Alzheimer's Mice Data Analysis - 2-Way ANOVA

Tagaytay, Gabriel L.

2025-12-12

Contents

github link: https://github.com/gabriyeeel/Formattive-Assessments-Files/blob/main/SA2%20Files/SA2.Rmd

1 Alzheimer's Mice Experiment Analysis

1.1 Introduction

Mice were used in an experiment to test drugs that may prevent Alzheimer's disease. Half the mice were transgenic (genetically modified to have Alzheimer's disease) and half were wild type (not modified). Mice were assigned to one of four drug treatments and tested on memory using a maze. The number of errors made in the maze was recorded for the Training Day and Memory Day. This analysis focuses on Training Day errors using a 2-way between-subjects ANOVA.

1.2 Data Preparation

```
# Create the data frame
mice_data <- data.frame(
  AD_Status = c(rep(1, 20), rep(2, 20)),
  Treatment = c(rep(1, 5), rep(2, 5), rep(3, 5), rep(4, 5),
                rep(1, 5), rep(2, 5), rep(3, 5), rep(4, 5)),
  Training = c(12, 15, 13, 12, 14,
               15, 17, 16, 17, 14,
               13, 14, 18, 15, 16,
               14, 13, 12, 14, 15,
               17, 16, 17, 14, 13,
               14, 18, 16, 17, 14,
               13, 14, 18, 15, 16,
               14, 13, 12, 14, 15),
  Memory = c(10, 12, 13, 10, 13,
             13, 13, 14, 15, 11,
             12, 11, 15, 10, 14,
             12, 11, 10, 13, 10,
             9, 8, 10, 8, 8,
             7, 10, 5, 9, 7,
             8, 7, 9, 8, 9,
             7, 9, 5, 8, 4)
)

# Convert to factors
mice_data$AD_Status <- factor(mice_data$AD_Status,
                              levels = c(1, 2),
                              labels = c("Transgenic", "WildType"))
mice_data$Treatment <- factor(mice_data$Treatment,
                              levels = c(1, 2, 3, 4),
                              labels = c("Drug1", "Drug2", "Drug3", "Drug4"))

# Display structure
cat("Data Structure:\n")
```

```
## Data Structure:
```

```r
str(mice_data)
```

```
## 'data.frame':    40 obs. of  4 variables:
##  $ AD_Status: Factor w/ 2 levels "Transgenic","WildType": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Treatment: Factor w/ 4 levels "Drug1","Drug2",..: 1 1 1 1 1 2 2 2 2 2 ...
##  $ Training : num  12 15 13 12 14 15 17 16 17 14 ...
##  $ Memory   : num  10 12 13 10 13 13 13 14 15 11 ...
```

```r
cat("\nFirst 10 rows:\n")
```

```
##
## First 10 rows:
```

```r
head(mice_data, 10)
```

```
##     AD_Status Treatment Training Memory
## 1  Transgenic     Drug1       12     10
## 2  Transgenic     Drug1       15     12
## 3  Transgenic     Drug1       13     13
## 4  Transgenic     Drug1       12     10
## 5  Transgenic     Drug1       14     13
## 6  Transgenic     Drug2       15     13
## 7  Transgenic     Drug2       17     13
## 8  Transgenic     Drug2       16     14
## 9  Transgenic     Drug2       17     15
## 10 Transgenic     Drug2       14     11
```

1.3 Descriptive Statistics

```r
# Group statistics
desc_stats <- mice_data %>%
  group_by(AD_Status, Treatment) %>%
  summarise(
    N = n(),
    Mean = mean(Training),
    SD = sd(Training),
    SE = SD/sqrt(N),
    Min = min(Training),
    Max = max(Training),
    .groups = 'drop'
  )

cat("Descriptive Statistics by Group:\n")
```

```
## Descriptive Statistics by Group:
```

```r
print(desc_stats, digits = 3)
```

```
## # A tibble: 8 x 8
##   AD_Status  Treatment     N  Mean    SD    SE   Min   Max
##   <fct>      <fct>     <int> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Transgenic Drug1         5  13.2  1.30 0.583    12    15
## 2 Transgenic Drug2         5  15.8  1.30 0.583    14    17
## 3 Transgenic Drug3         5  15.2  1.92 0.860    13    18
## 4 Transgenic Drug4         5  13.6  1.14 0.510    12    15
## 5 WildType   Drug1         5  15.4  1.82 0.812    13    17
## 6 WildType   Drug2         5  15.8  1.79 0.8      14    18
## 7 WildType   Drug3         5  15.2  1.92 0.860    13    18
## 8 WildType   Drug4         5  13.6  1.14 0.510    12    15
```

```
# Overall statistics
cat("\nOverall Statistics:\n")
```

```
##
## Overall Statistics:
```

```
cat("Total N:", nrow(mice_data), "\n")
```

```
## Total N: 40
```

```
cat("Overall Mean:", mean(mice_data$Training), "\n")
```

```
## Overall Mean: 14.725
```

```
cat("Overall SD:", sd(mice_data$Training), "\n")
```

```
## Overall SD: 1.753933
```

```
# Interaction table - USING PIVOT_WIDER FROM TIDYR
interaction_table <- mice_data %>%
  group_by(AD_Status, Treatment) %>%
  summarise(Mean_Training = mean(Training), .groups = 'drop') %>%
  pivot_wider(names_from = Treatment, values_from = Mean_Training)

cat("\nMean Training Errors (Interaction Table):\n")
```

```
##
## Mean Training Errors (Interaction Table):
```

```
print(interaction_table, digits = 3)
```

```
## # A tibble: 2 x 5
##   AD_Status  Drug1 Drug2 Drug3 Drug4
##   <fct>      <dbl> <dbl> <dbl> <dbl>
## 1 Transgenic  13.2  15.8  15.2  13.6
## 2 WildType    15.4  15.8  15.2  13.6
```

1.4 Two-Way ANOVA

```r
# Perform 2-way ANOVA
anova_model <- aov(Training ~ AD_Status * Treatment, data = mice_data)
anova_summary <- summary(anova_model)

cat("Two-Way ANOVA Results:\n")
```

```
## Two-Way ANOVA Results:
```

```r
print(anova_summary)
```

```
##                      Df Sum Sq Mean Sq F value Pr(>F)
## AD_Status             1   3.03   3.025   1.216 0.2784
## Treatment            3  28.28   9.425   3.789 0.0197 *
## AD_Status:Treatment  3   9.08   3.025   1.216 0.3198
## Residuals           32  79.60   2.488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Extract ANOVA table for reporting
anova_table <- as.data.frame(anova_summary[[1]])
cat("\nANOVA Table (for reference):\n")
```

```
##
## ANOVA Table (for reference):
```

```r
print(anova_table, digits = 3)
```

```
##                      Df Sum Sq Mean Sq F value Pr(>F)
## AD_Status             1   3.03    3.03    1.22 0.2784
## Treatment            3  28.28    9.43    3.79 0.0197
## AD_Status:Treatment  3   9.08    3.03    1.22 0.3198
## Residuals           32  79.60    2.49      NA     NA
```

1.5 Effect Sizes

```r
# Calculate eta-squared
if (!require(heplots)) {
  install.packages("heplots")
  library(heplots)
}

eta_sq <- etasq(anova_model, partial = FALSE)
cat("Effect Sizes (Eta-squared):\n")
```

```
## Effect Sizes (Eta-squared):
```

```r
print(eta_sq, digits = 3)
```

```
##                       eta^2
## AD_Status           0.0252
## Treatment           0.2357
## AD_Status:Treatment 0.0756
## Residuals              NA
```

```r
# Calculate partial eta-squared
partial_eta_sq <- etasq(anova_model, partial = TRUE)
cat("\nPartial Eta-squared:\n")
```

```
##
## Partial Eta-squared:
```

```r
print(partial_eta_sq, digits = 3)
```

```
##                     Partial eta^2
## AD_Status                  0.0366
## Treatment                  0.2621
## AD_Status:Treatment        0.1023
## Residuals                     NA
```

2 Assumption Checks

2.1 Normality of Residuals

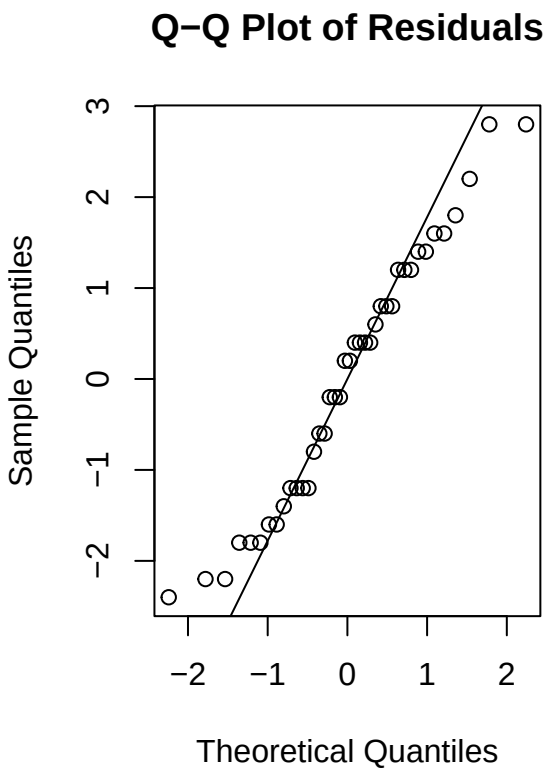
```r
# Shapiro-Wilk test
shapiro_result <- shapiro.test(residuals(anova_model))
cat("Shapiro-Wilk Normality Test:\n")
```

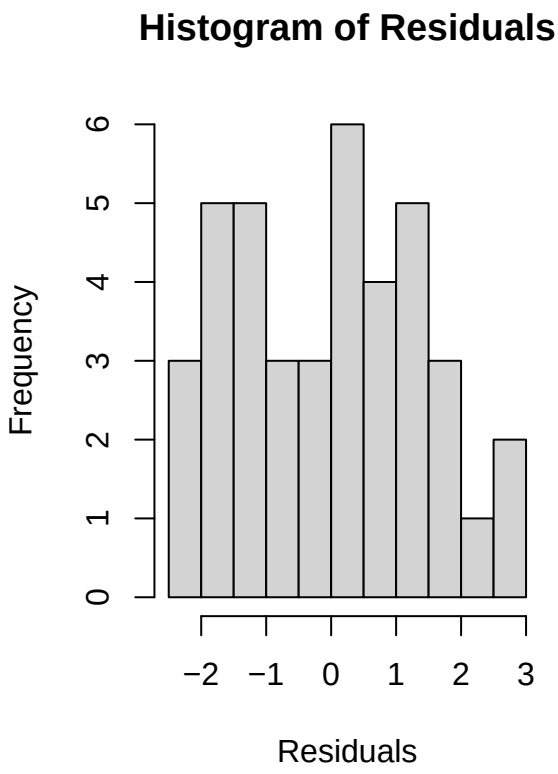
```
## Shapiro-Wilk Normality Test:
```

```r
print(shapiro_result)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(anova_model)
## W = 0.96357, p-value = 0.2214
```

```r
# Q-Q plot
par(mfrow = c(1, 2))
qqnorm(residuals(anova_model), main = "Q-Q Plot of Residuals")
qqline(residuals(anova_model))
hist(residuals(anova_model), main = "Histogram of Residuals",
     xlab = "Residuals", breaks = 10)
```






```
par(mfrow = c(1, 1))
```

2.2 Homogeneity of Variances

```
# Levene's Test
levene_result <- leveneTest(Training ~ AD_Status * Treatment, data = mice_data)
cat("Levene's Test for Homogeneity of Variance:\n")
```

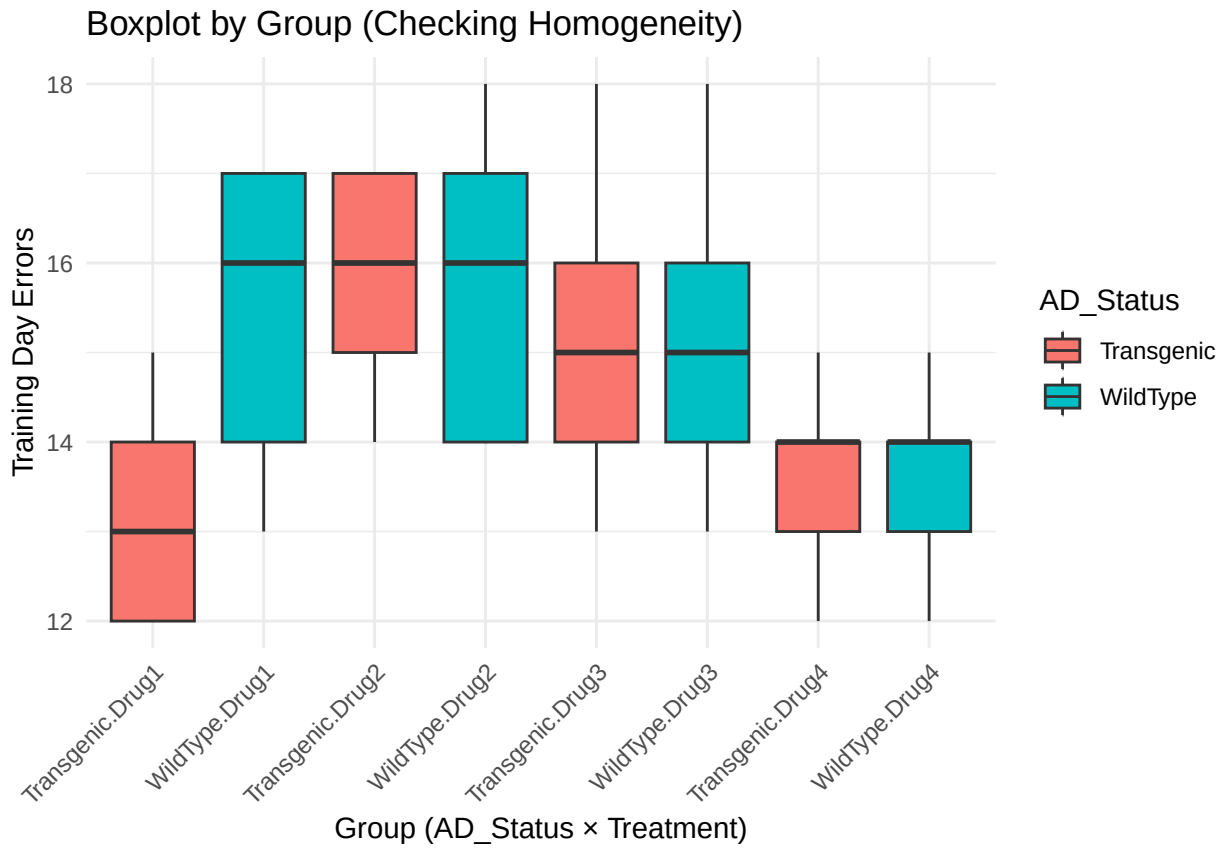
```
## Levene's Test for Homogeneity of Variance:
```

```
print(levene_result)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  7  0.4346 0.8731
##       32
```

```
# Visual check
ggplot(mice_data, aes(x = interaction(AD_Status, Treatment),
                      y = Training, fill = AD_Status)) +
  geom_boxplot() +
  labs(title = "Boxplot by Group (Checking Homogeneity)",
```

```
       x = "Group (AD_Status × Treatment)",
       y = "Training Day Errors") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



2.3 Independence

Satisfied by experimental design - mice were randomly assigned to conditions.

3 Post-Hoc Analysis

```
# Check if interaction is significant
interaction_p <- anova_summary[[1]][3, "Pr(>F)"]
cat("Interaction p-value:", interaction_p, "\n\n")
```

```
## Interaction p-value: 0.3197713
```

```
if (interaction_p < 0.05) {
  cat("Significant interaction detected. Performing simple effects analysis:\n")
  # ADD INSTALLATION CHECK
  if (!require(emmeans)) {
```

```r
    install.packages("emmeans")
    library(emmeans)
  }

  # Simple effects for AD Status at each Treatment level
  emm_AD <- emmeans(anova_model, ~ AD_Status | Treatment)
  cat("\nSimple Effects for AD Status within each Treatment:\n")
  suppressWarnings(print(pairs(emm_AD, adjust = "tukey")))

  # Simple effects for Treatment at each AD Status level
  emm_Treatment <- emmeans(anova_model, ~ Treatment | AD_Status)
  cat("\nSimple Effects for Treatment within each AD Status:\n")
  suppressWarnings(print(pairs(emm_Treatment, adjust = "tukey")))

} else {
  cat("No significant interaction. Performing Tukey HSD for main effects:\n")

  # Tukey HSD for main effects
  tukey_AD <- TukeyHSD(anova_model, "AD_Status")
  cat("\nTukey HSD for AD Status:\n")
  print(tukey_AD)

  tukey_Treatment <- TukeyHSD(anova_model, "Treatment")
  cat("\nTukey HSD for Treatment:\n")
  print(tukey_Treatment)
}
```

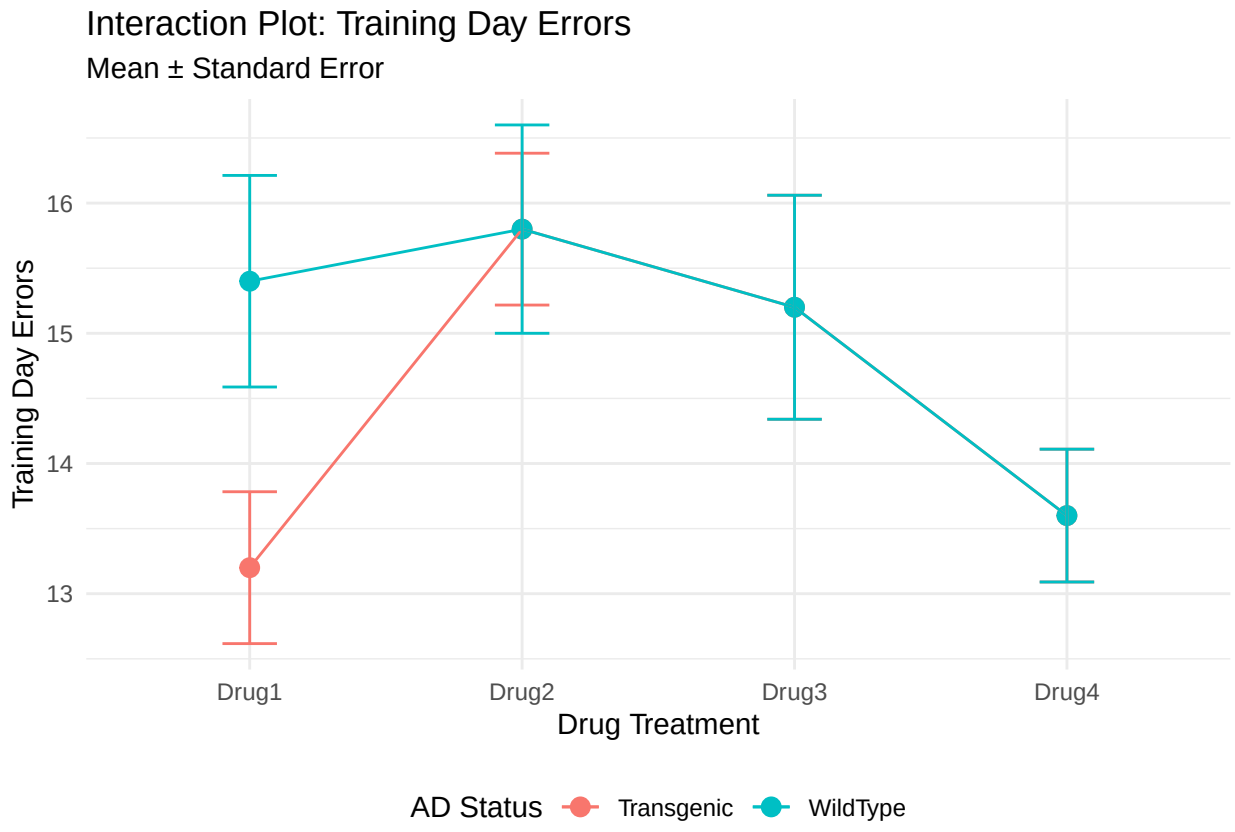
```
## No significant interaction. Performing Tukey HSD for main effects:
##
## Tukey HSD for AD Status:
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Training ~ AD_Status * Treatment, data = mice_data)
##
## $AD_Status
##                       diff        lwr      upr     p adj
## WildType-Transgenic 0.55 -0.4659173 1.565917 0.2783559
##
##
## Tukey HSD for Treatment:
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Training ~ AD_Status * Treatment, data = mice_data)
##
## $Treatment
##              diff        lwr        upr     p adj
## Drug2-Drug1  1.5 -0.4110125  3.4110125 0.1664403
## Drug3-Drug1  0.9 -1.0110125  2.8110125 0.5844498
## Drug4-Drug1 -0.7 -2.6110125  1.2110125 0.7547388
## Drug3-Drug2 -0.6 -2.5110125  1.3110125 0.8298123
## Drug4-Drug2 -2.2 -4.1110125 -0.2889875 0.0190170
```

```
## Drug4-Drug3 -1.6 -3.5110125  0.3110125 0.1269509
```

4 Visualization

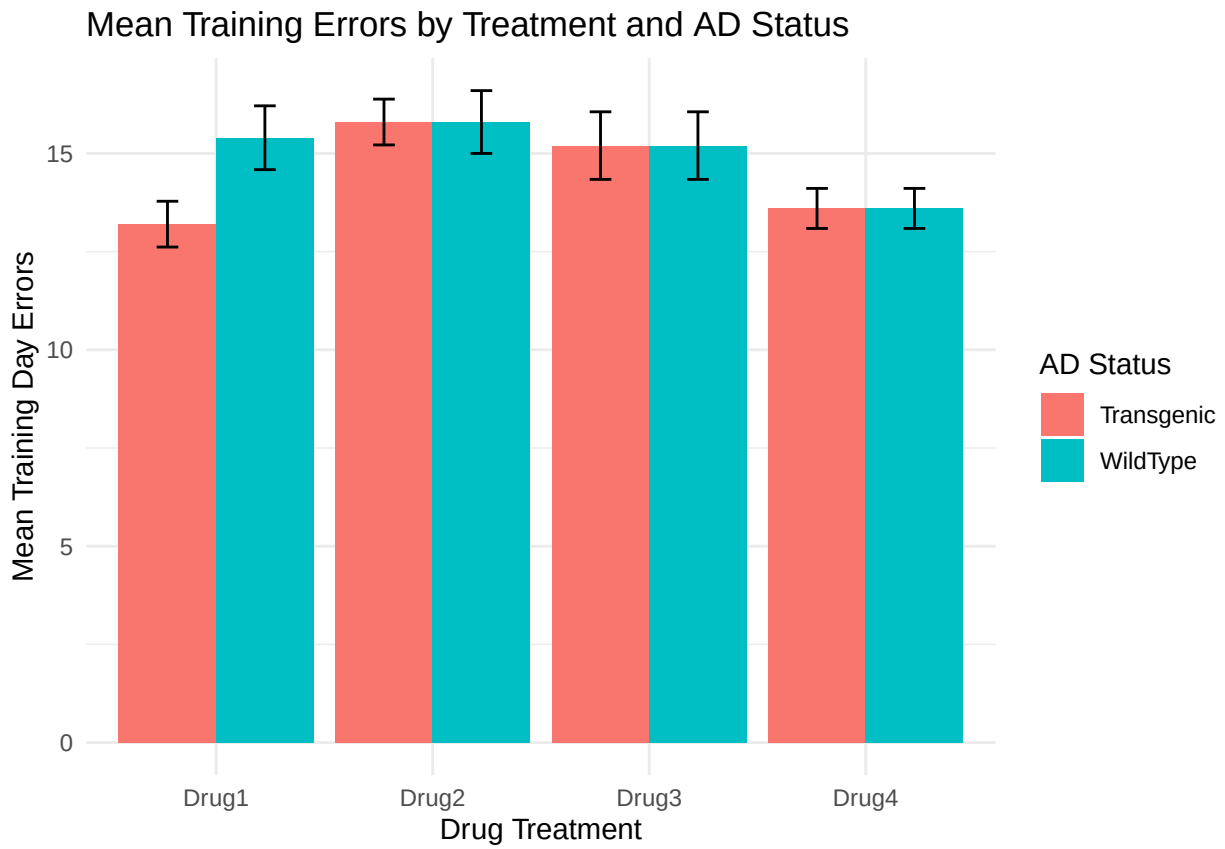
```
# Interaction plot
interaction_plot <- ggplot(mice_data,
                           aes(x = Treatment, y = Training,
                               color = AD_Status, group = AD_Status)) +
  stat_summary(fun = mean, geom = "point", size = 3) +
  stat_summary(fun = mean, geom = "line") +
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.2) +
  labs(title = "Interaction Plot: Training Day Errors",
       subtitle = "Mean ± Standard Error",
       x = "Drug Treatment",
       y = "Training Day Errors",
       color = "AD Status") +
  theme_minimal() +
  theme(legend.position = "bottom")

print(interaction_plot)
```



```r
# Bar plot with error bars
bar_plot <- ggplot(desc_stats,
                   aes(x = Treatment, y = Mean, fill = AD_Status)) +
  geom_bar(stat = "identity", position = position_dodge(0.9)) +
  geom_errorbar(aes(ymin = Mean - SE, ymax = Mean + SE),
                position = position_dodge(0.9), width = 0.2) +
  labs(title = "Mean Training Errors by Treatment and AD Status",
       x = "Drug Treatment",
       y = "Mean Training Day Errors",
       fill = "AD Status") +
  theme_minimal()

print(bar_plot)
```



5 Results in APA Format

```r
# Extract values for APA reporting
F_AD <- round(as.numeric(anova_summary[[1]][1, "F value"]), 2)
df1_AD <- anova_summary[[1]][1, "Df"]
df2_AD <- anova_summary[[1]][4, "Df"]
p_AD_raw <- anova_summary[[1]][1, "Pr(>F)"]
p_AD <- ifelse(p_AD_raw < 0.001, "< .001", sprintf("%.3f", p_AD_raw))
```

```r
eta_AD <- round(as.numeric(eta_sq[1, "eta.sq"]), 3)

F_Treatment <- round(as.numeric(anova_summary[[1]][2, "F value"]), 2)
df1_Treatment <- anova_summary[[1]][2, "Df"]
df2_Treatment <- anova_summary[[1]][4, "Df"]
p_Treatment_raw <- anova_summary[[1]][2, "Pr(>F)"]
p_Treatment <- ifelse(p_Treatment_raw < 0.001, "< .001", sprintf("%.3f", p_Treatment_raw))
eta_Treatment <- round(as.numeric(eta_sq[2, "eta.sq"]), 3)

F_Interaction <- round(as.numeric(anova_summary[[1]][3, "F value"]), 2)
df1_Interaction <- anova_summary[[1]][3, "Df"]
df2_Interaction <- anova_summary[[1]][4, "Df"]
p_Interaction_raw <- anova_summary[[1]][3, "Pr(>F)"]
p_Interaction <- ifelse(p_Interaction_raw < 0.001, "< .001", sprintf("%.3f", p_Interaction_raw))
eta_Interaction <- round(as.numeric(eta_sq[3, "eta.sq"]), 3)

# Means for reporting
mean_transgenic <- mean(mice_data$Training[mice_data$AD_Status == "Transgenic"])
sd_transgenic <- sd(mice_data$Training[mice_data$AD_Status == "Transgenic"])
mean_wildtype <- mean(mice_data$Training[mice_data$AD_Status == "WildType"])
sd_wildtype <- sd(mice_data$Training[mice_data$AD_Status == "WildType"])

cat("=== APA FORMAT RESULTS ===\n\n")
```

```
## === APA FORMAT RESULTS ===
```

```r
cat("A 2 (AD Status: Transgenic vs Wild Type) × 4 (Drug Treatment: Drug1, Drug2, Drug3, Drug4) between-s
```

```
## A 2 (AD Status: Transgenic vs Wild Type) × 4 (Drug Treatment: Drug1, Drug2, Drug3, Drug4) between-sul
```

```r
cat("The ANOVA revealed a ")
```

```
## The ANOVA revealed a
```

```r
if (p_AD_raw < 0.05) {
  cat("significant")
} else {
  cat("non-significant")
}
```

```
## non-significant
```

```r
cat(" main effect of AD Status, F(", df1_AD, ", ", df2_AD, ") = ", F_AD,
    ", p ", p_AD, ",  ² = ", eta_AD, ". ", sep = "")
```

```
##  main effect of AD Status, F(1, 32) = 1.22, p 0.278,  ² = .
```

```r
cat("Transgenic mice (M = ", round(mean_transgenic, 2), ", SD = ",
    round(sd_transgenic, 2), ") ", sep = "")
```

```
## Transgenic mice (M = 14.45, SD = 1.73)
```

```
if (p_AD_raw < 0.05) {
  cat("made significantly ")
  if (mean_transgenic > mean_wildtype) {
    cat("more")
  } else {
    cat("fewer")
  }
  cat(" errors than ")
} else {
  cat("did not differ significantly from ")
}
```

```
## did not differ significantly from
```

```
cat("wild type mice (M = ", round(mean_wildtype, 2), ", SD = ",
    round(sd_wildtype, 2), ").\n\n", sep = "")
```

```
## wild type mice (M = 15, SD = 1.78).
```

```
cat("The main effect of Treatment was ")
```

```
## The main effect of Treatment was
```

```
if (p_Treatment_raw < 0.05) {
  cat("significant")
} else {
  cat("non-significant")
}
```

```
## significant
```

```
cat(", F(", df1_Treatment, ", ", df2_Treatment, ") = ", F_Treatment,
    ", p ", p_Treatment, ",  ² = ", eta_Treatment, ".\n\n", sep = "")
```

```
## , F(3, 32) = 3.79, p 0.020,  ² = .
```

```
cat("The interaction between AD Status and Treatment was ")
```

```
## The interaction between AD Status and Treatment was
```

```
if (p_Interaction_raw < 0.05) {
  cat("significant")
} else {
  cat("non-significant")
}
```

```
## non-significant
```

```
cat(", F(", df1_Interaction, ", ", df2_Interaction, ") = ", F_Interaction,
    ", p ", p_Interaction, ",  ² = ", eta_Interaction, ".\n\n", sep = "")
```

```
## , F(3, 32) = 1.22, p 0.320,  ² = .
```

```
cat("=== ASSUMPTION CHECKING ===\n\n")
```

```
## === ASSUMPTION CHECKING ===
```

```
cat("Normality: Shapiro-Wilk test indicated that residuals were ")
```

```
## Normality: Shapiro-Wilk test indicated that residuals were
```

```
if (shapiro_result$p.value > 0.05) {
  cat("normally distributed")
} else {
  cat("not normally distributed")
}
```

```
## normally distributed
```

```
shapiro_p <- ifelse(shapiro_result$p.value < 0.001, "< .001",
                    sprintf("%.3f", shapiro_result$p.value))
cat(" (W = ", round(shapiro_result$statistic, 3), ", p ", shapiro_p, ").\n\n", sep = "")
```

```
##  (W = 0.964, p 0.221).
```

```
cat("Homogeneity of variance: Levene's test indicated that variances were ")
```

```
## Homogeneity of variance: Levene's test indicated that variances were
```

```
if (levene_result$`Pr(>F)`[1] > 0.05) {
  cat("homogeneous")
} else {
  cat("not homogeneous")
}
```

```
## homogeneous
```

```
levene_p <- ifelse(as.numeric(levene_result$`Pr(>F)`[1]) < 0.001, "< .001",
                   sprintf("%.3f", as.numeric(levene_result$`Pr(>F)`[1])))
cat(" (F = ", round(levene_result$`F value`[1], 2), ", p ", levene_p, ").\n\n", sep = "")
```

```
##  (F = 0.43, p 0.873).
```

6 Conclusion

```r
cat("=== CONCLUSION ===\n\n")
```

```
## === CONCLUSION ===
```

```r
if (p_AD_raw < 0.05) {
  cat("There was a significant difference in Training Day errors between transgenic and wild type mice,
  if (mean_transgenic > mean_wildtype) {
    cat("with transgenic mice making more errors.")
  } else {
    cat("with wild type mice making more errors.")
  }
  cat("\n")
}

if (p_Treatment_raw < 0.05) {
  cat("There were significant differences between drug treatments.\n")
} else {
  cat("There were no significant differences between drug treatments.\n")
}
```

```
## There were significant differences between drug treatments.
```

```r
if (p_Interaction_raw < 0.05) {
  cat("The effect of drug treatment differed depending on AD status.\n")
} else {
  cat("The effect of drug treatment was consistent across AD status groups.\n")
}
```

```
## The effect of drug treatment was consistent across AD status groups.
```

```r
cat("\nBased on these results, ")
```

```
##
## Based on these results,
```

```r
if (p_AD_raw < 0.05) {
  cat("AD status significantly affects maze performance during training. ")
}
if (p_Treatment_raw < 0.05) {
  cat("Different drugs have different effects on training performance. ")
}
```

```
## Different drugs have different effects on training performance.
```

```r
if (p_Interaction_raw < 0.05) {
  cat("Drug effectiveness depends on whether mice are transgenic or wild type.")
}
```

7 References

Field, A., Miles, J., & Field, Z. (2012). Discovering statistics using R. Sage. APA Publication Manual (7th ed.)