appendix (old)

Non-Fixed

```
# Install necessary libraries
packages <- c("tidyverse", "patchwork", "performance", "knitr")
for(pkg in packages) {
  if (!requireNamespace(pkg, quietly = TRUE)) {
    install.packages(pkg)
  }
}

# Load libraries
library(tidyverse)
library(patchwork)
library(performance)
library(knitr)

# Load data, cleaned_data.csv
# data <- read.csv("cleaned_data.csv", header = TRUE)
data <-read.csv(file.choose())

# Set factor levels to display in desired order
data$age_group <- factor(data$age_group,
                         levels = c("Y", "M", "O"),
                         labels = c("18-35", "36-54", "55+"))
data$treatment <- factor(data$treatment,
                         levels = c("W", "E", "ED", "C", "CD"),
                         labels = c("Water (Control)", "Energy Drink",
                                    "Energy Drink (Decaf)", "Coffee",
                                    "Coffee (Decaf)"))
attach(data)
```

```r
# Table 1: Summary of Problem Solving Test Scores
sum_data <- data.frame(Value = c(min(score), max(score), round(mean(score), 4),
                                 median(score), round(sd(score), 4), IQR(score)))
labs <- c("Min", "Max", "Mean", "Median", "SD", "IQR")
kable(t(sum_data), col.names = labs, align = "c",
      caption = "Summary of Problem Solving Test Scores")

# Figure 1: Problem Solving By Drink and Age Group
tr <- c("Water (Control)", "Energy Drink", "Energy Drink (Decaf)", "Coffee",
        "Coffee (Decaf")
age <- c("18-35", "36-54", "55+")
ggplot(data, aes(x = treatment, y = score, fill = age_group)) +
  geom_boxplot(position = position_dodge(width = 0.8)) +
  scale_x_discrete(labels = tr) +
  scale_y_continuous(limits = c(60, 100)) +
  labs(title = "Problem Solving Scores by Drink and Age Group",
       x = "Treatment (Drink Type)", y = "Problem Solving Test Scores",
       fill = "Age Group") +
  scale_fill_brewer(palette = "Dark2", breaks = age, labels = age) +
  theme_minimal()

# Figure 2, 3: Normality and Variance Homogeneity Graphs
# for one and two factor ANOVA
one_factor_model <- aov(score ~ treatment, data = data)
check_model(one_factor_model, check = c("normality", "homogeneity"))

two_factor_model <- aov(score ~ treatment + data$age_group, data = data)
check_model(two_factor_model, check = c("normality", "homogeneity"))

# Print results of assumption checks:
# 1. Check normality of residuals for the one-factor model
normality_results1 <- check_normality(one_factor_model)
print(normality_results1)

# 2. Check homogeneity of variances for the one-factor model
hetero_results1 <- check_heteroscedasticity(one_factor_model)
print(hetero_results1)

# 1. Check normality of residuals for the one-factor model again
normality_results2 <- check_normality(one_factor_model)
print(normality_results2)
```

```r
# 2. Check homogeneity of variances for the two-factor model
hetero_results2 <- check_heteroscedasticity(two_factor_model)
print(hetero_results2)

# Perform K-W Test
non_par <- kruskal.test(score~treatment, data)
formatted <- as.data.frame(tidy(non_par))
formatted$method <- NULL
col_names <- c("H-Statistic", "P Value", "DF")
kable(formatted, col.names = col_names,
      caption = "Kruskal-Wallis H-Test")
```

Fixed Appendix (old style)

Listing 1 Randomization Script for Assigning Treatments

```r
library(clipr)
groups <- c("Water", "Coffee", "Coffee (Decaf)", "Energy Drink",
          "Energy Drink (Decaf)")
labels <- rep(groups, 10)
shuffled <- sample(labels)
write_clip(shuffled, type = "text")
```

```r
# Ensure necessary packages are installed and loaded
packages <- c("tidyverse", "patchwork", "performance", "knitr", "see",
              "kableExtra", "broom")
for(pkg in packages) {
  if (!requireNamespace(pkg, quietly = TRUE)) {
    install.packages(pkg)
  }
}
library(tidyverse)
library(patchwork)
library(performance)
library(knitr)
library(see)
library(kableExtra)
library(broom)
```

```r
# Load the Data
data <- read.csv("cleaned_data.csv", header = TRUE)

# Put factor levels in desired order
data$age_group <- factor(data$age_group,
                         levels = c("Y", "M", "O"),
                         labels = c("18-35", "36-54", "55+"))
data$treatment <- factor(data$treatment,
                         levels = c("W", "E", "ED", "C", "CD"),
                         labels = c("Water (Control)", "Energy Drink",
                                    "Energy Drink (Decaf)", "Coffee",
                                    "Coffee (Decaf)"))
attach(data)

# Generate table for score summary data
sum_data <- data.frame(Value = c(min(score), max(score), round(mean(score), 4),
                                 median(score), round(sd(score), 4), IQR(score)))
labs <- c("Min", "Max", "Mean", "Median", "SD", "IQR")
kable(t(sum_data), col.names = labs, align = "c")

# Generate boxplot for scores explained by drink
ggplot(data, aes(x = treatment, y = score, fill = treatment)) +
  geom_boxplot() +
  scale_y_continuous(limits = c(60, 100)) +
  scale_fill_brewer(palette = "Set1") +
  labs(title = "Problem Solving Scores Categorized by Drink",
  x = "Treatment (Drink Type)", y = "Problem Solving Test Score") +
  theme(legend.position = "none")

# Check assumptions for one-way model
one_factor_model <- aov(score ~ treatment, data = data)
check_model(one_factor_model, check = c("normality", "homogeneity"))

# Perform non-parametric K-W test
non_par <- kruskal.test(score~treatment, data)
formatted <- as.data.frame(tidy(non_par))
formatted$method <- NULL
col_names <- c("H-Statistic", "P Value", "DF")
kable(formatted, col.names = col_names,
      caption = "Kruskal-Wallis H-Test")

# Generate boxplot for scores explained by drink and age
```

```r
age <- c("18-35", "36-54", "55+")
ggplot(data, aes(x = treatment, y = score, fill = age_group)) +
  geom_boxplot(position = position_dodge(width = 0.8)) +
  scale_y_continuous(limits = c(60, 100)) +
  labs(title = "Problem Solving Scores by Drink and Age Group",
       x = "Treatment (Drink Type)", y = "Problem Solving Test Scores",
       fill = "Age Group") +
  scale_fill_brewer(palette = "Blues", breaks = age, labels = age) +
  theme_minimal()

# Check assumptions for 2 factor model
two_factor_model <- aov(score ~ treatment + age_group, data = data)
check_model(two_factor_model, check = c("normality", "homogeneity"))
```