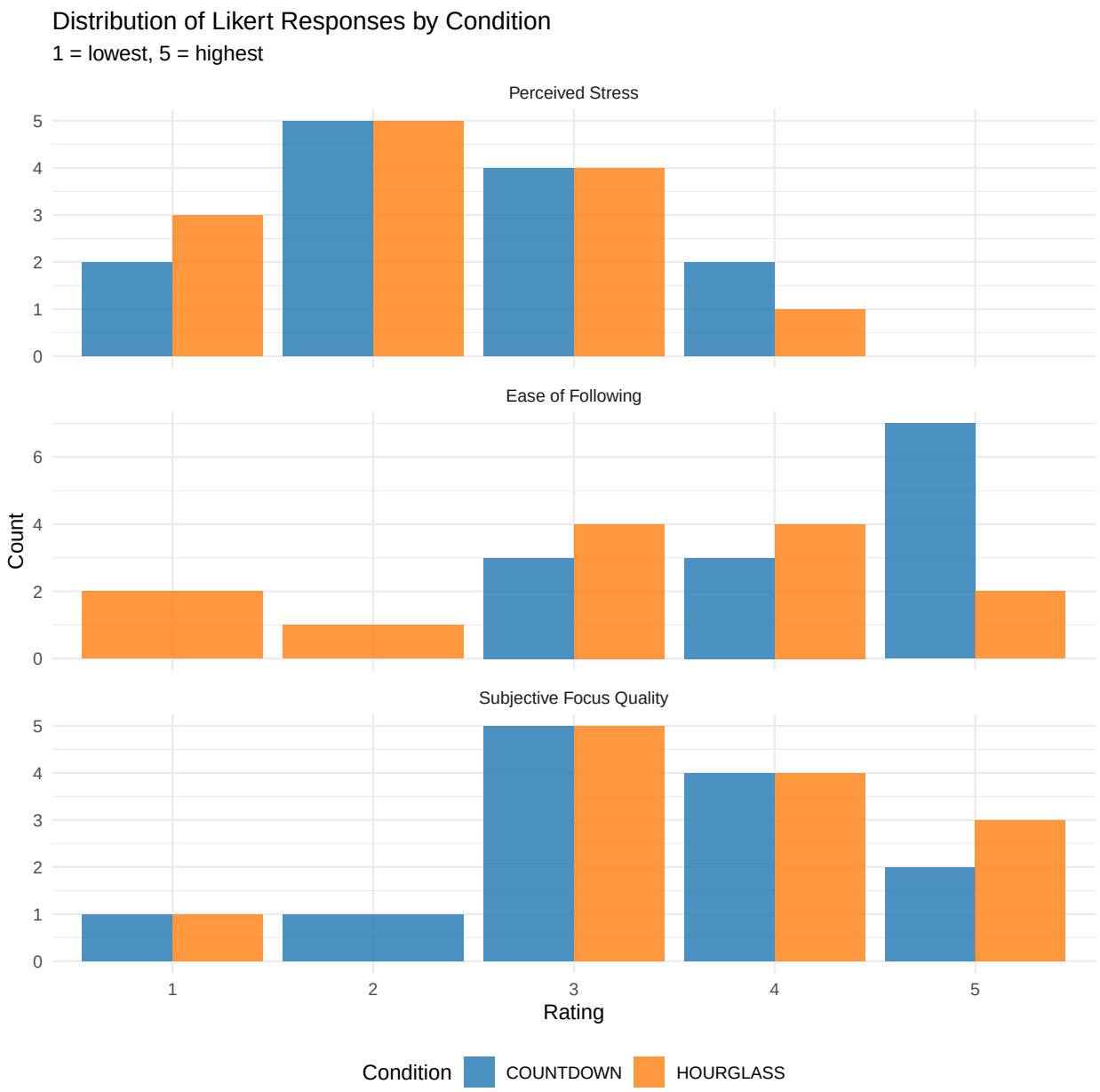

Figure 1: Distribution of Likert Responses by Condition

4.3 Check for Default Value Bias

Important methodological note: The survey UI pre-selects "3" as the default value. We check if this creates response bias.

```
# Count responses at each level
response_counts <- rated_sessions %>%
  summarise(
    stress_1 = sum(perceived_stress == 1),
    stress_2 = sum(perceived_stress == 2),
    stress_3 = sum(perceived_stress == 3),
    stress_4 = sum(perceived_stress == 4),
    stress_5 = sum(perceived_stress == 5),
    ease_1 = sum(ease_of_following == 1),
    ease_2 = sum(ease_of_following == 2),
    ease_3 = sum(ease_of_following == 3),
    ease_4 = sum(ease_of_following == 4),
    ease_5 = sum(ease_of_following == 5),
    quality_1 = sum(subjective_focus_quality == 1, na.rm = TRUE),
    quality_2 = sum(subjective_focus_quality == 2, na.rm = TRUE),
    quality_3 = sum(subjective_focus_quality == 3, na.rm = TRUE),
    quality_4 = sum(subjective_focus_quality == 4, na.rm = TRUE),
    quality_5 = sum(subjective_focus_quality == 5, na.rm = TRUE)
  )

# Calculate proportions
n_rated <- nrow(rated_sessions)
cat("Response Counts (N =", n_rated, "):\n\n")
```

```
## Response Counts (N = 26 ):
```

```
cat("Perceived Stress:\n")
```

```
## Perceived Stress:
```

```
cat("  1:", response_counts$stress_1, "(", round(100*response_counts$stress_1/n_rated, 1), "%)\n")
```

```
##   1: 5 ( 19.2 %)
```

```
cat("  2:", response_counts$stress_2, "(", round(100*response_counts$stress_2/n_rated, 1), "%)\n")
```

```
##   2: 10 ( 38.5 %)
```

```
cat("  3:", response_counts$stress_3, "(", round(100*response_counts$stress_3/n_rated, 1), "%) <-- defa
```

```
##   3: 8 ( 30.8 %) <-- default
```

```
cat("  4:", response_counts$stress_4, "(", round(100*response_counts$stress_4/n_rated, 1), "%)\n")
```

```
##   4: 3 ( 11.5 %)
```

```r
cat("  5:", response_counts$stress_5, "(", round(100*response_counts$stress_5/n_rated, 1), "%)\n")
```

```
##   5: 0 ( 0 %)
```

```r
cat("\nEase of Following:\n")
```

```
##
## Ease of Following:
```

```r
cat("  1:", response_counts$ease_1, "(", round(100*response_counts$ease_1/n_rated, 1), "%)\n")
```

```
##   1: 2 ( 7.7 %)
```

```r
cat("  2:", response_counts$ease_2, "(", round(100*response_counts$ease_2/n_rated, 1), "%)\n")
```

```
##   2: 1 ( 3.8 %)
```

```r
cat("  3:", response_counts$ease_3, "(", round(100*response_counts$ease_3/n_rated, 1), "%) <-- default\n
```

```
##   3: 7 ( 26.9 %) <-- default
```

```r
cat("  4:", response_counts$ease_4, "(", round(100*response_counts$ease_4/n_rated, 1), "%)\n")
```

```
##   4: 7 ( 26.9 %)
```

```r
cat("  5:", response_counts$ease_5, "(", round(100*response_counts$ease_5/n_rated, 1), "%)\n")
```

```
##   5: 9 ( 34.6 %)
```

```r
cat("\nSubjective Focus Quality:\n")
```

```
##
## Subjective Focus Quality:
```

```r
cat("  1:", response_counts$quality_1, "(", round(100*response_counts$quality_1/n_rated, 1), "%)\n")
```

```
##   1: 2 ( 7.7 %)
```

```r
cat("  2:", response_counts$quality_2, "(", round(100*response_counts$quality_2/n_rated, 1), "%)\n")
```

```
##   2: 1 ( 3.8 %)
```

```r
cat("  3:", response_counts$quality_3, "(", round(100*response_counts$quality_3/n_rated, 1), "%) <-- de
```

```
##   3: 10 ( 38.5 %) <-- default
```

```
cat("  4:", response_counts$quality_4, "(", round(100*response_counts$quality_4/n_rated, 1), "%)\n")
```

```
##   4: 8 ( 30.8 %)
```

```
cat("  5:", response_counts$quality_5, "(", round(100*response_counts$quality_5/n_rated, 1), "%)\n")
```

```
##   5: 5 ( 19.2 %)
```

```
# Binomial test: is "3" overrepresented vs expected 20% under uniform?
binom_stress <- binom.test(response_counts$stress_3, n_rated, p = 0.2, alternative = "greater")
binom_ease <- binom.test(response_counts$ease_3, n_rated, p = 0.2, alternative = "greater")

cat("\nBinomial test for '3' overrepresentation (vs 20% expected if uniform):\n")
```

```
##
## Binomial test for '3' overrepresentation (vs 20% expected if uniform):
```

```
cat("  Stress p-value:", round(binom_stress$p.value, 4), "\n")
```

```
##   Stress p-value: 0.1313
```

```
cat("  Ease p-value:", round(binom_ease$p.value, 4), "\n")
```

```
##   Ease p-value: 0.2526
```

4.4 Within-Person Comparison Setup

```
# Create paired data for participants who completed ratings for both conditions
paired_data <- rated_sessions %>%
  select(participant_id, condition, perceived_stress, ease_of_following, subjective_focus_quality) %>%
  pivot_wider(
    names_from = condition,
    values_from = c(perceived_stress, ease_of_following, subjective_focus_quality)
  ) %>%
  filter(
    !is.na(perceived_stress_COUNTDOWN) & !is.na(perceived_stress_HOURGLASS)
  ) %>%
  mutate(
    stress_diff = perceived_stress_HOURGLASS - perceived_stress_COUNTDOWN,
    ease_diff = ease_of_following_HOURGLASS - ease_of_following_COUNTDOWN,
    quality_diff = subjective_focus_quality_HOURGLASS - subjective_focus_quality_COUNTDOWN
  )

cat("Participants with ratings for both conditions:", nrow(paired_data), "\n\n")
```

```
## Participants with ratings for both conditions: 13
```

```r
cat("Within-Person Differences (HOURGLASS - COUNTDOWN):\n")
```

```
## Within-Person Differences (HOURGLASS - COUNTDOWN):
```

```r
cat("Perceived Stress: M =", round(mean(paired_data$stress_diff, na.rm = TRUE), 2),
    ", SD =", round(sd(paired_data$stress_diff, na.rm = TRUE), 2), "\n")
```

```
## Perceived Stress: M = -0.23 , SD = 1.17
```

```r
cat("Ease of Following: M =", round(mean(paired_data$ease_diff, na.rm = TRUE), 2),
    ", SD =", round(sd(paired_data$ease_diff, na.rm = TRUE), 2), "\n")
```

```
## Ease of Following: M = -1.08 , SD = 1.5
```

```r
cat("Focus Quality: M =", round(mean(paired_data$quality_diff, na.rm = TRUE), 2),
    ", SD =", round(sd(paired_data$quality_diff, na.rm = TRUE), 2), "\n")
```

```
## Focus Quality: M = 0.23 , SD = 1.24
```

4.5 Paired Plot Visualization

```r
# Visualize paired data
paired_long <- paired_data %>%
  select(participant_id,
         perceived_stress_COUNTDOWN, perceived_stress_HOURGLASS,
         ease_of_following_COUNTDOWN, ease_of_following_HOURGLASS,
         subjective_focus_quality_COUNTDOWN, subjective_focus_quality_HOURGLASS) %>%
  pivot_longer(
    cols = -participant_id,
    names_to = c("outcome", "condition"),
    names_pattern = "(.+)_(COUNTDOWN|HOURGLASS)",
    values_to = "rating"
  ) %>%
  mutate(
    outcome = factor(outcome,
                     levels = c("perceived_stress", "ease_of_following", "subjective_focus_quality"),
                     labels = c("Perceived Stress", "Ease of Following", "Focus Quality"))
  )

ggplot(paired_long, aes(x = condition, y = rating, group = participant_id)) +
  geom_line(alpha = 0.4, color = "gray50") +
  geom_point(aes(color = condition), size = 2) +
  facet_wrap(~outcome, scales = "free_y") +
  scale_color_manual(values = c("COUNTDOWN" = "#1f77b4", "HOURGLASS" = "#ff7f0e")) +
  labs(
    title = "Within-Person Comparison of Subjective Ratings",
    subtitle = "Each line connects the same participant's two sessions",
    x = "Condition",
    y = "Rating (1-5)"
  ) +
  theme(legend.position = "none")
```

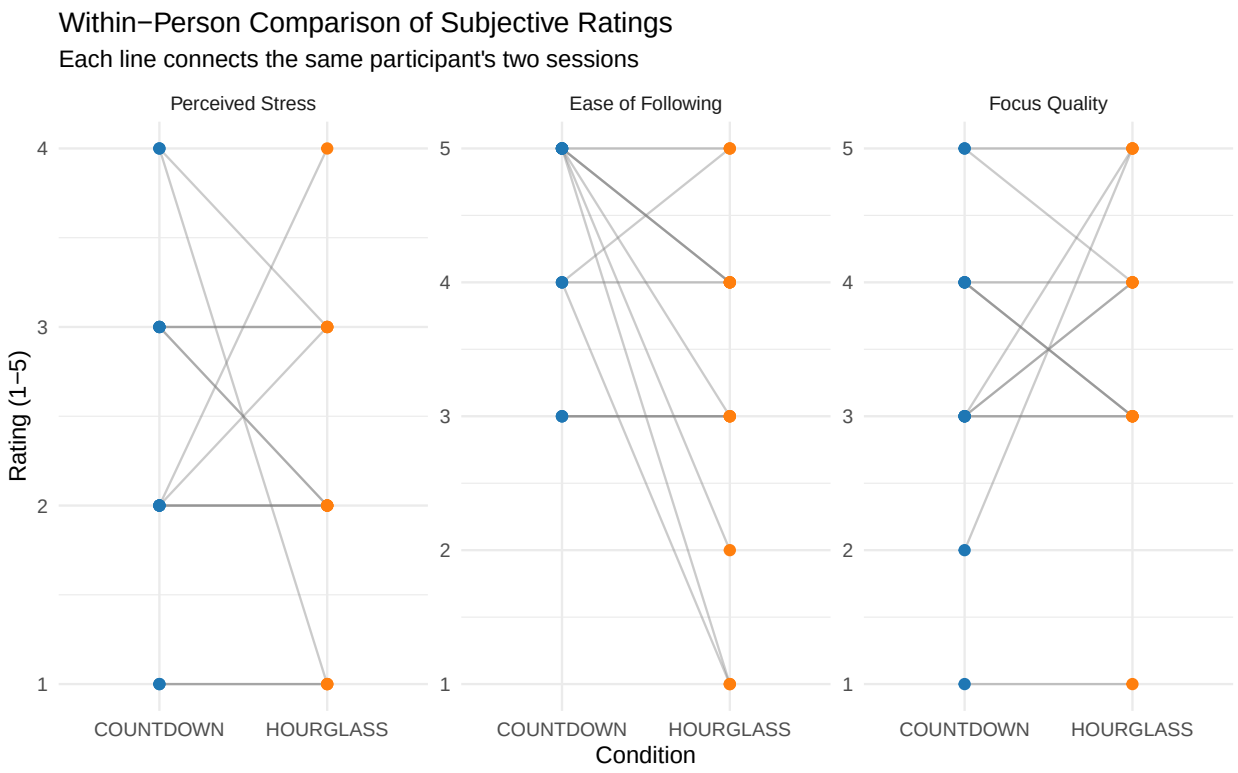

Figure 2: Within-Person Comparison of Subjective Ratings

4.6 Order Effects Check

```
# Check if session order affects subjective outcomes
order_effects <- rated_sessions %>%
  group_by(session_number) %>%
  summarise(
    n = n(),
    stress_mean = mean(perceived_stress, na.rm = TRUE),
    ease_mean = mean(ease_of_following, na.rm = TRUE),
    quality_mean = mean(subjective_focus_quality, na.rm = TRUE)
  )

order_effects %>%
  kable(
    caption = "Subjective Outcomes by Session Order",
    digits = 2,
    col.names = c("Session", "N", "Stress M", "Ease M", "Quality M")
  ) %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Table 2: Subjective Outcomes by Session Order

Session	N	Stress M	Ease M	Quality M

1	13	2.54	3.85	3.38
2	13	2.15	3.69	3.62

5. Statistical Models

5.1 Perceived Stress

Model 1a: Paired T-Test

```
if (nrow(paired_data) >= 2) {
  stress_ttest <- t.test(
    paired_data$perceived_stress_HOURGLASS,
    paired_data$perceived_stress_COUNTDOWN,
    paired = TRUE
  )

  cat("Paired T-Test: Perceived Stress\n")
  cat("Mean difference (HOURGLASS - COUNTDOWN):",
      round(stress_ttest$estimate, 3), "\n")
  cat("95% CI: [", round(stress_ttest$conf.int[1], 3), ",",
      round(stress_ttest$conf.int[2], 3), "]\n")
  cat("t =", round(stress_ttest$statistic, 3),
      ", df =", stress_ttest$parameter,
      ", p =", round(stress_ttest$p.value, 4), "\n")
} else {
  cat("Insufficient paired data for t-test\n")
  stress_ttest <- NULL
}
```

```
## Paired T-Test: Perceived Stress
## Mean difference (HOURGLASS - COUNTDOWN): -0.231
## 95% CI: [ -0.935 , 0.474 ]
## t = -0.714 , df = 12 , p = 0.489
```

Model 2a: Fixed Effects Regression

```
model_stress_fe <- lm(perceived_stress ~ condition + factor(participant_id),
                      data = rated_sessions)

stress_robust <- coeftest(model_stress_fe, vcov = vcovHC(model_stress_fe, type = "HC1"))

cat("Fixed Effects Model: Perceived Stress\n")
```

```
## Fixed Effects Model: Perceived Stress
```

```
cat("Coefficient (HOURGLASS vs COUNTDOWN):",
    round(stress_robust["conditionHOURGLASS", "Estimate"], 3), "\n")
```

```
## Coefficient (HOURGLASS vs COUNTDOWN): -0.231
```

```
cat("Robust SE:", round(stress_robust["conditionHOURGLASS", "Std. Error"], 3), "\n")
```

```
## Robust SE: 0.323
```

```
cat("t =", round(stress_robust["conditionHOURGLASS", "t value"], 3), "\n")
```

```
## t = -0.714
```

```
cat("p =", round(stress_robust["conditionHOURGLASS", "Pr(>|t|)"], 4), "\n")
```

```
## p = 0.489
```

Model 3a: With Baseline Covariate (Time Anxiety)

```
model_stress_cov <- lm(perceived_stress ~ condition + factor(participant_id) + time_anxiety_score,
                       data = rated_sessions %>% filter(!is.na(time_anxiety_score)))

stress_cov_robust <- coeftest(model_stress_cov, vcov = vcovHC(model_stress_cov, type = "HC1"))

cat("Fixed Effects + Time Anxiety: Perceived Stress\n")
```

```
## Fixed Effects + Time Anxiety: Perceived Stress
```

```
cat("Treatment effect (HOURGLASS):",
    round(stress_cov_robust["conditionHOURGLASS", "Estimate"], 3), "\n")
```

```
## Treatment effect (HOURGLASS): -0.231
```

```
cat("Robust SE:", round(stress_cov_robust["conditionHOURGLASS", "Std. Error"], 3), "\n")
```

```
## Robust SE: 0.323
```

```
cat("p =", round(stress_cov_robust["conditionHOURGLASS", "Pr(>|t|)"], 4), "\n")
```

```
## p = 0.489
```

5.2 Ease of Following

Model 1b: Paired T-Test

```r
if (nrow(paired_data) >= 2) {
  ease_ttest <- t.test(
    paired_data$ease_of_following_HOURGLASS,
    paired_data$ease_of_following_COUNTDOWN,
    paired = TRUE
  )

  cat("Paired T-Test: Ease of Following\n")
  cat("Mean difference (HOURGLASS - COUNTDOWN):",
      round(ease_ttest$estimate, 3), "\n")
  cat("95% CI: [", round(ease_ttest$conf.int[1], 3), ",",
      round(ease_ttest$conf.int[2], 3), "]\n")
  cat("t =", round(ease_ttest$statistic, 3),
      ", df =", ease_ttest$parameter,
      ", p =", round(ease_ttest$p.value, 4), "\n")
} else {
  ease_ttest <- NULL
}
```

```
## Paired T-Test: Ease of Following
## Mean difference (HOURGLASS - COUNTDOWN): -1.077
## 95% CI: [ -1.982 , -0.172 ]
## t = -2.592 , df = 12 , p = 0.0236
```

Model 2b: Fixed Effects Regression

```r
model_ease_fe <- lm(ease_of_following ~ condition + factor(participant_id),
                    data = rated_sessions)

ease_robust <- coeftest(model_ease_fe, vcov = vcovHC(model_ease_fe, type = "HC1"))

cat("Fixed Effects Model: Ease of Following\n")
```

```
## Fixed Effects Model: Ease of Following
```

```r
cat("Coefficient (HOURGLASS vs COUNTDOWN):",
    round(ease_robust["conditionHOURGLASS", "Estimate"], 3), "\n")
```

```
## Coefficient (HOURGLASS vs COUNTDOWN): -1.077
```

```r
cat("Robust SE:", round(ease_robust["conditionHOURGLASS", "Std. Error"], 3), "\n")
```

```
## Robust SE: 0.415
```

```r
cat("p =", round(ease_robust["conditionHOURGLASS", "Pr(>|t|)"], 4), "\n")
```

```
## p = 0.0236
```

5.3 Subjective Focus Quality

Model 1c: Paired T-Test

```
if (nrow(paired_data) >= 2) {
  quality_ttest <- t.test(
    paired_data$subjective_focus_quality_HOURGLASS,
    paired_data$subjective_focus_quality_COUNTDOWN,
    paired = TRUE
  )

  cat("Paired T-Test: Subjective Focus Quality\n")
  cat("Mean difference (HOURGLASS - COUNTDOWN):",
      round(quality_ttest$estimate, 3), "\n")
  cat("95% CI: [", round(quality_ttest$conf.int[1], 3), ",",
      round(quality_ttest$conf.int[2], 3), "]\n")
  cat("t =", round(quality_ttest$statistic, 3),
      ", df =", quality_ttest$parameter,
      ", p =", round(quality_ttest$p.value, 4), "\n")
} else {
  quality_ttest <- NULL
}
```

```
## Paired T-Test: Subjective Focus Quality
## Mean difference (HOURGLASS - COUNTDOWN): 0.231
## 95% CI: [ -0.516 , 0.977 ]
## t = 0.674 , df = 12 , p = 0.5133
```

Model 2c: Fixed Effects Regression

```
model_quality_fe <- lm(subjective_focus_quality ~ condition + factor(participant_id),
                       data = rated_sessions %>% filter(!is.na(subjective_focus_quality)))

quality_robust <- coeftest(model_quality_fe, vcov = vcovHC(model_quality_fe, type = "HC1"))

cat("Fixed Effects Model: Subjective Focus Quality\n")
```

```
## Fixed Effects Model: Subjective Focus Quality
```

```
cat("Coefficient (HOURGLASS vs COUNTDOWN):",
    round(quality_robust["conditionHOURGLASS", "Estimate"], 3), "\n")
```

```
## Coefficient (HOURGLASS vs COUNTDOWN): 0.231
```

```
cat("Robust SE:", round(quality_robust["conditionHOURGLASS", "Std. Error"], 3), "\n")
```

```
## Robust SE: 0.343
```

```r
cat("p =", round(quality_robust["conditionHOURGLASS", "Pr(>|t|)"], 4), "\n")
```

```
## p = 0.5133
```

6. Results Summary Table

```r
# Compile all results
results_summary <- data.frame(
  Outcome = c("Perceived Stress", "Perceived Stress", "Perceived Stress",
              "Ease of Following", "Ease of Following",
              "Focus Quality", "Focus Quality"),
  Model = c("Paired t-test", "Fixed Effects", "FE + Covariate",
            "Paired t-test", "Fixed Effects",
            "Paired t-test", "Fixed Effects"),
  Estimate = c(
    ifelse(!is.null(stress_ttest), round(stress_ttest$estimate, 3), NA),
    round(stress_robust["conditionHOURGLASS", "Estimate"], 3),
    round(stress_cov_robust["conditionHOURGLASS", "Estimate"], 3),
    ifelse(!is.null(ease_ttest), round(ease_ttest$estimate, 3), NA),
    round(ease_robust["conditionHOURGLASS", "Estimate"], 3),
    ifelse(!is.null(quality_ttest), round(quality_ttest$estimate, 3), NA),
    round(quality_robust["conditionHOURGLASS", "Estimate"], 3)
  ),
  SE = c(
    ifelse(!is.null(stress_ttest), round(sd(paired_data$stress_diff)/sqrt(nrow(paired_data)), 3), NA),
    round(stress_robust["conditionHOURGLASS", "Std. Error"], 3),
    round(stress_cov_robust["conditionHOURGLASS", "Std. Error"], 3),
    ifelse(!is.null(ease_ttest), round(sd(paired_data$ease_diff)/sqrt(nrow(paired_data)), 3), NA),
    round(ease_robust["conditionHOURGLASS", "Std. Error"], 3),
    ifelse(!is.null(quality_ttest), round(sd(paired_data$quality_diff, na.rm=TRUE)/sqrt(nrow(paired_data
    round(quality_robust["conditionHOURGLASS", "Std. Error"], 3)
  ),
  p_value = c(
    ifelse(!is.null(stress_ttest), round(stress_ttest$p.value, 4), NA),
    round(stress_robust["conditionHOURGLASS", "Pr(>|t|)"], 4),
    round(stress_cov_robust["conditionHOURGLASS", "Pr(>|t|)"], 4),
    ifelse(!is.null(ease_ttest), round(ease_ttest$p.value, 4), NA),
    round(ease_robust["conditionHOURGLASS", "Pr(>|t|)"], 4),
    ifelse(!is.null(quality_ttest), round(quality_ttest$p.value, 4), NA),
    round(quality_robust["conditionHOURGLASS", "Pr(>|t|)"], 4)
  )
)

results_summary %>%
  kable(
    caption = "Summary of Treatment Effects on Subjective Outcomes (HOURGLASS vs COUNTDOWN)",
    col.names = c("Outcome", "Model", "Estimate", "SE", "p-value"),
    digits = 4
  ) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed")) %>%
```

```
  pack_rows("Perceived Stress", 1, 3) %>%
  pack_rows("Ease of Following", 4, 5) %>%
  pack_rows("Focus Quality", 6, 7) %>%
  footnote(general = "Estimates represent effect of HOURGLASS relative to COUNTDOWN. Positive values in
```

Table 3: Summary of Treatment Effects on Subjective Outcomes (HOURGLASS vs COUNTDOWN)

Outcome	Model	Estimate	SE
Perceived Stress			
Perceived Stress	Paired t-test	-0.231	0.323
Perceived Stress	Fixed Effects	-0.231	0.323
Perceived Stress	FE + Covariate	-0.231	0.323
Ease of Following			
Ease of Following	Paired t-test	-1.077	0.415
Ease of Following	Fixed Effects	-1.077	0.415
Focus Quality			
Focus Quality	Paired t-test	0.231	0.343
Focus Quality	Fixed Effects	0.231	0.343

Note:
Estimates represent effect of HOURGLASS relative to COUNTDOWN. Positive values indicate higher ratings for HOURG

7. Preference Analysis (Post-Treatment)

```
if (nrow(post_treatment) > 0) {
  # Timer preference distribution
  pref_table <- post_treatment %>%
    group_by(preferred_timer) %>%
    summarise(
      n = n(),
      pct = n() / nrow(post_treatment) * 100
    )

  cat("Timer Preference Distribution (N =", nrow(post_treatment), "):\n")
  print(pref_table)

  # Test if preference differs from 50/50 (excluding NO_PREFERENCE)
  pref_binary <- post_treatment %>%
    filter(preferred_timer %in% c("COUNTDOWN", "HOURGLASS"))

  if (nrow(pref_binary) >= 2) {
    n_hourglass <- sum(pref_binary$preferred_timer == "HOURGLASS")
    n_total <- nrow(pref_binary)

    pref_binom <- binom.test(n_hourglass, n_total, p = 0.5)

    cat("\nBinomial test (HOURGLASS vs COUNTDOWN preference):\n")
    cat("HOURGLASS preferred:", n_hourglass, "of", n_total, "\n")
```

```
    cat("Proportion:", round(n_hourglass/n_total, 3), "\n")
    cat("95% CI: [", round(pref_binom$conf.int[1], 3), ",",
        round(pref_binom$conf.int[2], 3), "]\n")
    cat("p-value (vs 50/50):", round(pref_binom$p.value, 4), "\n")
  }

  # Would use again / recommend
  cat("\nUsage intentions:\n")
  cat("Would use again:", sum(post_treatment$would_use_again, na.rm = TRUE), "of",
      sum(!is.na(post_treatment$would_use_again)),
      "(", round(100*mean(post_treatment$would_use_again, na.rm = TRUE), 1), "%)\n")
  cat("Would recommend:", sum(post_treatment$recommend_to_others, na.rm = TRUE), "of",
      sum(!is.na(post_treatment$recommend_to_others)),
      "(", round(100*mean(post_treatment$recommend_to_others, na.rm = TRUE), 1), "%)\n")
} else {
  cat("No post-treatment survey data available.\n")
}
```

```
## Timer Preference Distribution (N = 12 ):
## # A tibble: 3 x 3
##   preferred_timer     n   pct
##   <chr>           <int> <dbl>
## 1 COUNTDOWN           5 41.7
## 2 HOURGLASS           6 50
## 3 NO_PREFERENCE       1  8.33
##
## Binomial test (HOURGLASS vs COUNTDOWN preference):
## HOURGLASS preferred: 6 of 11
## Proportion: 0.545
## 95% CI: [ 0.234 , 0.833 ]
## p-value (vs 50/50): 1
##
## Usage intentions:
## Would use again: 7 of 12 ( 58.3 %)
## Would recommend: 8 of 12 ( 66.7 %)
```

8. Qualitative Feedback Summary

8.1 Post-Session Comments

```
# Extract comments by condition
comments_df <- rated_sessions %>%
  filter(!is.na(comments) & comments != "") %>%
  select(condition, comments)

cat("Post-Session Comments by Condition:\n\n")
```

```
## Post-Session Comments by Condition:
```

```r
if (nrow(comments_df) > 0) {
  countdown_comments <- comments_df %>% filter(condition == "COUNTDOWN")
  hourglass_comments <- comments_df %>% filter(condition == "HOURGLASS")

  cat("COUNTDOWN (N =", nrow(countdown_comments), "):\n")
  if (nrow(countdown_comments) > 0) {
    for (i in 1:nrow(countdown_comments)) {
      cat("  -", countdown_comments$comments[i], "\n")
    }
  } else {
    cat("  (no comments)\n")
  }

  cat("\nHOURGLASS (N =", nrow(hourglass_comments), "):\n")
  if (nrow(hourglass_comments) > 0) {
    for (i in 1:nrow(hourglass_comments)) {
      cat("  -", hourglass_comments$comments[i], "\n")
    }
  } else {
    cat("  (no comments)\n")
  }
} else {
  cat("No comments recorded.\n")
}
```

```
## COUNTDOWN (N = 4 ):
##   - Seeing time go by creates more urgency for myself (in a good way)
##   - I liked this one bettel
##   - Timer was useful
##   - Once I got going on my task, I pretty much forgot about the timer and blew through the time
##
## HOURGLASS (N = 4 ):
##   - It seemed about the same
##   - I prefer having a number for the number of time left
##   - I didn't realize the timer was actually showing time completed. I didn't pay much attention to i
##   - I felt focused and motivated to work through the timer because it was a set goal to meet.
```

8.2 Post-Treatment Qualitative Feedback

```r
feedback_df <- post_treatment %>%
  filter(!is.na(qualitative_feedback) & qualitative_feedback != "")

cat("Post-Treatment Qualitative Feedback (N =", nrow(feedback_df), "):\n\n")
```

```
## Post-Treatment Qualitative Feedback (N = 12 ):
```

```r
if (nrow(feedback_df) > 0) {
  for (i in 1:nrow(feedback_df)) {
    cat("Participant", i, ":\n")
    cat("  ", feedback_df$qualitative_feedback[i], "\n\n")
```

```
  }
} else {
  cat("No qualitative feedback recorded.\n")
}
```

```
## Participant 1 :
##    The hourglass timer I didn't like as much since I had no idea how much time passed nor how much ti
##
## Participant 2 :
##    i like seeing a visual countdown, helps me track how long i have left
##
## Participant 3 :
##    The countdown timer was more stressful than the hourglass.
##
## Participant 4 :
##    I liked knowing exactly how much time is left, but this may be because this is what I am used to,
##
## Participant 5 :
##    Hourglass visualization is more relaxed as it doesn't have the minutes left displayed , that for m
##
## Participant 6 :
##    make the hour glass design and visualization a bit better.
## and make it an app or as a widget.
##
## Participant 7 :
##    I like to know the remaining time needed to focus, which is available in the countdown timer.
##
## Participant 8 :
##    Countdown was more distracting because of the numbers constantly changing
##
## Participant 9 :
##    Hourglass is more relaxing than countdown timer.
##
## Participant 10 :
##    I was probably better able to track the displacement of time with the countdown timer. However, wh
##
## Participant 11 :
##    I felt that without the countdown, I was less focused on how much time I had left, and I more focu
##
## Participant 12 :
##    The hourglass helped me retain my focus, but didn't apply as much pressure as the countdown!
```

9. CONSORT Flow Data

```
consort_data <- list(
  enrolled = raw_data$total_participants,
  completed_baseline = nrow(baseline),
  total_sessions = raw_data$total_sessions,
  hourglass_sessions = sum(sessions$condition == "HOURGLASS", na.rm = TRUE),
  countdown_sessions = sum(sessions$condition == "COUNTDOWN", na.rm = TRUE),
```

```
  with_ratings = nrow(rated_sessions),
  paired_complete = nrow(paired_data),
  completed_post_treatment = nrow(post_treatment)
)

cat("CONSORT Flow Data:\n")
```

```
## CONSORT Flow Data:
```

```
cat("Enrolled:", consort_data$enrolled, "\n")
```

```
## Enrolled: 25
```

```
cat("Completed Baseline Survey:", consort_data$completed_baseline, "\n")
```

```
## Completed Baseline Survey: 25
```

```
cat("Total Sessions Started:", consort_data$total_sessions, "\n")
```

```
## Total Sessions Started: 32
```

```
cat("  - COUNTDOWN:", consort_data$countdown_sessions, "\n")
```

```
##   - COUNTDOWN: 17
```

```
cat("  - HOURGLASS:", consort_data$hourglass_sessions, "\n")
```

```
##   - HOURGLASS: 15
```

```
cat("Sessions with Post-Session Ratings:", consort_data$with_ratings, "\n")
```

```
## Sessions with Post-Session Ratings: 26
```

```
cat("Participants with Both Conditions Rated:", consort_data$paired_complete, "\n")
```

```
## Participants with Both Conditions Rated: 13
```

```
cat("Completed Post-Treatment Survey:", consort_data$completed_post_treatment, "\n")
```

```
## Completed Post-Treatment Survey: 12
```

10. Effect Size Calculations

```r
# Cohen's d for within-subjects
cohens_d_within <- function(diff_scores) {
  mean(diff_scores, na.rm = TRUE) / sd(diff_scores, na.rm = TRUE)
}

if (nrow(paired_data) >= 2) {
  cat("Cohen's d (within-subjects, HOURGLASS - COUNTDOWN):\n")
  cat("  Perceived Stress: d =", round(cohens_d_within(paired_data$stress_diff), 3), "\n")
  cat("  Ease of Following: d =", round(cohens_d_within(paired_data$ease_diff), 3), "\n")
  cat("  Focus Quality: d =", round(cohens_d_within(paired_data$quality_diff), 3), "\n")

  cat("\nInterpretation: |d| < 0.2 = negligible, 0.2-0.5 = small, 0.5-0.8 = medium, > 0.8 = large\n")
}
```

```
## Cohen's d (within-subjects, HOURGLASS - COUNTDOWN):
##   Perceived Stress: d = -0.198
##   Ease of Following: d = -0.719
##   Focus Quality: d = 0.187
##
## Interpretation: |d| < 0.2 = negligible, 0.2-0.5 = small, 0.5-0.8 = medium, > 0.8 = large
```

11. Session Info

```r
sessionInfo()
```

```
## R version 4.5.0 (2025-04-11 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26200)
##
## Matrix products: default
##   LAPACK version 3.12.1
##
## 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/Los_Angeles
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] kableExtra_1.4.0 knitr_1.50       sandwich_3.1-1   lmtest_0.9-40
##  [5] zoo_1.8-14       lubridate_1.9.4  forcats_1.0.0    stringr_1.5.1
##  [9] dplyr_1.1.4      purrr_1.0.4      readr_2.1.5      tidyr_1.3.1
## [13] tibble_3.2.1     ggplot2_3.5.2    tidyverse_2.0.0  jsonlite_2.0.0
```

```
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.5         generics_0.1.4     xml2_1.3.8         stringi_1.8.7
##  [5] lattice_0.22-7     hms_1.1.3          digest_0.6.37      magrittr_2.0.3
##  [9] evaluate_1.0.3     grid_4.5.0         timechange_0.3.0   RColorBrewer_1.1-3
## [13] fastmap_1.2.0      viridisLite_0.4.2  scales_1.4.0       textshaping_1.0.1
## [17] cli_3.6.5          rlang_1.1.6        withr_3.0.2        yaml_2.3.10
## [21] tools_4.5.0        tzdb_0.5.0         vctrs_0.6.5        R6_2.6.1
## [25] lifecycle_1.0.4    pkgconfig_2.0.3    pillar_1.10.2      gtable_0.3.6
## [29] glue_1.8.0         systemfonts_1.3.1  xfun_0.52          tidyselect_1.2.1
## [33] rstudioapi_0.17.1  farver_2.1.2       htmltools_0.5.8.1  rmarkdown_2.29
## [37] svglite_2.2.2      labeling_0.4.3     compiler_4.5.0
```