STA304 The Outliers Code Appendix*

Vanshika Vanshika Navya Hooda Shea Munson
Alexia Mbagaya Chloe Syriac

December 1, 2024

Introduction

This appendex contains all code used in our technical report. It is ordered how it appears in the technical report, accompanied by any relevant comments.

```
library(tidyverse)
library(treemapify)
library(gridExtra)
library(patchwork)

# Split into Single Items
majTab=table(gsub("\\s+", "", unlist(strsplit(data$major,","))))
# Convert to Data Frame
majDF=as.data.frame(majTab)
# Refactor for Correct Plot Order
majDF$Var1=fct_relevel(majDF$Var1, c("STA", "MAT", "CSC", "ECO", "LSC", "NLA"))
majLabs=c("Applied Statistics", "Mathematics", "Computer Science", "Economics",
          "Life Sciences", "Not Listed")
statusCounts=as.data.frame(table(data$studentStatus))
statusLabs=c("Domestic", "International")
distanceDF=as.data.frame(table(data$campusDistance))
distanceLabs=c("On Campus", "0-5km", "6-10km", "11-15km", "16-20km", "21+km")

status <- ggplot(data=statusCounts, aes(area=Freq, fill=Var1, label=paste(Freq))) +
  geom_treemap(color="black", size=1.0) +
  labs(title="Student Status", x="n = 63", caption = "Figure 1") +
```

*Data and Report Available at: https://github.com/vanshikav2/Extra_Curricular_Activities_Research

```
  theme_minimal() +
  scale_fill_brewer(name="Options", labels=statusLabs, palette = "Set2") +
  geom_treemap_text(colour="black", place="center")


dist <- ggplot(data=distanceDF, aes(area=Freq, fill=Var1, label=paste(Freq))) +
  geom_treemap(color="black", size=1.0) +
  labs(title = "Distance From Campus", x="n = 63", caption = "Figure 2") +
  theme_minimal() +
  scale_fill_brewer(name="Distance", labels=distanceLabs, palette = "Set2") +
  geom_treemap_text(color="black", place="center") +
  theme(plot.title = element_text(hjust = 0.5))

maj <- ggplot(data=majDF, aes(x=Var1, y=Freq, fill=Var1)) +
  geom_bar(stat="identity", color="Black") +
  labs(x="Majors", y="Count", title = "Sample Majors Breakdown",
       caption = "Figure 3") +
  theme_light() +
  scale_fill_brewer(name="Options:", labels=majLabs, palette="Set2") +
  geom_text(aes(label=Freq), vjust=-.3, color="black") +
  theme(plot.title = element_text(hjust = 0.5),
        axis.title.y = element_text(size=12, angle=90),
        axis.text.x = element_text(size= 6))

(status + dist +  maj) + plot_layout(widths = c(1,1,1))
```

Research Question 1

```
library(tidyverse)
library(dplyr)
library(ggplot2)
library(car)
library(knitr)
library(kableExtra)
data <- read_csv("STA304_TheOutliers_CleanedData.csv")

#Deleting the studentID column and replacing NA values
data[is.na(data)] <- "NA"
data <- data %>% select(-studentID, -lectureSection)
```

```r
#Split `activityType` and `major` into individual categories
data <- data %>%
  separate_rows(activityType, sep = ",") %>%
  separate_rows(major, sep = ",")

data <- data %>% mutate(activityType = trimws(data$activityType),
                        major = trimws(data$major))


# Descriptive Analysis (What is the most preferred Extra Curricular Activity)
data <- data %>% mutate(activityType = as.factor(activityType))
activity_counts <- data %>%
  count(activityType, name = "Frequency") %>%
  arrange(desc(Frequency))

# Plot distribution of activity types
activity_labels <- c(
  "AAS" = "Athletics and Sports",
  "NEC" = "No Extracurricular Activities",
  "CLU" = "Clubs",
  "LSG" = "Leadership/Student Governments, Councils & Unions",
  "ACS" = "Academic Societies"
)

#Graph Showing the Activity Counts
ggplot(data = activity_counts, aes(x = activityType, y = Frequency,
                                   fill = activityType)) +
  geom_bar(stat = "identity", color = "black") +
  labs(
    x = "Activity Type",
    y = "Frequency",
    title = "Distribution of Extracurricular Activity Types"
  ) +
  theme_light() +
  scale_fill_brewer(
    name = "Activity Options:",
    labels = activity_labels,
    palette = "Set2"
  ) +
  geom_text(aes(label = Frequency), vjust = -0.3, size = 4)
```

Testing for the Significance of Majors on Activity Preference using the ANOVA Test

```r
# -----------------------------
# 2. ANOVA for Major
# -----------------------------
#Convert ActivityType to NumericValues
data <- data %>% mutate(activityTypeNumeric = as.numeric(factor(activityType)))

anova_major <- aov(activityTypeNumeric ~ major, data = data)
#summary(anova_major)

#Q-Q plot of Residuals
residuals <- residuals(anova_major)

qqnorm(residuals, col = "purple", main = "Q-Q Plot of Residuals for ANOVA Model
       for Major")

qqline(residuals, col = "black", lwd = 1)
```

```r
library(car)
library(knitr)
##Testing for Homogenity of Variances
data <- data %>%
  mutate(major = as.factor(major))
levene_test_major <- leveneTest(activityTypeNumeric ~ major, data = data)
levene_df <- as.data.frame(levene_test_major)

levene_df[is.na(levene_df)] <- ""

kable(levene_df, caption = "Levene Test on The Activity Preference by Major")
```

```r
anova_summary <- summary(anova_major)[[1]]
anova_df <- data.frame(anova_summary)
anova_df[is.na(anova_df)] <- ""
kable(anova_df, caption = "Anova Test on The Activity Preference by Major")
```

```r
activity_lbls <- c(
  "1" = "ACS",
  "2" = "AAS",
  "3" = "CLU",
  "4" = "LSG",
```

```
  "5" = "NEC"
)
```

Testing for the Significance of Gender on Activity Preference using the T-Test

```
##GenderIdentity
gender_data_F <- data %>% filter(genderIdentity == "F") %>% pull(activityTypeNumeric)
gender_data_M <- data %>% filter(genderIdentity == "M") %>% pull(activityTypeNumeric)

data <- data %>%
  mutate(genderIdentity = as.factor(genderIdentity))
shapiro_F <- shapiro.test(gender_data_F)
shapiro_M <- shapiro.test(gender_data_M)

shapiro_M_df <- as.data.frame(t(c(shapiro_M$statistic, shapiro_M$p.value)))
colnames(shapiro_M_df) <- c("W statistic", "p-value")
shapiro_F_df <- as.data.frame(t(c(shapiro_F$statistic, shapiro_F$p.value)))
colnames(shapiro_F_df) <- c("W statistic", "p-value")


shapiro_results <- data.frame(
  Gender = c("Males", "Females"),
  `W Statistic` = c(shapiro_F_df$`W statistic`, shapiro_M_df$`W statistic`),
  `p-value` = c(shapiro_F_df$`p-value`, shapiro_M_df$`p-value`)
)

kable(shapiro_results, col.names = c("Gender", "W Statistic", "p-value"),
      caption = "Shapiro-Wilk Test Results for Activity Preferences
      by Gender Groups")
```

```
levene_gender <- leveneTest(activityTypeNumeric ~ genderIdentity, data = data)

levene_gender_df <- as.data.frame(levene_gender)
levene_gender_df[is.na(levene_gender_df)] <- ""

kable(levene_gender_df, caption = "Levene Test for Equality of Variances in
      Activity Preferences by Gender Identity")
```

```
t_test_gender <- t.test(activityTypeNumeric ~ genderIdentity, data = data,
                        var.equal = FALSE)

t_test_result_df <- data.frame(T_Statistic = t_test_gender$statistic,
                               DF = t_test_gender$parameter,
                               p_value = t_test_gender$p.value, MEAN_FEMALE = t_test_gender$e

kable(t_test_result_df, caption = "Welch t-test on the Activity Proportion
      by Gender")
```

Testing for the Significance of Student Status on Activity Preference using the T-Test

```
##StudentStatus
status_data_D <- data %>% filter(studentStatus == "D") %>% pull(activityTypeNumeric)
status_data_I <- data %>% filter(studentStatus == "I") %>% pull(activityTypeNumeric)

shapiro_D <- shapiro.test(status_data_D)
shapiro_I <- shapiro.test(status_data_I)

shapiro_results <- data.frame(
  StudentStatus = c("D", "I"),
  W_Statistic = c(shapiro_D$statistic, shapiro_I$statistic),
  P_Value = c(shapiro_D$p.value, shapiro_I$p.value)
)
kable(shapiro_results, caption = "Shapiro Test on studentStatus")
```

```
data <- data %>%
  mutate(studentStatus = as.factor(studentStatus))
levene_status <- leveneTest(activityTypeNumeric ~ studentStatus, data = data)

levene_status_df <- as.data.frame(levene_status)
levene_status_df[is.na(levene_status_df)] <- ""

kable(levene_status_df, caption = "Levene Test on studentStatus")
```

```
wilcox_test <- wilcox.test(activityTypeNumeric ~ studentStatus, data = data)

wilcox_test_df <- data.frame(
  Statistic = wilcox_test$statistic,
  P_Value = wilcox_test$p.value)
```

```r
kable(
  wilcox_test_df,
  col.names = c("W Statistic", "P-Value"),
  caption = "Wilcoxon Rank-Sum Test Results for Activity Preferences by
  Student Status"
)
```

```r
# Boxplot for Major influence
plot1 <- ggplot(data, aes(x = major, y = activityTypeNumeric, fill = major)) +
  geom_boxplot() +
  labs(title = "Boxplot of Activity Preferences by Major",
       x = "Major",
       y = "Activity Type") +
  scale_y_continuous(
    breaks = 1:5,
    labels = activity_lbls
  )+
  theme_minimal() +
  theme(axis.text.y = element_text(size = 10),
    axis.title.y = element_text(size = 12),
    plot.margin = margin(10, 10, 10, 20)
  )

# Boxplot for Gender Influence
plot2 <- ggplot(data, aes(x = genderIdentity, y = activityTypeNumeric,
                          fill = genderIdentity)) +
  geom_boxplot() +
  labs(title = "Boxplot of Activity Preferences by Gender",
       x = "Gender Identity",
       y = "Activity Type (Numeric)") +
  scale_y_continuous(
    breaks = 1:5,
    labels = activity_lbls)+
  theme_minimal() +
  theme(axis.text.y = element_text(size = 10),
    axis.title.y = element_text(size = 12),
    plot.margin = margin(10, 10, 10, 20)
  )
# Boxplot for StudentStatus
plot3 <- ggplot(data, aes(x = studentStatus, y = activityTypeNumeric,
                          fill = studentStatus)) +
  geom_boxplot() +
```

```
  labs(title = "Boxplot of Activity Preferences by Student Status",
       x = "Student Status",
       y = "Activity Type (Numeric)") +
  scale_y_continuous(
    breaks = 1:5,
    labels = activity_lbls)+
  theme_minimal() +
  theme(axis.text.y = element_text(size = 10),
    axis.title.y = element_text(size = 12),
    plot.margin = margin(10, 10, 10, 20)
  )

grid.arrange(plot1, plot2, plot3, nrow = 2)
```

Research Question 2

Campus Distance and Activity Count

```
# Create summaries for campusDistance and activityCount
campus <- summary(data$campusDistance)
activity <- summary(data$activityCount)

# Combine the summaries into a data frame
summary_t <- data.frame(
  Statistic = c("Min", "1st Quartile", "Median", "Mean", "3rd Quartile", "Max"),
  CampusDistance = as.numeric(campus),
  ActivityCount = as.numeric(activity)
)

library(knitr)

kable(summary_t, format = "pipe", col.names = c("Statistic", "CampusDistance",
                                                "ActivityCount"))
```

```
# Load necessary libraries
library(ggplot2)

# Scatter plot with smoothing line to assess linearity
ggplot(data, aes(x = campusDistance, y = activityCount)) +
  geom_point(color ="darkgreen") +
```

Table 1: Correlation Test Results

```r
# Create a data frame to summarize the correlation test results
correlation_results <- data.frame(
  Test = c("Pearson's Correlation", "Spearman's Correlation"),
  Correlation_Coefficient = c(-0.018, 0.025),
  p_value = c(0.891, 0.847),
  Significance = c("Not Significant", "Not Significant")
)


kable(correlation_results, caption = "Summary of Correlation Test Results")
```

```r
  geom_smooth(method = "lm", col = "orange", se = FALSE) +
  labs(title = "Campus Distance vs. Activity Count with Linear Trend Line",
       x = "Campus Distance",
       y = "Activity Count")
```

```r
# Fit preliminary model
model <- lm(activityCount ~ campusDistance, data = data)

# Q-Q plot for normality of residuals
qqnorm(residuals(model))
qqline(residuals(model), col = "red")
```

```r
# Residuals vs Fitted plot to assess homoscedasticity
plot(model, which = 1)
```

Research Question 3

Time Commitment and Student Involvement

```r
library(rstanarm)

# Set up the plotting area for two plots side by side
par(mfrow = c(1, 2))
```

```r
# QQ plot for timeCommitment
qqnorm(data$timeCommitment, main = "QQ Plot (TC)",
       xlab = "Theoretical Quantiles", ylab = "Sample Quantiles")
qqline(data$timeCommitment, col = "red")

# QQ plot for studentInvolvement
qqnorm(data$studentInvolvement, main = "QQ Plot (SI)",
       xlab = "Theoretical Quantiles", ylab = "Sample Quantiles")
qqline(data$studentInvolvement, col = "blue")

# Reset the plotting layout to default (if needed for subsequent plots)
par(mfrow = c(1, 1))
```

```r
# Load necessary libraries
library(ggplot2)

# Create a scatter plot
ggplot(data, aes(x = timeCommitment, y = studentInvolvement)) +
  geom_point(color = "darkgreen") + # Set points to green
  geom_smooth(method = "lm", color = "orange", se = FALSE) + # Set line to orange
  ggtitle("Time Commitment and Student Involvement") +
  xlab("Time per week (hrs)") +
  ylab("Involvement (1-10)")
```

Time Commitment and Activity Count

```r
library(rstanarm)

# Set up the plotting area for two plots side by side
par(mfrow = c(1, 2))

# QQ plot for timeCommitment
qqnorm(data$timeCommitment, main = "QQ Plot (TC)",
       xlab = "Theoretical Quantiles", ylab = "Sample Quantiles")
qqline(data$timeCommitment, col = "red")

# QQ plot for activity count
qqnorm(data$activityCount, main = "QQ Plot (AC)",
       xlab = "Theoretical Quantiles", ylab = "Sample Quantiles")
```

Table 2: Correlation Test Results

```r
# Load the necessary packages
library(knitr)

# Ensure data is defined and contains the necessary columns
# Example: data <- data.frame(timeCommitment = ..., studentInvolvement = ...)

# Perform Spearman's correlation test
result <- cor.test(data$timeCommitment, data$studentInvolvement,
                   method = "spearman", exact = FALSE)

# Extract relevant values
p_value <- sprintf("%.3e", result$p.value)  # Format p-value in scientific notation
rho <- sprintf("%.3f", result$estimate)     # Format rho to 3 decimal places

# Create a data frame for the table
table_results <- data.frame(
  Statistic = c("Spearman Coefficient (rho)", "p-value"),
  Value = c(rho, p_value)
)

# Display the table with basic kable (without additional styling)
kable(table_results, caption = "Spearman's Rank Correlation Results")
```

```r
qqline(data$studentInvolvement, col = "blue")

# Reset the plotting layout to default (if needed for subsequent plots)
par(mfrow = c(1, 1))
```

```r
# Load necessary libraries
library(ggplot2)

# Create a scatter plot
ggplot(data, aes(x = timeCommitment, y = activityCount)) +
  geom_point(color = "darkgreen") +                       # Set points to green
  geom_smooth(method = "lm", color = "orange", se = FALSE) + # Set line to orange
  ggtitle("Time Commitment and Activity Count") +
  xlab("Time per week (hrs)") +
  ylab("Number of Activities")
```

```r
# Load the necessary packages
library(knitr)

# Ensure data is defined and contains the necessary columns
# Example: data <- data.frame(timeCommitment = ..., studentInvolvement = ...)

# Perform Spearman's correlation test
result <- cor.test(data$timeCommitment, data$activityCount,
                   method = "spearman", exact = FALSE)

# Extract relevant values
p_value <- sprintf("%.3e", result$p.value)  # Format p-value in scientific notation
rho <- sprintf("%.3f", result$estimate)     # Format rho to 3 decimal places

# Create a data frame for the table
table_results <- data.frame(
  Statistic = c("Spearman Coefficient (rho)", "p-value"),
  Value = c(rho, p_value)
)

# Display the table with basic kable (without additional styling)
kable(table_results, caption = "Spearman's Rank Correlation Results")
```