Forexam122

Kyaw Min Khaing

2025-01-28

## Introduction

This analysis demonstrates the Kruskal-Wallis H-Test on small data samples. We will rank the data, apply ties correction, compute the H statistic, and determine significance.

## Data Preparation

The table below shows the sample data for three groups (High, Medium, Low)

data <- data.frame(  
 High = c(21, 23, 18, 12, 19, 20),  
 Medium = c(19, 5, 10, 11, 9, NA),  
 Low = c(7, 8, 15, 3, 6, 4)  
)  
print(data)

## High Medium Low  
## 1 21 19 7  
## 2 23 5 8  
## 3 18 10 15  
## 4 12 11 3  
## 5 19 9 6  
## 6 20 NA 4

# Convert to long format for analysis

data\_long <- data %>%  
 pivot\_longer(cols = everything(), names\_to = "Group", values\_to = "Value") %>%  
 drop\_na()  
  
# Print the data  
print(data\_long)

## # A tibble: 17 × 2  
## Group Value  
## <chr> <dbl>  
## 1 High 21  
## 2 Medium 19  
## 3 Low 7  
## 4 High 23  
## 5 Medium 5  
## 6 Low 8  
## 7 High 18  
## 8 Medium 10  
## 9 Low 15  
## 10 High 12  
## 11 Medium 11  
## 12 Low 3  
## 13 High 19  
## 14 Medium 9  
## 15 Low 6  
## 16 High 20  
## 17 Low 4

## Ranking Data and Handling Ties

The Kruskal-Wallis test uses ranks rather than raw data values. We rank all values, including ties.

ranked\_data <- data\_long %>%  
 mutate(Rank = rank(Value))  
  
# Display ranked data  
print(ranked\_data)

## # A tibble: 17 × 3  
## Group Value Rank  
## <chr> <dbl> <dbl>  
## 1 High 21 16   
## 2 Medium 19 13.5  
## 3 Low 7 5   
## 4 High 23 17   
## 5 Medium 5 3   
## 6 Low 8 6   
## 7 High 18 12   
## 8 Medium 10 8   
## 9 Low 15 11   
## 10 High 12 10   
## 11 Medium 11 9   
## 12 Low 3 1   
## 13 High 19 13.5  
## 14 Medium 9 7   
## 15 Low 6 4   
## 16 High 20 15   
## 17 Low 4 2

## Compute Group Ranks and Sums

We calculate the rank sum for each group.

# Calculate rank sums  
rank\_sums <- ranked\_data %>%  
 group\_by(Group) %>%  
 summarise(  
 n = n(),  
 Rank\_Sum = sum(Rank)  
 )  
  
# Print rank sums  
print(rank\_sums)

## # A tibble: 3 × 3  
## Group n Rank\_Sum  
## <chr> <int> <dbl>  
## 1 High 6 83.5  
## 2 Low 6 29   
## 3 Medium 5 40.5

## Compute Ties Correction (C\_H)

The ties correction is applied to adjust for tied ranks. The formula for the ties correction is:

Where: - is the number of tied ranks in each group of ties. - is the total number of observations.

ties\_table <- ranked\_data %>%  
 count(Rank)  
  
C\_H <- 1 - sum((ties\_table$n^3 - ties\_table$n) / ((nrow(ranked\_data)^3) - nrow(ranked\_data)))  
  
# Print ties correction factor  
cat("Ties Correction Factor (C\_H):", C\_H, "\n")

## Ties Correction Factor (C\_H): 0.9987745

## Compute the H-Test Statistic

We calculate the Kruskal-Wallis H-test statistic using the formula:

Where: - is the total number of observations. - is the rank sum for each group. - is the number of observations in each group.

# Total number of observations  
N <- nrow(ranked\_data)  
# Calculate H statistic  
H <- 12 / (N \* (N + 1)) \* sum((rank\_sums$Rank\_Sum^2) / rank\_sums$n) - 3 \* (N + 1)  
H\_corrected <- H / C\_H # Apply ties correction  
# Print H statistic  
cat("H-Test Statistic (Corrected):", round(H\_corrected, 4), "\n")

## H-Test Statistic (Corrected): 9.9439

## Significance Test

Compare the H statistic to the Chi-Square distribution with degrees of freedom (where is the number of groups) to determine significance.

# Degrees of freedom  
k <- ranked\_data %>%  
 summarise(num\_groups = n\_distinct(Group)) %>%  
 pull(num\_groups)  
df <- k - 1  
  
# Critical value and p-value  
alpha <- 0.05  
p\_value <- pchisq(H\_corrected, df, lower.tail = FALSE)  
  
# Report significance  
cat("Degrees of Freedom (df):", df, "\n")

## Degrees of Freedom (df): 2

cat("p-value for Kruskal-Wallis H-Test:", round(p\_value, 4), "\n")

## p-value for Kruskal-Wallis H-Test: 0.0069

if (p\_value < alpha) {  
 cat("The result is significant at α =", alpha, "\n")  
} else {  
 cat("The result is not significant at α =", alpha, "\n")  
}

## The result is significant at α = 0.05

# Pairwise Mann-Whitney U tests  
pairwise\_results <- ranked\_data %>%  
 pairwise\_wilcox\_test(  
 Value ~ Group,  
 p.adjust.method = "bonferroni"  
 )  
  
# Display results  
print(pairwise\_results)

## # A tibble: 3 × 9  
## .y. group1 group2 n1 n2 statistic p p.adj p.adj.signif  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl> <dbl> <chr>   
## 1 Value High Low 6 6 35 0.004 0.013 \*   
## 2 Value High Medium 6 5 27.5 0.028 0.084 ns   
## 3 Value Low Medium 6 5 7 0.177 0.531 ns

Final Report

cat("### Final Report\n")

## ### Final Report

cat("Three groups were analyzed: High (n = 6), Medium (n = 5), and Low (n = 6).\n")

## Three groups were analyzed: High (n = 6), Medium (n = 5), and Low (n = 6).

cat("The corrected Kruskal-Wallis H statistic was", round(H\_corrected, 4), "with", df, "degrees of freedom.\n")

## The corrected Kruskal-Wallis H statistic was 9.9439 with 2 degrees of freedom.

cat("The p-value was", round(p\_value, 4), "at α =", alpha, ".\n")

## The p-value was 0.0069 at α = 0.05 .

if (p\_value < alpha) {  
 cat("We reject the null hypothesis and conclude that at least one group differs significantly.\n")  
} else {  
 cat("We fail to reject the null hypothesis, suggesting no significant differences between the groups.\n")  
}

## We reject the null hypothesis and conclude that at least one group differs significantly.

# Pairwise significance summary  
cat("\n### Pairwise Comparisons (Mann-Whitney U Test):\n")

##   
## ### Pairwise Comparisons (Mann-Whitney U Test):

cat("Results adjusted using Bonferroni correction:\n")

## Results adjusted using Bonferroni correction:

## Visualization

Visualize the ranked data for each group using a boxplot.

pairwise\_results <- data\_long %>%  
 pairwise\_wilcox\_test(Value ~ Group, p.adjust.method = "bonferroni")  
  
# Create the rank plot with significance brackets  
rank\_plot <- ggplot(ranked\_data, aes(x = Group, y = Rank, fill = Group)) +  
 geom\_boxplot(outlier.color = "red", outlier.shape = 8) +  
 labs(  
 title = "Rank Distribution by Group",  
 x = "Group",  
 y = "Rank"  
 ) +  
 theme\_minimal() +  
 geom\_signif(  
 comparisons = list(  
 c("High", "Medium"),  
 c("High", "Low"),  
 c("Medium", "Low")  
 ),  
 map\_signif\_level = TRUE,  
 test = "wilcox.test", # Perform Mann-Whitney U test directly  
 step\_increase = 0.1 # Adjust spacing to avoid overlapping  
 )  
  
# Print the plot  
print(rank\_plot)

## Warning in wilcox.test.default(c(16, 17, 12, 10, 13.5, 15), c(13.5, 3, 8, :  
## cannot compute exact p-value with ties

