HW2

Minh Luc, Devin Pham, Kyle Moore

Friday of Week 2, 04/08/2022

Contents

We all contributed equally for this homework.

Question 0

Member 1:

- Name: Minh Luc
- Student ID: A17209607

Member 2:

- Name: Kyle Moore
- Student ID: A14271413

Member 3:

- Name: Devin Pham
- Student ID: A17198936

Question 1

- **(a)**

```
# install the packages if needed by using
# install.packages("...")
library(tidyr)
library(readr)
```

```r
library(tidytuesdayR)
urlRemote <- 'https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/'
pathGithub <- 'data/2020/2020-07-28/'
fileName <- 'penguins.csv'
penguins <- paste0(urlRemote, pathGithub, fileName) %>% read.csv(header = TRUE)
dfr <- drop_na(as.data.frame(penguins))
head(dfr)
```

```
##   species    island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
## 1  Adelie Torgersen           39.1          18.7               181        3750
## 2  Adelie Torgersen           39.5          17.4               186        3800
## 3  Adelie Torgersen           40.3          18.0               195        3250
## 4  Adelie Torgersen           36.7          19.3               193        3450
## 5  Adelie Torgersen           39.3          20.6               190        3650
## 6  Adelie Torgersen           38.9          17.8               181        3625
##      sex year
## 1   male 2007
## 2 female 2007
## 3 female 2007
## 4 female 2007
## 5   male 2007
## 6 female 2007
```

- **(b)**

```r
nrow(dfr) # number of rows
```

```
## [1] 333
```

```r
ncol(dfr) # number of columns
```

```
## [1] 8
```

There are 333 rows and 8 columns in the dataframe(dfr).

Question 2

- **Find the mean vector, covariance matrix and correlation matrix of X:**

```r
X <- dfr[,3:6] # assign all rows, but only columns 3-6 to X

colMeans(X) # mean vector containing the means for each column in X
```

```
##    bill_length_mm     bill_depth_mm flipper_length_mm       body_mass_g
##          43.99279          17.16486         200.96697        4207.05706
```

```r
cov(X) # compute the covariance matrix of X
```

```
##                   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
## bill_length_mm         29.906333     -2.462091          50.05819   2595.6233
## bill_depth_mm          -2.462091      3.877888         -15.94725   -748.4561
## flipper_length_mm      50.058195    -15.947248         196.44168   9852.1916
## body_mass_g          2595.623304   -748.456122        9852.19165 648372.4877
```

```r
cor(X) # compute the correlation matrix of X
```

```
##                   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
## bill_length_mm         1.0000000    -0.2286256         0.6530956   0.5894511
## bill_depth_mm         -0.2286256     1.0000000        -0.5777917  -0.4720157
## flipper_length_mm      0.6530956    -0.5777917         1.0000000   0.8729789
## body_mass_g            0.5894511    -0.4720157         0.8729789   1.0000000
```

- The variance-covariance matrix is a symmetric matrix that represents how the variables are correlated: positively correlated, negatively correlated, or uncorrelated. The diagonal represents the variance of each variable itself. It is symmetric due to the fact that $\mathrm{Cov}(X, Y) = \mathrm{Cov}(Y, X)$.
- The correlation matrix is a standardized version of the variance-covariance matrix that represents the strength of the correlation between two variables where $-1 \leq correlation \leq 1$. Entries closer to 1 are more strongly positively correlated, those closer to -1 are strongly negatively correlated, and those near 0 are weakly or uncorrelated. The diagonals are all 1's because each variable is completely correlated with itself.

Question 3

- **(a)**:

```r
A <- cor(X) # assign the previous correlation matrix to A

2 * A # scalar multiplication of A by 2
```

```
##                   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
## bill_length_mm         2.0000000    -0.4572513          1.306191   1.1789022
## bill_depth_mm         -0.4572513     2.0000000         -1.155583  -0.9440313
## flipper_length_mm      1.3061913    -1.1555834          2.000000   1.7459578
## body_mass_g            1.1789022    -0.9440313          1.745958   2.0000000
```

- **(b)**:

```r
set.seed(3) # replace 1 by your own choice
B <- matrix(rnorm(16),nrow=4) # generates a random normal matrix and assigns it to B

C <- t(B) * B # assign B' * B to C

print(C)
```

```
##             [,1]         [,2]       [,3]        [,4]
## [1,]  0.92531590 -0.057271513 -0.3154259  0.82533946
## [2,] -0.05727151  0.000907452  0.1082558  0.28211422
## [3,] -0.31542594  0.108255761  0.5546996 -0.17199693
## [4,]  0.82533946  0.282114216 -0.1719969  0.09465248
```

  - C is symmetric.

- **(c)**:

```r
a <- 2
b <- 3

(a * A) + (b * B)
```

```
##                   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
## bill_length_mm        -0.8858002     0.1300972        -2.3503810  -0.9701732
## bill_depth_mm         -1.3348284     2.0903718         2.6465228  -0.1860742
```

```
## flipper_length_mm      2.0825559    -0.8993302         -0.2343448   2.2020949
## body_mass_g           -2.2774934     2.4057993         -1.6476979   1.0770307
```

- **(d)**:

```
eigen(A)
```

```
## eigen() decomposition
## $values
## [1] 2.7453557 0.7781172 0.3686425 0.1078846
##
## $vectors
##            [,1]        [,2]       [,3]       [,4]
## [1,]  0.4537532 -0.60019490  0.6424951  0.1451695
## [2,] -0.3990472 -0.79616951 -0.4258004 -0.1599044
## [3,]  0.5768250 -0.00578817 -0.2360952 -0.7819837
## [4,]  0.5496747 -0.07646366 -0.5917374  0.5846861
```

```
eigen(A)$values # eigen values
```

```
## [1] 2.7453557 0.7781172 0.3686425 0.1078846
```

```
eigen(A)$vectors  # eigen vectors
```

```
##            [,1]        [,2]       [,3]       [,4]
## [1,]  0.4537532 -0.60019490  0.6424951  0.1451695
## [2,] -0.3990472 -0.79616951 -0.4258004 -0.1599044
## [3,]  0.5768250 -0.00578817 -0.2360952 -0.7819837
## [4,]  0.5496747 -0.07646366 -0.5917374  0.5846861
```

```
library(expm)

sqrtm(A)  # square root the matrix A
```

```
##             [,1]        [,2]       [,3]       [,4]
## [1,]  0.91646783 -0.05222114  0.3073515  0.2507882
## [2,] -0.05222114  0.94148130 -0.2752149 -0.1874638
## [3,]  0.30735154 -0.27521488  0.7860240  0.4603890
## [4,]  0.25078816 -0.18746380  0.4603890  0.8306651
```

Question 4

- **Create a new dataset Y, where $Y_1 = 3X_1 + 2X_2, Y_2 = X_2 + X_3 + X_4$**

```
Y <- data.frame(matrix(ncol = 2, nrow = 333)) # create new empty dataframe
colnames(Y) <- c('Y_1', 'Y_2')  # label columns

Y['Y_1'] <- as.matrix((3* X[, 1]) + (2 * X[, 2])) # linear combination of X into new column
Y['Y_2'] <- as.matrix(X[, 2] + X[, 3] + X[, 4]) # linear combination of X into new column

head(Y) # show first rows of Y
```

```
##     Y_1    Y_2
## 1 154.7 3949.7
## 2 153.3 4003.4
## 3 156.9 3463.0
```

```
## 4 148.7 3662.3
## 5 159.1 3860.6
## 6 152.3 3823.8
```

```
colMeans(Y) # mean vector containing the means for each column in Y
```

```
##       Y_1       Y_2
##  166.3081 4425.1889
```

```
cov(Y) # compute the covariance matrix of Y
```

```
##           Y_1        Y_2
## Y_1  255.1235   6408.607
## Y_2 6408.6073 666748.384
```

Question 5

- **(a)**:

  We can find $\hat{a} = argminL(a)$ by setting $\frac{d}{da}L(a) = 0$

  First, we find $\frac{d}{da}L(a)$:

$$\begin{aligned}\frac{d}{da}L(a) &= \frac{d}{da}\frac{1}{n}\sum_{i=1}^{n}(X_i - a)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n} -2(X_i - a) \\ &= \frac{-2}{n}\sum_{i=1}^{n}(X_i - a)\end{aligned}$$

  Then we set the derivative equal to 0:

$$\begin{aligned}0 &= \frac{-2}{n}\sum_{i=1}^{n}(X_i - a) \\ 0 &= \sum_{i=1}^{n}(X_i - a)\end{aligned}$$

  We know that the sum of each sample component minus the sample mean equals 0, therefore $\hat{a} = \bar{X}$

- **(b)**:

  Plugging $a = \hat{a} = \bar{X}$ into $L(a) = \frac{1}{n}\sum_{i=1}^{n}(X_i - a)^2$ gives us:

$$L(\bar{X}) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

  We recognize this minimum value as the equation for the sample variance, therefore $L(\hat{a}) = S^2$

- **(c)**:

  $\hat{a}$ is an unbiased estimator of population mean because with many samples, the expectation of the sample mean would be equal to the population mean, or $E(\bar{X}) = \mu$.

This can be proven by:

$$
\begin{aligned}
E(\bar{X}) &= E\left(\frac{X_1 + X_2 + ... + X_n}{n}\right) \\
&= \frac{1}{n}E(X_1 + X_2 + ... + X_n) \\
&= \frac{1}{n}E(X_1) + E(X_2) + ... + E(X_n) \\
&= \frac{1}{n}(\mu + \mu + ... + \mu) \\
&= \frac{1}{n}n\mu \\
E(\bar{X}) &= \mu
\end{aligned}
$$

- **(d)**:

  $L(\hat{a})$ is a biased estimator of the population variance due to the fact that $0 = \sum_{i=1}^{n}(X_i - \bar{X})$, meaning we only need to solve $n-1$ of the deviations because the final one will always be set, or in other words, there are only $n-1$ degrees of freedom but we are dividing by n.
