CS544 Module 1 Assignment

Part1) 50 points

The following sample data shows the scores of the students in an exam:

59, 46, 76, 60, 49, 65, 82, 68, 99, 52

Using R, assign the above data as a *vector* in the same order to a variable, say, *scores*.
Do the following using R code **with only a single expression** for each case.

a) How many students took the exam?
Using indexing, show the expression for accessing the first two items.
Using indexing, show the expression for accessing the first and last items.
Using indexing, show the expression for accessing the middle two items.
The code should work for a vector of any size. You can assume there are an even number of values.

Sample output:

```
[1] 10
[1] 59 46
[1] 59 52
[1] 49 65
```

b) Use median(scores) to show the median of the data.
Using comparison operators, write the R expression for scores less than or equal to the median of the data.
Using comparison operators, write the R expression for scores greater than the median of the data.
Using the **sum** function, write the R expression for the number of scores less than or equal to the median of the data.
Using the **sum** function, write the R expression for the number of scores greater than the median of the data.

Sample output:

```
[1] 62.5
[1]  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE
[1] FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
[1] 5
[1] 5
```

c) Using logical indexing and the expressions from b), write the R expression for all the scores that are less than or equal to the median value of the data. Similarly, write the R expression for all the scores that are greater than or equal to the median.

Sample output:

```
[1] 59 46 60 49 52
[1] 76 65 82 68 99
```

d) Using logical indexing with TRUE and FALSE values, write the R expression for the odd indexed values from the scores. Similarly, write the R expression for the even indexed values from the scores. The code should work for any size input data. You can assume that there are

even number of values in scores. The code should take advantage of the automatic repetition of indices provided by R.

Sample output:

```
[1] 59 76 49 82 99
[1] 46 60 65 68 52
```

e) Using numeric indexing, write the R expression for the odd indexed values from the scores. Similarly, write the R expression for the even indexed values from the scores. The code should work for any size input data. You can assume that there are even number of values in scores. Use the **seq** function to generate the numeric indices.

Sample output:

```
[1] 59 76 49 82 99
[1] 46 60 65 68 52
```

f) Create a matrix of size 2 x 5 using the scores data. The first five values belong to the first row of the matrix. Assign the result to the variable, *scores.matrix*, and display the result.

Sample output:

```
     [,1] [,2] [,3] [,4] [,5]
[1,]   59   46   76   60   49
[2,]   65   82   68   99   52
```

g) Show the code for displaying the first and last columns of the matrix. The code should work for any size matrix.

Sample output:

```
     [,1] [,2]
[1,]   59   49
[2,]   65   52
```

h) Assign column names for the *scores.matrix* as Student_1, Student_2,… and row names as Quiz_1 and Quiz_2. The code should work for any size matrix, i.e., for any number of columns in the matrix and any number of rows.

Sample output:

```
       Student_1 Student_2 Student_3 Student_4 Student_5
Quiz_1        59        46        76        60        49
Quiz_2        65        82        68        99        52
```

i) Show the result for displaying the first and last columns of the matrix using the same code used in g)

Sample output:

```
       Student_1 Student_5
Quiz_1        59        49
Quiz_2        65        52
```

Part 2) 50 points

Create a data frame, say *dow*, using the column names: *Month, Open, High, Low*, and *Close* using the data shown in the snapshot below.

Month	Open	High	Low	Close
Jan	28639	29374	28170	28256
Feb	28320	29569	24681	25409
Mar	25591	27102	18214	21917
Apr	21227	24765	20735	24346
May	24121	24350	23361	24331

a) Write the R code for creating the above data frame with the above values and display the resulting data frame.

Sample Output:

```
  Month  Open  High   Low Close
1   Jan 28639 29374 28170 28256
2   Feb 28320 29569 24681 25409
3   Mar 25591 27102 18214 21917
4   Apr 21227 24765 20735 24346
5   May 24121 24350 23361 24331
```

b) Write the R expression to show the result of the **summary** function for *Open, High, Low*, and *Close*.

Sample Output:

```
      Open            High            Low            Close      
 Min.   :21227   Min.   :24350   Min.   :18214   Min.   :21917  
 1st Qu.:24121   1st Qu.:24765   1st Qu.:20735   1st Qu.:24331  
 Median :25591   Median :27102   Median :23361   Median :24346  
 Mean   :25580   Mean   :27032   Mean   :23032   Mean   :24852  
 3rd Qu.:28320   3rd Qu.:29374   3rd Qu.:24681   3rd Qu.:25409  
 Max.   :28639   Max.   :29569   Max.   :28170   Max.   :28256  
```

c) Write the R expression to show the data frame sliced using the columns *Month*, *Open*, and *Close*.

Sample Output:

```
  Month  Open Close
1   Jan 28639 28256
2   Feb 28320 25409
3   Mar 25591 21917
4   Apr 21227 24346
5   May 24121 24331
```

d) Write the R expression to show the data frame sliced using the first and last row. The code should work for a data frame of any size.

Sample Output:

```
  Month  Open  High   Low Close
1   Jan 28639 29374 28170 28256
5   May 24121 24350 23361 24331
```

e) Write the R expression to show the data frame sliced using the first and last row using the columns *Month*, *High*, and *Low*. The code should work for a data frame of any size.

Sample Output:

```
  Month  High   Low
1   Jan 29374 28170
5   May 24350 23361
```

f) Write the R expression to show all rows of the data frame whose *Low* is greater than 22,000. The code should work for a data frame of any size. Show one solution using **logical indexing**, and another solution using the **subset** function.

Sample Output:

```
  Month  Open  High   Low Close
1   Jan 28639 29374 28170 28256
2   Feb 28320 29569 24681 25409
5   May 24121 24350 23361 24331
```

g) Write the R expression to show all rows of the data frame whose *Open and Low* are both greater than 25,000. The code should work for a data frame of any size. Show one solution using **logical indexing**, and another solution using the **subset** function.

Sample Output:

```
  Month  Open  High   Low Close
1   Jan 28639 29374 28170 28256
```

h) Modify the data by adding a new column, *Volatility*, showing the difference between the *High* and *Low*. Write the R code and display the resulting data frame.

Sample Output:

```
  Month  Open  High   Low Close Volatility
1   Jan 28639 29374 28170 28256       1204
2   Feb 28320 29569 24681 25409       4888
3   Mar 25591 27102 18214 21917       8888
4   Apr 21227 24765 20735 24346       4030
5   May 24121 24350 23361 24331        989
```

i) Write the R expression to show the row(s) of the data with the maximum *Volatility*. Use **subset** and **max** functions.

Sample Output:

```
  Month  Open  High   Low Close Volatility
3   Mar 25591 27102 18214 21917       8888
```

j) Write the R expression to show the row(s) of the data with the minimum *Volatility*. Use **logical indexing** and **min** functions (**do not** use the subset function).

Sample Output:

```
  Month  Open  High   Low Close Volatility
5   May 24121 24350 23361 24331        989
```

Submission:

- You must work on your assignments individually. You are not allowed to copy the answers from the others.
- Each assignment has a strict deadline. However, you are still allowed to submit your assignment within 2 days after the deadline with a penalty. 15% of the credit will be deducted unless you made previous arrangements with your facilitator and professor. Assignments submitted 2 days after the deadline will not be graded.

When the term *lastName* is referenced, please replace it with your last name.

Create a folder, **CS544_HW1_lastName** and place the following files in this folder.

Provide all R code in a single file, **CS544_HW1_lastName.R**. Clearly mark each subpart of each question.

Provide the corresponding code and outputs from the R console in a single Word document, CS544_HW1_lastName.doc.

Archive the folder (**CS544_HW1_lastName.zip**). Upload the zip file to the Assignments section of Blackboard.