Assignment 1

Nguyen Xuan Binh

Homework Problem 1: Functions

In this exercise do not use the built-in functions cov, cor, cov2cor or any additional R packages.

```
setwd(getwd())
```

a)

Create an R function that takes a data matrix $\mathbf{X} \in R^{n \times p}, n > p$, as an argument and returns the unbiased estimator of the covariance matrix.

Model reference

```
A <- matrix(rnorm(30), ncol = 3)

print(cov(A))
```

```
##            [,1]       [,2]      [,3]
## [1,] 0.54194770 0.03532509 0.1517323
## [2,] 0.03532509 1.96516319 0.3636497
## [3,] 0.15173228 0.36364975 1.3417727
```

The function for unbiased covariance without using the R library

```
unbiased_cov <- function(matrix){
  average <- apply(matrix, 2, mean)
  centered <-  sweep(matrix, 2, average, "-")
  unbiased_covariance <- t(centered) %*% (centered) / (nrow(matrix) - 1)
  return(unbiased_covariance)
}

print(unbiased_cov(A))
```

```
##            [,1]       [,2]      [,3]
## [1,] 0.54194770 0.03532509 0.1517323
## [2,] 0.03532509 1.96516319 0.3636497
## [3,] 0.15173228 0.36364975 1.3417727
```

b)

Create an R function that takes a full-rank covariance matrix $\mathbf{A} \in R^{p \times p}$ as an argument and returns the square root of the inverse matrix such that $\mathbf{A}^{\frac{-1}{2}}\mathbf{A}^{\frac{-1}{2}} = \mathbf{A}$

A full-rank covariance matrix is a square matrix that has linearly independent columns and rows, which means it has the maximum possible rank (number of non-zero eigenvalues) which is equal to the number of rows (or columns) of the matrix. In other words, it is a matrix that has no zero eigenvalues, which indicates that all its columns are linearly independent, thus all its rows are also linearly independent. Additionally, it is a covariance matrix, which means it is symmetric.

Model reference

```r
# First install and load expm package
# install.packages("expm")
library(expm)
```

```
## Loading required package: Matrix

##
## Attaching package: 'expm'

## The following object is masked from 'package:Matrix':
##
##     expm
```

```r
# full-rank covariance matrix
A <- matrix(c(0.99,-0.17,0.30,-0.17,0.61,0.33,0.30,0.33,1.33), nrow = 3, ncol = 3)

# Find the inverse of the matrix
inverse_A <- solve(A)

# Find the square root of the inverse of the matrix
sqrt_inverse_A <- sqrtm(inverse_A)

sqrt_inverse_A
```

```
##            [,1]       [,2]       [,3]
## [1,]  1.0768396  0.2092707 -0.1800484
## [2,]  0.2092707  1.4301472 -0.2629218
## [3,] -0.1800484 -0.2629218  0.9539935
```

The square root of a matrix can be found by using the eigenvalue decomposition of the matrix. The eigenvalue decomposition is a way of decomposing a matrix into a canonical form, where the matrix is represented as a product of a matrix of eigenvectors and a diagonal matrix of eigenvalues.

The function for the square root of the inverse of a matrix without using the R library

```r
# Find the square root of the inverse of the full-rank covariance matrix
sqrt_inverse <- function(matrix){
```

```r
  inverse_matrix <- solve(matrix)

  # Find the eigenvalues and eigenvectors of the matrix
  result <- eigen(inverse_matrix)
  eigenvalues <- result$values
  eigenvectors <- result$vectors

  # Obtain the square roots ô the eigenvalues
  sqrt_eigenvalues <- sqrt(eigenvalues)

  # Form a diagonal matrix with the square root of eigenvalues
  sqrt_eigenvalues_matrix <- diag(sqrt_eigenvalues)

  # Multiply the matrix of eigenvectors with the diagonal matrix of the square roots of the eigenvalues
  sqrt_inverse_matrix <- eigenvectors %*% sqrt_eigenvalues_matrix %*% t(eigenvectors)

  return(sqrt_inverse_matrix)
}

print(sqrt_inverse(A))
```

```
##            [,1]       [,2]       [,3]
## [1,]  1.0768396  0.2092707 -0.1800484
## [2,]  0.2092707  1.4301472 -0.2629218
## [3,] -0.1800484 -0.2629218  0.9539935
```

This method will only work if the matrix is symmetric and its eigenvalues are real numbers, because the square root of real matrix is only defined for normal matrices. Thankfully, symmetry is guaranteed as it is a covariance matrix, and its eigenvalues are real numbers because it is a full-rank matrix.

c)

Create an R function that takes a full-rank covariance matrix **A** as an argument and returns the corresponding correlation matrix

Model reference

```r
correlation_matrix <- cov2cor(A)

print(correlation_matrix)
```

```
##            [,1]       [,2]      [,3]
## [1,]  1.0000000 -0.2187592 0.2614435
## [2,] -0.2187592  1.0000000 0.3663728
## [3,]  0.2614435  0.3663728 1.0000000
```

The function for correlation conversion from full-rank covariance matrix without using the R library

```r
# Find the correlation matrix from the full-rank covariance matrix
correlation <- function(cov_matrix){

  # Compute the standard deviations of the variables
  sqrt_diag <- diag(1/sqrt(diag(cov_matrix)))

  # Divide the covariance matrix by the outer product of the standard deviations
  cor_matrix <- sqrt_diag %*% cov_matrix %*% sqrt_diag

  return(cor_matrix)
}

print(correlation(A))
```

```
##            [,1]       [,2]      [,3]
## [1,]  1.0000000 -0.2187592 0.2614435
## [2,] -0.2187592  1.0000000 0.3663728
## [3,]  0.2614435  0.3663728 1.0000000
```

Read explanation from here: https://math.stackexchange.com/questions/186959/correlation-matrix-from-covariance-matrix