Lab 7 Solutions

01 October 2024

Name: Kaushik Raj V Nadar (208160499)

Problem 1

P(1) Consider the Webster, Gunst, and Mason (WGM) data set given in Table 9.4 of Montgomery Book.

Load Data

```
library(readxl)
dataset <- read_excel("Webster_Data.xlsx")

WGM_data <- as.data.frame(dataset)
```

(a) Normalise the data set by subtracting the associated mean and dividing by the square root of (n - 1) × the sample variance of each variables.

```
normalize_data <- function(data) {
  n <- nrow(data)
  norm_data <- scale(data, center = TRUE, scale = sqrt((n - 1) * apply(data, 2, var)))
  return(norm_data)
}

# Assuming WGM_data is the dataset from Table 9.4 in Montgomery
normalized_WGM_data <- normalize_data(WGM_data)
normalized_WGM_data
```

```
##                 y          x1          x2         x3         x4           x5
##  [1,] -0.01198334  0.48424800 -0.09901475 -0.1147079 -0.1499940  0.047460819
##  [2,] -0.03679554  0.48424800 -0.09901475 -0.1147079 -0.2219911 -0.055204427
##  [3,]  0.45668130  0.48424800 -0.09901475 -0.1147079 -0.2219911  0.440886033
##  [4,] -0.15808938 -0.22011273 -0.19802951  0.4970674 -0.1499940 -0.686433326
##  [5,] -0.13936493 -0.22011273 -0.19802951  0.4970674 -0.1499940 -0.099168133
##  [6,]  0.56755217 -0.22011273 -0.19802951  0.4970674 -0.1499940  0.003497113
##  [7,] -0.38536544 -0.04402255  0.49507377 -0.1911798 -0.1499940  0.410910779
##  [8,] -0.07590473 -0.04402255  0.49507377 -0.1911798 -0.1499940 -0.030724636
##  [9,]  0.26048966 -0.04402255  0.49507377 -0.1911798 -0.1499940  0.257037806
## [10,] -0.42382896 -0.22011273 -0.19802951 -0.1911798  0.4979801 -0.287013061
## [11,] -0.12728167 -0.22011273 -0.19802951 -0.1911798  0.4979801 -0.023480616
## [12,]  0.07389085 -0.22011273 -0.19802951 -0.1911798  0.4979801  0.022231647
##                x6
##  [1,] -0.122963251
##  [2,] -0.032710094
```

```
##  [3,] -0.008678188
##  [4,]  0.064485613
##  [5,] -0.061014338
##  [6,]  0.733106633
##  [7,] -0.532039689
##  [8,] -0.099465387
##  [9,]  0.197996200
## [10,] -0.283175955
## [11,] -0.016154781
## [12,]  0.160613236
## attr(,"scaled:center")
##         y        x1        x2        x3        x4        x5        x6
## 10.135917  2.500000  2.000000  2.500000  3.083333  0.351000  0.131250
## attr(,"scaled:scale")
##         y        x1        x2        x3        x4        x5        x6
## 10.841441 11.357817 10.099505 13.076697 13.889444  4.003302  1.872511
```

(b) Find the VIFs

```
calculate_vif_manual <- function(X) {
  # Compute the X^T X matrix
  XtX <- t(X) %*% X

  # Compute the inverse of (X^T X)
  XtX_inv <- solve(XtX)

  # The diagonal elements of the inverse of X^T X give the VIF
  vifs <- diag(XtX_inv)

  return(vifs)
}


X <- as.matrix(normalized_WGM_data[,-1])  # Exclude the response variable

vifs <- calculate_vif_manual(X)
vifs
```

```
##         x1         x2         x3         x4         x5         x6
## 182.051943 161.361942 266.263648 297.714658   1.919992   1.455265
```

(c) Find the condition number.

```
XtX <- t(X) %*% X
eigen_values <- eigen(XtX)$values
condition_number <- max(eigen_values) / min(eigen_values)
condition_number
```

```
## [1] 2195.908
```

(d) Find the eigen vectors and verify if it matches with the eigen vectors given in Table 9.6 of Montgomery Book.

```
eigen_vectors <- eigen(XtX)$vectors
eigen_vectors
```

```
##            [,1]        [,2]        [,3]        [,4]       [,5]        [,6]
## [1,] -0.3907189  0.33968212  0.67980398 -0.07990398  0.2510370 0.447679719
## [2,] -0.4556030  0.05392140 -0.70012501 -0.05768633  0.3444655 0.421140280
## [3,]  0.4826405  0.45332584 -0.16077736 -0.19102517 -0.4536372 0.541689124
## [4,]  0.1876590 -0.73546592  0.13587323  0.27645223 -0.0152087 0.573371872
## [5,] -0.4977330  0.09713874 -0.03185053  0.56356440 -0.6512834 0.006052127
## [6,]  0.3519499  0.35476494 -0.04864335  0.74817535  0.4337463 0.002166594
```

Problem 2

P(2) Find variance Decomposition Proportions for the WGM data and verify if it matches with the results reported in Table 9.7 of Montgomery Book.

```
calculate_vdp <- function(eigen_vectors, eigen_values, vifs) {
  p <- ncol(eigen_vectors)  # Number of predictors
  vdp_matrix <- matrix(0, nrow = p, ncol = p)  # Create a matrix to store VDPs

  # Loop over each predictor and each eigenvalue
  for (j in 1:p) {
    for (i in 1:p) {
      t_ji <- eigen_vectors[j, i]  # Element of the eigenvector matrix
      lam_i <- eigen_values[i]  # Square root of the eigenvalue
      vdp_matrix[i, j] <- (t_ji^2 / lam_i) / vifs[j]  # Calculate the VDP
    }
  }

  return(vdp_matrix)
}

# Compute the Variance Decomposition Proportions
vdp <- calculate_vdp(eigen_vectors, eigen_values, vifs)

# Display the VDP matrix
vdp
```

```
##              [,1]         [,2]         [,3]         [,4]         [,5]
## [1,] 0.0003452582 5.296422e-04 0.0003602021 4.870225e-05 0.0531256120
## [2,] 0.0004099188 1.165384e-05 0.0004991796 1.175094e-03 0.0031785813
## [3,] 0.0027529897 3.294447e-03 0.0001052859 6.725120e-05 0.0005730155
## [4,] 0.0000441702 2.597363e-05 0.0001726065 3.233165e-04 0.2083414193
## [5,] 0.0011242983 2.388316e-03 0.0025101916 2.523397e-06 0.7175332700
## [6,] 0.9953233648 9.937500e-01 0.9963525343 9.983831e-01 0.0172481021
##             [,6]
## [1,] 0.035045332
## [2,] 0.055935444
## [3,] 0.001763345
## [4,] 0.484454029
## [5,] 0.419885511
## [6,] 0.002916338
```