Assignment 1

glucksadam 300619334

July 2024

Packages

```r
library(knitr)
library(MASS)
library(testthat)
```

Project Files

Assignment 1 project files are available at this link.

Commit files are available at this link

Shapiro-Wilk Test Statistic

Function

This function follows the method:

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

where $x_{(i)}$ is an ordered statistic, $\bar{x}$ is the sample mean, and a_i is given by:

$$(a_1, ..., a_n) = \frac{m^T V^{-1}}{C}$$

$$C = (m^T V^{-1} V^{-1} m)^{1/2}$$

sources:

Wikipedia, other sources include:

Shapiro-Wilk Expanded Test, Charles Zaiontz 2014

An Extension of Shapiro and Wilk's W Test for Normality to Large Samples, J.P. Royston

`SW_test()` Function

```r
SW_test <- function(data, plot_qq = FALSE) {
  # Input validation
  if (!is.numeric(data)) {
    stop("Input data must be numeric.")
```

```r
}

if (length(data) < 3) {
  stop("Input data must have at least 3 values.")
}

if (any(is.na(data))) {
  stop("Input data contains NA values.")
}

if (any(is.infinite(data))) {
  stop("Input data contains infinite values.")
}

#Obtain sorted data, x_{(i)}
n <- length(data)
sorted_data <- sort(data)

# compute expected values for a normal dist
m <- qnorm((1:n - .375) / (n + .25))

# compute a covariance matrix
V <- matrix(0, n, n)
for (i in 1:n) {
  for (j in 1:n) {
    V[i, j] <- min(i, j) / n - (i * j) / n^2
  }
}

# Compute inverse of V
V_inv <- MASS::ginv(V)

# Compute C
#C <- sqrt(t(m) %*% V_inv %*% V_inv %*% m)

# Compute weights a
a <- t(m) %*% V_inv #V_inv %*% m
C <- sqrt(sum(a^2))
a <- a / C

# Compute the mean of the data
x_bar <- mean(data)

# Compute the Shapiro-Wilk test statistic W
W <- (sum(a * sorted_data)^2) / sum((sorted_data - mean(sorted_data))^2)

# Plot QQ-plot if required
if (plot_qq) {
  qqnorm(data)
  qqline(data, col = "blue", lwd = 2)
}

# Return W statistic, and QQ plot
```

```r
 #return(list("Shapiro-Wilk W-statistic:" = W))
  return(list("W" = W))

}
```

Test Data

Create Test File

```r
usethis::use_testthat()
```

Testing in RMarkdown

```r
### datasets

## generated data
data1 <- rnorm(300)
data2 <- rnorm(1000)
data3 <- c(rnorm(50), NA)
data4 <- c(rnorm(50), Inf)
data5 <- c("a", "b", "c")
data6 <- c(1, 2)
data7 <- rbinom(300, 1000, prob = 0.7)

## imported data
milk <- read.csv("Data/Milk.csv") #available on assignment repo
```

```r
context("Testing SW_test context")

## invalid inputs
test_that("function SW_test gives helpful errors",
          {
            expect_error(SW_test(data3), "Input data contains NA values.")

            expect_error(SW_test(data4), "Input data contains infinite values.")

            expect_error(SW_test(data5), "Input data must be numeric.")

            expect_error(SW_test(data6), "Input data must have at least 3 values.")

            })
```

```
## Test passed
```

```r
## valid inputs
test_that("function SW_test works with numeric dataframes", {

  expect_silent(SW_test(data1))

  expect_silent(SW_test(milk$Cost))

  expect_type(SW_test(data1), "list")

```

```r
  expect_true(typeof(SW_test(data1)[[1]])=="double", TRUE)
})
```

```
## Test passed
```

Testing using R file

note: the "R/test.R" file is a duplicate of the code above

```r
test_file("R/test.R")
```

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

Sample Output

```r
#basic output
SW_test(rnorm(100)) #sample size of 100
```

```
## $W
## [1] 0.08340613
```

```r
SW_test(milk$Cost) #sample size of 12
```

```
## $W
## [1] 0.1288595
```

```r
#W from a non-normal distribution
SW_test(data7)
```

```
## $W
## [1] 0.0001433813
```

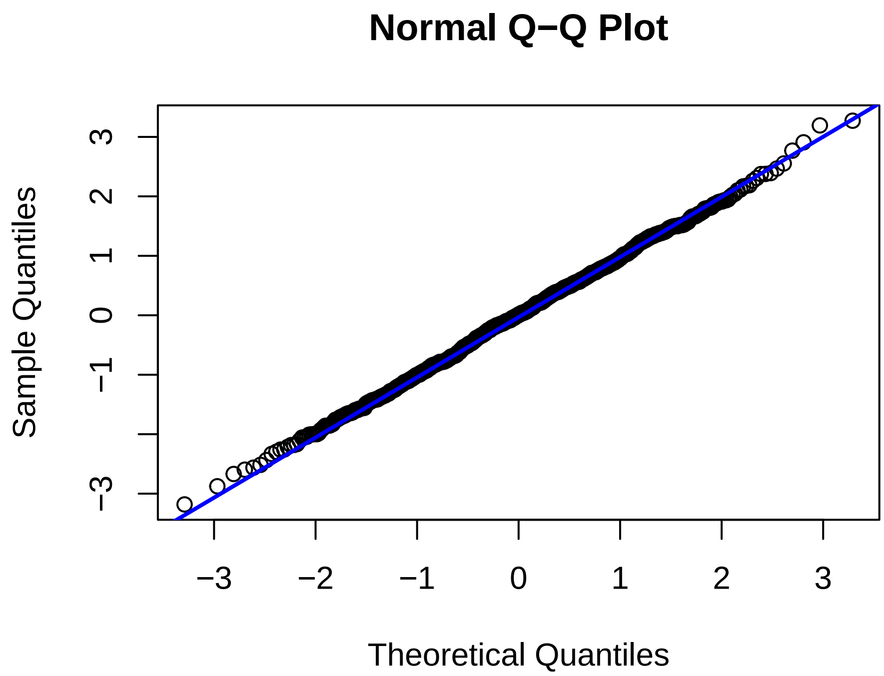
```r
#result is much smaller

## adding numbers to set
nums <- c(milk$Cost,runif(10, 2,6))
SW_test(nums)
```

```
## $W
## [1] 0.142839
```

A sample of 100 random normal has a W of 0.1052131. Datasets with smaller numbers, for example `milk` (n=12), and `nums` (n=5) have larger values for W. Adding more samples to `nums` brings down the value of W. It appears that low values of n are unstable.

```r
SW_test(data2, plot_qq = T)
```

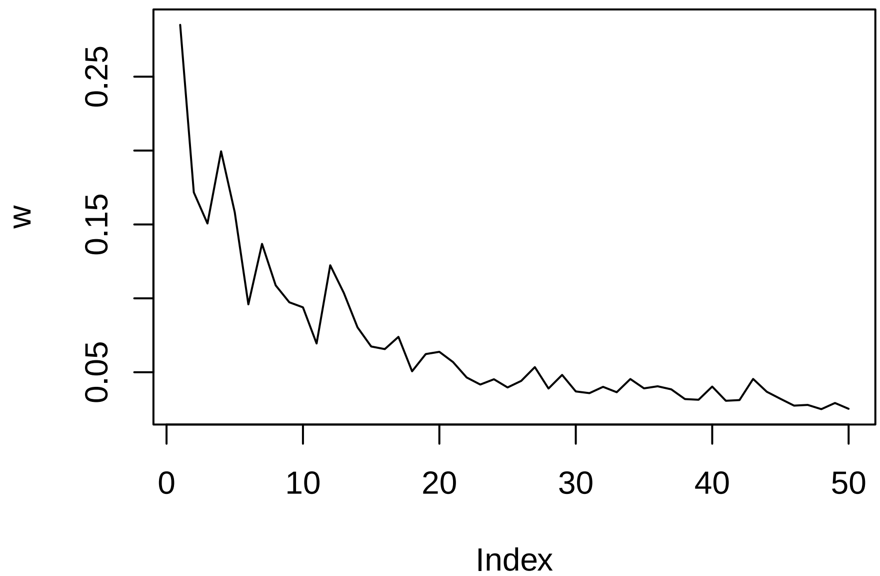



```
## $W
## [1] 0.0186
```

Increasing n-samples

```r
#what happens to W as the size of the data increases?
 w <- rep(NA,50)
for (i in 1:50) {
  data <- rnorm(10*i)
  dub <- SW_test(data)[[1]]
  w[i] <- dub[[1]]
}

plot(w, type="n")
lines(w)
```



As n increases, we get a smaller value for w, converging aroung 0.03.

compare QQ plots with increasing values of n

```
par(mfrow = c(2,3))
SW_test(rnorm(10), plot_qq = T)
```

```
## $W
## [1] 0.3335751
```

```
SW_test(rnorm(25), plot_qq = T)
```

```
## $W
## [1] 0.228222
```

```
SW_test(rnorm(50), plot_qq = T)
```

```
## $W
## [1] 0.1523558
```

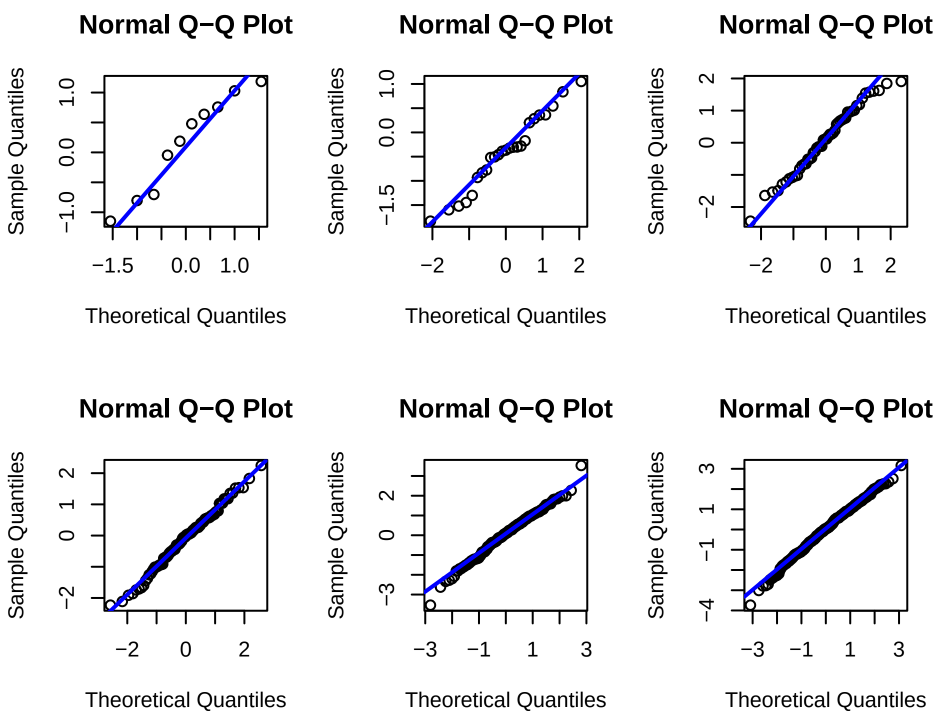
```
SW_test(rnorm(100), plot_qq = T)
```

```
## $W
## [1] 0.08472619
```

```
SW_test(rnorm(200), plot_qq = T)
```

```
## $W
## [1] 0.07147121
```

```
SW_test(rnorm(500), plot_qq = T)
```




```
## $W
## [1] 0.03373174
```

As n samples of random normal increase, the quantities converge onto the QQ line.

Final Thoughts

The function `SW_test()` derived from Wikipedia and online sources output a test statistic, W, as well as a QQ-plot — when default parameter `FALSE` is changed to `TRUE`. These test statistics are sensitive to the number of samples. Here we see that samples derived from a random normal distribution have higher values than those derived from a binomial distribution.

8 tests are run on the function, using two methods: code available in this document above, as well as a test file. Both use the same code. 4 Tests check error, and 4 tests check that the output is desired. All tests are passing, indicating that the correct errors are displayed, and the output is congruent with the functionality of the function.

Note: Assignment 1 project files are available at this link.

End of Assignment 1, DATA501

References

Wikipedia other sources include:

Shapiro-Wilk Expanded Test, Charles Zaiontz 2014

An Extension of Shapiro and Wilk's W Test for Normality to Large Samples, J.P. Royston