SP17_Intro to Data Science

Problem 3

Student: Lina Mi

First R Exercise: a Review of the R Introduction

1) Create a vector of 100 randomly distributed numbers between 0 and 100 using *runif* and save the vector to the variable *my_vec*, what information does *str* and *summary* tell you about *my_vec*? How do they differ?

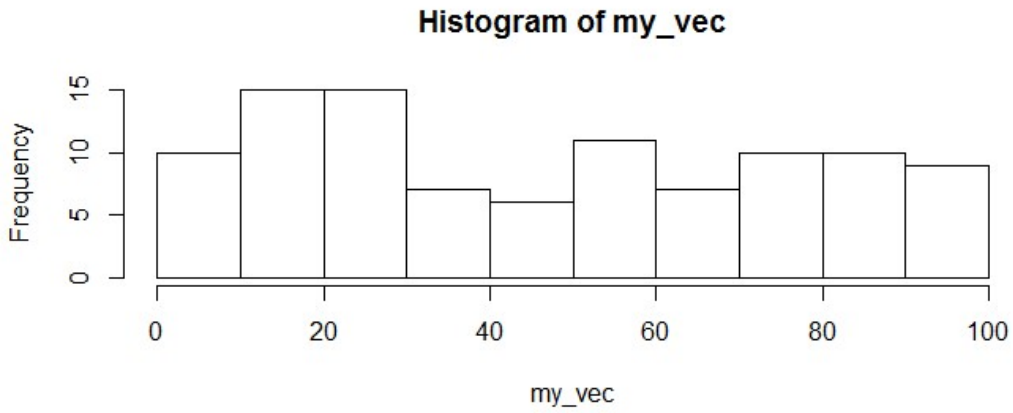
```
  my_vec<-runif(100, min=0, max = 100)
> my_vec
  [1] 59.64720468 90.60509573 17.30011790
  [4] 78.58810767 23.29343846 57.70482090
  [7] 84.08770333 13.22037769 89.58911896
 [10] 45.01373412 89.41425378 24.85451805
 [13]  8.36952936  4.86410747 97.98158670
 [16] 48.41677411 84.53930339 41.62936045
 [19] 48.93425428 18.32878175 75.91614679
 [22] 30.51433025 16.56782471  3.28091430
 [25] 13.65052082 17.71364114 51.95604505
 [28] 81.11207851 11.53620125 89.34217866
 [31] 57.53528811 14.65723943 90.28057964
 [34] 25.30024694 15.05976003 76.85471599
 [37] 23.01233311 30.53993280 51.85696122
 [40] 33.45996668 15.44349683 26.63695686
 [43] 35.07546168 57.84583788 80.86017952
 [46] 93.32703149 83.38633375 12.70027745
 [49] 64.94539515 69.03516576  3.20448244
 [52] 92.04891499 47.84688870 26.65205784
 [55] 85.65107163 22.91464778 79.19468733
 [58] 64.67748603 42.43346907  9.50682680
 [61]  0.34677039 53.11336690 52.43071159
 [64] 21.31855546 71.69320800 96.13435762
 [67] 51.82665996 17.45280223 56.25401349
 [70] 75.92581697 66.69713375 22.48729232
 [73] 34.58497624 31.98317599 90.48983976
 [76] 19.91983801 68.09630166 13.75177586
 [79] 10.69946869  9.28593958 91.64489552
 [82] 27.70604359 88.57938773 77.28646495
 [85] 79.50512362 20.56735931  4.81933230
 [88]  3.88159312 28.45741299 34.88098325
 [91] 73.74533254 25.16635812 51.74370031
 [94] 75.94447227 63.60845279 20.39406854
 [97] 99.30452821  0.04050434 20.65700251
[100] 63.40280906

 str(my_vec)
  num [1:100] 59.6 90.6 17.3 78.6 23.3 ...

summary(my_vec)
   Min. 1st Qu.  Median    Mean 3rd Qu.
 0.0405 20.2800 46.4300 46.6600 75.9300
   Max.
99.3000
```

From the outputs of *str* and *summary* functions, it can be seen that str gives the rang of elements of the vector *my_vec* and lists the first five elements of *my_vec*, *summary* function, however, gives the basic probability characters of *my_vec*, such as mean, median, first quartile and third quartile of *my_vec*

2) Try out a little plot, what does *hist(my_vec)* show you? What information does the helpfile for *hist* tell you about what you just did?



`hist(my_vec)` computes the histogram of the elements values of vector my_vec, such as that occurrence frequency of the elements between 0 and 10 is 10, and plot the resulting histogram if plot=TRUE..

3) Load up the *mtcars* data set using *data(mtcars).* Apply the following functions to *mtcars*: *class*, *str*, *summary*. What does these three functions tell you about *mtcars*?

```
   data(mtcars)
   > class(mtcars)
   [1] "data.frame"
> str(mtcars)
'data.frame':  32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
> summary(mtcars)
      mpg             cyl
 Min.   :10.40   Min.   :4.000
```

```
1st Qu.:15.43   1st Qu.:4.000
Median :19.20   Median :6.000
Mean   :20.09   Mean   :6.188
3rd Qu.:22.80   3rd Qu.:8.000
Max.   :33.90   Max.   :8.000
     disp            hp
Min.   : 71.1   Min.   : 52.0
1st Qu.:120.8   1st Qu.: 96.5
Median :196.3   Median :123.0
Mean   :230.7   Mean   :146.7
3rd Qu.:326.0   3rd Qu.:180.0
Max.   :472.0   Max.   :335.0
     drat            wt
Min.   :2.760   Min.   :1.513
1st Qu.:3.080   1st Qu.:2.581
Median :3.695   Median :3.325
Mean   :3.597   Mean   :3.217
3rd Qu.:3.920   3rd Qu.:3.610
Max.   :4.930   Max.   :5.424
     qsec            vs
Min.   :14.50   Min.   :0.0000
1st Qu.:16.89   1st Qu.:0.0000
Median :17.71   Median :0.0000
Mean   :17.85   Mean   :0.4375
3rd Qu.:18.90   3rd Qu.:1.0000
Max.   :22.90   Max.   :1.0000
      am              gear
Min.   :0.0000   Min.   :3.000
1st Qu.:0.0000   1st Qu.:3.000
Median :0.0000   Median :4.000
Mean   :0.4062   Mean   :3.688
3rd Qu.:1.0000   3rd Qu.:4.000
Max.   :1.0000   Max.   :5.000
     carb
Min.   :1.000
1st Qu.:2.000
Median :2.000
Mean   :2.812
3rd Qu.:4.000
Max.   :8.000
```

According to the outputs of three functions in R showed above, *class* returns the class type of dataset *mtcars*; *str* describes the basic characters of dataset *mtcars*, such as the number of variables and objects, and lists the first several elements for each variable; and first elements the *summary* provides the probability characters of dataset *mtcars*: mean, median, minimum, maximum, first and third quartile of each variable)

Look at the help file for the class *data.frame*. what does it tell you about these objects?

Data.frame is a share many of the properties of matrices and of lists.
fundamental data structure by R's modeling software. This structure tightly coupled collections of variables which

4) What kind of data are you thinking about working with for your final project? Give me a brief description of the data

I would like to choose a R data set, esoph, this data collected from a case-control study of esophageal cancer in ille-et-vilaine, France. This data frame recorded 88 age/alcohol/ tobacco combinations (88 rows) and five variables(five columns), the first variable is "agegp", representing age group study object belongs to; second variable is "alcgp", meaning alcohol consumption; third one is "tobgp", describing the tobacco consumption; the the fourth variable is "ncases", means number of cases; the last variable is "ncontrols", representing number of controls.

First Dynamic Report:

1) What are the column names and data types of the different columns in *iris*?

Column names of iris are *Sepal.Length, Sepal.width, Petal.Length, Petal.Width* and *Species* respectively; the data types of first four columns are numeric, the data type of last column is string

2) How many rows and columns does iris have?

According to the returns of *str(iris)*, There are 150 rows and 5 columns

3) Create a single vector (a new object) called "width" that is Sepal.Width column of iris.

```
>Width_vec<-iris$Sepal.Width
> Width_vec
  [1] 3.5 3.0 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1
 [11] 3.7 3.4 3.0 3.0 4.0 4.4 3.9 3.5 3.8 3.8
 [21] 3.4 3.7 3.6 3.3 3.4 3.0 3.4 3.5 3.4 3.2
 [31] 3.1 3.4 4.1 4.2 3.1 3.2 3.5 3.6 3.0 3.4
 [41] 3.5 2.3 3.2 3.5 3.8 3.0 3.8 3.2 3.7 3.3
 [51] 3.2 3.2 3.1 2.3 2.8 2.8 3.3 2.4 2.9 2.7
 [61] 2.0 3.0 2.2 2.9 2.9 3.1 3.0 2.7 2.2 2.5
 [71] 3.2 2.8 2.5 2.8 2.9 3.0 2.8 3.0 2.9 2.6
 [81] 2.4 2.4 2.7 2.7 3.0 3.4 3.1 2.3 3.0 2.5
 [91] 2.6 3.0 2.6 2.3 2.7 3.0 2.9 2.9 2.5 2.8
[101] 3.3 2.7 3.0 2.9 3.0 3.0 2.5 2.9 2.5 3.6
[111] 3.2 2.7 3.0 2.5 2.8 3.2 3.0 3.8 2.6 2.2
[121] 3.2 2.8 2.8 2.7 3.3 3.2 2.8 3.0 2.8 3.0
[131] 2.8 3.8 2.8 2.8 2.6 3.0 3.4 3.1 3.0 3.1
[141] 3.1 3.1 2.7 3.2 3.3 3.0 2.5 3.0 3.4 3.0
```

4) What is the 100th value in your 'Width' vector?

```
> Width_vec[100]
> [1] 2.8
```

5) What is the last value in your 'Width' vector? Can you write the code that return this value even if how long 'Width' is?

```
> Width_vec[length(Width_vec)]
> [1] 3
```

6) Select rows 10 to 20, with all columns in the iris dataset.

```
> iris[10:20,]
   Sepal.Length Sepal.Width Petal.Length
10          4.9         3.1          1.5
11          5.4         3.7          1.5
12          4.8         3.4          1.6
13          4.8         3.0          1.4
14          4.3         3.0          1.1
15          5.8         4.0          1.2
16          5.7         4.4          1.5
17          5.4         3.9          1.3
18          5.1         3.5          1.4
19          5.7         3.8          1.7
20          5.1         3.8          1.5
   Petal.Width Species
10         0.1  setosa
11         0.2  setosa
12         0.2  setosa
13         0.1  setosa
14         0.1  setosa
15         0.2  setosa
16         0.4  setosa
17         0.4  setosa
18         0.3  setosa
19         0.3  setosa
20         0.3  setosa
```

7) Select rows 10 to 20 with only Species, Petal.Width and Petal.Length. Can you do this two different ways?

```
1/ iris[10:20, c("Petal.Length","Petal.Width","Species")]
2/ iris[10:20, 3:5]
```

8) Select rows 1 to 10, 20 and 100 in iris dataset.

```
1/ Iris[c(1:10, 20:100),]
```

9) Select the first value in the Sepal.Length column of the iris dataset, try three different ways to do that

1/ iris$Sepal.Length[1]

2/ iris[1,'Sepal.Length']

3/ iris[1,1]

10) Without running the following code in R, try to determine which of following will return the first three rows of the Sepal.Length column in the iris data.fram? for each of the answers that *do not work* see if you explain why

a. Iris[c(1,2,3), 'Sepal.Length']
b. Iris[1,2,3, 'Sepal.Length']
c. Iris[(1,2,3), 'Sepal.Length']
d. Iris['Sepal.Length', c(1,2,3)]

Ans: only code a can return the first three rows of the "Sepal.Length" column in data.fram iris. in code b,and c, both 1,2,3 and (1,2,3) can not be used as argument to select first three rows, in code d, argument for choosing rows and that for choosing columns should were put in wrong order, the first argument following iris should be used for rows and second one for columns,