# ASS-1-DATA-501

Amish Verma 300598733

2024-07-26

## Introduction

The function developed in this assignment calculates the test statistic W,performs necessary input validation, and optionally generates a QQ plot to visually assess the normality of the data.

## Methodology

The Shapiro-Wilk test statistic (W) is calculated using the following formula:

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

where:

- $(x_{(i)})$ is the $(i)$-th order statistic (i.e., the $(i)$th smallest value in the sample).
- $(\bar{x})$ is the sample mean.

The coefficients $(a_i)$ are calculated using the vector $m$ and the covariance matrix $(V)$ of the order statistics.

## Implementaion

The core function, shapiro_Wilk_test, performs the following tasks:

- Input Validation: Ensures the input data is numeric, contains no NA or infinite values, and has at least three values. It also checks that the optional qqplot argument is logical (TRUE or FALSE).
- Weight Calculation: Computes the weights $a_i$ using the vector m and the covariance matrix V.
- Test Statistic Calculation: Calculates the Shapiro-Wilk test statistic (W).
- Q-Q Plot Generation (Optional): Generates a Q-Q plot if the qqplot argument is set to TRUE.

```
calculate_a <- function(n) {

  # Compute the expected values of the order statistics
  m <- qnorm((1:n - 0.375) / (n + 0.25))
```

```r
  # Construct the covariance matrix
  V <- matrix(0, n, n)
  for (i in 1:n) {
    for (j in 1:n) {
      V[i, j] <- min(i, j) - i * j / (n + 1)
    }
  }

  # Calculate the inverse of the covariance matrix
  V_inv <- solve(V)

  # Calculate the coefficients
  a <- as.numeric((m %*% V_inv) / sqrt(sum((m %*% V_inv)^2)))

  return(a)
}

# Function to calculate Shapiro-Wilk test statistic
shapiro_wilk_test <- function(data, qqplot = FALSE) {
  # Input validation
  if (!is.numeric(data)) {
    stop("\n Data must be numeric") # Added \n so that test passes! Weird stuff to catch
  }
  if (any(is.na(data))) {
    stop("\n Data contains NA values")
  }
  if (any(is.infinite(data))) {
    stop("\n Data contains infinite values")
  }
  if (length(data) < 3) {
    stop("\n Data must contain at least 3 values")
  }

  n <- length(data)
  if (n > 5000) {
    stop("Sample size must be between 3 and 5000")
  }

  if (!is.logical(qqplot)) {
    stop("Check the optional argument. By default, it's FALSE and can be set to
         TRUE")
  }
  # Calculate weights

  a <- calculate_a(n)

  # Order the data
  sorted_data <- sort(data)

  # Calculate the mean
  x_bar <- mean(data)

  # Calculate W
```

```
  W <- (sum(a * sorted_data)^2) / sum((sorted_data - x_bar)^2)
  #cat("The value of W:",W)

  if (qqplot) {
    qqnorm(data)
    qqline(data, col = 2)
  }
  return(W)
}
```

# Testing part

```
library(testthat)

usethis::use_testthat()
```

```
## v Setting active project to 'D:/VicUni/data501/Assignmnet-1/DATA501Assign1'
## v Leaving 'tests/testthat.R' unchanged
## * Call `use_test()` to initialize a basic test file and open it for editing.
```

## Invalid Input testing

- Non-numeric data: Ensuring the function raises an error when the input data is not numeric.
- Data with NA values: Ensuring the function raises an error when the input data contains NA values.
- Data with infinite values: Ensuring the function raises an error when the input data contains infinite values.
- Data with fewer than 3 values: Ensuring the function raises an error when the input data has fewer than 3 values.
- Incorrect qqplot argument: Ensuring the function raises an error when the qqplot argument is not logical (TRUE or FALSE).
- expect error: expect_error is used to test whether a function throws an error when provided with incorrect or invalid inputs. It ensures that the function handles such scenarios by raising appropriate error messages.

```
context("Testing the function")
## Invalid inputs testing
test_that("The function give valid erros for the inputs provided",
{
expect_error(shapiro_wilk_test(c("2","3","4","5")), "\n Data must be numeric")
expect_error(shapiro_wilk_test(c(2,3,4,NA)), "\n Data contains NA values")
expect_error(shapiro_wilk_test(c(2,3,4,Inf)), "\n Data contains infinite values")
expect_error(shapiro_wilk_test(c(2,3)), "\n Data must contain at least 3 values")
expect_error(shapiro_wilk_test(c(2,3,4,5),"ad"), "Check the optional argument. By default, it's FALSE ar
         TRUE")
})
```

```
## Test passed
```

### Valid Inputs Testing:

- Normal data: Testing the function with a sample from a normal distribution.
- Uniform data: Testing the function with a sample from a uniform distribution.
- Custom data: Testing the function with a custom numeric vector, both with and without the optional qqplot argument.
-

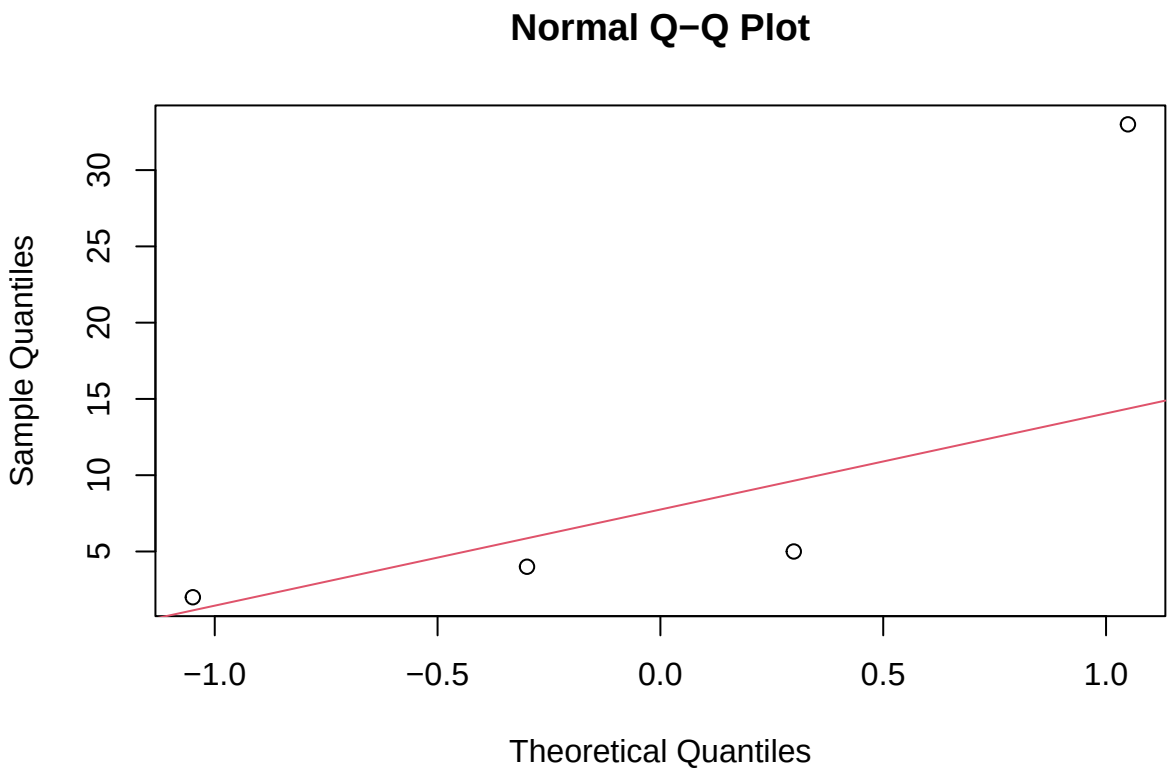
```
# Valid inputs testing
test_that("shapiro_wilk_test handles normal data correctly", {
  test_data <- rnorm(100)
  expect_silent(result <- shapiro_wilk_test(test_data))
  expect_is(result, "numeric")
})
```

```
## Test passed
```

```
test_that("shapiro_wilk_test handles uniform data correctly", {
  test_data <- runif(100)
  expect_silent(result <- shapiro_wilk_test(test_data))
  expect_is(result, "numeric")
})
```

```
## Test passed
```

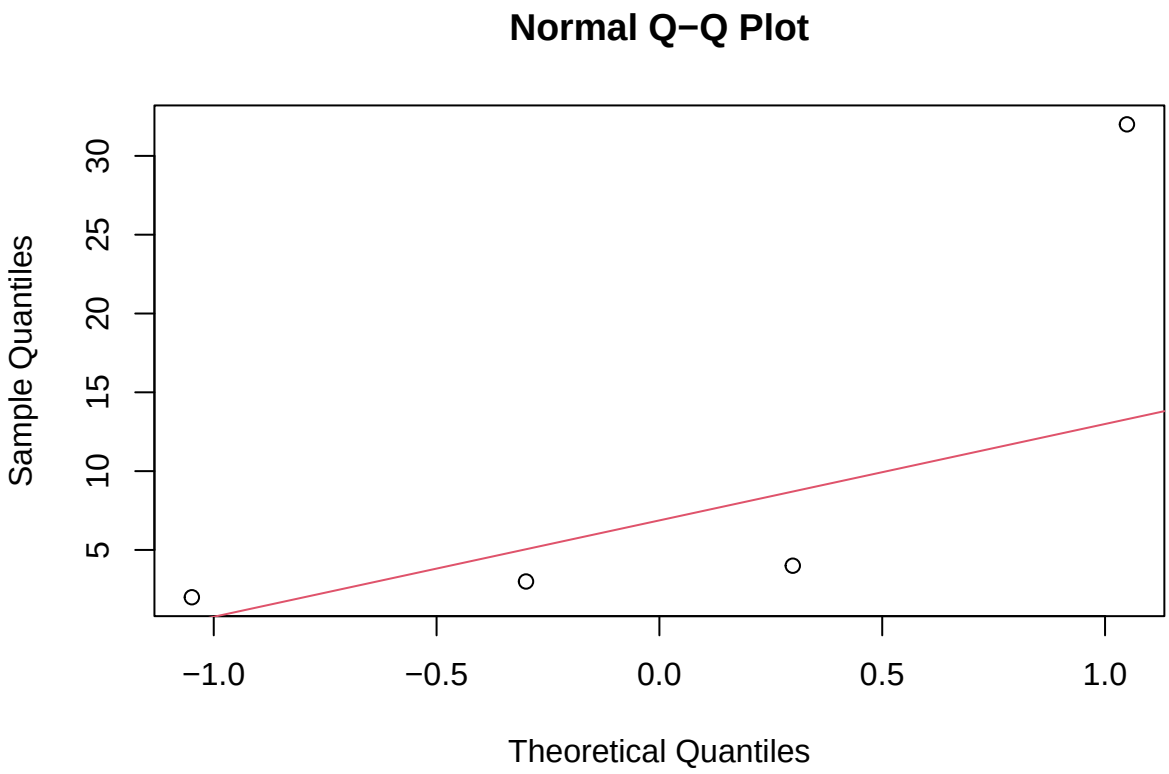
```
test_that("shapiro_wilk_test works fine with custom data",
{
expect_silent(shapiro_wilk_test(c(2,33,4,5),TRUE)) # With optional argument
expect_silent(shapiro_wilk_test(c(2,33,4,5))) # Without optional argument
})
```



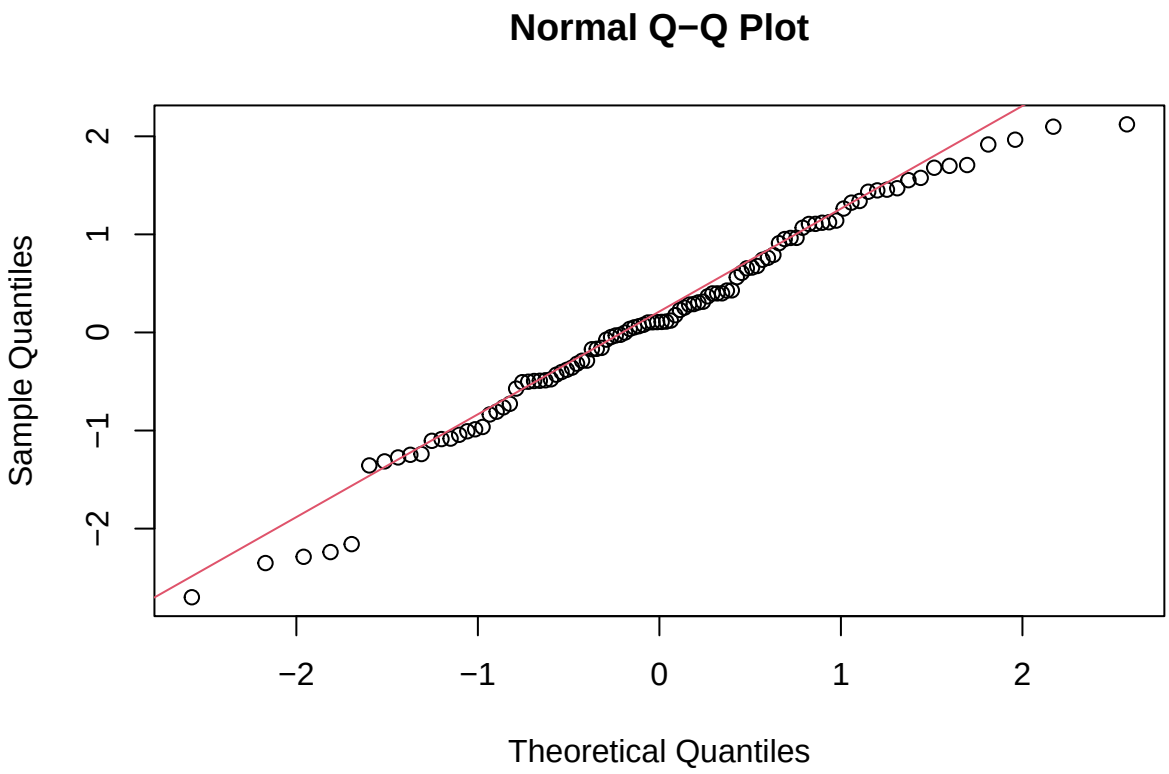

```
## Test passed
```

```
test_file("Ass-1-1.R")
```

```
## [ FAIL 0 | WARN 0 | SKIP 0 | PASS 0 ]
```




```r
# Example test results
test_data <- rnorm(100)
W <- shapiro_wilk_test(test_data, qqplot = TRUE)
```




```
print(W)
```

```
## [1] 0.08826313
```

## Github

The code for this project, including the implementation and tests, is available at this link.

Please click on the “link” word to go to the Github page.

### GIT BASH COMMANDS USE

- git init
- git status
- git add
- git commit -m “Message”
- git push origin main