Lab #1 Activities

Thanh Dat Nguyen

Instructions: Fill in your name in line 3. Fill in the code chunks below and answer the questions with text responses. Be sure when adding in text responses to never copy-paste symbols from outside of the document. Your responses must use code that was covered in class; other methods to solve the problems will not be accepted. Submit your knit pdf file to Crowdmark.

A reminder that the R code we have covered in class is available on our STAT 2150 A01 UM Learn page, under Content > Course Material. It is recommended that you knit to pdf after you fill in each code chunk.

Your knit pdf file should show the result answering each question. To do this, after creating an R object, you should also print it in a new line within the code chunk.

Question 1:

Create a vector (call it **x**) of the five values 6,4,5,7 and 2. Find the mean of these five values. (We say we are *passing in* the **x** vector to the **mean()** function.)

```
x = c(6,4,5,7,2)
mean(x)
```

```
## [1] 4.8
```

Question 2:

There are two datasets built-in to R named **state.area** and **state.name** referring to the 50 U.S. states. We can see what they contain by typing their names (see the knit PDF file):

```
state.area
```

```
##  [1]  51609 589757 113909  53104 158693 104247   5009   2057  58560  58876
## [11]   6450  83557  56400  36291  56290  82264  40395  48523  33215  10577
## [21]   8257  58216  84068  47716  69686 147138  77227 110540   9304   7836
## [31] 121666  49576  52586  70665  41222  69919  96981  45333   1214  31055
## [41]  77047  42244 267339  84916   9609  40815  68192  24181  56154  97914
```

```
state.name
```

```
##  [1] "Alabama"        "Alaska"         "Arizona"        "Arkansas"
##  [5] "California"     "Colorado"       "Connecticut"    "Delaware"
##  [9] "Florida"        "Georgia"        "Hawaii"         "Idaho"
## [13] "Illinois"       "Indiana"        "Iowa"           "Kansas"
## [17] "Kentucky"       "Louisiana"      "Maine"          "Maryland"
## [21] "Massachusetts"  "Michigan"       "Minnesota"      "Mississippi"
## [25] "Missouri"       "Montana"        "Nebraska"       "Nevada"
## [29] "New Hampshire"  "New Jersey"     "New Mexico"     "New York"
## [33] "North Carolina" "North Dakota"   "Ohio"           "Oklahoma"
```

```
## [37] "Oregon"         "Pennsylvania"   "Rhode Island"   "South Carolina"
## [41] "South Dakota"   "Tennessee"      "Texas"          "Utah"
## [45] "Vermont"        "Virginia"       "Washington"     "West Virginia"
## [49] "Wisconsin"      "Wyoming"
```

The datasets are linked, so, for example, the first element of **state.area** gives you the area in square miles of the first state in **state.name**. Write the R code that extracts (accesses) the appropriate element from the **state.area** dataset which gives Idaho's area.

```
area <- c(state.area)
name <- c(state.name)

name[12]
```

```
## [1] "Idaho"
```

```
area[12]
```

```
## [1] 83557
```

The three smallest states in the U.S. are Rhode Island, Delaware, and Connecticut. We would like to find the average area of these three states. Extract these three areas from **state.area** and then pass them into the **mean()** function. (You should not type the actual areas, but write the R code that extracts those areas.)

```
mean(area[39],area[8],area[7])
```

```
## [1] 1214
```

Question 3:

The FIFA World Cup takes place every four years. After Argentina won in 2022, the next World Cup will take place in 2026 in Canada, the U.S., and Mexico. Create a vector called **years** showing the years that all World Cups from 2026 to 2100 are scheduled to take place. Do not manually type all of the years.

```
years <- seq(2026,2100, by = 4)
years
```

```
##  [1] 2026 2030 2034 2038 2042 2046 2050 2054 2058 2062 2066 2070 2074 2078 2082
## [16] 2086 2090 2094 2098
```

Extract a year from the **years** vector showing when the seventh World Cup from now will take place.

```
years[7]
```

```
## [1] 2050
```

Question 4:

Write the R code that creates the following matrix (call it **mymat**), in a way that does not require manually typing each and every value.

$$\begin{bmatrix} 5 & 10 & 15 & 20 \\ 25 & 30 & 35 & 40 \\ 45 & 50 & 55 & 60 \\ 65 & 70 & 75 & 80 \\ 85 & 90 & 95 & 100 \\ 105 & 110 & 115 & 120 \end{bmatrix}$$

```
mymat <- matrix(seq(5,120,by = 5),nrow = 6, byrow = T)
mymat
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    5   10   15   20
## [2,]   25   30   35   40
## [3,]   45   50   55   60
## [4,]   65   70   75   80
## [5,]   85   90   95  100
## [6,]  105  110  115  120
```

Write the R code that extracts the third column of **mymat**.

```
for(i in 1:6){
  print(mymat[i,3])
}
```

```
## [1] 15
## [1] 35
## [1] 55
## [1] 75
## [1] 95
## [1] 115
```

Write the R code that creates the same output as the last code chunk, but uses the **seq()** function rather than accessing the **mymat** matrix.

```
mymat <- matrix(seq(5,120,by = 5),nrow = 6, byrow = T)
mymat
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    5   10   15   20
## [2,]   25   30   35   40
## [3,]   45   50   55   60
## [4,]   65   70   75   80
## [5,]   85   90   95  100
## [6,]  105  110  115  120
```

Write the R code that extracts the second column of **mymat**, stores it in an R object named **column2**, and then print the contents of **column2**. Also write the code that extracts the second row of **mymat**, stores it in an R object named **row2**, and then prints the contents of **row2**.

```r
column2 <- mymat[,2]
print(column2)
```

```
## [1]  10  30  50  70  90 110
```

```r
row2 <- mymat[2,]
print(row2)
```

```
## [1] 25 30 35 40
```

We can calculate the standard deviation of a vector of data values with the **sd()** function. For example:

```r
data = c(1,3,6)
sd(data)
```

```
## [1] 2.516611
```

Calculate the standard deviation of the **column2** and **row2** objects.

```r
sdCol2 = sd(column2)
sdRow2 = sd(row2)
sdCol2
```

```
## [1] 37.41657
```

```r
sdRow2
```

```
## [1] 6.454972
```

Looking at the values in the **column2** and **row2** objects, explain why it makes sense that the standard deviation of the **row2** object is much smaller.

the standard deviation of row 2 is smaller because its value is from the same row, likely closer together, while the values in column2 are from different rows, likely more varied, resulting in a larger standard deviation