STA 326 2.0 Programming and Data Analysis with R [*]

25 March 2020

Week 2: Answers

Question 1

```
data.vec <- c(244, 191, 160, 187, 180, 176, 174, 205, 211, 183, 211, 180, 194, 200)
names(data.vec) <- c(1:3, "A", "B", "C", "D",
                     "E", "F", "G", "H", "I",
                     "J", "K")
data.vec
```

```
  1   2   3   A   B   C   D   E   F   G   H   I   J   K
244 191 160 187 180 176 174 205 211 183 211 180 194 200
```

Question 2

```
x <- c(4, "a", TRUE) # Example for explicit coercison
class(x)
```

```
[1] "character"
```

Question 3

```
x <- c(3, 5, 1, 10, 12, 6)
x[x < 6] = 0
x
```

```
[1]  0  0  0 10 12  6
```

Question 4

```
weight <- c(60, 72, 57, 90, 95, 72, 70)
## Method 1: Your own code
sqrt(sum((weight-mean(weight))^2)/(length(weight)-1))
```

```
[1] 14.17409
```

```
## Method 2: built-in function
sd(weight)
```

```
[1] 14.17409
```

Question 5

```
mat <- matrix(c(1, 1, 3, 5, 2, 6, -2, -1, -3), ncol=3)
colnames(mat) <- c("a1", "b1", "c1")
```

```
rownames(mat) <- c("A", "B", "C")
mat
```

```
  a1 b1 c1
A  1  5 -2
B  1  2 -1
C  3  6 -3
```

Question 6

```
y <- matrix(c(1, 3, 0, 9, 5, -1), nrow=3, byrow=T)
x <- matrix(c(3, 4, -2, 6), nrow=2, byrow=T)
y
```

```
     [,1] [,2]
[1,]    1    3
[2,]    0    9
[3,]    5   -1
```

```
x
```

```
     [,1] [,2]
[1,]    3    4
[2,]   -2    6
```

```
## Matrix Multiplication
y%*%x
```

```
     [,1] [,2]
[1,]   -3   22
[2,]  -18   54
[3,]   17   14
```

```
## Transpose of x
t(x)
```

```
     [,1] [,2]
[1,]    3   -2
[2,]    4    6
```

```
## Inverse
solve(x)
```

```
           [,1]       [,2]
[1,] 0.23076923 -0.1538462
[2,] 0.07692308  0.1153846
```

```
x[1, ] # gives the first row  of x
```

```
[1] 3 4
```

```
x[2, ] # gives the second row of x
```

```
[1] -2  6
```

```
x[, 2] # gives the second column of x
```

```
[1] 4 6
```

```
y[1, 2] # give the element corresponds to the 1st row and 2nd column
```

```
[1] 3
```

```
y[, 2:3] # gives an error because y contains only two columns
```

```
y[ ,1:2] # extract elements corresponds to the 1st and the 2nd columns
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    0    9
## [3,]    5   -1
```

Question 7

```
# Y variable - Sales
sales <- c(2580, 11942, 9845, 27800, 18926, 4800, 14550)

# X variable - advertising
advertising <- c(1.2, 2.6, 2.2, 3.2, 2.9, 1.5, 2.7)

## Using your own function to estimate beta_0 and beta_1
sales.mat <- matrix(sales, ncol = 1)
advertising.mat <- matrix(c(rep(1, length(advertising)), advertising), ncol=2)

solve(t(advertising.mat)%*%advertising.mat)%*%t(advertising.mat)%*%sales.mat
```

```
          [,1]
[1,] -12348.56
[2,]  10851.71
```

```
## You can use built-in "lm" function to check your answer

lm(sales ~ advertising)
```

```

Call:
lm(formula = sales ~ advertising)

Coefficients:
(Intercept)  advertising  
     -12349        10852  
```

Question 8

```
x <- -1:6
x
```

```
[1] -1  0  1  2  3  4  5  6
```

```
# a
x[x < 0]
```

```
[1] -1
```

```
# b
x[x > 0]
```

```
[1] 1 2 3 4 5 6
```

```
# c
x[x <= 3]
```

```
[1] -1  0  1  2  3
```

```
# d
x[x < 0|x > 4]
```

```
[1] -1  5  6
```

```
# e
x[1]
```

```
[1] -1
```

```
# f
x[c(2, 4)]
```

```
[1] 0 2
```

```
# g
x[!x==0]
```

```
[1] -1  1  2  3  4  5  6
```

Question 9

part a

```
height <- 58:72
weight <- c(115, 117, 120, 123, 126, 129, 132, 135, 139, 142, 146, 150, 154, 159, 164)
mat.height.weight <- matrix(c(height, weight), ncol=2)
mat.height.weight
```

```
      [,1] [,2]
 [1,]   58  115
 [2,]   59  117
 [3,]   60  120
 [4,]   61  123
 [5,]   62  126
 [6,]   63  129
 [7,]   64  132
 [8,]   65  135
 [9,]   66  139
[10,]   67  142
[11,]   68  146
[12,]   69  150
[13,]   70  154
[14,]   71  159
[15,]   72  164
```

part b

```
t(mat.height.weight)
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
[1,]   58   59   60   61   62   63   64   65   66    67    68    69    70    71
[2,]  115  117  120  123  126  129  132  135  139   142   146   150   154   159
     [,15]
[1,]    72
[2,]   164
```

part c

```
# Method 1: convert the matrix mat.height.weight to a dataframe
dataframe.height.weight <- as.data.frame(mat.height.weight)
colnames(dataframe.height.weight) <- c("height", "weight")
dataframe.height.weight
```

```
##    height weight
## 1      58    115
## 2      59    117
## 3      60    120
## 4      61    123
## 5      62    126
## 6      63    129
## 7      64    132
## 8      65    135
## 9      66    139
## 10     67    142
## 11     68    146
## 12     69    150
## 13     70    154
## 14     71    159
## 15     72    164
```

```
# Method 2
dataframe.height.weight <- data.frame(height=height, weight=weight)
dataframe.height.weight
```

```
##    height weight
## 1      58    115
## 2      59    117
## 3      60    120
## 4      61    123
## 5      62    126
## 6      63    129
## 7      64    132
## 8      65    135
## 9      66    139
## 10     67    142
## 11     68    146
## 12     69    150
## 13     70    154
## 14     71    159
## 15     72    164
```

part d

```
dataframe.height.weight[8, ]
```

```
  height weight
8     65    135
```

Question 10

```
## Method 1
data("mtcars")
summary(mtcars$cyl)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  4.000   4.000   6.000   6.188   8.000   8.000 
```

```
## Method 2
data(mtcars)
attach(mtcars)
summary(cyl) # When you attach the data frame you can use the variable name itself. I prefer method 1
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  4.000   4.000   6.000   6.188   8.000   8.000 
```