

NavCity Demographic and Cognitive Correlates Analysis - Three Groups

Your Name

2025-10-10

Load Required Libraries

```
library(tidyverse)
library(broom)
library(knitr)
library(kableExtra)
library(reshape2)
library(car)
library(rstatix)      # For Games-Howell and other post-hoc tests
library(FSA)          # For Dunn's test
library(effectsize)   # For effect size calculations
```

Data Import and Preparation

```
# Load all data files
demographic_data <- read_csv("/Volumes/YB_Drive/NavAging_Paper/data/demographic_data.csv")
non_nav <- read_csv("/Volumes/YB_Drive/NavAging_Paper/data/non_nav_data.csv")
ya_nav <- read_csv("/Volumes/YB_Drive/NavAging_Paper/data/ya_averaged_results.csv")
oa_nav <- read_csv("/Volumes/YB_Drive/NavAging_Paper/data/oa_averaged_results.csv")

# Combine YA and OA navigation data
nav_data <- bind_rows(
  ya_nav %>% mutate(Group = "YA"),
  oa_nav %>% mutate(Group = "OA")
)

# Average navigation metrics across all blocks for each participant
nav_averaged <- nav_data %>%
  group_by(Participant, Group) %>%
  summarise(
    Mean_Speed = mean(Speed, na.rm = TRUE),
    Mean_Distance = mean(Distance, na.rm = TRUE),
    Mean_Navigation_Time = mean(Navigation_Time, na.rm = TRUE),
    .groups = "drop"
)
```

```

# Merge all datasets
merged_temp <- demographic_data %>%
  full_join(non_nav, by = "Participant", suffix = c("_demo", "_non_nav")) %>%
  mutate(Group = coalesce(Group_non_nav,
                          if_else(Group_demo == 1, "YA", "OA"))) %>%
  select(-Group_demo, -Group_non_nav)

merged_data <- merged_temp %>%
  left_join(nav_averaged, by = c("Participant", "Group"))

# Calculate change scores
merged_data <- merged_data %>%
  mutate(
    SSS_Diff = SSS_Post - SSS_Pre,
    SSQ_Diff = SSQ_Post - SSQ_Pre
  )

# CREATE THREE-GROUP VARIABLE
# Define your criteria for splitting OA into High and Low
# Example: Split by median NARA score (or your preferred measure)
oa_data <- merged_data %>% filter(Group == "OA")
nara_median <- median(oa_data$NARA, na.rm = TRUE)

merged_data <- merged_data %>%
  mutate(Group_3Level = case_when(
    Group == "YA" ~ "YA",
    Group == "OA" & NARA >= nara_median ~ "OA_High",
    Group == "OA" & NARA < nara_median ~ "OA_Low",
    TRUE ~ NA_character_
  ))

# Convert to factor with proper ordering
merged_data$Group_3Level <- factor(merged_data$Group_3Level,
                                    levels = c("YA", "OA_High", "OA_Low"))

# Display sample of merged data
cat("Total participants:", nrow(merged_data), "\n")

## Total participants: 60

cat("YA participants:", sum(merged_data$Group_3Level == "YA", na.rm = TRUE), "\n")

## YA participants: 30

cat("OA_High participants:", sum(merged_data$Group_3Level == "OA_High", na.rm = TRUE), "\n")

## OA_High participants: 15

cat("OA_Low participants:", sum(merged_data$Group_3Level == "OA_Low", na.rm = TRUE), "\n\n")

## OA_Low participants: 15

```

```

head(merged_data) %>%
  select(Participant, Group_3Level, Gender, Handedness, SBSOD, Mean_Speed) %>%
  kable() %>%
  kable_styling()

```

Participant	Group_3Level	Gender	Handedness	SBSOD	Mean_Speed
BNC01	YA	W	R	2.20	10.750260
BNC02	YA	W	R	3.40	16.197016
BNC03	YA	M	M	5.20	12.966692
BNC04	YA	W	R	5.27	8.770131
BNC05	YA	W	R	4.87	17.149700
BNC06	OA_High	M	R	6.87	14.922086

Part 1: Group Comparisons (YA vs. OA_High vs. OA_Low)

Test 1: Gender Distribution

```

# Create contingency table
gender_table <- table(merged_data$Gender, merged_data$Group_3Level)

cat("Contingency Table:\n")

```

```
## Contingency Table:
```

```
print(gender_table)
```

```
##
##      YA OA_High OA_Low
##      M    13      6      6
##      W    17      9      9
```

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

```
# Add row and column totals for clarity
gender_table_with_totals <- addmargins(gender_table)
cat("With Totals:\n")
```

```
## With Totals:
```

```
print(gender_table_with_totals)
```

```
##
##      YA OA_High OA_Low Sum
##      M    13      6      6  25
##      W    17      9      9  35
##      Sum  30     15     15  60
```

```

cat("\n")

# Calculate proportions within each group
cat("Proportions by Group:\n")

## Proportions by Group:

prop_table <- prop.table(gender_table, margin = 2)
print(round(prop_table, 3))

## 
##          YA OA_High OA_Low
##   M 0.433    0.400  0.400
##   W 0.567    0.600  0.600

cat("\n")

# Perform chi-square test (without Yates' continuity correction)
chi_test <- chisq.test(gender_table, correct = FALSE)

cat("Chi-Square Test Results:\n")

## Chi-Square Test Results:

cat("  ^  =", chi_test$statistic, "\n")

##  ^  = 0.06857143

cat("df  =", chi_test$parameter, "\n")

## df  = 2

cat("p-value  =", chi_test$p.value, "\n")

## p-value = 0.9662954

# Check expected frequencies
cat("\nExpected Frequencies:\n")

## 
##          YA OA_High OA_Low
##   M 12.5     6.25   6.25
##   W 17.5     8.75   8.75

```

```

cat("\n")

# Calculate effect size (Cramér's V)
n <- sum(gender_table)
min_dim <- min(nrow(gender_table) - 1, ncol(gender_table) - 1)
cramers_v <- sqrt(chi_test$statistic / (n * min_dim))
cat("Cramér's V =", round(cramers_v, 3), "\n")

## Cramér's V = 0.034

# Interpretation
cat("\n--- INTERPRETATION ---\n")

## 
## --- INTERPRETATION ---

if(chi_test$p.value < 0.05) {
  cat("Result: SIGNIFICANT difference in gender distribution across groups (p < 0.05)\n")
} else {
  cat("Result: NO significant difference in gender distribution across groups (p >= 0.05)\n")
}

## Result: NO significant difference in gender distribution across groups (p >= 0.05)

# Check assumption
if(min(chi_test$expected) < 5) {
  cat("WARNING: Some expected frequencies < 5. Consider Fisher's exact test or combining categories.\n")
}

```

Test 2: Handedness Distribution

```

# Create collapsed category: Right vs. Non-Right
merged_data <- merged_data %>%
  mutate(Handedness_Collapsed = ifelse(Handedness == "R", "Right-handed", "Non-right-handed"))

# Create contingency table
handedness_table <- table(merged_data$Handedness_Collapsed, merged_data$Group_3Level)

cat("Collapsed Contingency Table (Right vs. Non-Right):\n")

## Collapsed Contingency Table (Right vs. Non-Right):

print(handedness_table)

##
##          YA OA_High OA_Low
## Non-right-handed  5      2     2
## Right-handed    25     13    13

```

```

cat("\n")

# Add row and column totals
handedness_table_with_totals <- addmargins(handedness_table)
cat("With Totals:\n")

## With Totals:

print(handedness_table_with_totals)

##                                     YA OA_High OA_Low Sum
##   Non-right-handed    5      2      2   9
##   Right-handed       25     13     13  51
##   Sum                 30     15     15  60

cat("\n")

# Calculate proportions within each group
cat("Proportions by Group:\n")

## Proportions by Group:

prop_table <- prop.table(handedness_table, margin = 2)
print(round(prop_table, 3))

##                                     YA OA_High OA_Low
##   Non-right-handed  0.167   0.133   0.133
##   Right-handed      0.833   0.867   0.867

cat("\n")

# Perform chi-square test
chi_test <- chisq.test(handedness_table, correct = FALSE)

cat("Chi-Square Test Results:\n")

## Chi-Square Test Results:

cat("  =", chi_test$statistic, "\n")

##  = 0.130719

cat("df =", chi_test$parameter, "\n")

## df = 2

```

```

cat("p-value =", chi_test$p.value, "\n")

## p-value = 0.9367307

# Check expected frequencies
cat("\nExpected Frequencies:\n")

## 
## Expected Frequencies:

print(round(chi_test$expected, 2))

## 
##          YA OA_High OA_Low
## Non-right-handed 4.5    2.25   2.25
## Right-handed    25.5   12.75  12.75

cat("\n")

# Calculate effect size
n <- sum(handedness_table)
min_dim <- min(nrow(handedness_table) - 1, ncol(handedness_table) - 1)
cramers_v <- sqrt(chi_test$statistic / (n * min_dim))
cat("Cramér's V =", round(cramers_v, 3), "\n")

## Cramér's V = 0.047

# Interpretation
cat("\n--- INTERPRETATION ---\n")

## 
## --- INTERPRETATION ---

if(chi_test$p.value < 0.05) {
  cat("Result: SIGNIFICANT difference in handedness distribution across groups (p < 0.05)\n")
} else {
  cat("Result: NO significant difference in handedness distribution across groups (p >= 0.05)\n")
}

## Result: NO significant difference in handedness distribution across groups (p >= 0.05)

if(min(chi_test$expected) < 5) {
  cat("  WARNING: Some expected frequencies < 5.\n")
}

##     WARNING: Some expected frequencies < 5.

```

Test 3: Prior VR Experience Distribution

```

cat("VR Experience: Prior VR Exposure\n")

## VR Experience: Prior VR Exposure

cat("Note: Ordinal scale (0 = never, 1 = 1-3 times, 2 = >3 times)\n")

## Note: Ordinal scale (0 = never, 1 = 1-3 times, 2 = >3 times)

cat("Analysis: Kruskal-Wallis test (appropriate for ordinal data with 3 groups)\n\n")

## Analysis: Kruskal-Wallis test (appropriate for ordinal data with 3 groups)

# Summary statistics by group
vr_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(VR_Experience_Quantified)),
    Median = median(VR_Experience_Quantified, na.rm = TRUE),
    Q1 = quantile(VR_Experience_Quantified, 0.25, na.rm = TRUE),
    Q3 = quantile(VR_Experience_Quantified, 0.75, na.rm = TRUE),
    Mean = mean(VR_Experience_Quantified, na.rm = TRUE),
    SD = sd(VR_Experience_Quantified, na.rm = TRUE),
    Min = min(VR_Experience_Quantified, na.rm = TRUE),
    Max = max(VR_Experience_Quantified, na.rm = TRUE)
  )

cat("Summary Statistics:\n")

## Summary Statistics:

print(vr_summary)

## # A tibble: 3 x 9
##   Group_3Level     n Median     Q1     Q3   Mean     SD   Min   Max
##   <fct>     <int>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 YA          30     1     0     2  0.933  0.868     0     2
## 2 OA_High     15     0     0     0  0.2    0.414     0     1
## 3 OA_Low      15     0     0     1  0.467  0.640     0     2

cat("\n")

# Filter out NA values for testing
vr_data_clean <- merged_data %>%
  filter(!is.na(VR_Experience_Quantified), !is.na(Group_3Level))

# Kruskal-Wallis test
kw_test <- kruskal.test(VR_Experience_Quantified ~ Group_3Level, data = vr_data_clean)

cat("Kruskal-Wallis Test Results:\n")

```

```

## Kruskal-Wallis Test Results:

cat("H statistic =", kw_test$statistic, "\n")

## H statistic = 8.926015

cat("df =", kw_test$parameter, "\n")

## df = 2

cat("p-value =", kw_test$p.value, "\n")

## p-value = 0.01152764

# Effect size (epsilon-squared)
epsilon_sq <- kw_test$statistic / ((nrow(vr_data_clean)^2 - 1) / (nrow(vr_data_clean) + 1))
cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

## Epsilon-squared = 0.151

# Post-hoc: Dunn's test with Holm correction
if(kw_test$p.value < 0.05) {
  cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
  dunn_results <- dunnTest(VR_Experience_Quantified ~ Group_3Level,
                            data = vr_data_clean, method = "holm")
  print(dunn_results$res)
  cat("\n")
}

## Post-hoc Comparisons (Dunn's test with Holm correction):
##          Comparison      Z     P.unadj     P.adj
## 1  OA_High - OA_Low -1.011366 0.311841269 0.31184127
## 2    OA_High - YA -2.879292 0.003985685 0.01195706
## 3    OA_Low - YA -1.711468 0.086994838 0.17398968

# Interpretation
cat("\n--- INTERPRETATION ---\n")

## --- INTERPRETATION ---

if(kw_test$p.value < 0.05) {
  cat("Result: SIGNIFICANT difference in VR experience across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in VR experience across groups (p >= 0.05)\n")
}

## Result: SIGNIFICANT difference in VR experience across groups (p < 0.05)
## See post-hoc tests above for pairwise comparisons

```

Test 4: Weekly Video Game Usage

```
cat("Video Game Usage (numeric variable)\n\n")

## Video Game Usage (numeric variable)

# Summary statistics by group
vg_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(Video_Game_Experience_Quantified)),
    Mean = mean(Video_Game_Experience_Quantified, na.rm = TRUE),
    SD = sd(Video_Game_Experience_Quantified, na.rm = TRUE),
    Median = median(Video_Game_Experience_Quantified, na.rm = TRUE),
    Q1 = quantile(Video_Game_Experience_Quantified, 0.25, na.rm = TRUE),
    Q3 = quantile(Video_Game_Experience_Quantified, 0.75, na.rm = TRUE),
    Min = min(Video_Game_Experience_Quantified, na.rm = TRUE),
    Max = max(Video_Game_Experience_Quantified, na.rm = TRUE)
  )

cat("Summary Statistics:\n")

## Summary Statistics:

print(vg_summary)

## # A tibble: 3 x 9
##   Group_3Level     n   Mean    SD Median    Q1    Q3   Min   Max
##   <fct>     <int> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 YA          30  2.7  3.39    1.5    0  3.75    0    10
## 2 OA_High     15  0.133 0.516    0      0    0      0     2
## 3 OA_Low      15  1     2.65    0      0    0      0     8

cat("\n")

# Filter out NA values for testing
vg_data_clean <- merged_data %>%
  filter(!is.na(Video_Game_Experience_Quantified), !is.na(Group_3Level))

# Check normality for all groups
cat("Normality Tests (Shapiro-Wilk):\n")

## Normality Tests (Shapiro-Wilk):

for(grp in levels(vg_data_clean$Group_3Level)) {
  data_subset <- vg_data_clean %>% filter(Group_3Level == grp) %>%
    pull(Video_Game_Experience_Quantified)
  shapiro_result <- shapiro.test(data_subset)
  cat(grp, ": W =", round(shapiro_result$statistic, 4), ", p =", shapiro_result$p.value)
  if(shapiro_result$p.value < 0.05) cat(" (non-normal)") else cat(" (normal)")
  cat("\n")
}
```

```

## YA : W = 0.7793 , p = 2.812079e-05 (non-normal)
## OA_High : W = 0.2841 , p = 9.834313e-08 (non-normal)
## OA_Low : W = 0.4227 , p = 8.898791e-07 (non-normal)

cat("\n")

# Check homogeneity of variance
levene_test <- leveneTest(Video_Game_Experience_Quantified ~ Group_3Level,
                           data = vg_data_clean)
cat("Levene's Test for Homogeneity of Variance:\n")

## Levene's Test for Homogeneity of Variance:

cat("F =", round(levene_test$`F value`[1], 4), ", p =", levene_test$`Pr(>F)`[1])

## F = 7.3654 , p = 0.001428301

if(levene_test$`Pr(>F)`[1] < 0.05) cat(" (variances unequal)") else cat(" (variances equal)")

## (variances unequal)

cat("\n\n")

# Decide which test to use
use_welch <- levene_test$`Pr(>F)`[1] < 0.05
normality_check <- vg_data_clean %>%
  group_by(Group_3Level) %>%
  summarise(p_val = shapiro.test(Video_Game_Experience_Quantified)$p.value) %>%
  pull(p_val)
use_nonparametric <- any(normality_check < 0.05)

if(use_nonparametric) {
  cat(" Data violates normality assumption. Using Kruskal-Wallis test.\n\n")
  kw_test <- kruskal.test(Video_Game_Experience_Quantified ~ Group_3Level,
                          data = vg_data_clean)
  cat("Kruskal-Wallis Test Results:\n")
  cat("H statistic =", kw_test$statistic, "\n")
  cat("df =", kw_test$parameter, "\n")
  cat("p-value =", kw_test$p.value, "\n")

  # Effect size
  epsilon_sq <- kw_test$statistic / ((nrow(vg_data_clean)^2 - 1) /
                                         (nrow(vg_data_clean) + 1))
  cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

  p_value <- kw_test$p.value

  # Post-hoc tests
  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
    dunn_results <- dunnTest(Video_Game_Experience_Quantified ~ Group_3Level,

```

```

        data = vg_data_clean, method = "holm")
print(dunn_results$res)
cat("\n")
}

} else if(use_welch) {
  cat("Variances are unequal. Using Welch's ANOVA.\n\n")
  Welch_anova <- oneway.test(Video_Game_Experience_Quantified ~ Group_3Level,
                             data = vg_data_clean, var.equal = FALSE)
  cat("Welch's ANOVA Results:\n")
  cat("F =", Welch_anova$statistic, "\n")
  cat("df1 =", Welch_anova$parameter[1], ", df2 =", Welch_anova$parameter[2], "\n")
  cat("p-value =", Welch_anova$p.value, "\n\n")

  p_value <- Welch_anova$p.value

  # Post-hoc: Games-Howell
  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Games-Howell):\n")
    games_howell <- games_howell_test(vg_data_clean,
                                         Video_Game_Experience_Quantified ~ Group_3Level)
    print(games_howell)
    cat("\n")
  }

} else {
  cat("Assumptions met. Using standard one-way ANOVA.\n\n")
  anova_model <- aov(Video_Game_Experience_Quantified ~ Group_3Level,
                      data = vg_data_clean)
  anova_summary <- summary(anova_model)

  cat("One-way ANOVA Results:\n")
  print(anova_summary)
  cat("\n")

  # Effect size (eta-squared)
  eta_sq <- effectsize::eta_squared(anova_model)
  cat("Eta-squared =", round(eta_sq$Eta2[1], 3), "\n\n")

  p_value <- anova_summary[[1]]$`Pr(>F)`[1]

  # Post-hoc: Tukey HSD
  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Tukey HSD):\n")
    tukey_results <- TukeyHSD(anova_model)
    print(tukey_results)
    cat("\n")
  }
}

## Data violates normality assumption. Using Kruskal-Wallis test.
## Kruskal-Wallis Test Results:

```

```

## H statistic = 12.37901
## df = 2
## p-value = 0.002050841
## Epsilon-squared = 0.21
##
## Post-hoc Comparisons (Dunn's test with Holm correction):
##          Comparison      Z     P.unadj      P.adj
## 1 OA_High - OA_Low -0.5894117 0.555585130 0.555585130
## 2 OA_High - YA -3.1724463 0.001511605 0.004534816
## 3 OA_Low - YA -2.4918523 0.012707889 0.025415778

# Interpretation
cat("\n--- INTERPRETATION ---\n")

##
## --- INTERPRETATION ---

if(p_value < 0.05) {
  cat("Result: SIGNIFICANT difference in video game usage across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in video game usage across groups (p >= 0.05)\n")
}

## Result: SIGNIFICANT difference in video game usage across groups (p < 0.05)
## See post-hoc tests above for pairwise comparisons

```

Test 5: Weekly Exercise Frequency

```

cat("Exercise Frequency (numeric variable)\n\n")

## Exercise Frequency (numeric variable)

# Summary statistics by group
exercise_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(Exercise_Quantified)),
    Mean = mean(Exercise_Quantified, na.rm = TRUE),
    SD = sd(Exercise_Quantified, na.rm = TRUE),
    Median = median(Exercise_Quantified, na.rm = TRUE),
    Q1 = quantile(Exercise_Quantified, 0.25, na.rm = TRUE),
    Q3 = quantile(Exercise_Quantified, 0.75, na.rm = TRUE),
    Min = min(Exercise_Quantified, na.rm = TRUE),
    Max = max(Exercise_Quantified, na.rm = TRUE)
  )

cat("Summary Statistics:\n")

## Summary Statistics:

```

```

print(exercise_summary)

## # A tibble: 3 x 9
##   Group_3Level     n   Mean    SD Median    Q1    Q3   Min   Max
##   <fct>     <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 YA          30  4.69  3.30    3.9  2.62    6    0   15
## 2 OA_High      15  4.87  3.80    4    2.25    6    0   14
## 3 OA_Low       15  5.20  2.56    5    3.25    7    0   10

cat("\n")

# Filter out NA values for testing
exercise_data_clean <- merged_data %>%
  filter(!is.na(Exercise_Quantified), !is.na(Group_3Level))

# Check normality
cat("Normality Tests (Shapiro-Wilk):\n")

## Normality Tests (Shapiro-Wilk):

for(grp in levels(exercise_data_clean$Group_3Level)) {
  data_subset <- exercise_data_clean %>% filter(Group_3Level == grp) %>%
    pull(Exercise_Quantified)
  shapiro_result <- shapiro.test(data_subset)
  cat(grp, ": W =", round(shapiro_result$statistic, 4), ", p =", shapiro_result$p.value)
  if(shapiro_result$p.value < 0.05) cat(" (non-normal)") else cat(" (normal)")
  cat("\n")
}

## YA : W = 0.889 , p = 0.004559315 (non-normal)
## OA_High : W = 0.8968 , p = 0.0849313 (normal)
## OA_Low : W = 0.98 , p = 0.9696899 (normal)

cat("\n")

# Check homogeneity of variance
levene_test <- leveneTest(Exercise_Quantified ~ Group_3Level, data = exercise_data_clean)
cat("Levene's Test for Homogeneity of Variance:\n")

## Levene's Test for Homogeneity of Variance:

cat("F =", round(levene_test$`F value`[1], 4), ", p =", levene_test$`Pr(>F)`[1])

## F = 0.4105 , p = 0.6652644

if(levene_test$`Pr(>F)`[1] < 0.05) cat(" (variances unequal)") else cat(" (variances equal)")

## (variances equal)

```

```

cat("\n\n")

# Decide which test to use
use_welch <- levene_test$`Pr(>F)`[1] < 0.05
normality_check <- exercise_data_clean %>%
  group_by(Group_3Level) %>%
  summarise(p_val = shapiro.test(Exercise_Quantified)$p.value) %>%
  pull(p_val)
use_nonparametric <- any(normality_check < 0.05)

if(use_nonparametric) {
  cat("  Data violates normality assumption. Using Kruskal-Wallis test.\n\n")
  kw_test <- kruskal.test(Exercise_Quantified ~ Group_3Level, data = exercise_data_clean)
  cat("Kruskal-Wallis Test Results:\n")
  cat("H statistic =", kw_test$statistic, "\n")
  cat("df =", kw_test$parameter, "\n")
  cat("p-value =", kw_test$p.value, "\n")

  epsilon_sq <- kw_test$statistic / ((nrow(exercise_data_clean)^2 - 1) /
                                         (nrow(exercise_data_clean) + 1))
  cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

  p_value <- kw_test$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
    dunn_results <- dunnTest(Exercise_Quantified ~ Group_3Level,
                               data = exercise_data_clean, method = "holm")
    print(dunn_results$res)
    cat("\n")
  }
}

} else if(use_welch) {
  cat("Variances are unequal. Using Welch's ANOVA.\n\n")
  welch_anova <- oneway.test(Exercise_Quantified ~ Group_3Level,
                             data = exercise_data_clean, var.equal = FALSE)
  cat("Welch's ANOVA Results:\n")
  cat("F =", welch_anova$statistic, "\n")
  cat("df1 =", welch_anova$parameter[1], ", df2 =", welch_anova$parameter[2], "\n")
  cat("p-value =", welch_anova$p.value, "\n\n")

  p_value <- welch_anova$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Games-Howell):\n")
    games_howell <- games_howell_test(exercise_data_clean,
                                         Exercise_Quantified ~ Group_3Level)
    print(games_howell)
    cat("\n")
  }
}

} else {
  cat("Assumptions met. Using standard one-way ANOVA.\n\n")
}

```

```

anova_model <- aov(Exercise_Quantified ~ Group_3Level, data = exercise_data_clean)
anova_summary <- summary(anova_model)

cat("One-way ANOVA Results:\n")
print(anova_summary)
cat("\n")

eta_sq <- effectsize::eta_squared(anova_model)
cat("Eta-squared =", round(eta_sq$Eta2[1], 3), "\n\n")

p_value <- anova_summary[[1]]$`Pr(>F)`[1]

if(p_value < 0.05) {
  cat("Post-hoc Comparisons (Tukey HSD):\n")
  tukey_results <- TukeyHSD(anova_model)
  print(tukey_results)
  cat("\n")
}
}

## Data violates normality assumption. Using Kruskal-Wallis test.
##
## Kruskal-Wallis Test Results:
## H statistic = 1.378944
## df = 2
## p-value = 0.501841
## Epsilon-squared = 0.023

cat("\n--- INTERPRETATION ---\n")

##
## --- INTERPRETATION ---

if(p_value < 0.05) {
  cat("Result: SIGNIFICANT difference in exercise frequency across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in exercise frequency across groups (p >= 0.05)\n")
}

## Result: NO significant difference in exercise frequency across groups (p >= 0.05)

```

Test 6: SBSOD Scores

```

cat("SBSOD: Santa Barbara Sense of Direction Scale\n\n")

## SBSOD: Santa Barbara Sense of Direction Scale

```

```

# Summary statistics by group
sbsod_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(SBSOD)),
    Mean = mean(SBSOD, na.rm = TRUE),
    SD = sd(SBSOD, na.rm = TRUE),
    Median = median(SBSOD, na.rm = TRUE),
    Q1 = quantile(SBSOD, 0.25, na.rm = TRUE),
    Q3 = quantile(SBSOD, 0.75, na.rm = TRUE),
    Min = min(SBSOD, na.rm = TRUE),
    Max = max(SBSOD, na.rm = TRUE)
  )

cat("Summary Statistics:\n")

## Summary Statistics:

print(sbsod_summary)

## # A tibble: 3 x 9
##   Group_3Level     n   Mean    SD Median    Q1    Q3   Min   Max
##   <fct>     <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 YA          30  4.35  1.10    4.5  3.55  5.32  2.13   6
## 2 OA_High     15  5.27  0.900   5.07  4.96  5.7   3.6   6.87
## 3 OA_Low      15  4.90  0.943   5.2   4.2   5.46  3.27  6.13

cat("\n")

# Filter out NA values for testing
sbsod_data_clean <- merged_data %>%
  filter(!is.na(SBSOD), !is.na(Group_3Level))

# Check normality
cat("Normality Tests (Shapiro-Wilk):\n")

## Normality Tests (Shapiro-Wilk):

for(grp in levels(sbsod_data_clean$Group_3Level)) {
  data_subset <- sbsod_data_clean %>% filter(Group_3Level == grp) %>% pull(SBSOD)
  shapiro_result <- shapiro.test(data_subset)
  cat(grp, ": W =", round(shapiro_result$statistic, 4), ", p =", shapiro_result$p.value)
  if(shapiro_result$p.value < 0.05) cat(" (non-normal)") else cat(" (normal)")
  cat("\n")
}

## YA : W = 0.9522 , p = 0.1939842 (normal)
## OA_High : W = 0.9521 , p = 0.5579543 (normal)
## OA_Low : W = 0.9245 , p = 0.22579 (normal)

```

```

cat("\n")

# Check homogeneity of variance
levene_test <- leveneTest(SBSOD ~ Group_3Level, data = sbsod_data_clean)
cat("Levene's Test for Homogeneity of Variance:\n")

## Levene's Test for Homogeneity of Variance:

cat("F =", round(levene_test$`F value`[1], 4), ", p =", levene_test$`Pr(>F)`[1])

## F = 1.1152 , p = 0.3348988

if(levene_test$`Pr(>F)`[1] < 0.05) cat(" (variances unequal)") else cat(" (variances equal)")

## (variances equal)

cat("\n\n")

# Decide which test to use
use_welch <- levene_test$`Pr(>F)`[1] < 0.05
normality_check <- sbsod_data_clean %>%
  group_by(Group_3Level) %>%
  summarise(p_val = shapiro.test(SBSOD)$p.value) %>%
  pull(p_val)
use_nonparametric <- any(normality_check < 0.05)

if(use_nonparametric) {
  cat(" Data violates normality assumption. Using Kruskal-Wallis test.\n\n")
  kw_test <- kruskal.test(SBSOD ~ Group_3Level, data = sbsod_data_clean)
  cat("Kruskal-Wallis Test Results:\n")
  cat("H statistic =", kw_test$statistic, "\n")
  cat("df =", kw_test$parameter, "\n")
  cat("p-value =", kw_test$p.value, "\n")

  epsilon_sq <- kw_test$statistic / ((nrow(sbsod_data_clean)^2 - 1) /
                                         (nrow(sbsod_data_clean) + 1))
  cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

  p_value <- kw_test$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
    dunn_results <- dunnTest(SBSOD ~ Group_3Level, data = sbsod_data_clean, method = "holm")
    print(dunn_results$res)
    cat("\n")
  }
}

} else if(use_welch) {
  cat("Variances are unequal. Using Welch's ANOVA.\n\n")
  Welch_anova <- oneway.test(SBSOD ~ Group_3Level, data = sbsod_data_clean, var.equal = FALSE)
  cat("Welch's ANOVA Results:\n")
}

```

```

cat("F =", welch_anova$statistic, "\n")
cat("df1 =", welch_anova$parameter[1], ", df2 =", welch_anova$parameter[2], "\n")
cat("p-value =", welch_anova$p.value, "\n\n")

p_value <- welch_anova$p.value

if(p_value < 0.05) {
  cat("Post-hoc Comparisons (Games-Howell):\n")
  games_howell <- games_howell_test(sbsod_data_clean, SBSOD ~ Group_3Level)
  print(games_howell)
  cat("\n")
}

} else {
  cat("Assumptions met. Using standard one-way ANOVA.\n\n")
  anova_model <- aov(SBSOD ~ Group_3Level, data = sbsod_data_clean)
  anova_summary <- summary(anova_model)

  cat("One-way ANOVA Results:\n")
  print(anova_summary)
  cat("\n")

  eta_sq <- effectsize::eta_squared(anova_model)
  cat("Eta-squared =", round(eta_sq$Eta2[1], 3), "\n\n")

  p_value <- anova_summary[[1]]$`Pr(>F)`[1]

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Tukey HSD):\n")
    tukey_results <- TukeyHSD(anova_model)
    print(tukey_results)
    cat("\n")
  }
}

```

```

## Assumptions met. Using standard one-way ANOVA.
##
## One-way ANOVA Results:
##           Df Sum Sq Mean Sq F value Pr(>F)
## Group_3Level  2   9.11   4.556   4.402 0.0167 *
## Residuals    57  59.00   1.035
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Eta-squared = 0.134
##
## Post-hoc Comparisons (Tukey HSD):
##   Tukey multiple comparisons of means
##   95% family-wise confidence level
##
## Fit: aov(formula = SBSOD ~ Group_3Level, data = sbsod_data_clean)
##
## $Group_3Level

```

```

##          diff      lwr      upr     p adj
## OA_High-YA    0.9200000  0.1458110 1.6941890 0.0160652
## OA_Low-YA     0.5466667 -0.2275224 1.3208557 0.2143167
## OA_Low-OA_High -0.3733333 -1.2672898 0.5206232 0.5767877

cat("\n--- INTERPRETATION ---\n")

## --- INTERPRETATION ---

if(p_value < 0.05) {
  cat("Result: SIGNIFICANT difference in SBSOD scores across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in SBSOD scores across groups (p >= 0.05)\n")
}

## Result: SIGNIFICANT difference in SBSOD scores across groups (p < 0.05)
## See post-hoc tests above for pairwise comparisons

```

Test 7: PSQI Scores

```

cat("PSQI: Pittsburgh Sleep Quality Index\n")

## PSQI: Pittsburgh Sleep Quality Index

cat("Note: Higher scores indicate WORSE sleep quality (range 0-21)\n\n")

## Note: Higher scores indicate WORSE sleep quality (range 0-21)

# Summary statistics by group
psqi_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(PSQI)),
    Mean = mean(PSQI, na.rm = TRUE),
    SD = sd(PSQI, na.rm = TRUE),
    Median = median(PSQI, na.rm = TRUE),
    Q1 = quantile(PSQI, 0.25, na.rm = TRUE),
    Q3 = quantile(PSQI, 0.75, na.rm = TRUE),
    Min = min(PSQI, na.rm = TRUE),
    Max = max(PSQI, na.rm = TRUE)
  )

cat("Summary Statistics:\n")

## Summary Statistics:

```

```

print(psqi_summary)

## # A tibble: 3 x 9
##   Group_3Level     n   Mean    SD Median    Q1    Q3   Min   Max
##   <fct>     <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 YA          30   5.1   1.95    5   4     6    2    12
## 2 OA_High      15   5.27  2.71    6   3.5   7.5   1     9
## 3 OA_Low       15   4.4   3.11    4   2     7    0    11

cat("\n")

# Filter out NA values for testing
psqi_data_clean <- merged_data %>%
  filter(!is.na(PSQI), !is.na(Group_3Level))

# Check normality
cat("Normality Tests (Shapiro-Wilk):\n")

## Normality Tests (Shapiro-Wilk):

for(grp in levels(psqi_data_clean$Group_3Level)) {
  data_subset <- psqi_data_clean %>% filter(Group_3Level == grp) %>% pull(PSQI)
  shapiro_result <- shapiro.test(data_subset)
  cat(grp, ": W =", round(shapiro_result$statistic, 4), ", p =", shapiro_result$p.value)
  if(shapiro_result$p.value < 0.05) cat(" (non-normal)") else cat(" (normal)")
  cat("\n")
}

## YA : W = 0.8731 , p = 0.001963064 (non-normal)
## OA_High : W = 0.9381 , p = 0.3589076 (normal)
## OA_Low : W = 0.9315 , p = 0.2874813 (normal)

cat("\n")

# Check homogeneity of variance
levene_test <- leveneTest(PSQI ~ Group_3Level, data = psqi_data_clean)
cat("Levene's Test for Homogeneity of Variance:\n")

## Levene's Test for Homogeneity of Variance:

cat("F =", round(levene_test$`F value`[1], 4), ", p =", levene_test$`Pr(>F)`[1])

## F = 3.6327 , p = 0.03274076

if(levene_test$`Pr(>F)`[1] < 0.05) cat(" (variances unequal)") else cat(" (variances equal)")

## (variances unequal)

```

```

cat("\n\n")

# Decide which test to use
use_welch <- levene_test$`Pr(>F)`[1] < 0.05
normality_check <- psqi_data_clean %>%
  group_by(Group_3Level) %>%
  summarise(p_val = shapiro.test(PSQI)$p.value) %>%
  pull(p_val)
use_nonparametric <- any(normality_check < 0.05)

if(use_nonparametric) {
  cat("  Data violates normality assumption. Using Kruskal-Wallis test.\n\n")
  kw_test <- kruskal.test(PSQI ~ Group_3Level, data = psqi_data_clean)
  cat("Kruskal-Wallis Test Results:\n")
  cat("H statistic =", kw_test$statistic, "\n")
  cat("df =", kw_test$parameter, "\n")
  cat("p-value =", kw_test$p.value, "\n")

  epsilon_sq <- kw_test$statistic / ((nrow(psqi_data_clean)^2 - 1) /
                                         (nrow(psqi_data_clean) + 1))
  cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

  p_value <- kw_test$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
    dunn_results <- dunnTest(PSQI ~ Group_3Level, data = psqi_data_clean, method = "holm")
    print(dunn_results$res)
    cat("\n")
  }
}

} else if(use_welch) {
  cat("Variances are unequal. Using Welch's ANOVA.\n\n")
  welch_anova <- oneway.test(PSQI ~ Group_3Level, data = psqi_data_clean, var.equal = FALSE)
  cat("Welch's ANOVA Results:\n")
  cat("F =", welch_anova$statistic, "\n")
  cat("df1 =", welch_anova$parameter[1], ", df2 =", welch_anova$parameter[2], "\n")
  cat("p-value =", welch_anova$p.value, "\n\n")

  p_value <- welch_anova$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Games-Howell):\n")
    games_howell <- games_howell_test(psqi_data_clean, PSQI ~ Group_3Level)
    print(games_howell)
    cat("\n")
  }
}

} else {
  cat("Assumptions met. Using standard one-way ANOVA.\n\n")
  anova_model <- aov(PSQI ~ Group_3Level, data = psqi_data_clean)
  anova_summary <- summary(anova_model)
}

```

```

cat("One-way ANOVA Results:\n")
print(anova_summary)
cat("\n")

eta_sq <- effectsize::eta_squared(anova_model)
cat("Eta-squared =", round(eta_sq$Eta2[1], 3), "\n\n")

p_value <- anova_summary[[1]]$`Pr(>F)`[1]

if(p_value < 0.05) {
  cat("Post-hoc Comparisons (Tukey HSD):\n")
  tukey_results <- TukeyHSD(anova_model)
  print(tukey_results)
  cat("\n")
}
}

## Data violates normality assumption. Using Kruskal-Wallis test.
## 
## Kruskal-Wallis Test Results:
## H statistic = 1.355847
## df = 2
## p-value = 0.5076701
## Epsilon-squared = 0.023

cat("\n--- INTERPRETATION ---\n")

## 
## --- INTERPRETATION ---

if(p_value < 0.05) {
  cat("Result: SIGNIFICANT difference in sleep quality across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in sleep quality across groups (p >= 0.05)\n")
}

## Result: NO significant difference in sleep quality across groups (p >= 0.05)

```

Test 8: Trails A Completion Time

```

cat("Trails A: Completion Time\n")

## Trails A: Completion Time

cat("Note: Trail Making Test Part A measures processing speed\n\n")

## Note: Trail Making Test Part A measures processing speed

```

```

# Summary statistics by group
trails_a_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(Trails_A_CT)),
    Mean = mean(Trails_A_CT, na.rm = TRUE),
    SD = sd(Trails_A_CT, na.rm = TRUE),
    Median = median(Trails_A_CT, na.rm = TRUE),
    Q1 = quantile(Trails_A_CT, 0.25, na.rm = TRUE),
    Q3 = quantile(Trails_A_CT, 0.75, na.rm = TRUE),
    Min = min(Trails_A_CT, na.rm = TRUE),
    Max = max(Trails_A_CT, na.rm = TRUE)
  )

cat("Summary Statistics:\n")

## Summary Statistics:

print(trails_a_summary)

## # A tibble: 3 x 9
##   Group_3Level     n   Mean    SD Median    Q1    Q3   Min   Max
##   <fct>     <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 YA          30  25.9  4.56  24.6  22.9  30.0  17.1  35.2
## 2 OA_High     15  33.5  6.29  35.7  28.9  36.4  22.4  45
## 3 OA_Low      15  35.0  7.50  34.1  30.4  36    24.3  48.6

cat("\n")

# Filter out NA values for testing
trails_a_clean <- merged_data %>%
  filter(!is.na(Trails_A_CT), !is.na(Group_3Level))

# Check normality
cat("Normality Tests (Shapiro-Wilk):\n")

## Normality Tests (Shapiro-Wilk):

for(grp in levels(trails_a_clean$Group_3Level)) {
  data_subset <- trails_a_clean %>% filter(Group_3Level == grp) %>% pull(Trails_A_CT)
  shapiro_result <- shapiro.test(data_subset)
  cat(grp, ": W =", round(shapiro_result$statistic, 4), ", p =", shapiro_result$p.value)
  if(shapiro_result$p.value < 0.05) cat(" (non-normal)") else cat(" (normal)")
  cat("\n")
}

## YA : W = 0.9444 , p = 0.119447 (normal)
## OA_High : W = 0.9229 , p = 0.2131439 (normal)
## OA_Low : W = 0.8726 , p = 0.03683581 (non-normal)

```

```

cat("\n")

# Check homogeneity of variance
levene_test <- leveneTest(Trails_A_CT ~ Group_3Level, data = trails_a_clean)
cat("Levene's Test for Homogeneity of Variance:\n")

## Levene's Test for Homogeneity of Variance:

cat("F =", round(levene_test$`F value`[1], 4), ", p =", levene_test$`Pr(>F)`[1])

## F = 0.9064 , p = 0.4097212

if(levene_test$`Pr(>F)`[1] < 0.05) cat(" (variances unequal)") else cat(" (variances equal)")

## (variances equal)

cat("\n\n")

# Decide which test to use
use_welch <- levene_test$`Pr(>F)`[1] < 0.05
normality_check <- trails_a_clean %>%
  group_by(Group_3Level) %>%
  summarise(p_val = shapiro.test(Trails_A_CT)$p.value) %>%
  pull(p_val)
use_nonparametric <- any(normality_check < 0.05)

if(use_nonparametric) {
  cat(" Data violates normality assumption. Using Kruskal-Wallis test.\n\n")
  kw_test <- kruskal.test(Trails_A_CT ~ Group_3Level, data = trails_a_clean)
  cat("Kruskal-Wallis Test Results:\n")
  cat("H statistic =", kw_test$statistic, "\n")
  cat("df =", kw_test$parameter, "\n")
  cat("p-value =", kw_test$p.value, "\n")

  epsilon_sq <- kw_test$statistic / ((nrow(trails_a_clean)^2 - 1) / (nrow(trails_a_clean) + 1))
  cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

  p_value <- kw_test$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
    dunn_results <- dunnTest(Trails_A_CT ~ Group_3Level, data = trails_a_clean, method = "holm")
    print(dunn_results$res)
    cat("\n")
  }
}

} else if(use_welch) {
  cat("Variances are unequal. Using Welch's ANOVA.\n\n")
  welch_anova <- oneway.test(Trails_A_CT ~ Group_3Level, data = trails_a_clean, var.equal = FALSE)
  cat("Welch's ANOVA Results:\n")
  cat("F =", welch_anova$statistic, "\n")
}

```

```

cat("df1 =", welch_anova$parameter[1], ", df2 =", welch_anova$parameter[2], "\n")
cat("p-value =", welch_anova$p.value, "\n\n")

p_value <- welch_anova$p.value

if(p_value < 0.05) {
  cat("Post-hoc Comparisons (Games-Howell):\n")
  games_howell <- games_howell_test(trails_a_clean, Trails_A_CT ~ Group_3Level)
  print(games_howell)
  cat("\n")
}

} else {
  cat("Assumptions met. Using standard one-way ANOVA.\n\n")
  anova_model <- aov(Trails_A_CT ~ Group_3Level, data = trails_a_clean)
  anova_summary <- summary(anova_model)

  cat("One-way ANOVA Results:\n")
  print(anova_summary)
  cat("\n")

  eta_sq <- effectsize::eta_squared(anova_model)
  cat("Eta-squared =", round(eta_sq$Eta2[1], 3), "\n\n")

  p_value <- anova_summary[[1]]$`Pr(>F)`[1]

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Tukey HSD):\n")
    tukey_results <- TukeyHSD(anova_model)
    print(tukey_results)
    cat("\n")
  }
}
}

## Data violates normality assumption. Using Kruskal-Wallis test.
##
## Kruskal-Wallis Test Results:
## H statistic = 21.90834
## df = 2
## p-value = 1.748495e-05
## Epsilon-squared = 0.371
##
## Post-hoc Comparisons (Dunn's test with Holm correction):
## Comparison Z P.unadj P.adj
## 1 OA_High - OA_Low -0.0888703 0.9291849863 0.9291849863
## 2 OA_High - YA 3.7697239 0.0001634282 0.0003268565
## 3 OA_Low - YA 3.8723425 0.0001077943 0.0003233830

cat("\n--- INTERPRETATION ---\n")

##
## --- INTERPRETATION ---

```

```

if(p_value < 0.05) {
  cat("Result: SIGNIFICANT difference in Trails A completion time across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in Trails A completion time across groups (p >= 0.05)\n")
}

```

```

## Result: SIGNIFICANT difference in Trails A completion time across groups (p < 0.05)
## See post-hoc tests above for pairwise comparisons

```

Test 9: Trails B Completion Time

```
cat("Trails B: Completion Time\n")
```

```
## Trails B: Completion Time
```

```
cat("Note: Trail Making Test Part B measures executive function and cognitive flexibility\n\n")
```

```
## Note: Trail Making Test Part B measures executive function and cognitive flexibility
```

```
# Summary statistics by group
trails_b_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(Trails_B_CT)),
    Mean = mean(Trails_B_CT, na.rm = TRUE),
    SD = sd(Trails_B_CT, na.rm = TRUE),
    Median = median(Trails_B_CT, na.rm = TRUE),
    Q1 = quantile(Trails_B_CT, 0.25, na.rm = TRUE),
    Q3 = quantile(Trails_B_CT, 0.75, na.rm = TRUE),
    Min = min(Trails_B_CT, na.rm = TRUE),
    Max = max(Trails_B_CT, na.rm = TRUE)
  )
```

```
cat("Summary Statistics:\n")
```

```
## Summary Statistics:
```

```
print(trails_b_summary)
```

```
## # A tibble: 3 x 9
##   Group_3Level     n   Mean     SD Median     Q1     Q3   Min   Max
##   <fct>     <int> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 YA          30  36.9   11.1   35.5  29.7   38    21.5  75.1
## 2 OA_High     15  56.5   23.7   48.4   46     58.6  33.5  132.
## 3 OA_Low      15  62.5   33.8   48.9   43.8   70.0  26.4  160.
```

```

cat("\n")

# Filter out NA values for testing
trails_b_clean <- merged_data %>%
  filter(!is.na(Trails_B_CT), !is.na(Group_3Level))

# Check normality
cat("Normality Tests (Shapiro-Wilk):\n")

## Normality Tests (Shapiro-Wilk):

for(grp in levels(trails_b_clean$Group_3Level)) {
  data_subset <- trails_b_clean %>% filter(Group_3Level == grp) %>% pull(Trails_B_CT)
  shapiro_result <- shapiro.test(data_subset)
  cat(grp, ": W =", round(shapiro_result$statistic, 4), ", p =", shapiro_result$p.value)
  if(shapiro_result$p.value < 0.05) cat(" (non-normal)") else cat(" (normal)")
  cat("\n")
}

## YA : W = 0.8805 , p = 0.002891261 (non-normal)
## OA_High : W = 0.7164 , p = 0.0003740666 (non-normal)
## OA_Low : W = 0.8015 , p = 0.003848452 (non-normal)

cat("\n")

# Check homogeneity of variance
levene_test <- leveneTest(Trails_B_CT ~ Group_3Level, data = trails_b_clean)
cat("Levene's Test for Homogeneity of Variance:\n")

## Levene's Test for Homogeneity of Variance:

cat("F =", round(levene_test$`F value`[1], 4), ", p =", levene_test$`Pr(>F)`[1])

## F = 2.6216 , p = 0.08143511

if(levene_test$`Pr(>F)`[1] < 0.05) cat(" (variances unequal)") else cat(" (variances equal)")

## (variances equal)

cat("\n\n")

# Decide which test to use
use_welch <- levene_test$`Pr(>F)`[1] < 0.05
normality_check <- trails_b_clean %>%
  group_by(Group_3Level) %>%
  summarise(p_val = shapiro.test(Trails_B_CT)$p.value) %>%
  pull(p_val)
use_nonparametric <- any(normality_check < 0.05)

```

```

if(use_nonparametric) {
  cat(" Data violates normality assumption. Using Kruskal-Wallis test.\n\n")
  kw_test <- kruskal.test(Trails_B_CT ~ Group_3Level, data = trails_b_clean)
  cat("Kruskal-Wallis Test Results:\n")
  cat("H statistic =", kw_test$statistic, "\n")
  cat("df =", kw_test$parameter, "\n")
  cat("p-value =", kw_test$p.value, "\n")

  epsilon_sq <- kw_test$statistic / ((nrow(trails_b_clean)^2 - 1) / (nrow(trails_b_clean) + 1))
  cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

  p_value <- kw_test$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
    dunn_results <- dunnTest(Trails_B_CT ~ Group_3Level, data = trails_b_clean, method = "holm")
    print(dunn_results$res)
    cat("\n")
  }
}

} else if(use_welch) {
  cat("Variances are unequal. Using Welch's ANOVA.\n\n")
  welch_anova <- oneway.test(Trails_B_CT ~ Group_3Level, data = trails_b_clean, var.equal = FALSE)
  cat("Welch's ANOVA Results:\n")
  cat("F =", welch_anova$statistic, "\n")
  cat("df1 =", welch_anova$parameter[1], ", df2 =", welch_anova$parameter[2], "\n")
  cat("p-value =", welch_anova$p.value, "\n\n")

  p_value <- welch_anova$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Games-Howell):\n")
    games_howell <- games_howell_test(trails_b_clean, Trails_B_CT ~ Group_3Level)
    print(games_howell)
    cat("\n")
  }
}

} else {
  cat("Assumptions met. Using standard one-way ANOVA.\n\n")
  anova_model <- aov(Trails_B_CT ~ Group_3Level, data = trails_b_clean)
  anova_summary <- summary(anova_model)

  cat("One-way ANOVA Results:\n")
  print(anova_summary)
  cat("\n")

  eta_sq <- effectsize::eta_squared(anova_model)
  cat("Eta-squared =", round(eta_sq$Eta2[1], 3), "\n\n")

  p_value <- anova_summary[[1]]$`Pr(>F)`[1]

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Tukey HSD):\n")
  }
}

```

```

tukey_results <- TukeyHSD(anova_model)
print(tukey_results)
cat("\n")
}

## Data violates normality assumption. Using Kruskal-Wallis test.
##
## Kruskal-Wallis Test Results:
## H statistic = 19.48377
## df = 2
## p-value = 5.876973e-05
## Epsilon-squared = 0.33
##
## Post-hoc Comparisons (Dunn's test with Holm correction):
## Comparison Z P.unadj P.adj
## 1 OA_High - OA_Low 0.06795775 0.9458192643 0.9458192643
## 2 OA_High - YA 3.64285852 0.0002696271 0.0008088812
## 3 OA_Low - YA 3.56438766 0.0003647066 0.0007294132

cat("\n--- INTERPRETATION ---\n")

##
## --- INTERPRETATION ---

if(p_value < 0.05) {
  cat("Result: SIGNIFICANT difference in Trails B completion time across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in Trails B completion time across groups (p >= 0.05)\n")
}

## Result: SIGNIFICANT difference in Trails B completion time across groups (p < 0.05)
## See post-hoc tests above for pairwise comparisons

```

Test 10: Trails B-A Difference

```

cat("Trails B-A Difference\n")

## Trails B-A Difference

cat("Note: Difference score (Trails B - Trails A) isolates executive function component\n\n")

## Note: Difference score (Trails B - Trails A) isolates executive function component

```

```

# Calculate Trails_BA_Diff
merged_data <- merged_data %>%
  mutate(Trails_BA_Diff = Trails_B_CT - Trails_A_CT)

# Summary statistics by group
trails_ba_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(Trails_BA_Diff)),
    Mean = mean(Trails_BA_Diff, na.rm = TRUE),
    SD = sd(Trails_BA_Diff, na.rm = TRUE),
    Median = median(Trails_BA_Diff, na.rm = TRUE),
    Q1 = quantile(Trails_BA_Diff, 0.25, na.rm = TRUE),
    Q3 = quantile(Trails_BA_Diff, 0.75, na.rm = TRUE),
    Min = min(Trails_BA_Diff, na.rm = TRUE),
    Max = max(Trails_BA_Diff, na.rm = TRUE)
  )

cat("Summary Statistics:\n")

## Summary Statistics:

print(trails_ba_summary)

## # A tibble: 3 x 9
##   Group_3Level     n   Mean    SD Median    Q1    Q3   Min   Max
##   <fct>     <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 YA           30  11.0  9.81  8.45  5.13 14.8 -2.5  39.9
## 2 OA_High      15  23.0 24.8   18   11.3 22.5 -3     104.
## 3 OA_Low       15  27.6 28.4   17.2  12.1 32.4 -2.20 111.

cat("\n")

# Filter out NA values for testing
trails_ba_clean <- merged_data %>%
  filter(!is.na(Trails_BA_Diff), !is.na(Group_3Level))

# Check normality
cat("Normality Tests (Shapiro-Wilk):\n")

## Normality Tests (Shapiro-Wilk):

for(grp in levels(trails_ba_clean$Group_3Level)) {
  data_subset <- trails_ba_clean %>% filter(Group_3Level == grp) %>% pull(Trails_BA_Diff)
  shapiro_result <- shapiro.test(data_subset)
  cat(grp, ": W =", round(shapiro_result$statistic, 4), ", p =", shapiro_result$p.value)
  if(shapiro_result$p.value < 0.05) cat(" (non-normal)") else cat(" (normal)")
  cat("\n")
}

## YA : W = 0.8963 , p = 0.006828979 (non-normal)
## OA_High : W = 0.6913 , p = 0.0002009516 (non-normal)
## OA_Low : W = 0.772 , p = 0.001641013 (non-normal)

```

```

cat("\n")

# Check homogeneity of variance
levene_test <- leveneTest(Trails_BA_Diff ~ Group_3Level, data = trails_ba_clean)
cat("Levene's Test for Homogeneity of Variance:\n")

## Levene's Test for Homogeneity of Variance:

cat("F =", round(levene_test$`F value`[1], 4), ", p =", levene_test$`Pr(>F)`[1])

## F = 1.9199 , p = 0.1559831

if(levene_test$`Pr(>F)`[1] < 0.05) cat(" (variances unequal)") else cat(" (variances equal)")

## (variances equal)

cat("\n\n")

# Decide which test to use
use_welch <- levene_test$`Pr(>F)`[1] < 0.05
normality_check <- trails_ba_clean %>%
  group_by(Group_3Level) %>%
  summarise(p_val = shapiro.test(Trails_BA_Diff)$p.value) %>%
  pull(p_val)
use_nonparametric <- any(normality_check < 0.05)

if(use_nonparametric) {
  cat(" Data violates normality assumption. Using Kruskal-Wallis test.\n\n")
  kw_test <- kruskal.test(Trails_BA_Diff ~ Group_3Level, data = trails_ba_clean)
  cat("Kruskal-Wallis Test Results:\n")
  cat("H statistic =", kw_test$statistic, "\n")
  cat("df =", kw_test$parameter, "\n")
  cat("p-value =", kw_test$p.value, "\n")

  epsilon_sq <- kw_test$statistic / ((nrow(trails_ba_clean)^2 - 1) / (nrow(trails_ba_clean) + 1))
  cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

  p_value <- kw_test$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
    dunn_results <- dunnTest(Trails_BA_Diff ~ Group_3Level, data = trails_ba_clean, method = "holm")
    print(dunn_results$res)
    cat("\n")
  }
}

} else if(use_welch) {
  cat("Variances are unequal. Using Welch's ANOVA.\n\n")
  welch_anova <- oneway.test(Trails_BA_Diff ~ Group_3Level, data = trails_ba_clean, var.equal = FALSE)
  cat("Welch's ANOVA Results:\n")
  cat("F =", welch_anova$statistic, "\n")
}

```

```

cat("df1 =", welch_anova$parameter[1], ", df2 =", welch_anova$parameter[2], "\n")
cat("p-value =", welch_anova$p.value, "\n\n")

p_value <- welch_anova$p.value

if(p_value < 0.05) {
  cat("Post-hoc Comparisons (Games-Howell):\n")
  games_howell <- games_howell_test(trails_ba_clean, Trails_BA_Diff ~ Group_3Level)
  print(games_howell)
  cat("\n")
}

} else {
  cat("Assumptions met. Using standard one-way ANOVA.\n\n")
  anova_model <- aov(Trails_BA_Diff ~ Group_3Level, data = trails_ba_clean)
  anova_summary <- summary(anova_model)

  cat("One-way ANOVA Results:\n")
  print(anova_summary)
  cat("\n")

  eta_sq <- effectsize::eta_squared(anova_model)
  cat("Eta-squared =", round(eta_sq$Eta2[1], 3), "\n\n")

  p_value <- anova_summary[[1]]$`Pr(>F)`[1]

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Tukey HSD):\n")
    tukey_results <- TukeyHSD(anova_model)
    print(tukey_results)
    cat("\n")
  }
}
}

## Data violates normality assumption. Using Kruskal-Wallis test.
##
## Kruskal-Wallis Test Results:
## H statistic = 10.02947
## df = 2
## p-value = 0.006639392
## Epsilon-squared = 0.17
##
## Post-hoc Comparisons (Dunn's test with Holm correction):
##          Comparison      Z     P.unadj     P.adj
## 1 OA_High - OA_Low -0.3345334 0.737977123 0.73797712
## 2 OA_High - YA   2.3780723 0.017403415 0.03480683
## 3 OA_Low - YA   2.7643581 0.005703491 0.01711047

cat("\n--- INTERPRETATION ---\n")

##
## --- INTERPRETATION ---

```

```

if(p_value < 0.05) {
  cat("Result: SIGNIFICANT difference in Trails B-A difference across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in Trails B-A difference across groups (p >= 0.05)\n")
}

```

```

## Result: SIGNIFICANT difference in Trails B-A difference across groups (p < 0.05)
## See post-hoc tests above for pairwise comparisons

```

Test 11: Corsi Block Total Score

```

cat("Corsi Block Test: Total Score\n")

## Corsi Block Test: Total Score

cat("Note: Corsi measures visuospatial working memory\n\n")

```

Note: Corsi measures visuospatial working memory

```

# Summary statistics by group
corsi_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(Corsi_Score_Total)),
    Mean = mean(Corsi_Score_Total, na.rm = TRUE),
    SD = sd(Corsi_Score_Total, na.rm = TRUE),
    Median = median(Corsi_Score_Total, na.rm = TRUE),
    Q1 = quantile(Corsi_Score_Total, 0.25, na.rm = TRUE),
    Q3 = quantile(Corsi_Score_Total, 0.75, na.rm = TRUE),
    Min = min(Corsi_Score_Total, na.rm = TRUE),
    Max = max(Corsi_Score_Total, na.rm = TRUE)
  )

```

```
cat("Summary Statistics:\n")
```

Summary Statistics:

```
print(corsi_summary)
```

```

## # A tibble: 3 x 9
##   Group_3Level     n   Mean    SD Median     Q1     Q3   Min   Max
##   <fct>       <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 YA             30  89.2  10.4   87.5  83.2   95    66   115
## 2 OA_High        15  71.4   9.52   75     64.5   77.5   54    87
## 3 OA_Low         15  70.3  10.4    71     61.5   75.5   54    89

```

```

cat("\n")

# Filter out NA values for testing
corsi_clean <- merged_data %>%
  filter(!is.na(Corsi_Score_Total), !is.na(Group_3Level))

# Check normality
cat("Normality Tests (Shapiro-Wilk):\n")

## Normality Tests (Shapiro-Wilk):

for(grp in levels(corsi_clean$Group_3Level)) {
  data_subset <- corsi_clean %>% filter(Group_3Level == grp) %>% pull(Corsi_Score_Total)
  shapiro_result <- shapiro.test(data_subset)
  cat(grp, ": W =", round(shapiro_result$statistic, 4), ", p =", shapiro_result$p.value)
  if(shapiro_result$p.value < 0.05) cat(" (non-normal)") else cat(" (normal)")
  cat("\n")
}

## YA : W = 0.9669 , p = 0.4592182 (normal)
## OA_High : W = 0.9589 , p = 0.6728767 (normal)
## OA_Low : W = 0.9665 , p = 0.8036213 (normal)

cat("\n")

# Check homogeneity of variance
levene_test <- leveneTest(Corsi_Score_Total ~ Group_3Level, data = corsi_clean)
cat("Levene's Test for Homogeneity of Variance:\n")

## Levene's Test for Homogeneity of Variance:

cat("F =", round(levene_test$`F value`[1], 4), ", p =", levene_test$`Pr(>F)`[1])

## F = 0.0811 , p = 0.9221783

if(levene_test$`Pr(>F)`[1] < 0.05) cat(" (variances unequal)") else cat(" (variances equal)")

## (variances equal)

cat("\n\n")

# Decide which test to use
use_welch <- levene_test$`Pr(>F)`[1] < 0.05
normality_check <- corsi_clean %>%
  group_by(Group_3Level) %>%
  summarise(p_val = shapiro.test(Corsi_Score_Total)$p.value) %>%
  pull(p_val)
use_nonparametric <- any(normality_check < 0.05)

```

```

if(use_nonparametric) {
  cat(" Data violates normality assumption. Using Kruskal-Wallis test.\n\n")
  kw_test <- kruskal.test(Corsi_Score_Total ~ Group_3Level, data = corsi_clean)
  cat("Kruskal-Wallis Test Results:\n")
  cat("H statistic =", kw_test$statistic, "\n")
  cat("df =", kw_test$parameter, "\n")
  cat("p-value =", kw_test$p.value, "\n")

  epsilon_sq <- kw_test$statistic / ((nrow(corsi_clean)^2 - 1) / (nrow(corsi_clean) + 1))
  cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

  p_value <- kw_test$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
    dunn_results <- dunnTest(Corsi_Score_Total ~ Group_3Level, data = corsi_clean, method = "holm")
    print(dunn_results$res)
    cat("\n")
  }
}

} else if(use_welch) {
  cat("Variances are unequal. Using Welch's ANOVA.\n\n")
  welch_anova <- oneway.test(Corsi_Score_Total ~ Group_3Level, data = corsi_clean, var.equal = FALSE)
  cat("Welch's ANOVA Results:\n")
  cat("F =", welch_anova$statistic, "\n")
  cat("df1 =", welch_anova$parameter[1], ", df2 =", welch_anova$parameter[2], "\n")
  cat("p-value =", welch_anova$p.value, "\n\n")

  p_value <- welch_anova$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Games-Howell):\n")
    games_howell <- games_howell_test(corsi_clean, Corsi_Score_Total ~ Group_3Level)
    print(games_howell)
    cat("\n")
  }
}

} else {
  cat("Assumptions met. Using standard one-way ANOVA.\n\n")
  anova_model <- aov(Corsi_Score_Total ~ Group_3Level, data = corsi_clean)
  anova_summary <- summary(anova_model)

  cat("One-way ANOVA Results:\n")
  print(anova_summary)
  cat("\n")

  eta_sq <- effectsize::eta_squared(anova_model)
  cat("Eta-squared =", round(eta_sq$Eta2[1], 3), "\n\n")

  p_value <- anova_summary[[1]]$`Pr(>F)`[1]

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Tukey HSD):\n")
  }
}

```

```

tukey_results <- TukeyHSD(anova_model)
print(tukey_results)
cat("\n")
}

## Assumptions met. Using standard one-way ANOVA.
##
## One-way ANOVA Results:
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Group_3Level  2   5050   2525.1     24.2 2.46e-08 ***
## Residuals    57   5948    104.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Eta-squared = 0.459
##
## Post-hoc Comparisons (Tukey HSD):
##   Tukey multiple comparisons of means
##   95% family-wise confidence level
##
## Fit: aov(formula = Corsi_Score_Total ~ Group_3Level, data = corsi_clean)
##
## $Group_3Level
##          diff      lwr      upr   p adj
## OA_High-YA -17.800000 -25.57337 -10.026629 0.00000027
## OA_Low-YA   -18.866667 -26.64004 -11.093295 0.00000008
## OA_Low-OA_High -1.066667 -10.04258  7.909249 0.9559559

cat("\n--- INTERPRETATION ---\n")

##
## --- INTERPRETATION ---

if(p_value < 0.05) {
  cat("Result: SIGNIFICANT difference in Corsi total score across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in Corsi total score across groups (p >= 0.05)\n")
}

## Result: SIGNIFICANT difference in Corsi total score across groups (p < 0.05)
## See post-hoc tests above for pairwise comparisons

```

Test 12: SSQ Pre-Test Scores

```
cat("SSQ Pre-Test: Simulator Sickness Questionnaire (Before VR)\n")
```

```
## SSQ Pre-Test: Simulator Sickness Questionnaire (Before VR)
```

```

cat("Note: Higher scores indicate MORE simulator sickness symptoms\\n\\n")

## Note: Higher scores indicate MORE simulator sickness symptoms

# Summary statistics by group
ssq_pre_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(SSQ_Pre)),
    Mean = mean(SSQ_Pre, na.rm = TRUE),
    SD = sd(SSQ_Pre, na.rm = TRUE),
    Median = median(SSQ_Pre, na.rm = TRUE),
    Q1 = quantile(SSQ_Pre, 0.25, na.rm = TRUE),
    Q3 = quantile(SSQ_Pre, 0.75, na.rm = TRUE),
    Min = min(SSQ_Pre, na.rm = TRUE),
    Max = max(SSQ_Pre, na.rm = TRUE)
  )

cat("Summary Statistics:\\n")

## Summary Statistics:

print(ssq_pre_summary)

## # A tibble: 3 x 9
##   Group_3Level     n   Mean    SD Median    Q1    Q3   Min   Max
##   <fct>     <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 YA           30  2.53  2.27     2    0     4     0     7
## 2 OA_High      15  2.67  4.39     0    0     3     0    13
## 3 OA_Low       15  2.87  2.83     2   0.5    4.5    0     9

cat("\\n")

# Filter out NA values for testing
ssq_pre_clean <- merged_data %>%
  filter(!is.na(SSQ_Pre), !is.na(Group_3Level))

# Check normality
cat("Normality Tests (Shapiro-Wilk):\\n")

## Normality Tests (Shapiro-Wilk):

for(grp in levels(ssq_pre_clean$Group_3Level)) {
  data_subset <- ssq_pre_clean %>% filter(Group_3Level == grp) %>% pull(SSQ_Pre)
  shapiro_result <- shapiro.test(data_subset)
  cat(grp, ": W =", round(shapiro_result$statistic, 4), ", p =", shapiro_result$p.value)
  if(shapiro_result$p.value < 0.05) cat("(non-normal)") else cat("(normal)")
  cat("\\n")
}

```

```

## YA : W = 0.8979 , p = 0.007457235 (non-normal)
## OA_High : W = 0.6747 , p = 0.000135048 (non-normal)
## OA_Low : W = 0.8837 , p = 0.05387559 (normal)

cat("\n")

# Check homogeneity of variance
levene_test <- leveneTest(SSQ_Pre ~ Group_3Level, data = ssq_pre_clean)
cat("Levene's Test for Homogeneity of Variance:\n")

## Levene's Test for Homogeneity of Variance:

cat("F =", round(levene_test$`F value`[1], 4), ", p =", levene_test$`Pr(>F)`[1])

## F = 0.5361 , p = 0.5879252

if(levene_test$`Pr(>F)`[1] < 0.05) cat(" (variances unequal)") else cat(" (variances equal)")

## (variances equal)

cat("\n\n")

# Decide which test to use
use_welch <- levene_test$`Pr(>F)`[1] < 0.05
normality_check <- ssq_pre_clean %>%
  group_by(Group_3Level) %>%
  summarise(p_val = shapiro.test(SSQ_Pre)$p.value) %>%
  pull(p_val)
use_nonparametric <- any(normality_check < 0.05)

if(use_nonparametric) {
  cat(" Data violates normality assumption. Using Kruskal-Wallis test.\n\n")
  kw_test <- kruskal.test(SSQ_Pre ~ Group_3Level, data = ssq_pre_clean)
  cat("Kruskal-Wallis Test Results:\n")
  cat("H statistic =", kw_test$statistic, "\n")
  cat("df =", kw_test$parameter, "\n")
  cat("p-value =", kw_test$p.value, "\n")

  epsilon_sq <- kw_test$statistic / ((nrow(ssq_pre_clean)^2 - 1) / (nrow(ssq_pre_clean) + 1))
  cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

  p_value <- kw_test$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
    dunn_results <- dunnTest(SSQ_Pre ~ Group_3Level, data = ssq_pre_clean, method = "holm")
    print(dunn_results$res)
    cat("\n")
  }
}

} else if(use_welch) {

```

```

cat("Variances are unequal. Using Welch's ANOVA.\n\n")
welch_anova <- oneway.test(SSQ_Pre ~ Group_3Level, data = ssq_pre_clean, var.equal = FALSE)
cat("Welch's ANOVA Results:\n")
cat("F =", welch_anova$statistic, "\n")
cat("df1 =", welch_anova$parameter[1], ", df2 =", welch_anova$parameter[2], "\n")
cat("p-value =", welch_anova$p.value, "\n\n")

p_value <- welch_anova$p.value

if(p_value < 0.05) {
  cat("Post-hoc Comparisons (Games-Howell):\n")
  games_howell <- games_howell_test(ssq_pre_clean, SSQ_Pre ~ Group_3Level)
  print(games_howell)
  cat("\n")
}

} else {
  cat("Assumptions met. Using standard one-way ANOVA.\n\n")
  anova_model <- aov(SSQ_Pre ~ Group_3Level, data = ssq_pre_clean)
  anova_summary <- summary(anova_model)

  cat("One-way ANOVA Results:\n")
  print(anova_summary)
  cat("\n")

  eta_sq <- effectsize::eta_squared(anova_model)
  cat("Eta-squared =", round(eta_sq$Eta2[1], 3), "\n\n")

  p_value <- anova_summary[[1]]$`Pr(>F)`[1]

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Tukey HSD):\n")
    tukey_results <- TukeyHSD(anova_model)
    print(tukey_results)
    cat("\n")
  }
}
}

## Data violates normality assumption. Using Kruskal-Wallis test.
##
## Kruskal-Wallis Test Results:
## H statistic = 1.478232
## df = 2
## p-value = 0.4775359
## Epsilon-squared = 0.025

cat("\n--- INTERPRETATION ---\n")

##
## --- INTERPRETATION ---

```

```

if(p_value < 0.05) {
  cat("Result: SIGNIFICANT difference in pre-test SSQ scores across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in pre-test SSQ scores across groups (p >= 0.05)\n")
}

```

```
## Result: NO significant difference in pre-test SSQ scores across groups (p >= 0.05)
```

Test 13: SSQ Post-Test Scores

```
cat("SSQ Post-Test: Simulator Sickness Questionnaire (After VR)\n")
```

```
## SSQ Post-Test: Simulator Sickness Questionnaire (After VR)
```

```
cat("Note: Higher scores indicate MORE simulator sickness symptoms\n\n")
```

```
## Note: Higher scores indicate MORE simulator sickness symptoms
```

```
# Summary statistics by group
ssq_post_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(SSQ_Post)),
    Mean = mean(SSQ_Post, na.rm = TRUE),
    SD = sd(SSQ_Post, na.rm = TRUE),
    Median = median(SSQ_Post, na.rm = TRUE),
    Q1 = quantile(SSQ_Post, 0.25, na.rm = TRUE),
    Q3 = quantile(SSQ_Post, 0.75, na.rm = TRUE),
    Min = min(SSQ_Post, na.rm = TRUE),
    Max = max(SSQ_Post, na.rm = TRUE)
  )
```

```
cat("Summary Statistics:\n")
```

```
## Summary Statistics:
```

```
print(ssq_post_summary)
```

```
## # A tibble: 3 x 9
##   Group_3Level     n   Mean     SD Median     Q1     Q3   Min   Max
##   <fct>     <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 YA           30  3.53  4.13  1.5    0   6.75    0   14
## 2 OA_High      15  3.4   3.85   2    0.5    5    0   12
## 3 OA_Low       15  4    4.91   2    0   6.5    0   17
```

```

cat("\n")

# Filter out NA values for testing
ssq_post_clean <- merged_data %>%
  filter(!is.na(SSQ_Post), !is.na(Group_3Level))

# Check normality
cat("Normality Tests (Shapiro-Wilk):\n")

## Normality Tests (Shapiro-Wilk):

for(grp in levels(ssq_post_clean$Group_3Level)) {
  data_subset <- ssq_post_clean %>% filter(Group_3Level == grp) %>% pull(SSQ_Post)
  shapiro_result <- shapiro.test(data_subset)
  cat(grp, ": W =", round(shapiro_result$statistic, 4), ", p =", shapiro_result$p.value)
  if(shapiro_result$p.value < 0.05) cat(" (non-normal)") else cat(" (normal)")
  cat("\n")
}

## YA : W = 0.8275 , p = 0.0002166613 (non-normal)
## OA_High : W = 0.8321 , p = 0.009784022 (non-normal)
## OA_Low : W = 0.8175 , p = 0.006227948 (non-normal)

cat("\n")

# Check homogeneity of variance
levene_test <- leveneTest(SSQ_Post ~ Group_3Level, data = ssq_post_clean)
cat("Levene's Test for Homogeneity of Variance:\n")

## Levene's Test for Homogeneity of Variance:

cat("F =", round(levene_test$`F value`[1], 4), ", p =", levene_test$`Pr(>F)`[1])

## F = 0.1982 , p = 0.8207582

if(levene_test$`Pr(>F)`[1] < 0.05) cat(" (variances unequal)") else cat(" (variances equal)")

## (variances equal)

cat("\n\n")

# Decide which test to use
use_welch <- levene_test$`Pr(>F)`[1] < 0.05
normality_check <- ssq_post_clean %>%
  group_by(Group_3Level) %>%
  summarise(p_val = shapiro.test(SSQ_Post)$p.value) %>%
  pull(p_val)
use_nonparametric <- any(normality_check < 0.05)

```

```

if(use_nonparametric) {
  cat("  Data violates normality assumption. Using Kruskal-Wallis test.\n\n")
  kw_test <- kruskal.test(SSQ_Post ~ Group_3Level, data = ssq_post_clean)
  cat("Kruskal-Wallis Test Results:\n")
  cat("H statistic =", kw_test$statistic, "\n")
  cat("df =", kw_test$parameter, "\n")
  cat("p-value =", kw_test$p.value, "\n")

  epsilon_sq <- kw_test$statistic / ((nrow(ssq_post_clean)^2 - 1) / (nrow(ssq_post_clean) + 1))
  cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

  p_value <- kw_test$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
    dunn_results <- dunnTest(SSQ_Post ~ Group_3Level, data = ssq_post_clean, method = "holm")
    print(dunn_results$res)
    cat("\n")
  }
}

} else if(use_welch) {
  cat("Variances are unequal. Using Welch's ANOVA.\n\n")
  welch_anova <- oneway.test(SSQ_Post ~ Group_3Level, data = ssq_post_clean, var.equal = FALSE)
  cat("Welch's ANOVA Results:\n")
  cat("F =", welch_anova$statistic, "\n")
  cat("df1 =", welch_anova$parameter[1], ", df2 =", welch_anova$parameter[2], "\n")
  cat("p-value =", welch_anova$p.value, "\n\n")

  p_value <- welch_anova$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Games-Howell):\n")
    games_howell <- games_howell_test(ssq_post_clean, SSQ_Post ~ Group_3Level)
    print(games_howell)
    cat("\n")
  }
}

} else {
  cat("Assumptions met. Using standard one-way ANOVA.\n\n")
  anova_model <- aov(SSQ_Post ~ Group_3Level, data = ssq_post_clean)
  anova_summary <- summary(anova_model)

  cat("One-way ANOVA Results:\n")
  print(anova_summary)
  cat("\n")

  eta_sq <- effectsize::eta_squared(anova_model)
  cat("Eta-squared =", round(eta_sq$Eta2[1], 3), "\n\n")

  p_value <- anova_summary[[1]]$`Pr(>F)`[1]

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Tukey HSD):\n")
  }
}

```

```

tukey_results <- TukeyHSD(anova_model)
print(tukey_results)
cat("\n")
}

## Data violates normality assumption. Using Kruskal-Wallis test.
##
## Kruskal-Wallis Test Results:
## H statistic = 0.106699
## df = 2
## p-value = 0.9480486
## Epsilon-squared = 0.002

cat("\n--- INTERPRETATION ---\n")

##
## --- INTERPRETATION ---

if(p_value < 0.05) {
  cat("Result: SIGNIFICANT difference in post-test SSQ scores across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in post-test SSQ scores across groups (p >= 0.05)\n")
}

## Result: NO significant difference in post-test SSQ scores across groups (p >= 0.05)

```

Test 14: SSQ Difference (Post - Pre)

```

cat("SSQ Difference: Post - Pre\n")

## SSQ Difference: Post - Pre

cat("Note: Positive values indicate INCREASE in simulator sickness after VR\n\n")

## Note: Positive values indicate INCREASE in simulator sickness after VR

# Summary statistics by group
ssq_diff_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(SSQ_Diff)),
    Mean = mean(SSQ_Diff, na.rm = TRUE),
    SD = sd(SSQ_Diff, na.rm = TRUE),
    Median = median(SSQ_Diff, na.rm = TRUE),
    Q1 = quantile(SSQ_Diff, 0.25, na.rm = TRUE),
    Q3 = quantile(SSQ_Diff, 0.75, na.rm = TRUE),

```

```

    Min = min(SSQ_Diff, na.rm = TRUE),
    Max = max(SSQ_Diff, na.rm = TRUE)
)

cat("Summary Statistics:\n")

## Summary Statistics:

print(ssq_diff_summary)

## # A tibble: 3 x 9
##   Group_3Level     n   Mean    SD Median    Q1    Q3   Min   Max
##   <fct>       <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 YA            30    1    3.31    0     0     3    -5    10
## 2 OA_High       15   0.733   4.57    0    -0.5    3    -6    12
## 3 OA_Low        15   1.13    4.27    0    -1.5    2    -4    10

cat("\n")

# Filter out NA values for testing
ssq_diff_clean <- merged_data %>%
  filter(!is.na(SSQ_Diff), !is.na(Group_3Level))

# Check normality
cat("Normality Tests (Shapiro-Wilk):\n")

## Normality Tests (Shapiro-Wilk):

for(grp in levels(ssq_diff_clean$Group_3Level)) {
  data_subset <- ssq_diff_clean %>% filter(Group_3Level == grp) %>% pull(SSQ_Diff)
  shapiro_result <- shapiro.test(data_subset)
  cat(grp, ": W =", round(shapiro_result$statistic, 4), ", p =", shapiro_result$p.value)
  if(shapiro_result$p.value < 0.05) cat(" (non-normal)") else cat(" (normal)")
  cat("\n")
}

## YA : W = 0.942 , p = 0.1026877 (normal)
## OA_High : W = 0.9079 , p = 0.1257675 (normal)
## OA_Low : W = 0.8909 , p = 0.06917766 (normal)

cat("\n")

# Check homogeneity of variance
levene_test <- leveneTest(SSQ_Diff ~ Group_3Level, data = ssq_diff_clean)
cat("Levene's Test for Homogeneity of Variance:\n")

## Levene's Test for Homogeneity of Variance:

```

```

cat("F =", round(levene_test$`F value`[1], 4), ", p =", levene_test$`Pr(>F)`[1])

## F = 0.3872 , p = 0.6807084

if(levene_test$`Pr(>F)`[1] < 0.05) cat(" (variances unequal)") else cat(" (variances equal)")

## (variances equal)

cat("\n\n")

# Decide which test to use
use_welch <- levene_test$`Pr(>F)`[1] < 0.05
normality_check <- ssq_diff_clean %>%
  group_by(Group_3Level) %>%
  summarise(p_val = shapiro.test(SSQ_Diff)$p.value) %>%
  pull(p_val)
use_nonparametric <- any(normality_check < 0.05)

if(use_nonparametric) {
  cat(" Data violates normality assumption. Using Kruskal-Wallis test.\n\n")
  kw_test <- kruskal.test(SSQ_Diff ~ Group_3Level, data = ssq_diff_clean)
  cat("Kruskal-Wallis Test Results:\n")
  cat("H statistic =", kw_test$statistic, "\n")
  cat("df =", kw_test$parameter, "\n")
  cat("p-value =", kw_test$p.value, "\n")

  epsilon_sq <- kw_test$statistic / ((nrow(ssq_diff_clean)^2 - 1) / (nrow(ssq_diff_clean) + 1))
  cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

  p_value <- kw_test$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
    dunn_results <- dunnTest(SSQ_Diff ~ Group_3Level, data = ssq_diff_clean, method = "holm")
    print(dunn_results$res)
    cat("\n")
  }
}

} else if(use_welch) {
  cat("Variances are unequal. Using Welch's ANOVA.\n\n")
  welch_anova <- oneway.test(SSQ_Diff ~ Group_3Level, data = ssq_diff_clean, var.equal = FALSE)
  cat("Welch's ANOVA Results:\n")
  cat("F =", welch_anova$statistic, "\n")
  cat("df1 =", welch_anova$parameter[1], ", df2 =", welch_anova$parameter[2], "\n")
  cat("p-value =", welch_anova$p.value, "\n\n")

  p_value <- welch_anova$p.value

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Games-Howell):\n")
    games_howell <- games_howell_test(ssq_diff_clean, SSQ_Diff ~ Group_3Level)
    print(games_howell)
  }
}

```

```

        cat("\n")
    }

} else {
  cat("Assumptions met. Using standard one-way ANOVA.\n\n")
  anova_model <- aov(SSQ_Diff ~ Group_3Level, data = ssq_diff_clean)
  anova_summary <- summary(anova_model)

  cat("One-way ANOVA Results:\n")
  print(anova_summary)
  cat("\n")

  eta_sq <- effectsize::eta_squared(anova_model)
  cat("Eta-squared =", round(eta_sq$Eta2[1], 3), "\n\n")

  p_value <- anova_summary[[1]]$`Pr(>F)`[1]

  if(p_value < 0.05) {
    cat("Post-hoc Comparisons (Tukey HSD):\n")
    tukey_results <- TukeyHSD(anova_model)
    print(tukey_results)
    cat("\n")
  }
}

## Assumptions met. Using standard one-way ANOVA.
##
## One-way ANOVA Results:
##           Df Sum Sq Mean Sq F value Pr(>F)
## Group_3Level  2   1.3   0.633   0.042  0.959
## Residuals    57  866.7  15.205
##
## Eta-squared = 0.001

cat("\n--- INTERPRETATION ---\n")

##
## --- INTERPRETATION ---

if(p_value < 0.05) {
  cat("Result: SIGNIFICANT difference in SSQ change scores across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in SSQ change scores across groups (p >= 0.05)\n")
}

## Result: NO significant difference in SSQ change scores across groups (p >= 0.05)

```

Test 15: SSS Pre-Test Scores (Ordinal)

```

cat("SSS Pre-Test: Stanford Sleepiness Scale (Before VR)\n")

## SSS Pre-Test: Stanford Sleepiness Scale (Before VR)

cat("Note: Ordinal scale (1 = alert, 7 = very sleepy)\n")

## Note: Ordinal scale (1 = alert, 7 = very sleepy)

cat("Analysis: Kruskal-Wallis test (appropriate for ordinal data)\n\n")

## Analysis: Kruskal-Wallis test (appropriate for ordinal data)

# Summary statistics by group
sss_pre_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(SSS_Pre)),
    Median = median(SSS_Pre, na.rm = TRUE),
    Q1 = quantile(SSS_Pre, 0.25, na.rm = TRUE),
    Q3 = quantile(SSS_Pre, 0.75, na.rm = TRUE),
    Mean = mean(SSS_Pre, na.rm = TRUE),
    SD = sd(SSS_Pre, na.rm = TRUE),
    Min = min(SSS_Pre, na.rm = TRUE),
    Max = max(SSS_Pre, na.rm = TRUE)
  )

cat("Summary Statistics:\n")

## Summary Statistics:

print(sss_pre_summary)

## # A tibble: 3 x 9
##   Group_3Level     n Median     Q1     Q3   Mean     SD   Min   Max
##   <fct>     <int>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 YA          30      2      1      2    1.05    1.05    1      6
## 2 OA_High     15      1      1     1.5    1.33    0.617   1      3
## 3 OA_Low      15      1      1     1.5    1.33    0.617   1      3

cat("\n")

# Filter out NA values for testing
sss_pre_clean <- merged_data %>%
  filter(!is.na(SSS_Pre), !is.na(Group_3Level))

# Kruskal-Wallis test
kw_test <- kruskal.test(SSS_Pre ~ Group_3Level, data = sss_pre_clean)

cat("Kruskal-Wallis Test Results:\n")

```

```

## Kruskal-Wallis Test Results:

cat("H statistic =", kw_test$statistic, "\n")

## H statistic = 9.587349

cat("df =", kw_test$parameter, "\n")

## df = 2

cat("p-value =", kw_test$p.value, "\n")

## p-value = 0.008281969

# Effect size (epsilon-squared)
epsilon_sq <- kw_test$statistic / ((nrow(sss_pre_clean)^2 - 1) / (nrow(sss_pre_clean) + 1))
cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

## Epsilon-squared = 0.162

# Post-hoc: Dunn's test with Holm correction
if(kw_test$p.value < 0.05) {
  cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
  dunn_results <- dunnTest(SSS_Pre ~ Group_3Level, data = sss_pre_clean, method = "holm")
  print(dunn_results$res)
  cat("\n")
}

## Post-hoc Comparisons (Dunn's test with Holm correction):
##          Comparison      Z    P.unadj     P.adj
## 1 OA_High - OA_Low  0.0000000 1.000000000 1.000000000
## 2      OA_High - YA -2.528155  0.01146638  0.03439915
## 3      OA_Low - YA -2.528155  0.01146638  0.02293277

# Interpretation
cat("\n--- INTERPRETATION ---\n")

## --- INTERPRETATION ---

if(kw_test$p.value < 0.05) {
  cat("Result: SIGNIFICANT difference in pre-test sleepiness across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in pre-test sleepiness across groups (p >= 0.05)\n")
}

## Result: SIGNIFICANT difference in pre-test sleepiness across groups (p < 0.05)
## See post-hoc tests above for pairwise comparisons

```

Test 16: SSS Post-Test Scores (Ordinal)

```
cat("SSS Post-Test: Stanford Sleepiness Scale (After VR)\n")

## SSS Post-Test: Stanford Sleepiness Scale (After VR)

cat("Note: Ordinal scale (1 = alert, 7 = very sleepy)\n")

## Note: Ordinal scale (1 = alert, 7 = very sleepy)

cat("Analysis: Kruskal-Wallis test (appropriate for ordinal data)\n\n")

## Analysis: Kruskal-Wallis test (appropriate for ordinal data)

# Summary statistics by group
sss_post_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(SSS_Post)),
    Median = median(SSS_Post, na.rm = TRUE),
    Q1 = quantile(SSS_Post, 0.25, na.rm = TRUE),
    Q3 = quantile(SSS_Post, 0.75, na.rm = TRUE),
    Mean = mean(SSS_Post, na.rm = TRUE),
    SD = sd(SSS_Post, na.rm = TRUE),
    Min = min(SSS_Post, na.rm = TRUE),
    Max = max(SSS_Post, na.rm = TRUE)
  )

cat("Summary Statistics:\n")

## Summary Statistics:

print(sss_post_summary)

## # A tibble: 3 x 9
##   Group_3Level     n Median     Q1     Q3   Mean     SD   Min   Max
##   <fct>     <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 YA         30     1     1     3   1.83  1.12     0     4
## 2 OA_High    15     1     1     2   1.53  0.915    1     4
## 3 OA_Low     15     1     1    1.5   1.33  0.617    1     3

cat("\n")

# Filter out NA values for testing
sss_post_clean <- merged_data %>%
  filter(!is.na(SSS_Post), !is.na(Group_3Level))

# Kruskal-Wallis test
kw_test <- kruskal.test(SSS_Post ~ Group_3Level, data = sss_post_clean)

cat("Kruskal-Wallis Test Results:\n")
```

```

## Kruskal-Wallis Test Results:

cat("H statistic =", kw_test$statistic, "\n")

## H statistic = 2.203462

cat("df =", kw_test$parameter, "\n")

## df = 2

cat("p-value =", kw_test$p.value, "\n")

## p-value = 0.3322954

# Effect size (epsilon-squared)
epsilon_sq <- kw_test$statistic / ((nrow(sss_post_clean)^2 - 1) / (nrow(sss_post_clean) + 1))
cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

## Epsilon-squared = 0.037

# Post-hoc: Dunn's test with Holm correction
if(kw_test$p.value < 0.05) {
  cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
  dunn_results <- dunnTest(SSS_Post ~ Group_3Level, data = sss_post_clean, method = "holm")
  print(dunn_results$res)
  cat("\n")
}

# Interpretation
cat("\n--- INTERPRETATION ---\n")

## 
## --- INTERPRETATION ---

if(kw_test$p.value < 0.05) {
  cat("Result: SIGNIFICANT difference in post-test sleepiness across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in post-test sleepiness across groups (p >= 0.05)\n")
}

## Result: NO significant difference in post-test sleepiness across groups (p >= 0.05)

```

Test 17: SSS Difference (Post - Pre) (Ordinal)

```

cat("SSS Difference: Post - Pre\n")

## SSS Difference: Post - Pre

```

```

cat("Note: Positive values indicate INCREASED sleepiness after VR\n")

## Note: Positive values indicate INCREASED sleepiness after VR

cat("Analysis: Kruskal-Wallis test (appropriate for ordinal difference scores)\n\n")

## Analysis: Kruskal-Wallis test (appropriate for ordinal difference scores)

# Summary statistics by group
sss_diff_summary <- merged_data %>%
  group_by(Group_3Level) %>%
  summarise(
    n = sum(!is.na(SSS_Diff)),
    Median = median(SSS_Diff, na.rm = TRUE),
    Q1 = quantile(SSS_Diff, 0.25, na.rm = TRUE),
    Q3 = quantile(SSS_Diff, 0.75, na.rm = TRUE),
    Mean = mean(SSS_Diff, na.rm = TRUE),
    SD = sd(SSS_Diff, na.rm = TRUE),
    Min = min(SSS_Diff, na.rm = TRUE),
    Max = max(SSS_Diff, na.rm = TRUE)
  )

cat("Summary Statistics:\n")

## Summary Statistics:

print(sss_diff_summary)

## # A tibble: 3 x 9
##   Group_3Level     n Median    Q1    Q3    Mean     SD    Min    Max
##   <fct>     <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 YA          30    0 -0.75  0.75 -0.167  1.46    -5     2
## 2 OA_High     15    0    0    0    0.2    0.414    0     1
## 3 OA_Low      15    0    0    0.5    0    0.845    -2     1

cat("\n")

# Filter out NA values for testing
sss_diff_clean <- merged_data %>%
  filter(!is.na(SSS_Diff), !is.na(Group_3Level))

# Kruskal-Wallis test
kw_test <- kruskal.test(SSS_Diff ~ Group_3Level, data = sss_diff_clean)

cat("Kruskal-Wallis Test Results:\n")

## Kruskal-Wallis Test Results:

```

```

cat("H statistic =", kw_test$statistic, "\n")

## H statistic = 0.592614

cat("df =", kw_test$parameter, "\n")

## df = 2

cat("p-value =", kw_test$p.value, "\n")

## p-value = 0.7435591

# Effect size (epsilon-squared)
epsilon_sq <- kw_test$statistic / ((nrow(sss_diff_clean)^2 - 1) / (nrow(sss_diff_clean) + 1))
cat("Epsilon-squared =", round(epsilon_sq, 3), "\n\n")

## Epsilon-squared = 0.01

# Post-hoc: Dunn's test with Holm correction
if(kw_test$p.value < 0.05) {
  cat("Post-hoc Comparisons (Dunn's test with Holm correction):\n")
  dunn_results <- dunnTest(SSS_Diff ~ Group_3Level, data = sss_diff_clean, method = "holm")
  print(dunn_results$res)
  cat("\n")
}

# Interpretation
cat("\n--- INTERPRETATION ---\n")

## --- INTERPRETATION ---

if(kw_test$p.value < 0.05) {
  cat("Result: SIGNIFICANT difference in sleepiness change across groups (p < 0.05)\n")
  cat("See post-hoc tests above for pairwise comparisons\n")
} else {
  cat("Result: NO significant difference in sleepiness change across groups (p >= 0.05)\n")
}

## Result: NO significant difference in sleepiness change across groups (p >= 0.05)

```