IS 3003

# RESEARCH PAPER

## A NOVEL APPROACH FOR NON-PARAMETRIC TWO-WAY ANOVA

GROUP 04
16TH MARCH 2025

# Table of Contents

# Existing Non-Parametric Alternatives to Two-Way ANOVA

In factorial experiments, two-way ANOVA examines the effects of two categorical independent variables on a dependent variable. However traditional ANOVA assumes normality, homogeneity of Variance, and independent of observations. When these assumptions are violated, non parametric alternatives are preferred. In here explores some non parametric methods.

## Aligned Rank Transform (ART) ANOVA

Aligned Rank Transform (ART) ANOVA is a non-parametric statistical method that extends the ANOVA framework to handle ordinal or non normal data. It aligns and ranks data for each main effect and interaction term separately, allowing factorial designs to be analyzed without assuming normality or equal variances.

### *Assumptions of ART ANOVA*

- The data is ordinal or continuous but violates parametric assumptions.
- The design includes one or more categorical independent variables (factors).
- The dependent variable has no strict requirement for normality or homogeneity of variance.
- Independence of observations within groups.

### *ART ANOVA suitable when*

- The dependent variable is ordinal (eg: Likert-Scale data).
- Assumptions of Parametric ANOVA (normality and homogeneity of variance) are violated.
- When want to analyze interaction effects in a factorial design.

### *Why ART ANOVA best:*

It allows for testing both main effects and interaction effects in a non - parametric framework, making it ideal for factorial designs.

### *Steps to Perform ART ANOVA:*

1. Align and Rank the data for each main effect and interaction term (Using software like R).
2. Run a standard ANOVA on the transformed (aligned and ranked) data.
3. Analyze the main effects and interaction effects from ANOVA output.
4. Interpret results in the context of the study.

```
library(ARTool)

Warning: package 'ARTool' was built under R version 4.4.3

# Load the dataset
data("warpbreaks")

# Convert factors
warpbreaks$wool <- as.factor(warpbreaks$wool)
warpbreaks$tension <- as.factor(warpbreaks$tension)

# Apply ART transformation
art_model <- art(breaks ~ wool * tension, data = warpbreaks)

# Perform ART ANOVA
anova_results <- anova(art_model)
print(anova_results)

Analysis of Variance of Aligned Rank Transformed Data

Table Type: Anova Table (Type III tests)
Model: No Repeated Measures (lm)
Response: art(breaks)

               Df Df.res F value    Pr(>F)   
1 wool          1     48  3.0180 0.0887605  .
2 tension       2     48  6.0888 0.0043989 **
3 wool:tension  2     48  3.2502 0.0474455  *
---
Signif. codes:   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Pairwise comparisons for 'wool'
art.con(art_model, "wool")

NOTE: Results may be misleading due to involvement in interactions

 contrast estimate   SE df t.ratio p.value
 A - B         7.56 4.35 48   1.737  0.0888

Results are averaged over the levels of: tension

# Pairwise comparisons for 'tension'
art.con(art_model, "tension")

NOTE: Results may be misleading due to involvement in interactions

 contrast estimate   SE df t.ratio p.value
 L - M        10.22 4.89 48   2.090  0.1025
 L - H        16.94 4.89 48   3.465  0.0032
 M - H         6.72 4.89 48   1.375  0.3620

Results are averaged over the levels of: wool
P value adjustment: tukey method for comparing a family of 3 estimates
```

### *Limitations*

- Aligning and ranking may distort data relationships.
- Some loss of interpretability due to data transformation.

## Scheirer Ray Hare Test

Scheirer Ray Hare Test is a statistical test that can be used to examine whether the measure is affected by two or more factors. Since it does not require a normal distribution of the data, it is one of the non parametric method. It is an extension of the kuskal-wallis test.

### *Assumptions of Scheirer Ray Hare Test*

- The data comes from independent groups.
- The response variable is at least ordinal (can be ranked).
- No interaction effects between factors (only test the main effects of each factor).

### Scheirer Ray Hare Test suitable when:

- When the data is not normally distributed.
- When the sample size is small.
- when testing only main effects( Since interactions are not considered).

### Steps to Perform:

1. Rank the data across all observations.
2. Calculate sums of ranks: Find the sum of ranks for each level of the two factors.
3. Compute the test statistic
4. Compare with chi square distribution: The test statistics follows a chi-square distribution.
5. Make a Decision

```
library(rcompanion)

Warning: package 'rcompanion' was built under R version 4.4.3

# Perform the Scheirer-Ray-Hare Test
scheirer_test <- scheirerRayHare(breaks ~ wool * tension, data = warpbreaks)


DV:  breaks
Observations:  54
D:  0.9980941
MS total:  247.5

print(scheirer_test)

             Df Sum Sq       H  p.value
wool          1  327.6  1.3261 0.249508
tension       2 2670.2 10.8093 0.004496
wool:tension  2  899.8  3.6427 0.161810
Residuals    48 9194.9
```

### Limitations

- Does not test for interaction effects.
- Less powerful than ART ANOVA in factorial designs.

## Rank Transform (RT) Approach as a non-parametric alternative to two-way ANOVA

Rank Transform (RT) Approach is a non-parametric alternative to two-way ANOVA, particularly when normality, homogeneity of variance, or interval scaling are violated. In the RT approach,the data is transformed into ranks, then standard parametric methods (e.g., ANOVA) are applied to the ranked data. This method is simple, robust, and widely used in non-parametric statistics.

### *What is the Rank Transform Approach?*

The RT method transforms the observed data into ranks, where each data point is substituted with its rank within the overall dataset. Once the data is ranked, standard parametric methods,like ANOVA, can be applied to this ranked data. This method is especially beneficial when:

- The data does not follow a normal distribution.
- There are outliers or extreme values present.
- The data is ordinal or fails to meet the criteria for interval/ratio scaling.

### *Advantages of the Rank Transform Approach*

- **Robustness**: Less sensitive to outliers and non-normality.
- **Simplicity**: Easy to implement using standard statistical software.
- **Versatility**: Can be applied to a wide range of data types, including ordinal data.

### *Limitations of the Rank Transform Approach*

- **Loss of Information**: By ranking the data, we lose the original scale, which can result in a significant loss of valuable information.
- **Interpretation**: Since the results are based on ranks, understanding them can be less straightforward compared to using the raw data.
- **Interaction Effects**: The Rank Transform approach might not effectively capture interaction effects in factorial designs, which could lead to oversight in analysis.

### *Comparison with Other Non-Parametric Alternatives*

- **Kruskal-Wallis Test**: A non-parametric alternative for one-way ANOVA, but not suitable for factorial designs.
- **Aligned Rank Transform (ART)**: An extension of the RT approach specifically designed for factorial designs with interactions.
- **Permutation Tests**: Another non-parametric alternative, but computationally intensive.

The Rank Transform Approach is a simple and effective non-parametric alternative to two-way ANOVA, especially when parametric assumptions are violated.

### *Steps to Apply the Rank Transform Approach for Two-Way ANOVA*

#### *Step 1: Rank the Data*

- Start by consolidating all observations from the dataset and ranking them. When there are tied values, assign them the average rank.

#### *Step 2: Use Two-Way ANOVA on Ranked Data*

- Use the ranked data as your response variable in a standard two-way ANOVA model. Make sure to include both main effects and interaction terms.

#### *Step 3: Interpret the Results*

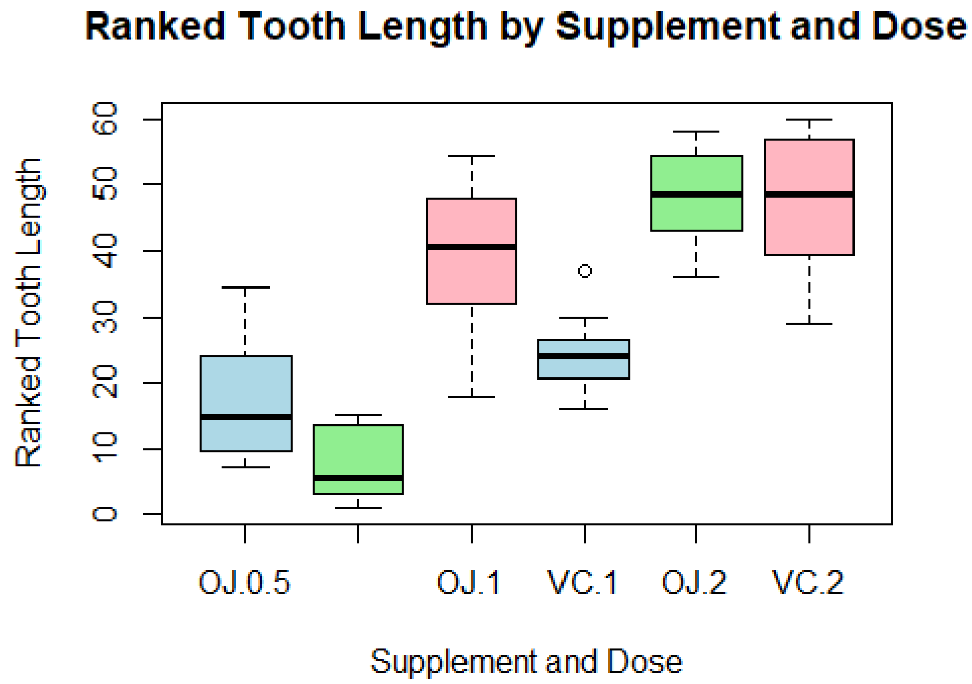
- Analyze the ANOV A results (F-tests, p-values) as you would typically do, keeping in mind that the calculations are based on ranks instead of the original raw data.

**Example:**

```
# Load the ToothGrowth dataset
data("ToothGrowth")

# Inspect the dataset
head(ToothGrowth)  # View the first few rows

   len supp dose
1  4.2   VC  0.5
2 11.5   VC  0.5
3  7.3   VC  0.5
4  5.8   VC  0.5
```

```
5  6.4   VC  0.5
6 10.0   VC  0.5

summary(ToothGrowth)  # Summary statistics

      len          supp         dose      
 Min.   : 4.20   OJ:30   Min.   :0.500  
 1st Qu.:13.07   VC:30   1st Qu.:0.500  
 Median :19.25           Median :1.000  
 Mean   :18.81           Mean   :1.167  
 3rd Qu.:25.27           3rd Qu.:2.000  
 Max.   :33.90           Max.   :2.000  

sum(is.na(ToothGrowth))  # Check for missing values

[1] 0

# Rank the response variable (tooth length)
ToothGrowth$Ranked_Length <- rank(ToothGrowth$len)

# Perform two-way ANOVA on the ranked data
anova_model_ranked <- aov(Ranked_Length ~ supp * dose, data = ToothGrowth)
summary(anova_model_ranked)  # Summarize ANOVA results for ranked data

            Df Sum Sq Mean Sq F value   Pr(>F)    
supp         1   1050    1050  11.656   0.0012 ** 
dose         1  11589   11589 128.640 3.92e-16 ***
supp:dose    1    298     298   3.304   0.0745 .  
Residuals   56   5045      90                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Post-hoc analysis (Tukey's HSD) on ranked data
TukeyHSD(anova_model_ranked)  # Perform Tukey's HSD test on ranked data

Warning in replications(paste("~", xx), data = mf): non-factors ignored: dose

Warning in replications(paste("~", xx), data = mf): non-factors ignored:
supp,
dose

Warning in TukeyHSD.aov(anova_model_ranked): 'which' specified some non-
factors
which will be dropped

  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Ranked_Length ~ supp * dose, data = ToothGrowth)

$supp
```

```
            diff       lwr       upr     p adj
VC-OJ -8.366667 -13.27592 -3.457416 0.0011964

# Visualize the ranked data
boxplot(Ranked_Length ~ supp * dose, data = ToothGrowth,
        main = "Ranked Tooth Length by Supplement and Dose",
        xlab = "Supplement and Dose",
        ylab = "Ranked Tooth Length",
        col = c("lightblue", "lightgreen", "lightpink"))
```



## Permutation Tests

### *What are Permutation Tests in Two-Way ANOVA?*

A permutation test for two-way ANOVA, often referred to as PERMANOVA (Permutational Multivariate Analysis of Variance), serves as a non-parametric alternative to the traditional two-way ANOV A. Unlike standard methods, which assume normality and equal variance, permutation tests rely on randomly shuffling the data labels to test whether the observed group differences are statistically significant.

### *Assumptions of Permutation Tests*

1. Exchangeability of Observations
2. Fixed Factor Structure
3. Homogeneity of Multivariate Dispersion (For PERMANOVA on Distance Data)
4. Sufficient Number of Permutations

### *When to Use a Permutation Test*

1. When the normality assumption is violated.
2. When homogeneity of variance is not met.
3. When sample sizes are small.
4. When working with multivariate or distance-based data.
5. When testing for interaction effects nonparametrically.
6. When data contains outliers or does not meet linearity assumptions.

### *Steps in Conducting a Permutation Test*

*Step 1: Define the Hypotheses*

- **Null Hypothesis (H0):** There are no significant differences among the groups.
- **Alternative Hypothesis (Ha):** At least one group shows a significant difference.

*Step 2: Compute the Observed Test Statistic*

- Execute a standard two-way ANOV A on the original dataset.
- Extract the **F-statistics** for both the main effects and interactions.

*Step 3: Randomly Shuffle Group Labels*

- Shuffle the data labels randomly while maintaining the structure of the dataset, simulating the condition where the null hypothesis holds true.

*Step 4: Recalculate the Test Statistic for Each Permutation*

- After each shuffle, **recalculate the F-statistics.**
- Repeat this process numerous times (e.g., 1,000 or 10,000 times) to generate a **null distribution** of F-values.

*Step 5: Compare the Observed Statistic to the Permutation Distribution*

- Assess how extreme the observed F-value is in comparison to the permuted values.
- The **p-value** is equal to the number of permuted F-values that are greater than or equal to the observed F-value, divided by the total number of permutations.

*Step 6: Make a Decision*

- If the p-value is less than the significance level (e.g., α = 0.05), reject the null hypothesis (H0) and conclude that at least one factor has a significant impact.
- If the p-value is greater than or equal to the significance level, fail to reject the null hypothesis (H0), indicating no significant effect was detected

**Example:**

```r
# Load dataset
data(iris)

# Step 1: Define Hypotheses
# Null Hypothesis (H0): Species and Sepal Length Category have no effect on Petal Length
# Alternative Hypothesis (Ha): At least one factor significantly affects Petal Length

# Create a categorical variable for Sepal Length
iris$Sepal_Length_Cat <- ifelse(iris$Sepal.Length > median(iris$Sepal.Length), "Long", "Short")

# Step 2: Compute Observed Test Statistic (F-values)
observed_anova <- aov(Petal.Length ~ Species * Sepal_Length_Cat, data = iris)
```

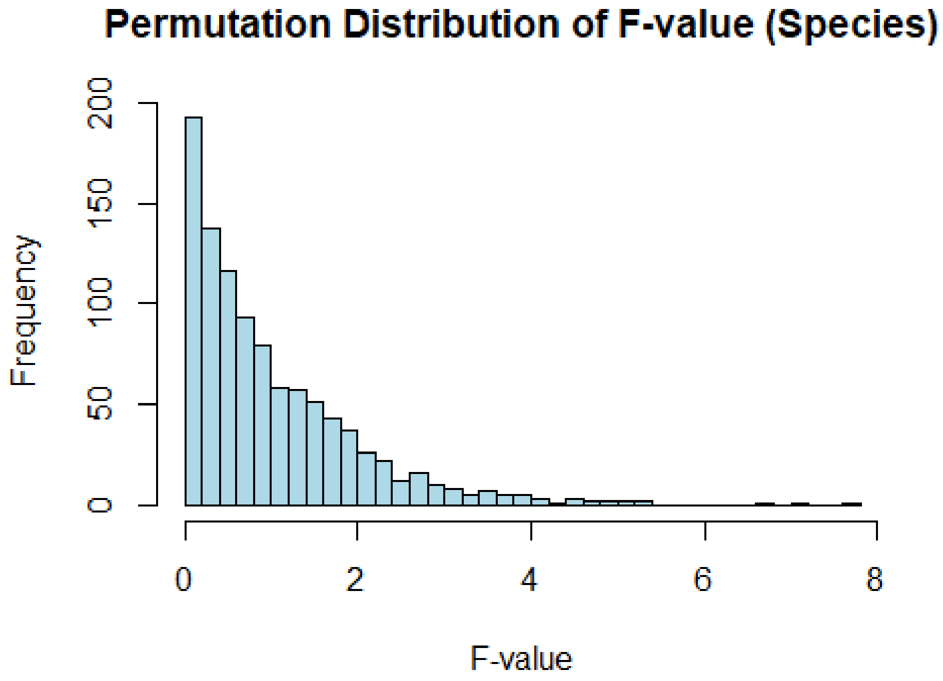
```r
observed_F_values <- summary(observed_anova)[[1]]$`F value`[1:3]  # Extract F-values
print(observed_F_values)
```

```
[1] 1.619143e+03 5.663563e+01 4.367319e-02
```

```r
# Step 3 & 4: Shuffle Labels & Compute Permutation Distribution
set.seed(123)  # For reproducibility
num_permutations <- 1000  # Number of permutations
permuted_F_matrix <- matrix(NA, nrow = num_permutations, ncol = 3)  # Store permuted F-values

for (i in 1:num_permutations) {
  iris$Species_perm <- sample(iris$Species)  # Shuffle species labels
  iris$Sepal_Length_Cat_perm <- sample(iris$Sepal_Length_Cat)  # Shuffle sepal length category

  permuted_anova <- aov(Petal.Length ~ Species_perm * Sepal_Length_Cat_perm, data = iris)
  permuted_F_matrix[i, ] <- summary(permuted_anova)[[1]]$`F value`[1:3]
}

# Step 5: Compute p-values
p_values <- colMeans(permuted_F_matrix >= observed_F_values)

# Step 6: Make a Decision
effect_names <- c("Species", "Sepal Length Category", "Interaction")
for (i in 1:3) {
  if (p_values[i] < 0.05) {
    cat(effect_names[i], ": Significant Effect (p =", p_values[i], ")\n")
  } else {
    cat(effect_names[i], ": No Significant Effect (p =", p_values[i], ")\n")
  }
}
```

```
Species : No Significant Effect (p = 0.32 )
Sepal Length Category : No Significant Effect (p = 0.275 )
Interaction : No Significant Effect (p = 0.317 )
```

```r
# Visualization: Permutation Distribution for Species Effect
hist(permuted_F_matrix[, 1], breaks = 30, col = "lightblue",
     main = "Permutation Distribution of F-value (Species)",
     xlab = "F-value")
abline(v = observed_F_values[1], col = "red", lwd = 2)  # Mark observed F-value
```


Permutation Distribution of F-value (Species)
Frequency
0
50
100
150
200
0
2
4
6
8
F-value

# Novel Approach for Non-Parametric Two-way ANOVA

## Introduction to Novel Development

Traditional two-way Analysis of Variance (ANOVA) is widely used to evaluate the effects of two categorical independent variables on a continuous dependent variable. However, it relies on assumptions of normality and homogeneity of variances, which may not hold in many real-world scenarios. To address these limitations, this study explores and develops a non-parametric alternative to two-way ANOVA that makes minimal distributional assumptions while maintaining interpretability and computational feasibility.

The methodological innovation in this study involves a **bootstrap-permutation approach** that:

- Accounts for **interaction effects** between factors,
- Avoids strict **parametric assumptions**,
- Provides **robust effect size estimates** using Eta Squared ($\eta^2$),
- Uses **resampling techniques** to assess statistical significance and variability.

The study follows a structured process:

1. **Theoretical Development** – Formulating a mathematical framework, defining test statistics, and deriving a sampling distribution.
2. **Implementation** – Developing R functions to perform the non-parametric test, including diagnostic and visualization tools.
3. **Validation** – Comparing the method with traditional parametric two-way ANOVA, conducting power analysis, and testing robustness under assumption violations.

By leveraging bootstrap confidence intervals and permutation-based p-values, this approach enhances statistical inference in cases where standard ANOVA assumptions do not hold. The resulting method provides a flexible and interpretable alternative, especially useful in heterogeneous or skewed data scenarios.

## Our Approach

A novel approach for non-parametric two-way ANOVA was developed by leveraging two resampling techniques: Bootstrapping and Permutation. In this analysis, a non-parametric bootstrap method was first applied to estimate the sampling distribution of eta squared ($\eta^2$), an effect size measure representing the proportion of variance explained by the model. Through repeated resampling with replacement, bootstrap samples were generated to approximate the distribution of $\eta^2$ while addressing non-normality, heteroscedasticity, and violations of homogeneity of variances. This approach allowed for the estimation of standard errors and confidence intervals without relying on strict parametric assumptions, making it particularly useful in cases where the theoretical distribution of the statistic is unknown or complex.

Following bootstrapping, permutation tests were conducted to assess the significance of the observed $\eta^2$ while accounting for both main effects and interaction effects. By randomly shuffling data labels, a distribution of $\eta^2$ under the null hypothesis—where no effect is assumed—was generated. The observed $\eta^2$ was then compared to this distribution to evaluate the likelihood of obtaining such an effect by chance. This method utilized $\eta^2$-based p-values to determine statistical significance, providing a non-parametric framework for hypothesis testing without imposing restrictive assumptions on the data.

## Why Our Method Works?

The proposed method offers a novel non-parametric approach to two-way ANOVA by entirely eliminating reliance on parametric assumptions and data transformations, making it highly adaptable to diverse datasets. Traditional ANOVA assumes normality, homogeneity of variances, and independent errors, which are often violated in real-world data. Many existing non-parametric alternatives either require rank-based transformations (e.g., the Aligned Rank Transform) or impose restrictions on interactions. In contrast, the proposed method directly estimates the sampling distribution of eta squared ($\eta^2$) using a **non-parametric bootstrap**, which preserves the original data structure while accounting for non-normality, heteroscedasticity, and complex variance patterns.

Following this, permutation tests provide an assumption-free means of hypothesis testing by generating a null distribution for $\eta^2$ through random label shuffling. This ensures that **both main effects and interaction effects** are tested without requiring parametric distributions or variance equalization. Unlike traditional permutation-based ANOVA, which often relies on F-statistics derived from assumptions of normality, the proposed approach evaluates effect sizes directly, ensuring robust and interpretable results. Moreover, by utilizing **$\eta^2$-based p-values**, significance is assessed in a manner that is **scale-independent and resistant to the distortions introduced by data transformations**.

The combination of bootstrapping and permutation in this sequential manner enables both accurate effect size estimation and rigorous hypothesis testing, making it a **truly non-parametric alternative** to two-way ANOVA. This approach is particularly advantageous for complex datasets where normality and homogeneity assumptions fail, yet an interpretable measure of variance explained and statistical significance is still required.

# Mathematical Framework for Bootstrap-Permutation Non-Parametric Two-Way ANOVA

## Step 1: Assumptions

1. Exchangeability of Observations
2. Fixed Factor Structure
3. Sufficient Number of Permutations

## Step 2: Define the Hypotheses

- Null Hypothesis (H0): There are no significant differences among the groups.
- Alternative Hypothesis (Ha): At least one group shows a significant difference.

## Step 3: Compute Observed Test Statistic (Eta Squared)

The test statistic used is **Eta Squared (**η²), which is defined as:

$$\eta^2 = \frac{SS_{\text{Effect}}}{SS_{\text{Total}}}$$

where: *$SS_{Effect}$* represents the sum of squares for the effect of interest (main effects or interaction*).* *$SS_{Total}$* is the total sum of squares in the dataset.

### *Main Effects and Interaction*

For **factor A**:

$$\eta_A^2 = \frac{SS_A}{SS_{\text{Total}}}$$

For **factor B**:

$$\eta_B^2 = \frac{SS_B}{SS_{\text{Total}}}$$

For **interaction ( A*B ):**

$$\eta_{A\times B}^2 = \frac{SS_{A\times B}}{SS_{\text{Total}}}$$

## Step 4: Permutation Testing for Significance

To determine **p-values**, permutation testing is conducted by random shuffling of data labels and recalculating η² under the null hypothesis.

### *Permutation Procedure*

1. **Shuffle labels**: Factor labels (A and B) are randomly permuted while keeping η² fixed.
2. **Recompute** η²: The permuted dataset's η² is calculated.
3. **Repeat for P permutations** (e.g., P = 5000) to build the null distribution.
4. **Compute p-values**: The proportion of times η² exceeds the observed η² is calculated as:

$$p = \frac{\sum_{p=1}^{P} I\left(\eta^2_{\text{perm},p} \geq \eta^2_{\text{obs}}\right)}{P}$$

where *I(.)* is an **indicator function** that equals 1 if $\eta^2_{perm}$ >= $\eta^2_{obs}$ , otherwise 0. - $\eta^2_{obs}$ is the observed effect size.

- If the p-value is less than the significance level (e.g., α = 0.05), reject the null hypothesis (H0) and conclude that there is a significant effect.
- If the p-value is greater than or equal to the significance level, fail to reject the null hypothesis (H0), suggesting no strong evidence for a significant effect.

## Justification of Eta Squared as Test Statistic

The selection of **Eta Squared** η² as the test statistic is motivated by its robustness, interpretability, and compatibility with both bootstrapping and permutation approaches.

## *Comparison between F value and Eta Squared*

### F-Ratio (F-Statistic)

#### *Definition*

The **F-ratio**, or **F-statistic**, quantifies the ratio of between-group variance to within-group variance in ANOVA. It is calculated as:

$$F = \frac{MS_{\text{Effect}}}{MS_{\text{Error}}}$$

where:
- $MS_{\text{Effect}}$ is the mean square for the factor of interest.
- $MS_{\text{Error}}$ is the mean square for residual variance.

#### *Purpose*

- Used to test whether a factor significantly influences the response variable.
- A larger F-value suggests a stronger effect relative to random error.

#### *Interpretation*

- If F is significantly larger than 1, the factor likely has a meaningful effect.
- Statistical significance is determined by comparing F to the F-distribution and computing a p-value.

#### *Assumptions*

- Normality of residuals.
- Homogeneity of variance.
- Independence of observations.

## Eta Squared (η²)

### *Definition*

Eta Squared measures the proportion of total variance explained by a factor. It is calculated as:

$$\eta^2 = \frac{SS_{\text{Effect}}}{SS_{\text{Total}}}$$

where:
- $SS_{\text{Effect}}$ is the sum of squares for the factor.
- $SS_{\text{Total}}$ is the total sum of squares.

### *Purpose*

- Measures effect size rather than statistical significance.
- Indicates the proportion of variation in the response variable explained by the factor.

### *Interpretation*

- Ranges from 0 to 1, with the following common thresholds (Cohen, 1988):
    - ( $\eta^2$= 0.01 ) → Small effect
    - ( $\eta^2$ = 0.06 ) → Medium effect
    - ( $\eta^2$ = 0.14+ ) → Large effect
- Unlike F, $\eta^2$ does not depend on sample size and can be used in both parametric and non-parametric contexts.

### *Assumptions*

- None required for interpretation.
- Suitable for non-parametric methods and bootstrap-based resampling approaches.

### *Key Properties of Eta Squared*

- **Robust to non-normality**: Unlike F-statistics, $\eta^2$ does not assume normality.
- **Interpretable**: It directly measures the proportion of variance explained by each factor.
- **Comparable across models**: $\eta^2$ can be applied in various designs, making it effective for bootstrapping and permutation methods.

### *Why Use Eta Squared?*

- **Effect Size Measure**
  - Unlike p-values, $\eta^2$ quantifies the variance in $Y$ explained by each factor.
  - It provides insight into practical significance beyond statistical significance.
- **Interpretability**
  - $\eta^2$ values range from 0 to 1, making it easy to understand:
    - **0**: No effect
    - **0.01**: Small effect
    - **0.06**: Medium effect
    - **0.14+**: Large effect (Cohen, 1988)
- **Consistency Across Bootstrap & Permutation**
  - $\eta^2$ allows direct comparison between observed and resampled values.
  - Permutation tests examine how $\eta^2$ behaves under the null hypothesis, while bootstrapping estimates its variability.

### *Empirical Estimation of Eta Squared*

The **sampling distribution** of $\eta^2$ is estimated using **Bootstrapping** to obtain a robust estimate of $\eta^2$ 's variability. **Permutation Testing** is used to approximate the null distribution for significance testing.

# Implementation in R

Before executing the non-parametric two-way ANOVA method, it is important to assess the distribution of data. After assessing the distribution of data, key assumptions must be checked. Since our method does not require normality and homoscedasticity assumptions, diagnostic tests can help justify the use of a non-parametric approach.

The following function checks homoscedasticity and normality of residuals.

```r
homoscedasticity_tests <- function(data) {
  fit <- lm(Y ~ A * B, data = data)

  cat("\n### Breusch-Pagan Test ###\n")
  print(bptest(fit))

  cat("\n### Levene’s Test ###\n")
  print(leveneTest(Y ~ A * B, data = data))

  # Diagnostic plots
  par(mfrow = c(2,2))
  qqnorm(residuals(fit), main = "QQ Plot of Residuals")
  qqline(residuals(fit), col = "red")
  plot(fitted(fit), residuals(fit), main="Residuals vs Fitted Values",
xlab="Fitted Values", ylab="Residuals")
  abline(h = 0, col = "red")
  hist(residuals(fit), main = "Histogram of Residuals", breaks = 10, col =
"lightblue", border = "black")
  plot(density(residuals(fit)), main = "Density of Residuals", col = "blue")

  # Normality test
  cat("\n### Shapiro-Wilk Test for Normality ###\n")
  print(shapiro.test(residuals(fit)))
}
```

Breusch-Pagan Test: If p-value is < 0.05, variance is not equal, suggesting heteroscedasticity.

Levene’s Test: A significant result indicates variance differences between groups.

Shapiro-Wilk Test: If p-value is < 0.05, residuals deviate from normality.

A violation of any assumption strengthens the justification for using a non-parametric approach rather than standard ANOVA.

Outliers of influential observations can bias statistical results. The Cook's distance method identifies points that excessively influence the model.

```
influence_analysis <- function(data) {
  fit <- lm(Y ~ A * B, data = data)
  cooksd <- cooks.distance(fit)
  plot(cooksd, type="h", main="Cook's Distance")
  abline(h = 4/length(cooksd), col="red")
}
```

High Cook's Distance (> 4/N): Suggests a potential influential point.

Visual Analysis: Large spikes in the plot indicate high-leverage observations.

After verifying assumption violations, it is justified to move forward with the non parametric two-way ANOVA approach.

## Bootstrap Approach

Multiple non-parametric bootstrap methods are executed to handle different conditions in the data, selecting the best method based on the results of assumption checks.

### 1. Choosing the Best Bootstrap Type

The function choose_bootstrap_type() selects the best method as: Residual Bootstrap if normality holds, Wild Bootstrap if there is heteroscedasticity, Pairwise Bootstrap if normality is violated, Winsorized Bootstrap if outliers exist.

```
choose_bootstrap_type <- function(data) {
  fit <- lm(Y ~ A * B, data = data)

  bp_test <- bptest(fit)$p.value
  levene_test <- leveneTest(Y ~ A * B, data = data)[1, "Pr(>F)"]

  shapiro_test <- if (nrow(data) < 5000) shapiro.test(residuals(fit))$p.value
else NA

  outliers <- sum(abs(scale(residuals(fit))) > 3)
  if (bp_test < 0.05 || levene_test < 0.05) {
    if (outliers > 0) {
      return("winsorized")
    } else {
      return("wild")
    }
  } else {
    if (!is.na(shapiro_test) && shapiro_test > 0.05) {
      return("residual")
    } else {
      return("pairwise")
    }
  }
}
```

## 2. Define test statistic: Eta Squared

The key test statistic Eta squared () quantified the proportion of variance explained by factors.

```
eta_squared <- function(fit) {
  ss_effect <- sum((fitted(fit) - mean(fit$model$Y))^2)
  ss_total <- sum((fit$model$Y - mean(fit$model$Y))^2)
  return(ss_effect / ss_total)
}
```

## 3. Estimate distribution of test statistic

i. Residual Bootstrap - Resample residuals to create new datasets

ii. Wild Bootstrap - Adjust residuals using random sign flips to maintain heteroscedasticity

iii. Pairwise Bootstrap - Resample entire rows of data instead of residuals

iv. Winsorized Bootstrap - Trim extreme residuals to reduce outlier influence

```
bootstrap_functions <- list(
  residual = function(data, num_samples = 1000) {
    fit <- lm(Y ~ A * B, data = data)
    residuals <- residuals(fit)
    fitted_values <- fitted(fit)

    boot_stats <- replicate(num_samples, {
      data$Y <- fitted_values + sample(residuals, replace = TRUE)
      eta_squared(lm(Y ~ A * B, data = data))
    })
    return(boot_stats)
  },

  wild = function(data, num_samples = 1000) {
    fit <- lm(Y ~ A * B, data = data)
    residuals <- residuals(fit)
    fitted_values <- fitted(fit)

    boot_stats <- replicate(num_samples, {
      weights <- sample(c(-1, 1), length(residuals), replace = TRUE)
      data$Y <- fitted_values + residuals * weights
      eta_squared(lm(Y ~ A * B, data = data))
    })
    return(boot_stats)
  },

  pairwise = function(data, num_samples = 1000) {
    boot_stats <- replicate(num_samples, {
      boot_sample <- data[sample(1:nrow(data), replace = TRUE), ]
      eta_squared(lm(Y ~ A * B, data = boot_sample))
    })
    return(boot_stats)
  },

  winsorized = function(data, num_samples = 1000, trim = 0.05) {
    fit <- lm(Y ~ A * B, data = data)
    residuals <- residuals(fit)
    fitted_values <- fitted(fit)

    lower_bound <- quantile(residuals, trim)
    upper_bound <- quantile(residuals, 1 - trim)
    winsorized_residuals <- pmax(pmin(residuals, upper_bound), lower_bound)

    boot_stats <- replicate(num_samples, {
      data$Y <- fitted_values + sample(winsorized_residuals, replace = TRUE)
      eta_squared(lm(Y ~ A * B, data = data))
    })
    return(boot_stats)
  }
)
```

### 4. Execution of Bootstrap Procedure

After the best bootstrap method is chosen, it is executed to produce bootstrap confidence intervals and standard errors for effect size estimation.

### 5. Visualization of Bootstrap Distribution

The bootstrap distribution of the test statistic eta squared can be visualized using ggplot package. A peak in the curve shows where most estimates of eta squared are concentrated, suggesting insights into how strong the effects are in our analysis.

### 6. Interpretation of Results

- Mean $\eta^2$: Average proportion of variance explained across bootstrap samples.
- Bootstrap Confidence Interval: Range of plausible values for $\eta^2$.
- Standard Error: Measure of variability in $\eta^2$ estimation.
- Visual Validation: Bootstrap density plot confirms stability of distribution.

## Permutation Approach

The purpose of permutation testing is to assess the effect of independent factors A, B and their interactions A*B, on the dependent variable Y, by randomly assigning values and recalculating the effect size; eta squared.

The permutation approach involves following key functions:

## 1. Coefficient Extraction

Using this function, the coefficient name for a given term from the fitted linear model is extracted. This is later used to find the exact terms that are needed to check significance.

```r
get_coef_name <- function(fit, term) {
  coef_names <- names(coef(fit))

  match <- grep(term, coef_names, value = TRUE)

  if (length(match) > 0) {
    return(match[1])  # Return the first match
  } else {
    stop(paste("Coefficient", term, "not found in model"))
  }
}
```

## 2. Eta squared calculation

The proportion of variance explained by factors A, B, and interaction A*B is computed using the ANOVA sum of squares.

```r
calculate_eta_squared <- function(fit, data) {
  anova_table <- anova(fit)
  ss_A <- anova_table["A", "Sum Sq"]
  ss_B <- anova_table["B", "Sum Sq"]
  ss_interaction <- anova_table["A:B", "Sum Sq"]

  total_ss <- sum((data$Y - mean(data$Y))^2)

  eta_A <- ss_A / total_ss
  eta_B <- ss_B / total_ss
  eta_interaction <- ss_interaction / total_ss

  return(list(eta_A = eta_A, eta_B = eta_B, eta_interaction =
eta_interaction))
}
```

## 3. Data permutation

Here data is randomly shuffled based on the type of permutation.

“main_A” : Permute factor A while other variables are kept constant.

“main_B” : Permute factor B while other variables are kept constant.

“interaction” : Permute dependent Y to test interaction.

```
permute_data <- function(data, type) {
  permuted_data <- as.data.table(copy(data))

  if (type == "main_A") {
    permuted_data[, A := sample(A)]
  } else if (type == "main_B") {
    permuted_data[, B := sample(B)]
  } else if (type == "interaction") {
    permuted_data[, Y := sample(Y)]
  }

  permuted_data[, A := factor(A, levels = levels(data$A))]
  permuted_data[, B := factor(B, levels = levels(data$B))]

  return(permuted_data)
}
```

## 4. Permutation Execution

The Permutation process is repeated B times (generally 1000) to generate a null distribution of eta squared values for each effect. This is later compared with the observed distribution of eta squared.

```
permute_and_fit <- function(data, type, B) {
  results <- replicate(B, {
    permuted_data <- permute_data(data, type)
    fit_perm <- lm(Y ~ A * B, data = permuted_data)
    ss_eta <- calculate_eta_squared(fit_perm, permuted_data)
    return(list(eta_A = ss_eta$eta_A, eta_B = ss_eta$eta_B, eta_interaction =
ss_eta$eta_interaction))
  }, simplify = FALSE)

  return(results)
}
```

## 5. Summarization of Results

Here, the mean, standard deviation, and confidence intervals of permuted eta squared values are computed.

```
summarize_permutation_results <- function(results) {
  eta_A_values <- sapply(results, function(x) x$eta_A)
  eta_B_values <- sapply(results, function(x) x$eta_B)
  eta_interaction_values <- sapply(results, function(x) x$eta_interaction)

  summary_A <- c(mean = mean(eta_A_values), sd = sd(eta_A_values), ci_95 =
quantile(eta_A_values, c(0.025, 0.975)))
  summary_B <- c(mean = mean(eta_B_values), sd = sd(eta_B_values), ci_95 =
quantile(eta_B_values, c(0.025, 0.975)))
  summary_interaction <- c(mean = mean(eta_interaction_values), sd =
sd(eta_interaction_values), ci_95 = quantile(eta_interaction_values, c(0.025,
0.975)))

  return(list(summary_A = summary_A, summary_B = summary_B,
summary_interaction = summary_interaction))
}
```

## 6. Sequential Permutation Testing

The permutation test is performed sequentially, shuffling A, B, and interaction terms, fitting the model and calculating eta squared distributions. Density plots are also generated for easier visual understanding of the distribution.

```
stratified_perm_test_sequential <- function(data, B = 1000) {
  fit <- lm(Y ~ A * B, data = data)
  eta_squared_obs <- calculate_eta_squared(fit, data)

  # Run permutations sequentially
  perm_results <- vector("list", B)
  for (i in 1:B) {
    perm_data_A <- permute_data(data, "main_A")
    perm_data_B <- permute_data(data, "main_B")
    perm_data_AB <- permute_data(data, "interaction")

    fit_A <- lm(Y ~ A * B, data = perm_data_A)
    fit_B <- lm(Y ~ A * B, data = perm_data_B)
    fit_AB <- lm(Y ~ A * B, data = perm_data_AB)

    perm_eta_A <- calculate_eta_squared(fit_A, perm_data_A)$eta_A
    perm_eta_B <- calculate_eta_squared(fit_B, perm_data_B)$eta_B
    perm_eta_interaction <- calculate_eta_squared(fit_AB,
perm_data_AB)$eta_interaction

    perm_results[[i]] <- list(
      eta_A = perm_eta_A,
      eta_B = perm_eta_B,
      eta_interaction = perm_eta_interaction
    )
  }

  # Extract permuted values
  perm_eta_A <- sapply(perm_results, `[[`, "eta_A")
  perm_eta_B <- sapply(perm_results, `[[`, "eta_B")
  perm_eta_interaction <- sapply(perm_results, `[[`, "eta_interaction")

  # Compute p-values (checking greater than observed)
  p_eta_A <- mean(perm_eta_A >= eta_squared_obs$eta_A)
  p_eta_B <- mean(perm_eta_B >= eta_squared_obs$eta_B)
  p_eta_interaction <- mean(perm_eta_interaction >=
eta_squared_obs$eta_interaction)

  # Create separate data frames for visualization
  perm_df_A <- data.frame(eta = perm_eta_A, type = "A")
  perm_df_B <- data.frame(eta = perm_eta_B, type = "B")
  perm_df_interaction <- data.frame(eta = perm_eta_interaction, type =
"Interaction")

  # Main Effect A Plot
  plot_A <- ggplot(perm_df_A, aes(x = eta)) +
    geom_density(fill = "blue", alpha = 0.5) +
    geom_vline(aes(xintercept = eta_squared_obs$eta_A), color = "darkblue",
linetype = "dashed", linewidth = 1) +
    labs(title = "Main Effect A", x = "Eta Squared for Main Effect A", y =
"Density") +
    theme_minimal()

  # Main Effect B Plot
  plot_B <- ggplot(perm_df_B, aes(x = eta)) +
    geom_density(fill = "red", alpha = 0.5) +
    geom_vline(aes(xintercept = eta_squared_obs$eta_B), color = "darkred",
linetype = "dashed", linewidth = 1) +
    labs(title = "Main Effect B", x = "Eta Squared for Main Effect B", y =
"Density") +
    theme_minimal()
  # Interaction Plot
  plot_interaction <- ggplot(perm_df_interaction, aes(x = eta)) +
    geom_density(fill = "green", alpha = 0.5) +
    geom_vline(aes(xintercept = eta_squared_obs$eta_interaction), color =
"darkgreen", linetype = "dashed", linewidth = 1) +
    labs(title = "Interaction (A:B)", x = "Eta Squared for Interaction A:B",
y = "Density") +
    theme_minimal()

  # Show the plots
  print(plot_A)
  print(plot_B)
  print(plot_interaction)

  # Return the results and p-values
  return(list(
    eta_A = eta_squared_obs$eta_A,
    eta_B = eta_squared_obs$eta_B,
    eta_interaction = eta_squared_obs$eta_interaction,
    p_eta_A = p_eta_A,
    p_eta_B = p_eta_B,
    p_eta_interaction = p_eta_interaction
  ))
}
```

For ease of understanding, the final results will be showcased in a table similar to that of ANOVA, making it easier to compare the results of the novel approach with the traditional methods.

The permutation-based p values or the proportion of permuted values greater than or equal to the observed eta squared values, determine the significance of the effects. The vertical line on the null distribution visualization indicate the observed effect.

## Validation

The final results obtained from the Bootstrap-Permutation approach, can be validated by comparing them with the results of traditional ANOVA and its famous non-parametric alternative ART ANOVA.

## Test Robustness

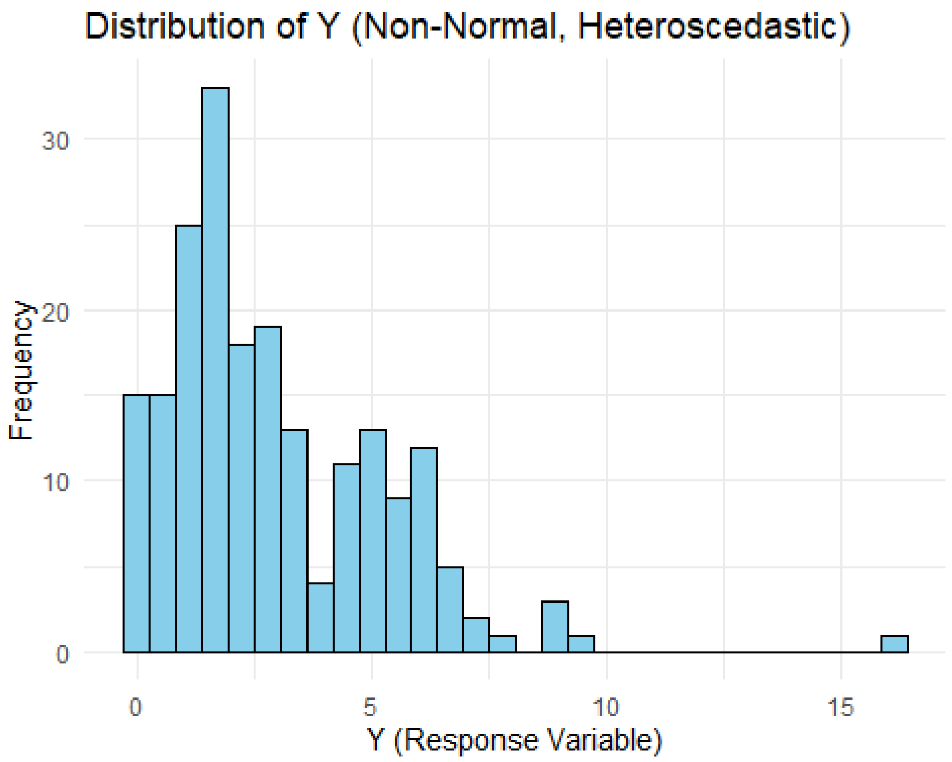
A dataset of synthetic is generated such that the data is of non-normal nature, making it viable to use our novel non-parametric approach to assess the effects of the two factors.

### *Generating synthetic data*

```
simulate_non_normal_hetero_interaction <- function(n) {

  A <- factor(rep(c("A1", "A2"), each = n / 2))
  B <- factor(rep(c("B1", "B2"), times = n / 2))

  mean_A <- ifelse(A == "A2", 1, 0)
  mean_B <- ifelse(B == "B2", 1.5, 0)
  interaction_effect <- ifelse(A == "A2" & B == "B2", 2, 0)

  residuals <- rexp(n, rate = 1)

  residuals <- residuals * (ifelse(A == "A2", 1.5, 1))

  Y <- mean_A + mean_B + interaction_effect + residuals

  # Return the simulated data
  data.frame(A, B, Y)
}

# Generate the simulated data
set.seed(4423)
data <- simulate_non_normal_hetero_interaction(n = 200)
```

### *Visualize Distribution of original data*

```
library(ggplot2)

Warning: package 'ggplot2' was built under R version 4.4.2

ggplot(data, aes(x = Y)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  theme_minimal() +
  labs(title = "Distribution of Y (Non-Normal, Heteroscedastic)",
       x = "Y (Response Variable)", y = "Frequency")
```



### *Checking assumptions*

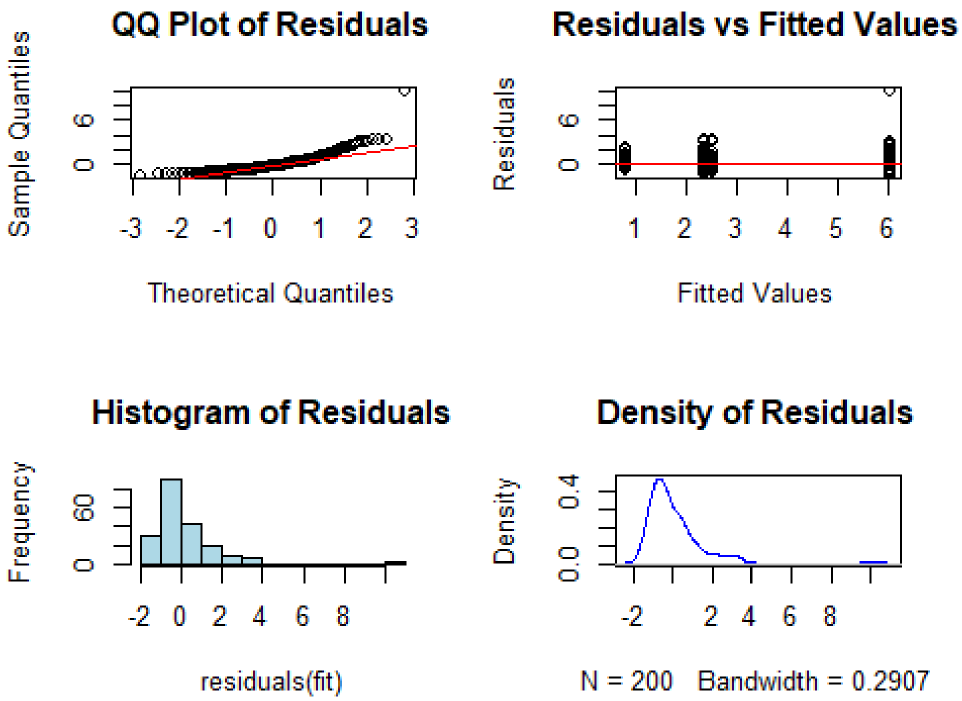
Now the homoscedasticity assumptions and the presence of influential observations is checked as follows.

```
# Assumption Diagnostic results
library(lmtest)
Warning: package 'lmtest' was built under R version 4.4.2
Loading required package: zoo
Warning: package 'zoo' was built under R version 4.4.2

Attaching package: 'zoo'
The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
library(car)
```

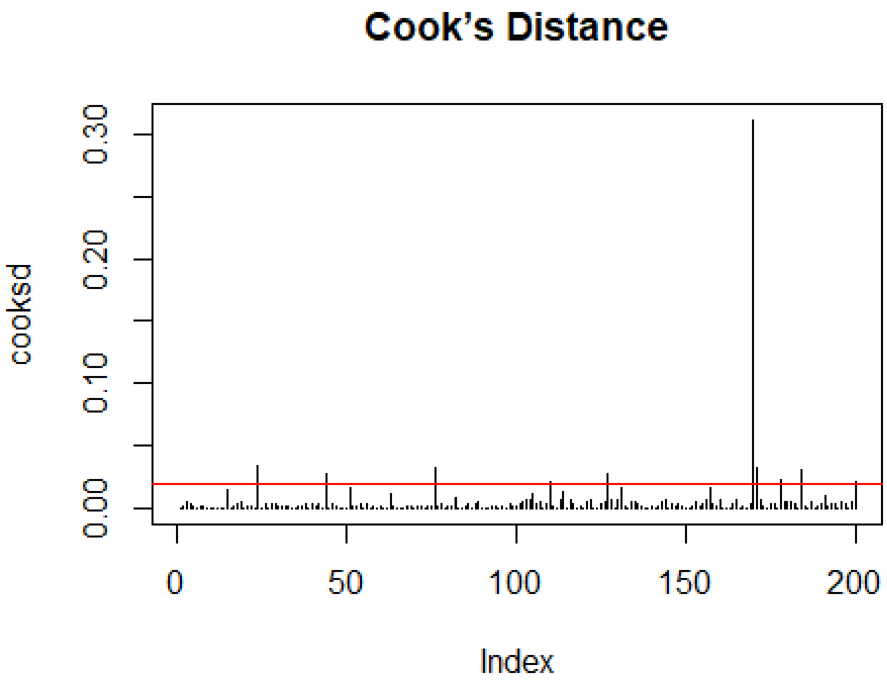
```
Warning: package 'car' was built under R version 4.4.2

Loading required package: carData

Warning: package 'carData' was built under R version 4.4.2
```

```r
homoscedasticity_tests <- function(data) {
  fit <- lm(Y ~ A * B, data = data)

  cat("\n### Breusch-Pagan Test ###\n")
  print(bptest(fit))

  cat("\n### Levene’s Test ###\n")
  print(leveneTest(Y ~ A * B, data = data))

  # Diagnostic plots
  par(mfrow = c(2,2))
  qqnorm(residuals(fit), main = "QQ Plot of Residuals")
  qqline(residuals(fit), col = "red")
  plot(fitted(fit), residuals(fit), main="Residuals vs Fitted Values",
xlab="Fitted Values", ylab="Residuals")
  abline(h = 0, col = "red")
  hist(residuals(fit), main = "Histogram of Residuals", breaks = 10, col =
"lightblue", border = "black")
  plot(density(residuals(fit)), main = "Density of Residuals", col = "blue")

  # Normality test
  cat("\n### Shapiro-Wilk Test for Normality ###\n")
  print(shapiro.test(residuals(fit)))
}

homoscedasticity_tests(data)
```

```
### Breusch-Pagan Test ###

    studentized Breusch-Pagan test

data:  fit
BP = 4.6381, df = 3, p-value = 0.2003


### Levene’s Test ###
Levene's Test for Homogeneity of Variance (center = median)
       Df F value  Pr(>F)
group   3  2.8539 0.03841 *
      196
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
### Shapiro-Wilk Test for Normality ###

    Shapiro-Wilk normality test

data:  residuals(fit)
W = 0.77691, p-value = 3.677e-16

influence_analysis(data)
```

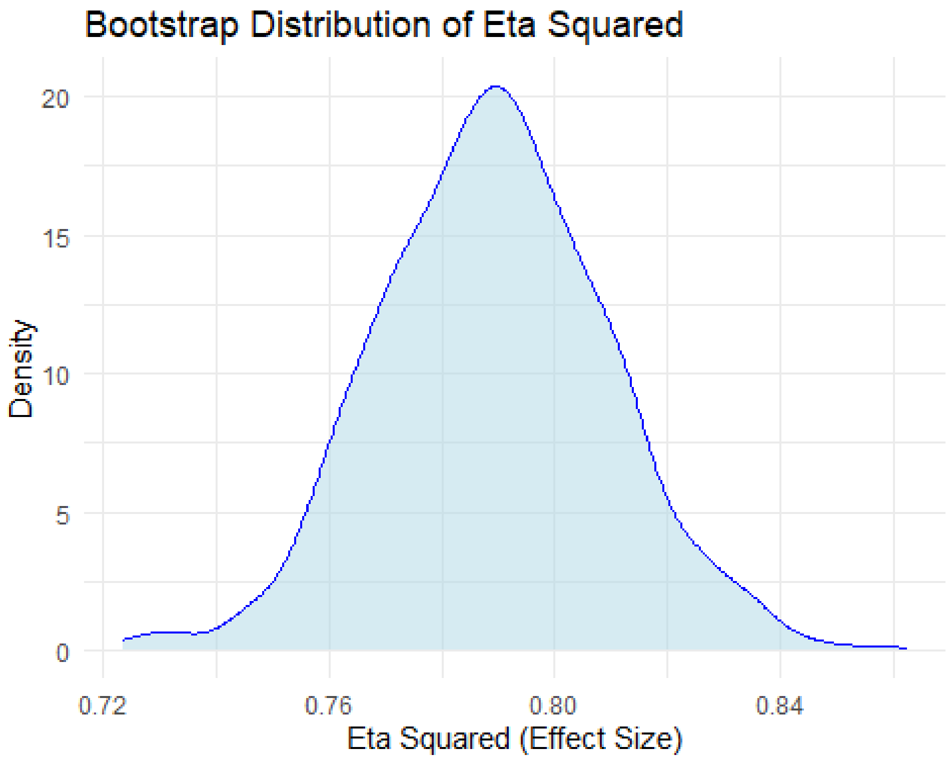


Although the Breusch-Pagan test does not result in significant results, both Levene's test and Shapiro-Wilk Test results in p values less than 0.05, indicating that there is a reasonable justification to apply a non-parametric approach to this data.

### *Bootstrap Implementation*

```
bootstrap_type <- choose_bootstrap_type(data)
print(paste("Using", bootstrap_type, "bootstrap"))

[1] "Using winsorized bootstrap"

results <- bootstrap_functions[[bootstrap_type]](data, num_samples = 1000)

boot_ci <- quantile(results, c(0.025, 0.975))
boot_se <- sd(results)

print(paste("Bootstrap 95% CI:", boot_ci[1], "to", boot_ci[2]))

[1] "Bootstrap 95% CI: 0.750580491823835 to 0.830491421427131"

print(paste("Bootstrap Standard Error:", boot_se))

[1] "Bootstrap Standard Error: 0.0202501722820937"
```

```
boot_df <- data.frame(eta = results, type = "Bootstrap")

ggplot(boot_df, aes(x = eta)) +
  geom_density(fill = "lightblue", color = "blue", alpha = 0.5) +
  labs(title = "Bootstrap Distribution of Eta Squared",
       x = "Eta Squared (Effect Size)",
       y = "Density") +
  theme_minimal()
```

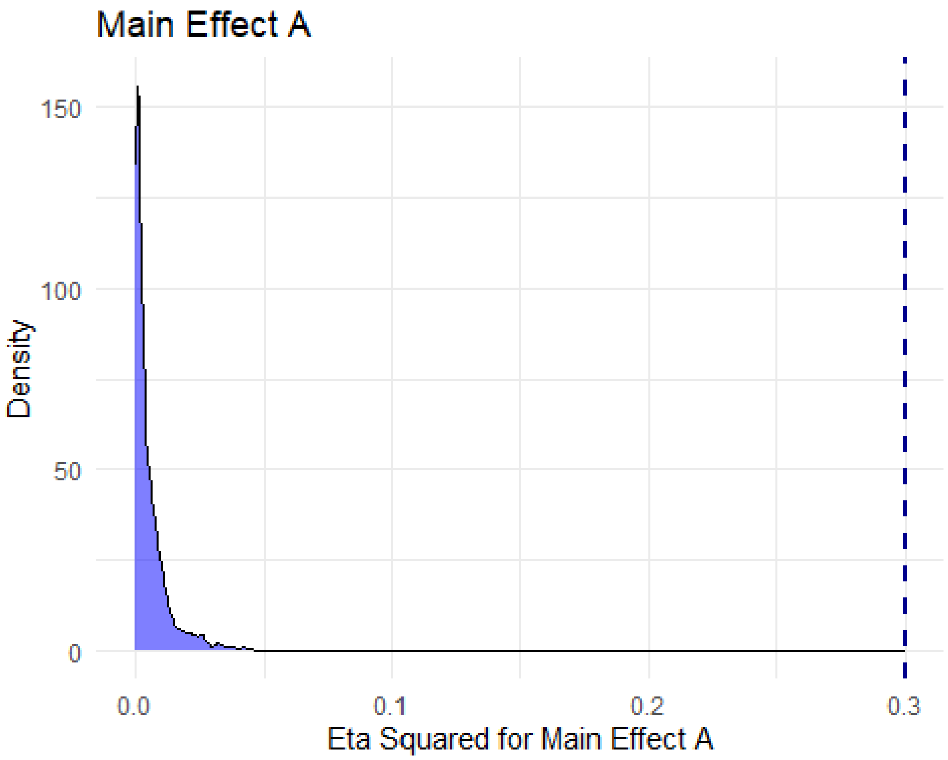


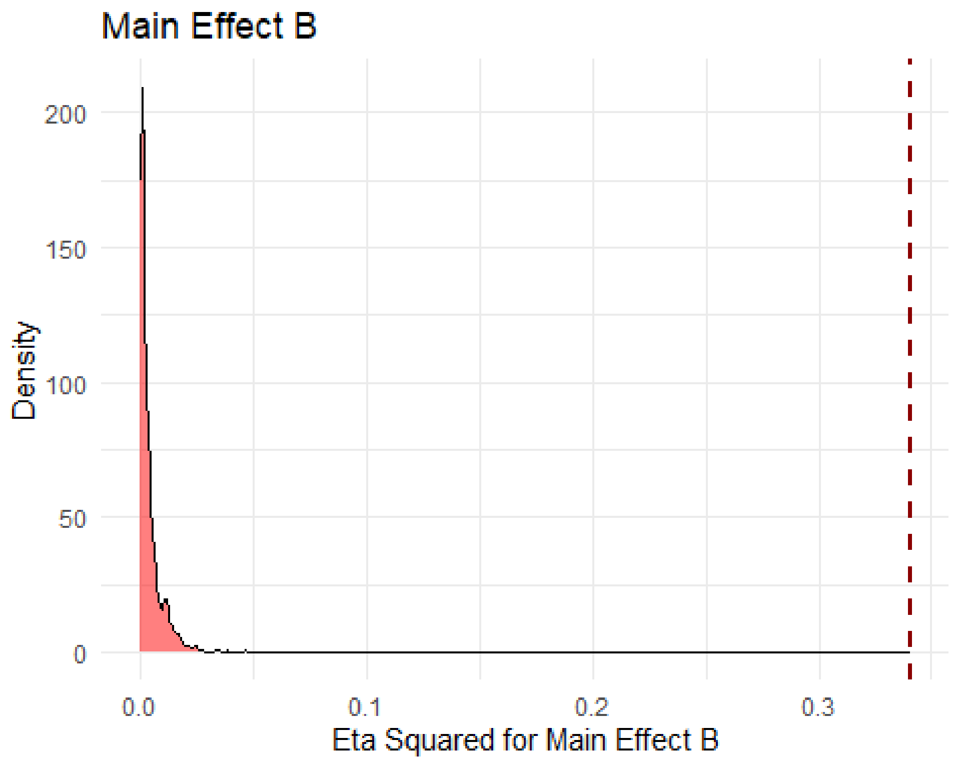
```
results_df <- data.frame(
  bootstrap_type = bootstrap_type,
  mean_eta = mean(results),
  ci_lower = boot_ci[1],
  ci_upper = boot_ci[2],
  se_eta = boot_se
)
print(results_df)

     bootstrap_type  mean_eta  ci_lower  ci_upper     se_eta
2.5%     winsorized 0.7890454 0.7505805 0.8304914 0.02025017
```

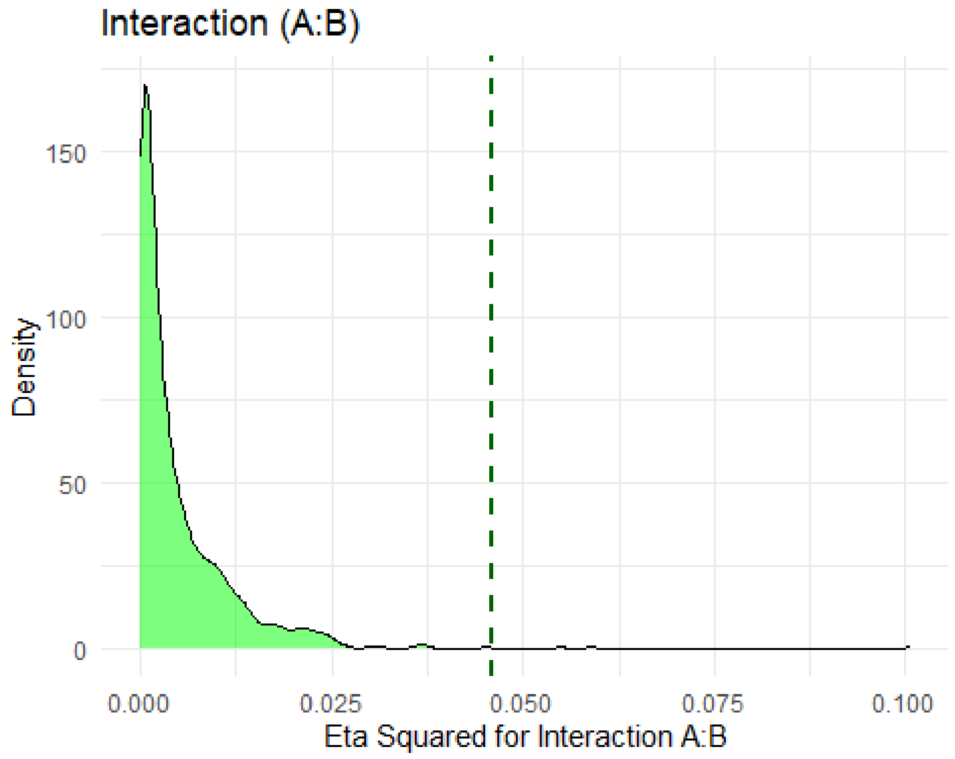
Due to the presence of outliers, ‘Winsorized Bootstrap’ was chosen as the best approach. Moving forward, the 95% Bootstrap Confidence Interval was calculated with 0.02 standard error approximately.

### *Permutation Approach*

```
library(stats)
library(data.table)

Warning: package 'data.table' was built under R version 4.4.2


Attaching package: 'data.table'

The following objects are masked from 'package:zoo':

    yearmon, yearqtr

# Run the sequential permutation test
results <- stratified_perm_test_sequential(data, B = 1000)
```







```
# Create a data frame to mimic an ANOVA table
anova_table <- data.frame(
  Term = c("A", "B", "A:B (Interaction)"),
  Eta_Squared = c(results$eta_A, results$eta_B, results$eta_interaction),
  P_Value = c(results$p_eta_A, results$p_eta_B, results$p_eta_interaction),
  Significance = c(
    ifelse(results$p_eta_A < 0.05, "***", ""),
    ifelse(results$p_eta_B < 0.05, "***", ""),
    ifelse(results$p_eta_interaction < 0.05, "**", "")
  )
)

# Print the table in a clear format
print(anova_table)

               Term Eta_Squared P_Value Significance
1                 A  0.30042806   0.000          ***
2                 B  0.34076958   0.000          ***
3 A:B (Interaction)  0.04590757   0.003           **
```

The results from the novel non parametric approach for two-way ANOVA show that, all effects A, B, and interaction A*B are significant at 5% level of significance.

### *Validation*

```
#If assumptions are met
fit_anova <- aov(Y ~ A * B, data = data)
anova(fit_anova)

Analysis of Variance Table

Response: Y
           Df Sum Sq Mean Sq F value    Pr(>F)
A           1 320.86  320.86 188.191 < 2.2e-16 ***
B           1 363.95  363.95 213.461 < 2.2e-16 ***
A:B         1  49.03   49.03  28.757 2.297e-07 ***
Residuals 196 334.18    1.70
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#For non-normal data
library(ARTool)
art_model <- art(Y ~ A * B, data = data)
anova(art_model)

Analysis of Variance of Aligned Rank Transformed Data

Table Type: Anova Table (Type III tests)
Model: No Repeated Measures (lm)
Response: art(Y)
```

```
      Df Df.res F value     Pr(>F)
1 A    1    196 325.450 < 2.22e-16 ***
2 B    1    196 323.403 < 2.22e-16 ***
3 A:B  1    196  49.532 3.2001e-11 ***
---
Signif. codes:   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both the traditional ANOVA and ART ANOVA approaches show that the effects A, B, and A*B interaction are significant at 5% level of significance.

## Power Analysis

Since power analysis requires many simulations, parallel processing is used to speed up computations by running multiple simulations simultaneously. Experimental data with 2 factors of 2 levels each are created.

First, the power analysis is implemented for normally distributed response variable.

```
library(future.apply)

Warning: package 'future.apply' was built under R version 4.4.2

Loading required package: future

Warning: package 'future' was built under R version 4.4.3

plan(multisession)

power_analysis_permutation_future <- function(n_per_group, effect_size,
                                              num_simulations, alpha, B) {
  B <- as.integer(B)
  if (is.na(B) || B <= 0) stop("B must be a positive integer")

  results <- future_lapply(1:num_simulations, function(i) {
    A <- rep(c("A1", "A2"), each = n_per_group * 2)
    B_factor <- rep(c("B1", "B2"), times = n_per_group * 2)

    #For normal data
    Y <- rnorm(length(A), mean = ifelse(A == "A1" & B_factor == "B1", 0 +
effect_size, 0), sd = 1)

    data <- data.frame(A = factor(A), B = factor(B_factor), Y = Y)
    res <- stratified_perm_test_sequential(data, B = B)

    c(p_A = as.numeric(res$p_eta_A < alpha),
      p_B = as.numeric(res$p_eta_B < alpha),
      p_interaction = as.numeric(res$p_eta_interaction < alpha))
  }, future.seed = TRUE)

  results <- do.call(rbind, results)

  return(list(
    estimated_power_A = mean(results[, "p_A"]),
    estimated_power_B = mean(results[, "p_B"]),
    estimated_power_interaction = mean(results[, "p_interaction"]),
    num_simulations = num_simulations,
    alpha = alpha
  ))
}

set.seed(123)
power_results_future <- power_analysis_permutation_future(n_per_group = 100,
effect_size = 0.3,
                                                          num_simulations =
200, alpha = 0.05, B = 500)
print(power_results_future)

$estimated_power_A
[1] 0.35

$estimated_power_B
[1] 0.3

$estimated_power_interaction
[1] 0.33

$num_simulations
[1] 200

$alpha
[1] 0.05
```

For normal data, the implemented approach has low power, approximately around 30%. Since method was specifically crafted to suit data of non-normal nature, next the power analysis is conducted for skewed response generated using exponential distribution.

```
library(future.apply)

plan(multisession)

power_analysis_permutation_future <- function(n_per_group, effect_size,
                                              num_simulations, alpha, B) {
  B <- as.integer(B)
  if (is.na(B) || B <= 0) stop("B must be a positive integer")

  results <- future_lapply(1:num_simulations, function(i) {
    A <- rep(c("A1", "A2"), each = n_per_group * 2)
    B_factor <- rep(c("B1", "B2"), times = n_per_group * 2)

    #For non-normal data
    Y <- rexp(length(A), rate = 1 / ifelse(A == "A1" & B_factor == "B1",
effect_size, 1))

    data <- data.frame(A = factor(A), B = factor(B_factor), Y = Y)

    res <- stratified_perm_test_sequential(data, B = B)

    c(p_A = as.numeric(res$p_eta_A < alpha),
      p_B = as.numeric(res$p_eta_B < alpha),
      p_interaction = as.numeric(res$p_eta_interaction < alpha))
  }, future.seed = TRUE)

  results <- do.call(rbind, results)

  return(list(
    estimated_power_A = mean(results[, "p_A"]),
    estimated_power_B = mean(results[, "p_B"]),
    estimated_power_interaction = mean(results[, "p_interaction"]),
    num_simulations = num_simulations,
    alpha = alpha
  ))
}

set.seed(123)
power_results_future <- power_analysis_permutation_future(n_per_group = 100,
effect_size = 0.3,
                                                          num_simulations =
200, alpha = 0.05, B = 500)
print(power_results_future)

$estimated_power_A
[1] 0.98

$estimated_power_B
[1] 0.98
```

```
$estimated_power_interaction
[1] 0.99

$num_simulations
[1] 200

$alpha
[1] 0.05
```

Now, the method has approximately 98% power, implying the novel approach is very powerful for skewed, non-nomral data making it a strong alternative for non-parametric two-way ANOVA in such cases. But power is low for normally distributed data suggesting that, it is better to stick with traditional ANOVA in these situations.

# Conclusion

The Bootstrap and Permutation based Non-Parametric Two-way ANOVA approach studied shows strong adaptability to different data distributions of non-normal nature. The power analysis and the robustness check on data where traditional assumptions are violated, further confirm the validity of the approach.

# Challenges

One of the main difficulties faced with the bootstrap-permutation method was the balance between precision and computational speed. Although bootstrapping offers a strong estimate of the sampling distribution of $\eta^2$, and permutation provides a significance test without assumptions, both techniques are computationally demanding, especially when dealing with large datasets and a high number of resamples. The sequential nature of these resampling methods further complicates this situation, as each iteration relies on the result of the previous computation, hindering potential efficiency improvements that could be realized through parallel processing (Efron & Tibshirani, 1994). Moreover, the significant memory resources needed to store and manage numerous resampled datasets represent a considerable challenge, particularly in high-dimensional contexts.

To address these challenges, alternative methods like using F-statistics, even in non-parametric settings, can be helpful, as they are computationally efficient while still capturing effect sizes. Techniques such as Monte Carlo approximations, stratified resampling, and adaptive methods can also reduce the computational burden without losing much accuracy. Additionally, using parallel computing or cloud-based resources can improve efficiency, though they require specialized implementation. Ultimately, a balance must be found between statistical accuracy and computational feasibility, especially for large datasets.

# References

1. Scheirer, C. J., Ray, W. S., & Hare, N. (1976). The analysis of ranked data derived from completely randomized factorial designs. *Biometrics, 32*(2), 429-434. https://doi.org/10.2307/2529511
2. Aligned Rank Transform Techniques for Analysis of Variance and Multiple Comparisons. ResearchGate
3. Rank Transformations as a Bridge Between Parametric and Nonparametric Statistics - Rejoinder. Retrieved from ResearchGate
4. Permutational Analysis of Variance Wikipedia
5. Anderson, M. J., & Ter Braak, C. J. F. (n.d.). Randomization Tests with R. University of Vermont
6. The R Companion: Friedman Test. R Companion
7. The R Companion: Aligned Rank Transform. R Companion
8. Semantic Scholar
9. JSTOR
10. Conover, W. J., & Iman, R. L. (1981). Rank transformations as a bridge between parametric and nonparametric statistics. The American Statistician, 35(3), 124-129. https://doi.org/10.1080/00031305.1981.10479327
11. ARTool. Aligned rank transform for nonparametric factorial ANOVA. Retrieved from https://depts.washington.edu/acelab/proj/art/index.html
12. Anderson, M. J., & Ter Braak, C. J. F. (n.d.). Permutation tests for multi-factorial analysis of variance. Retrieved from https://www.uvm.edu/~statdhtx/StatPages/R/RandomizationTestsWithR/andersonTerBrack.pdf
13. Stack Exchange - Cross Validated. Discussions on nonparametric two-way ANOVA and permutation tests. Retrieved from https://stats.stackexchange.com/
14. Holbert, C. (2022, April 16). Nonparametric two-way ANOVA. Retrieved from https://www.cfholbert.com/blog/nonparametric_two_way_anova/
15. Efron, B., & Tibshirani, R. (1994). An introduction to the bootstrap. Chapman & Hall.
16. Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). Lawrence Erlbaum Associates.

# Appendix

Group Members and Their Contribution:

- Tishani Wijekoon – s16379
  - Novel Approach, Mathematical Framework, Implementation in R
- Lasandi Perera – s16388
  - Existing methods – ART, Schierer-Hay Test
- Anupama Chandrapala – s16295
  - Existing methods – RT methods, Permutation methods

The R codes, core functions, documentation, vignettes and the R package can be found online in the GitHub Repository under the following link:

https://github.com/TishaniSakalya/Permova