HW3

Minh Luc, Devin Pham, Kyle Moore

Friday of Week 3, 04/15/2022

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

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

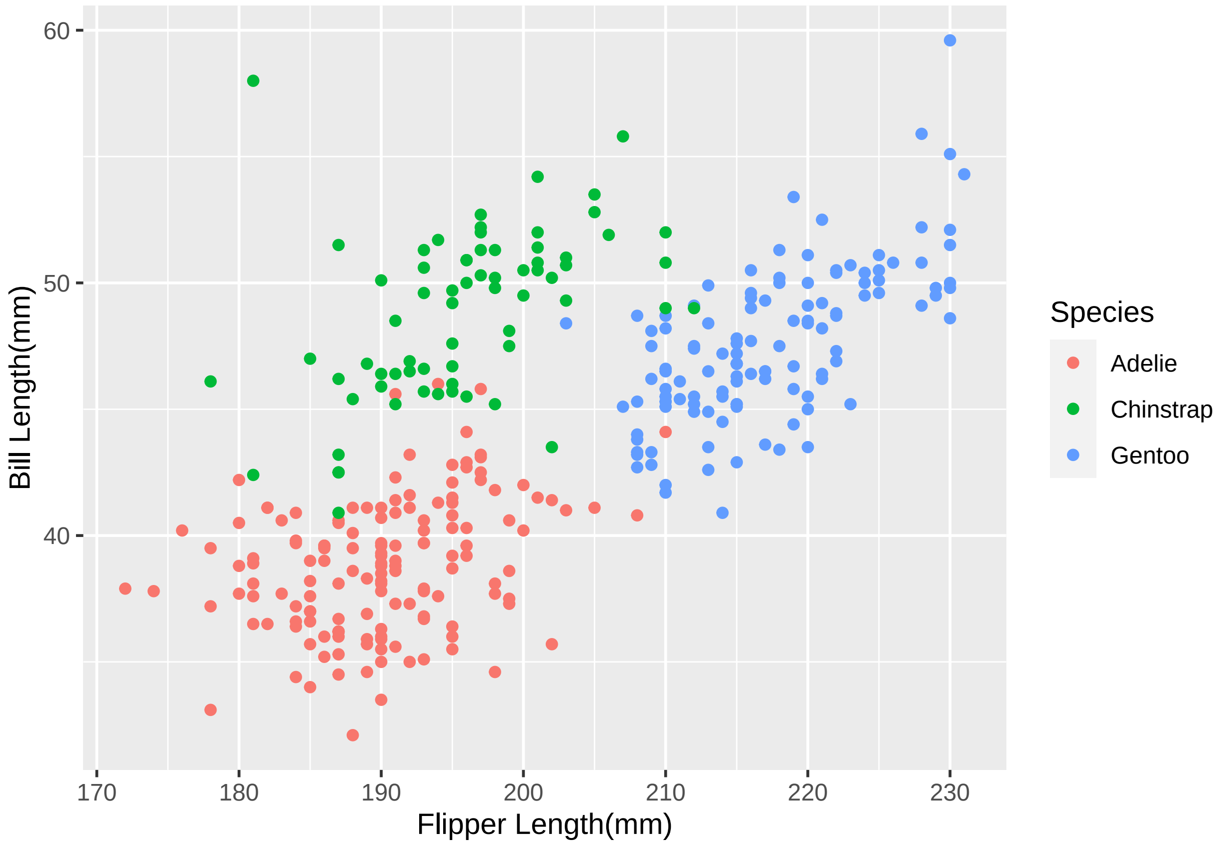
```
dfr <- drop_na(as.data.frame(penguins))
head(dfr)
```

```
##   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
```

Question 1

- **Answer:**

```
library(ggplot2)

scatter_plot <- ggplot(dfr, aes(x = flipper_length_mm, y = bill_length_mm, col = species)) + geom_point
scatter_plot + labs(x = 'Flipper Length(mm)', y = 'Bill Length(mm)', color = "Species")
```



Question 2

- **Answer:**

  We can see that the Chinstrap species doesn't follow the same linear pattern as the other two species, it is above them, meaning that species has a different relationship between its flipper and bill lengths. If species had no effect on the relationship then they would all follow the same pattern. We can also tell that the sizes of the different species are noticeably different when viewing the samples distributions and their relative location in the scatter plot.

Question 3

- **(a)** The sample mean, $\bar{X}$, is an unbiased estimator of the population mean, μ.

```
sample_mean <- mean(dfr$flipper_length_mm)  # sample mean of the flipper length column
print(sample_mean)
```

```
## [1] 200.967
```

- **(b)**

```
n <- nrow(dfr)  # number of rows, also number of observations
sample_sd <- sd(dfr$flipper_length_mm)  # sample standard deviation
```

```r
print(n)
```

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

```r
print(sample_sd)
```

```
## [1] 14.01577
```

```r
# two-tailed t-score with 95% confidence times the sample standard error
margin <- qt(0.05/2,df=n-1) * (sample_sd/sqrt(n))

print(margin)
```

```
## [1] -1.510876
```

```r
# find the lower and upper confidence intervals
lower_interval <- sample_mean - margin
upper_interval <- sample_mean + margin

print(lower_interval)
```

```
## [1] 202.4778
```

```r
print(upper_interval)
```

```
## [1] 199.4561
```

- **(c)**

```r
t.test(dfr$flipper_length_mm, mu = 35)  # t-test of flipper length with a mu of 35
```

```
##
##  One Sample t-test
##
## data:  dfr$flipper_length_mm
## t = 216.09, df = 332, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 35
## 95 percent confidence interval:
##  199.4561 202.4778
## sample estimates:
## mean of x
##   200.967
```

Here we got a p-value of 2.2E-16 which is ≈ 0 and our $|t|$ is large, therefore we reject the null hypothesis that $\mu_0 = 35$.

Question 4

- **(a)**

  By looking at the corresponding components of $\boldsymbol{\mu}, \boldsymbol{\Sigma}$, we see that X_1 is normally distributed with $\mu = 2$ and $Var(X_1) = 3$, or $X_1 \sim N(2, 3)$

- **(b)**

  By checking $\boldsymbol{\Sigma}$ for the covariances of X_1 and X_3, X_2 and X_3, we see that $\text{Cov}(X_1, X_3) = \text{Cov}(X_2, X_3) = 0$ which means they are independent.

- **(c)**

  Using $\boldsymbol{a}^T\boldsymbol{X} = a_1X_1 + a_2X_2 + a_3X_3$, we see that $a = (2, -3, 1)^T$

  Since $\boldsymbol{X} \sim N_3(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $\boldsymbol{a}^T\boldsymbol{X} \sim N(\boldsymbol{a}^T\boldsymbol{\mu}, \boldsymbol{a}^T\boldsymbol{\Sigma}\boldsymbol{a})$.

  ```
  sigma <- cbind(c(3,-1,0),c(-1,4,0),c(0,0,2))
  mu <- c(2,-1,4)
  a <- c(2,-3,1)

  print(mu)
  ```

  ```
  ## [1]  2 -1  4
  ```

  ```
  print(sigma)
  ```

  ```
  ##      [,1] [,2] [,3]
  ## [1,]    3   -1    0
  ## [2,]   -1    4    0
  ## [3,]    0    0    2
  ```

  ```
  print(a)
  ```

  ```
  ## [1]  2 -3  1
  ```

  ```
  t(a) %*% mu   # a^T * mu
  ```

  ```
  ##      [,1]
  ## [1,]   11
  ```

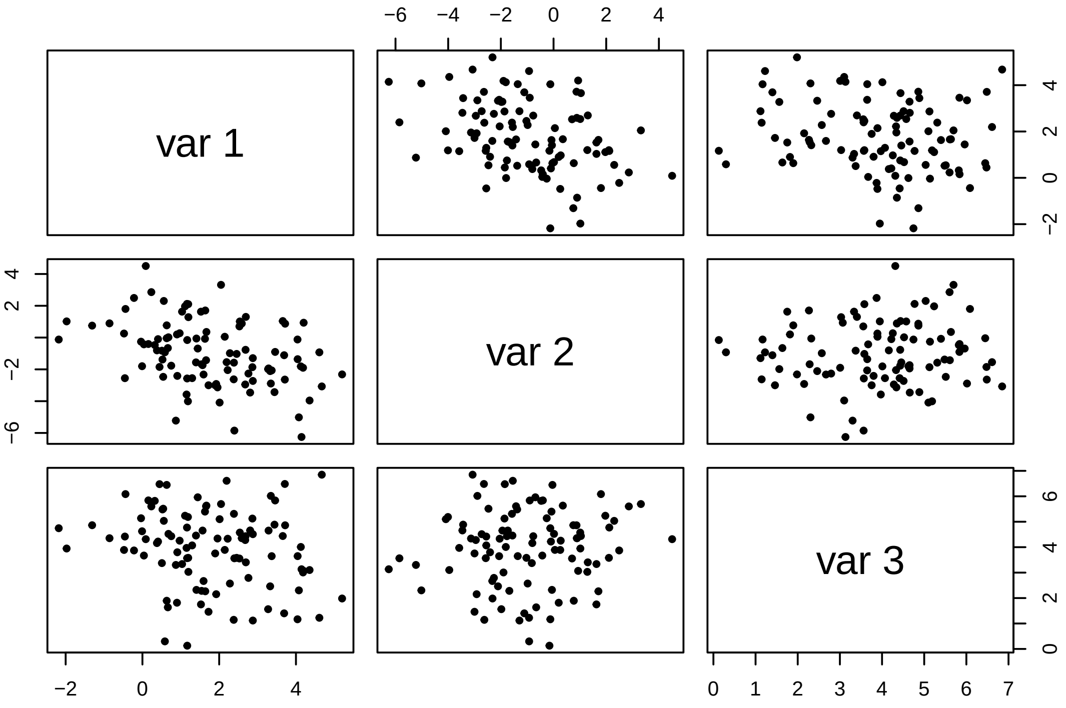
  ```
  t(a) %*% sigma %*% a  # a^T * Sigma * a
  ```

  ```
  ##      [,1]
  ## [1,]   62
  ```

  Using the results above: $\boldsymbol{a}^T\boldsymbol{X} \sim N(\boldsymbol{a}^T\boldsymbol{\mu}, \boldsymbol{a}^T\boldsymbol{\Sigma}\boldsymbol{a}) = \boldsymbol{a}^T\boldsymbol{X} \sim N(11, 62)$

- **(d)**

  ```
  library("mvtnorm")

  m = 9

  set.seed(m)

  random_vectors <- rmvnorm(n=100, mean=mu, sigma=sigma) # 100 random vectors

  pairs(random_vectors, pch=20) # plot pairwise scatter plots
  ```



- **(e)**

```
sample_mean_vector <- colMeans(random_vectors)  # find the sample mean vector
print(sample_mean_vector)
```

```
## [1]  1.7839686 -0.9883291  3.9379285
```

```
sample_cov_matrix <- cov(random_vectors)  # find the sample covariance matrix
print(sample_cov_matrix)
```

```
##            [,1]       [,2]       [,3]
## [1,]  2.3155819 -1.1562479 -0.4446808
## [2,] -1.1562479  4.0640247  0.2180145
## [3,] -0.4446808  0.2180145  2.2589710
```

```
T2 <- 100 * t(sample_mean_vector - mu) %*% solve(sample_cov_matrix) %*% (sample_mean_vector - mu)

print(T2)
```

```
##          [,1]
## [1,] 2.775337
```

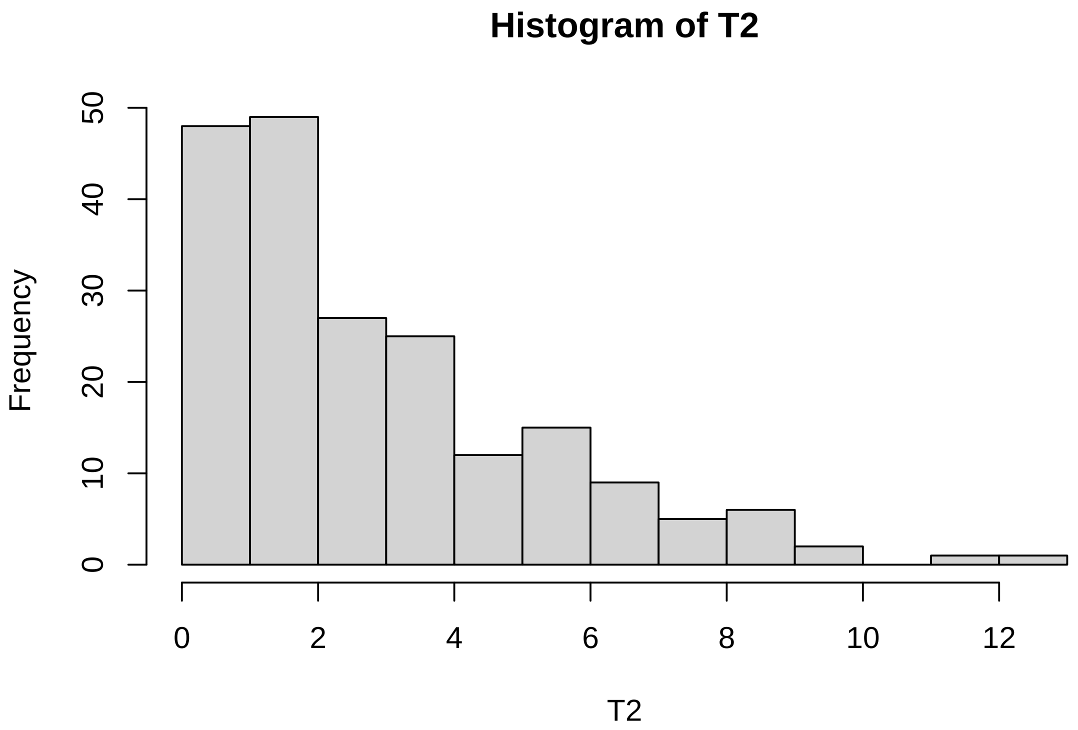
- **(f)**

```
B <- 200
T2 <- rep(0,B)

for (b in 1:B) {
  set.seed(m+b) # set new random seed
  random_vectors <- rmvnorm(n=100, mean=mu, sigma=sigma)  # generate random vectors
  sample_mean_vector <- colMeans(random_vectors)  # find sample mean vector
  sample_cov_matrix <- cov(random_vectors)  # find sample covariance matrix

  #find new T^2 and assign it to the index of the current iteration
  T2[b] <- 100 * t(sample_mean_vector - mu) %*% solve(sample_cov_matrix) %*% (sample_mean_vector
}

print(T2)
```

```
##   [1]  8.84775776  1.51022672  3.51997624  1.57235235  7.72374925  1.93736673
##   [7]  3.04205503  3.15589010  3.85173048  0.04172466  0.65514520  4.34793249
##  [13]  3.17808492  3.10648370  5.37714059  1.42934705  5.60167386  0.67231695
##  [19]  5.18567453  5.17934517  7.10342916  1.77796182  0.55936770  1.50706237
##  [25]  1.34995012  7.51006109  1.71698539  0.74261428  3.10360343  0.82775874
##  [31]  6.76168725  1.45375559  0.97624347  2.78965728  5.02947028  1.45820742
##  [37]  2.53718309  5.15932544  1.27578543  1.96729294  6.13067520  1.35600875
##  [43]  3.46601300  2.26886444  1.24846556  3.22603313  4.53897916  2.77599581
##  [49]  0.42925286  6.21051433  6.47556264  3.04001071  2.59510776  1.95340309
##  [55]  4.12488467  0.67956645  3.27647142  1.73208109  0.30273226  1.17609377
##  [61]  5.35341373  3.14226220  1.86075735  0.47815364  3.23000603 11.75789041
##  [67]  0.33181533  0.51872154  0.11252993  4.90185660  0.77188818  0.71215677
##  [73]  1.04923005  3.14858443  5.53529118  4.84456978  0.78876988  1.90186208
##  [79]  0.10826794  0.65776046  8.46651149  2.79864003  1.77970857  6.27276451
##  [85]  0.68427121  8.15561745  2.07413943  3.11743094  1.38537628  1.67254523
##  [91]  1.90687993  2.03913497  3.16649965  4.96890198  0.68169658  0.90039046
##  [97]  2.18658514  1.55754479  5.21555683  0.47886307  5.24174557  0.67239568
## [103]  4.22554684  0.90197483  0.72535645  6.28800675  3.27561107  1.18245521
## [109]  3.26392581  0.74819811  5.33032419  1.87260690  1.29819473  1.45235831
## [115]  1.54271819  0.84350343  0.23997332  3.37185022  1.40880520  4.66257737
## [121]  0.41483361  1.14989189  0.58017046  0.86565911  3.23596829  8.24183157
## [127]  5.08534482  1.81025952  1.62555364  0.84228261  1.28167185  2.30776778
## [133]  0.40040784  2.30902323  2.96775345  9.01510370  1.26511808  1.96499048
## [139]  1.29084727  4.98391205  0.55358598  2.90061169  2.88272584  6.71501676
## [145]  0.85813935  3.56901428  6.43155275  2.48023310  2.35099324  0.89671364
## [151]  1.64217870  0.20970798  0.69768977  0.64244675  1.32026517  5.07314247
## [157]  8.87107516  0.84231363  2.32896179  0.86619038  0.22169714  1.22039132
## [163]  8.32181104  9.57939579  1.74980895  2.69419393  1.49609640  1.21348110
## [169]  3.00993800  7.93179078  2.55831193  5.93369279  1.82868901  2.22536826
## [175]  2.18344573  5.18694335  4.49370808  3.43274309  0.39764860  2.39964662
## [181]  2.86243343  2.05847065  2.22097819  3.00718718  1.03146860  1.35689667
## [187]  4.10179856 12.20513430  1.43914409  4.87272301  0.32119496  2.51842246
## [193]  3.35796990  6.95104744  0.66279911  0.68427396  0.17404412  7.79417089
## [199]  2.44617307  1.05925301
```

- (g)

```
hist(T2)  # create histogram of T2
```


Histogram of T2
Frequency
0
10
20
30
40
50
0
2
4
6
8
10
12
T2