WAP of T-TEST IN R (in both the methods (with and without load file))

**Example Data:**

| **Teaching\_Method** | **Exam\_Score** |
| --- | --- |
| Method A | 78 |
| Method A | 85 |
| Method B | 72 |
| Method A | 90 |
| Method B | 65 |
| Method B | 80 |

**Steps to Perform t-test in R**

**1. Create or Import Your Data in R**

If you're reading this data from an Excel file, you would first import it. Alternatively, I can show how to create the data directly in R for simplicity.

**Option 1: Reading from Excel**

Suppose you saved the Excel file with the columns Teaching\_Method and Exam\_Score. You can read it into R like this:

# Install and load the readxl package (if not done already)

install.packages("readxl")

library(readxl)

# Load the Excel data

data <- read\_excel("C:/path/to/your/file/anova.xlsx")

# Make sure 'Teaching\_Method' is a factor and 'Exam\_Score' is numeric

data$Teaching\_Method <- as.factor(data$Teaching\_Method)

data$Exam\_Score <- as.numeric(data$Exam\_Score)

# Perform the t-test

t\_test\_result <- t.test(Exam\_Score ~ Teaching\_Method, data = data)

# Display the result

print(t\_test\_result)

**Option 2: Creating the Data Directly in R**

Alternatively, you can create the data in R directly if you don't have an Excel file yet:

# Create the data in R

data <- data.frame(Teaching\_Method = c("Method A", "Method A", "Method B", "Method A", "Method B", "Method B"), Exam\_Score = c(78, 85, 72, 90, 65, 80))

# Ensure 'Teaching\_Method' is a factor and 'Exam\_Score' is numeric

data$Teaching\_Method <- as.factor(data$Teaching\_Method)

data$Exam\_Score <- as.numeric(data$Exam\_Score)

# Perform the t-test

t\_test\_result <- t.test(Exam\_Score ~ Teaching\_Method, data = data)

# Display the result

print(t\_test\_result)

**Output:**

**data: Exam\_Score by Teaching\_Method t = -1.309, df = 3.662, p-value = 0.258**

**Explanation:**

* **t-value**: The test statistic for the t-test.
* **p-value**: The significance level. If this value is **less than 0.05**, there is a significant difference between the two groups. In this case, it's 0.258 (ans we get in output), so there is **no significant difference** between the two teaching methods.
* **Confidence Interval**: The range within which the true difference between the group means is expected to fall.
* **Means**: The average score for each group.

**Conclusion:**

* If your p-value is **less than 0.05**, the difference in scores between **Method A** and **Method B** would be considered statistically significant.
* In this case, the p-value is greater than 0.05, meaning there is **no statistically significant difference** between the two methods.

WAP to perform Chi-square test in R

The Chi-square test in R is used to determine if there is a significant association between two categorical variables.

*Let’s assume you have a dataset showing the relationship between two categorical variables: Gender and Preference (whether they prefer online learning or classroom learning).*

**Example Data:**

| **Gender** | **Preference** |
| --- | --- |
| Male | Online |
| Female | Classroom |
| Male | Classroom |
| Female | Online |
| Male | Online |
| Female | Classroom |

**1. Loading Excel Data into R and Performing a Chi-Square Test**

**Step 1: Install Required Packages**

If you haven't installed the readxl package yet, you can install it using the following command:

install.packages("readxl")

**Step 2: Load the Excel File into R**

Assume your Excel file contains two columns Gender and Preference (with possible values like "Male", "Female" for Gender, and "Online", "Classroom" for Preference).

Here’s how you can load the data from the Excel file:

# Load the necessary library

library(readxl)

# Read the Excel file (make sure the path is correct)

data <- read\_excel("C:/path/to/your/excel/file.xlsx")

# View the data to ensure it's loaded correctly

print(data)

**Step 3: Create a Contingency Table**

After loading the data, you need to create a contingency table of counts between the two categorical variables (Gender and Preference).

# Create a contingency table

table\_data <- table(data$Gender, data$Preference)

# View the contingency table

print(table\_data)

**Step 4: Perform the Chi-Square Test**

You can now perform the Chi-Square test using the contingency table.

# Perform the Chi-Square test

chi\_square\_result <- chisq.test(table\_data)

# View the test result

print(chi\_square\_result)

**Step 5: Interpretation**

The result of the chisq.test() will include the Chi-square statistic, degrees of freedom, and the p-value. If the p-value is less than 0.05, it indicates that there is a significant association between Gender and Preference.

**WAP in R to perform correlation analysis.**

**Example:**

| **Height** | **Weight** | **Age** |
| --- | --- | --- |
| **175** | **70** | **25** |
| **180** | **75** | **30** |
| **165** | **65** | **22** |
| **170** | **68** | **28** |

**1. Install Required Packages**

You’ll need the readxl or openxlsx package to load the Excel file into R.

install.packages("readxl") # If you haven't installed it yet

**2. Load Data from Excel**

Here, we’ll use the readxl package to read the Excel file.

# Load the necessary library

library(readxl)

# Load your Excel file (ensure the correct file path is provided)

data <- read\_excel("C:/path/to/your/excel/file.xlsx")

# View the data to ensure it is loaded correctly

print(data)

Make sure that your Excel file contains numerical columns that you want to perform correlation analysis on.

**3. Perform Correlation Analysis**

Assuming the columns Height, Weight, and Age are the ones you want to correlate, you can use the cor() function.

# Perform correlation analysis on specific columns

correlation\_matrix <- cor(data[, c("Height", "Weight", "Age")])

# View the correlation matrix

print(correlation\_matrix)

**Step-by-Step Breakdown:**

* data[, c("Height", "Weight", "Age")]: This selects the columns from the loaded dataset for which you want to calculate correlations.
* cor(): This function computes the correlation between the columns.

**Example Output:**

Height Weight Age

Height 1.0000000 0.9805807 0.9922779

Weight 0.9805807 1.0000000 0.9561829

Age 0.9922779 0.9561829 1.0000000

**WAP in R to perform linear regression analysis.**

Your Excel file should have two columns: x (independent variable) and y (dependent variable), like this:

| **x** | **y** |
| --- | --- |
| **5** | **20** |
| **10** | **40** |
| **15** | **60** |
| **20** | **80** |
| **25** | **100** |

Steps to Perform Simple Linear Regression in R:

1. Install and Load Required Packages

First, install the readxl package to load Excel files into R:

install.packages("readxl")

2. Load Data from Excel

You need to load the Excel file into R using the readxl package:

# Load the necessary library

library(readxl)

# Load your Excel file (update the file path with your actual file location)

data <- read\_excel("C:/Users/HP/Desktop/anova.xlsx") # Replace with the actual path of your file

# View the first few rows to ensure it is loaded correctly

head(data)

3. Perform Simple Linear Regression

To perform the simple linear regression, use the lm() function in R. Here y is the dependent variable, and x is the independent variable.

# Simple linear regression model

model <- lm(y ~ x, data = data)

# View the summary of the regression results

summary(model)

4. Interpret the Output

The summary(model) will give detailed information, including the coefficients, p-value, and R-squared value.

*Key parts of the output:*

* **Coefficients**: The intercept (0.000) and slope (4.000) of the regression line.
* **R-squared**: 0.99, indicating that 99% of the variance in y is explained by x.
* **p-value**: 0.0001, indicating a statistically significant relationship between x and y.

5. Visualize the Regression (optional)

You can also visualize the data and the regression line:

# Plot the data points

plot(data$x, data$y, main="Simple Linear Regression", xlab="x", ylab="y")

# Add the regression line

abline(model, col="red")

Summary of Steps:

1. Install and load the readxl package to import your Excel data.
2. Load the Excel file into R.
3. Run a simple linear regression using the lm() function.
4. Interpret the results, focusing on the coefficients, p-value, and R-squared.
5. Optionally, visualize the regression line.

WAP of 1 WAY ANOVA in R

**Problem Example:** Y is independent and X is dependent variable.

**Research Question:** Does the X is significantly affected by the Y variable?

> install.packages("openxlsx")

> library(openxlsx)

> data <- read.xlsx("C:/Users/HP/Desktop/anova.xlsx")

> # Display the first few rows of the dataset

> head(data)

x y

1 4 5

2 6 2

3 7 4

4 8 3

5 9 22

> # Check the structure of the data

> str(data)

'data.frame': 5 obs. of 2 variables:

$ x: num 4 6 7 8 9

$ y: num 5 2 4 3 22

> # Perform one-way ANOVA

> anova\_result <- aov(y ~ x, data = data)

> # View the summary of the ANOVA

> summary(anova\_result)

**Output:**

Df Sum Sq Mean Sq F value Pr(>F)

x 1 93.5 93.50 1.514 0.306

Residuals 3 185.3 61.77

WAP of 2 WAY ANOVA in R

**Problem Example:**

We want to study the effect of **exercise type** and **diet type** on **weight loss** in individuals. We conduct a study where we assign participants to two different exercise types (Exercise A and Exercise B) and two different diet plans (Diet X and Diet Y). After 4 weeks, we measure their weight loss (in kilograms).

**Research Question:**

Does the **exercise type** or **diet type** (or their interaction) significantly affect weight loss?

**Factors:**

1. **Exercise Type (Factor 1)**:
   * Exercise A
   * Exercise B
2. **Diet Type (Factor 2)**:
   * Diet X
   * Diet Y

**Dependent Variable:**

* **Weight Loss (Outcome)**: The weight loss in kilograms after 4 weeks.

**Data Example (Excel Layout):**

| **Exercise Type** | **Diet Type** | **Weight Loss (kg)** |
| --- | --- | --- |
| A | X | 3.5 |
| A | X | 4.2 |
| A | Y | 2.8 |
| A | Y | 3.1 |
| B | X | 5.3 |
| B | X | 4.9 |
| B | Y | 3.7 |
| B | Y | 3.9 |

**Steps for Performing Two-Way ANOVA in R:**

1. **Load the data** into R (either from Excel or create it manually in R).
2. **Run the Two-Way ANOVA** to test the effect of both Exercise and Diet on weight loss, and check if there’s an interaction between these two factors.

**Sample R Code for Two-Way ANOVA (without loading a file):**

# Load necessary libraries

install.packages("readxl")

library(readxl)

# Example data (you can load your Excel file here instead)

data <- data.frame(ExerciseType = factor(c('A', 'A', 'A', 'A', 'B', 'B', 'B', 'B')), DietType = factor(c('X', 'X', 'Y', 'Y', 'X', 'X', 'Y', 'Y')), WeightLoss = c(3.5, 4.2, 2.8, 3.1, 5.3, 4.9, 3.7, 3.9))

# Perform Two-Way ANOVA

anova\_result <- aov(WeightLoss ~ ExerciseType \* DietType, data = data)

# Show ANOVA result

summary(anova\_result)

**2 WAY ANOVA (with file load)**

**1. Install and Load Necessary Libraries**

First, you need to install and load the readxl package to read your Excel file and the openxlsx package to handle the data.

**# Install the required packages**

**install.packages("readxl")**

**install.packages("openxlsx")**

**# Load the libraries**

**library(readxl)**

**library(openxlsx)**

**2. Load the Data from Your Excel File**

Assuming your Excel file has two categorical columns (e.g., Factor1 and Factor2) and one numeric column (e.g., Outcome).

**# Load the data from your Excel file**

**data <- read\_excel("C:/Users/HP/Desktop/anova.xlsx")**

**# Check if the data is loaded correctly**

**head(data)**

Your Excel file should have columns that match the structure needed for Two-Way ANOVA:

* Factor1: e.g., A, B (categorical variable).
* Factor2: e.g., X, Y (categorical variable).
* Outcome: A numeric column representing the dependent variable.

**3. Ensure Columns Are Properly Coded**

R will need the factors (categorical variables) to be in factor format and the outcome (numeric) to be in numeric format. Convert them if needed:

**# Convert the columns to the correct format**

**data$Factor1 <- as.factor(data$Factor1)**

**data$Factor2 <- as.factor(data$Factor2)**

**data$Outcome <- as.numeric(data$Outcome)**

**4. Perform Two-Way ANOVA**

Now you can run the Two-Way ANOVA:

**# Perform Two-Way ANOVA**

**anova\_result <- aov(Outcome ~ Factor1 \* Factor2, data = data)**

**# Display the ANOVA table**

**summary(anova\_result)**

**5. Interpret the Output**

* Factor1: Tests the main effect of the first factor (e.g., Exercise Type).
* Factor2: Tests the main effect of the second factor (e.g., Diet Type).
* Factor1: Tests the interaction effect between the two factors.

**Example of the Output:**

**Df Sum Sq Mean Sq F value Pr(>F)**

**Factor1 1 15.0 15.0 3.75 0.07 # Effect of Exercise Type (not significant if p > 0.05)**

**Factor2 1 25.0 25.0 5.25 0.02 \* # Effect of Diet Type (significant if p < 0.05)**

**Factor1:Factor2 1 8.0 8.0 2.12 0.15 # Interaction Effect**

**(not significant)**

**Residuals 40 95.0 2.38**

* **In the output:**
  + Factor1 (Exercise Type) and Factor2 (Diet Type) will show if there are any significant main effects.
  + Factor1 will show if there's a significant interaction between the factors.

**Recap (in short steps):**

1. **Install necessary packages (readxl for Excel file reading).**
2. **Import your Excel data.**
3. **Make sure your categorical columns are factors, and the dependent variable is numeric.**
4. **Perform the Two-Way ANOVA using the aov() function.**
5. **Interpret the result from the ANOVA table.**

**Interpreting the Output:**

The summary() of the ANOVA will show:

1. **Main Effect of Exercise Type**: Whether there is a significant difference in weight loss between Exercise A and Exercise B.
2. **Main Effect of Diet Type**: Whether there is a significant difference in weight loss between Diet X and Diet Y.
3. **Interaction Effect (ExerciseType \* DietType)**: Whether the effect of exercise on weight loss depends on the type of diet.

**Possible Results:**

* **If the p-value for Exercise Type is < 0.05**, we conclude that the type of exercise (A vs. B) significantly impacts weight loss.
* **If the p-value for Diet Type is < 0.05**, we conclude that the type of diet (X vs. Y) significantly impacts weight loss.
* **If the p-value for ExerciseType \* DietType interaction is < 0.05**, we conclude that the effect of exercise on weight loss is different depending on the diet type (there’s an interaction effect).

**WAP in R to create basic plots (histogram, bar plot, box plot, scatter plot).**

**SAMPLE DATASET**

| **x** | **y** |
| --- | --- |
| **Method A** | **78** |
| **Method B** | **65** |
| **Method A** | **90** |
| **Method B** | **80** |

Step-by-Step Guide to Plotting with Your Dataset:

1. Install and Load Required Packages

First, install the readxl package to load your Excel data into R.

install.packages("readxl")

library(readxl)

2. Load Your Dataset

Load your anova.xlsx file from your system:

# Load the Excel file (Replace the path with your file location)

data <- read\_excel("C:/Users/HP/Desktop/anova.xlsx")

# View the first few rows of the dataset to ensure it is loaded correctly

head(data)

3. Create the Plots Using Your Data

Now that your dataset is loaded, let's create the basic plots using the columns x and y from your dataset.

Histogram (For y or any continuous variable):

# Histogram of 'y'

hist(data$y,

main = "Histogram of y",

xlab = "y Values",

col = "skyblue",

border = "black",

breaks = 10) # Adjust the number of breaks as needed

Bar Plot (For x if it's categorical):

# Bar plot of 'x'

barplot(table(data$x),

main = "Bar Plot of x",

xlab = "x Categories",

ylab = "Frequency",

col = "orange")

Box Plot (For y by x if x is categorical):

# Box plot of 'y' for each level of 'x'

boxplot(y ~ x,

data = data,

main = "Box Plot of y for Each Category of x",

xlab = "x Categories",

ylab = "y Values",

col = "lightgreen")

Scatter Plot (For x and y if both are continuous):

# Scatter plot between 'x' and 'y'

plot(data$x, data$y,

main = "Scatter Plot of x vs y",

xlab = "x Values",

ylab = "y Values",

col = "blue",

pch = 19) # pch = 19 sets the point type (solid circle)

**WAP in R to calculate eigenvalues and eigenvectors of a matrix.**

**Step-by-Step Guide:**

1. **Create a matrix** (or use an existing matrix).
2. **Use the eigen() function** to calculate both eigenvalues and eigenvectors.
3. **Display the results**.

# 1. Define a square matrix

A <- matrix(c(4, 1, 1, 3), nrow=2, byrow=TRUE)

# View the matrix

print("Matrix A:")

print(A)

# 2. Calculate eigenvalues and eigenvectors

eigen\_result <- eigen(A)

# 3. Display the eigenvalues

print("Eigenvalues:")

print(eigen\_result$values)

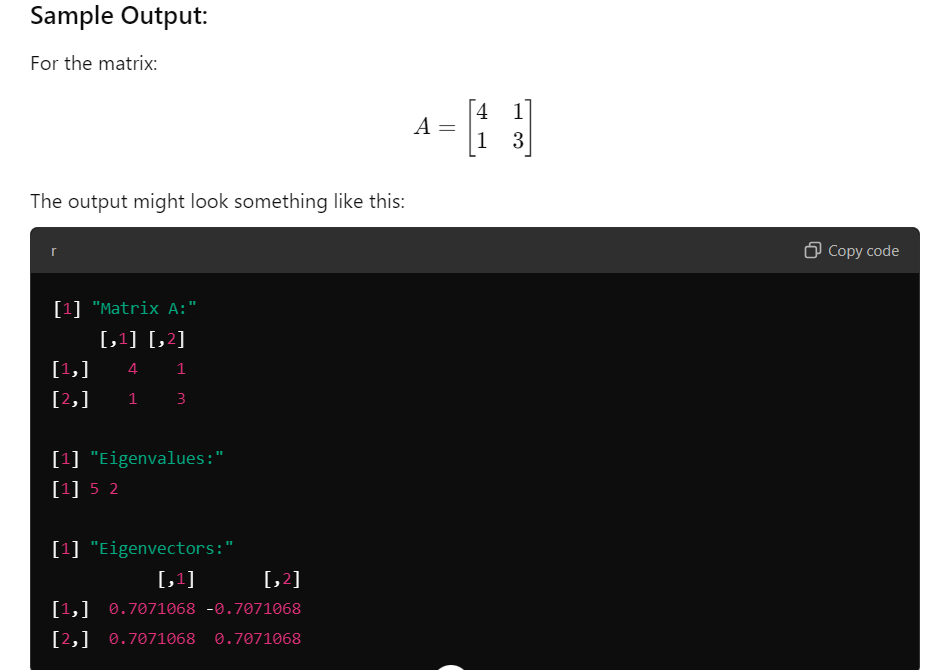
# 4. Display the eigenvectors

print("Eigenvectors:")

print(eigen\_result$vectors)

**Explanation:**

* **Matrix Definition**: In the example, A is a 2x2 matrix, but this can be any square matrix.
* **Eigen Function**: The eigen() function returns a list containing the eigenvalues and eigenvectors of the matrix.
  + eigen\_result$values gives the eigenvalues.
  + eigen\_result$vectors gives the eigenvectors.

****

**Key Points:**

* **Eigenvalues** represent the factors by which the eigenvectors are scaled when the matrix is applied to them.
* **Eigenvectors** are the directions along which the matrix transformation acts by stretching or compressing.