

LAB ASSIGNMENT-2 FEB 9

ITA0443-STATISTICS WITH R PROGRAMMING FOR REAL TIME PROBLEMS

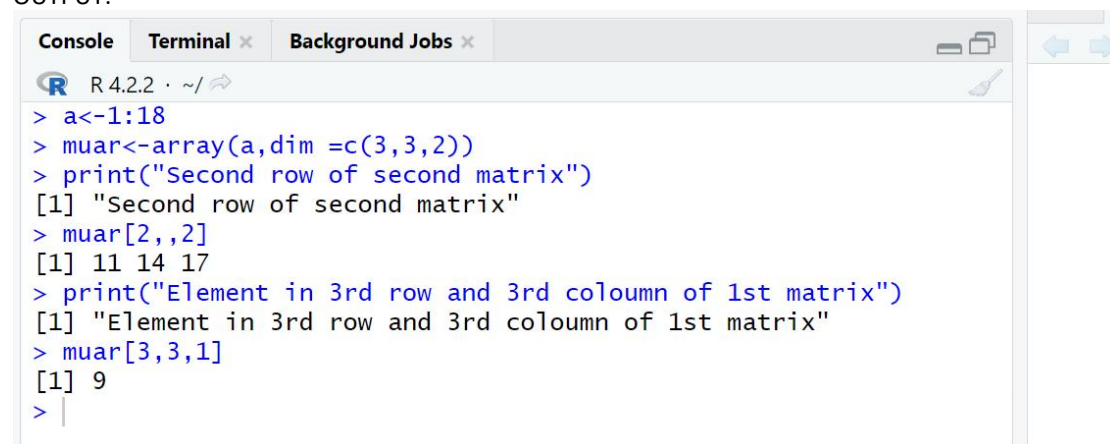
GITHUBLINK:-<https://github.com/Vamsim29/ITA0443-STATISTICS-WITH-R-PROGRAMMING>

1. Write a R program to create an array of two 3x3 matrices each with 3 rows and 3 columns from two given two vectors. Print the second row of the second matrix of the array and the element in the 3rd row and 3rd column of the 1st matrix.

PROGRAM:-

```
a<-1:18
muar<-array(a,dim =c(3,3,2))
print("Second row of second matrix")
muar[2,,2]
print("Element in 3rd row and 3rd coloumn of 1st matrix")
muar[3,3,1]
```

OUTPUT:-

The screenshot shows an R console window with the following output:

```
> a<-1:18
> muar<-array(a,dim =c(3,3,2))
> print("Second row of second matrix")
[1] "Second row of second matrix"
> muar[2,,2]
[1] 11 14 17
> print("Element in 3rd row and 3rd coloumn of 1st matrix")
[1] "Element in 3rd row and 3rd coloumn of 1st matrix"
> muar[3,3,1]
[1] 9
>
```

2. Write a R program to combine three arrays so that the first row of the first array is followed by the first row of the second array and then first row of the third array.

PROGRAM:-

```
a<-matrix(letters[1:9],nrow = 3,ncol = 3)
b<-matrix(1:9,nrow = 3,ncol = 3)
c<-matrix(LETTERS[1:9],nrow = 3,ncol = 3)
a
b
c
m = matrix(t(cbind(a,b,c)),byrow = 9,ncol = 3)
m
```

OUTPUT:-

```
Console Terminal x Background Jobs x
R 4.2.2 · ~/
> a<-matrix(letters[1:9],nrow = 3,ncol = 3)
> b<-matrix(1:9,nrow = 3,ncol = 3)
> c<-matrix(LETTERS[1:9],nrow = 3,ncol = 3)
> a
      [,1] [,2] [,3]
[1,] "a"  "d"  "g"
[2,] "b"  "e"  "h"
[3,] "c"  "f"  "i"
> b
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> c
      [,1] [,2] [,3]
[1,] "A"  "D"  "G"
[2,] "B"  "E"  "H"
[3,] "C"  "F"  "I"

> m = matrix(t(cbind(a,b,c)),byrow = 9,ncol = 3)
> m
      [,1] [,2] [,3]
[1,] "a"  "d"  "g"
[2,] "1"  "4"  "7"
[3,] "A"  "D"  "G"
[4,] "b"  "e"  "h"
[5,] "2"  "5"  "8"
[6,] "B"  "E"  "H"
[7,] "c"  "f"  "i"
[8,] "3"  "6"  "9"
[9,] "C"  "F"  "I"
```

3. Write a R program to create an array using four given columns, three given rows, and two given tables and display the content of the array.

PROGRAM:-

```
d<-array(1:24,dim = c(4,3,2))
d
```

OUTPUT:-

```

Console Terminal x Background Jobs x
R 4.2.2 · ~/
> d<-array(1:24,dim = c(4,3,2))
> d
, , 1
      [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12
, , 2
      [,1] [,2] [,3]
[1,]   13   17   21
[2,]   14   18   22
[3,]   15   19   23
[4,]   16   20   24

```

4. Write a R program to create a two-dimensional 5x3 array of sequence of even integers greater than 50.

PROGRAM:-

```

d<-array(seq(from=52,length.out=15,by=2),dim = c(5,3))
d

```

OUTPUT:-

```

Console Terminal x Background Jobs x
R 4.2.2 · ~/
> d<-array(seq(from=52,length.out=15,by=2),dim = c(5,3))
> d
      [,1] [,2] [,3]
[1,]   52   62   72
[2,]   54   64   74
[3,]   56   66   76
[4,]   58   68   78
[5,]   60   70   80
>

```

Use Below Data frame from question 5 to 9

```

exam_data = data.frame(
  name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'),
  score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
  attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
  qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)

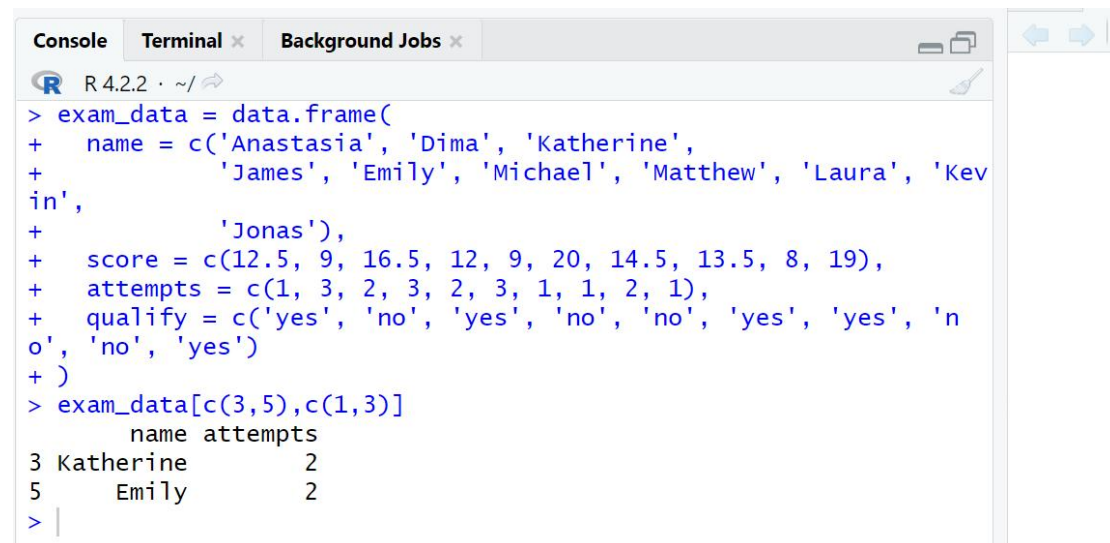
```

5. Write a R program to extract 3 rd and 5 th rows with 1 st and 3 rd columns from a given data frame

PROGRAM:-

```
exam_data = data.frame(
  name = c('Anastasia', 'Dima', 'Katherine',
           'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin',
           'Jonas'),
  score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
  attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
  qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
exam_data[c(3,5),c(1,3)]
```

OUTPUT:-



```
R 4.2.2 · ~/
> exam_data = data.frame(
+   name = c('Anastasia', 'Dima', 'Katherine',
+           'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin',
+           'Jonas'),
+   score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
+   attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
+   qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
+ )
> exam_data[c(3,5),c(1,3)]
      name attempts
3 Katherine        2
5   Emily         2
> |
```

6. Write a R program to add a new column named country in a given data frame

PROGRAM:-

```
exam_data = data.frame(
  name = c('Anastasia', 'Dima', 'Katherine',
           'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin',
           'Jonas'),
  score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
  attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
  qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
Country<-c("USA","USA","USA","USA","UK","USA","USA","India","USA","USA")
newexam_data<-cbind(exam_data,Country)
newexam_data
```

OUTPUT:-

```

Console Terminal Background Jobs
R 4.2.2 · ~/
+ name = c('Anastasia', 'Dima', 'Katherine',
+         'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'K
evin',
+         'Jonas'),
+ score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
+ attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
+ qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'n
o', 'no', 'yes')
+ )
> Country<-c("USA", "USA", "USA", "USA", "UK", "USA", "USA", "India", "U
SA", "USA")
> newexam_data<-cbind(exam_data, Country)
> newexam_data
      name score attempts qualify Country
1 Anastasia 12.5         1    yes      USA
2      Dima   9.0         3     no      USA
3 Katherine 16.5         2    yes      USA
4      James 12.0         3     no      USA
5      Emily  9.0         2     no      UK
6   Michael 20.0         3    yes      USA
7   Matthew 14.5         1    yes      USA
8      Laura 13.5         1     no    India
9      Kevin  8.0         2     no      USA
10     Jonas 19.0         1    yes      USA

```

7. Write a R program to add new row(s) to an existing data frame

```
new_exam_data = data.frame(name = c('Robert', 'Sophia'), score = c(10.5, 9),
                           attempts = c(1, 3), qualify = c('yes', 'no'))
```

PROGRAM:-

```

exam_data = data.frame(
  name = c('Anastasia', 'Dima', 'Katherine',
           'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin',
           'Jonas'),
  score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
  attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
  qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
Country<-c("USA", "USA", "USA", "USA", "UK", "USA", "USA", "India", "USA", "USA")
newexam_data<-cbind(exam_data, Country)
new_exam_data = data.frame(name = c('Robert', 'Sophia'), score = c(10.5, 9),
                           attempts = c(1, 3), qualify = c('yes', 'no'))
final_exam_data<-rbind(exam_data, new_exam_data)
final_exam_data

```

OUTPUT:-

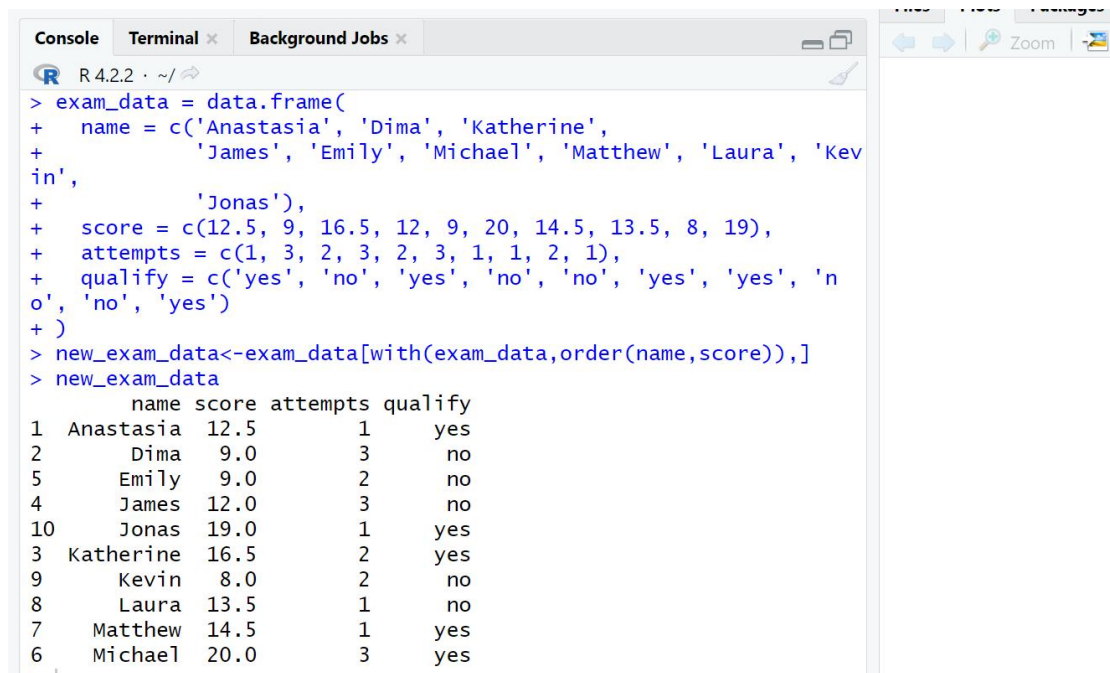
```
Console Terminal Background Jobs
R 4.2.2 ~/  
o', 'no', 'yes')  
+ )  
> Country<-c("USA","USA","USA","USA","UK","USA","USA","India","U  
SA","USA")  
> newexam_data<-cbind(exam_data,Country)  
> new_exam_data = data.frame(name = c('Robert', 'Sophia'),score  
= c(10.5, 9),  
+                               attempts = c(1,3),qualify = c('ye  
s', 'no'))  
> final_exam_data<-rbind(exam_data,new_exam_data)  
> final_exam_data  
      name score attempts qualify  
1 Anastasia 12.5         1     yes  
2 Dima      9.0         3     no  
3 Katherine 16.5         2     yes  
4 James    12.0         3     no  
5 Emily     9.0         2     no  
6 Michael  20.0         3     yes  
7 Matthew  14.5         1     yes  
8 Laura    13.5         1     no  
9 Kevin     8.0         2     no  
10 Jonas   19.0         1     yes  
11 Robert  10.5         1     yes  
12 Sophia   9.0         3     no
```

8. Write a R program to sort a given data frame by name and score

PROGRAM:-

```
exam_data = data.frame(  
  name = c('Anastasia', 'Dima', 'Katherine',  
           'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin',  
           'Jonas'),  
  score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),  
  attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),  
  qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')  
)  
new_exam_data<-exam_data[with(exam_data,order(name,score)),]  
new_exam_data
```

OUTPUT:-



The screenshot shows an R console window with the following code and output:

```
> exam_data = data.frame(
+   name = c('Anastasia', 'Dima', 'Katherine',
+           'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin',
+           'Jonas'),
+   score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
+   attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
+   qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
+ )
> new_exam_data<-exam_data[with(exam_data,order(name,score)),]
> new_exam_data
```

	name	score	attempts	qualify
1	Anastasia	12.5	1	yes
2	Dima	9.0	3	no
5	Emily	9.0	2	no
4	James	12.0	3	no
10	Jonas	19.0	1	yes
3	Katherine	16.5	2	yes
9	Kevin	8.0	2	no
8	Laura	13.5	1	no
7	Matthew	14.5	1	yes
6	Michael	20.0	3	yes

9. Write a R program to save the information of a data frame in a file and display the information of the file.

PROGRAM:-

```
exam_data = data.frame(
  name = c('Anastasia', 'Dima', 'Katherine',
           'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin',
           'Jonas'),
  score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
  attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
  qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
print("The original data frame")
exam_data
save(exam_data, file = "hellodata.rda")
load("hellodata.rda")
file.info("hellodata.rda")
```

OUTPUT:-


```
Console Terminal Background Jobs
R 4.2.2 · ~/
+ qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
+ )
> print("The original data frame")
[1] "The original data frame"
> exam_data
  name score attempts qualify
1 Anastasia 12.5      1     yes
2 Dima      9.0      3     no
3 Katherine 16.5      2     yes
4 James    12.0      3     no
5 Emily     9.0      2     no
6 Michael  20.0      3     yes
7 Matthew  14.5      1     yes
8 Laura    13.5      1     no
9 Kevin     8.0      2     no
10 Jonas   19.0      1     yes
> save(exam_data, file = "hellodata.rda")
> load("hellodata.rda")
> file.info("hellodata.rda")
      size isdir mode                mtime
hellodata.rda 302 FALSE 666 2023-02-10 10:33:44
               ctime                atime exe
hellodata.rda 2023-02-10 10:32:11 2023-02-10 10:33:45 no
```

10. Write a R program to call the (built-in) data `airquality`. Check whether it is a data frame or not? Order the entire data frame by the first and second column. Remove the variables 'Solar.R' and 'Wind' and display the data frame.

PROGRAM:-

```
ar<-airquality
mode(ar)
ar[,c("Solar.R")]=NULL
ar[,c("Wind")]=NULL
ar
```

OUTPUT:-

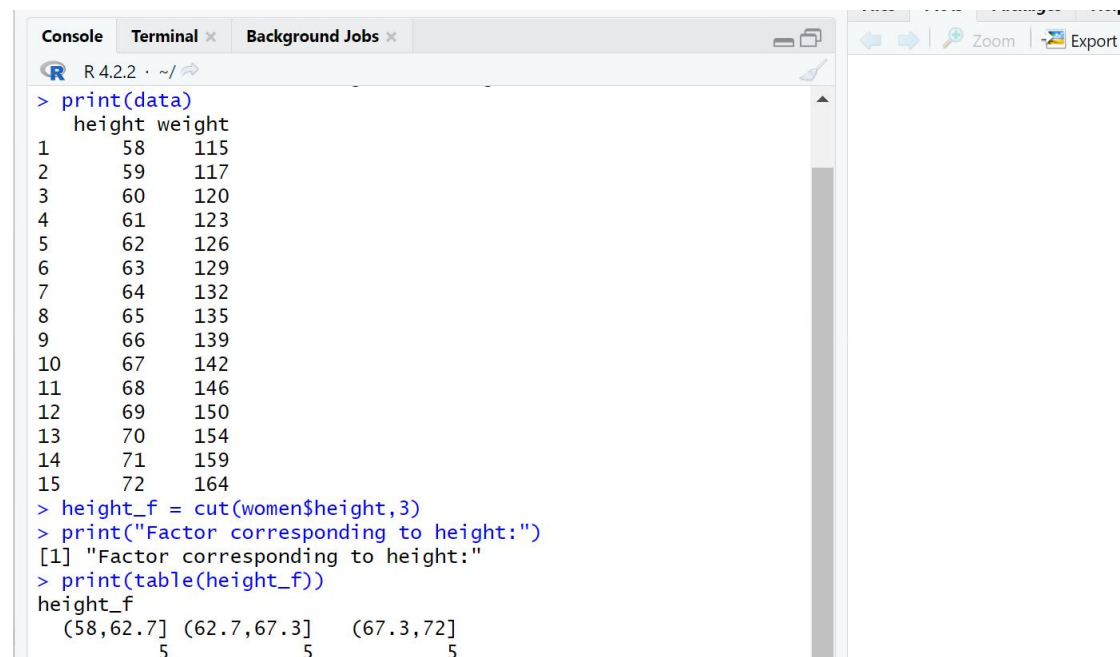
```
Console Terminal Background Jobs
R 4.2.2 · ~/
> ar<-airquality
> mode(ar)
[1] "list"
> ar[,c("Solar.R")]=NULL
> ar[,c("Wind")]=NULL
> ar
  Ozone Temp Month Day
1    41  67     5   1
2    36  72     5   2
3    12  74     5   3
4    18  62     5   4
5    NA  56     5   5
6    28  66     5   6
7    23  65     5   7
8    19  59     5   8
9     8  61     5   9
10   NA  69     5  10
11     7  74     5  11
12    16  69     5  12
13    11  66     5  13
```


11. Write a R program to create a factor corresponding to height of women data set , which inbuilt in R, contains height and weights for a sample of women.

PROGRAM:-

```
data = women
print("Women data set of height and weights:")
print(data)
height_f = cut(women$height,3)
print("Factor corresponding to height:")
print(table(height_f))
```

OUTPUT:-



The screenshot shows the R console output for the program. It displays the data for the 'women' dataset, the creation of the 'height_f' factor using the 'cut' function, and the resulting factor levels and their frequencies.

```
R 4.2.2 · ~/
> print(data)
  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
> height_f = cut(women$height,3)
> print("Factor corresponding to height:")
[1] "Factor corresponding to height:"
> print(table(height_f))
height_f
(58,62.7] (62.7,67.3] (67.3,72]
         5          5          5
```

12. Write a R program to extract the five of the levels of factor created from a random sample from the LETTERS (Part of the base R distribution.)

PROGRAM:-

```
data<-sample(LETTERS,size = 20,replace = TRUE)
print("printing the original data")
data
fac<-factor(data)
print("printing original factors")
fac
print("only five levels")
print(table(data[1:5]))
```

OUTPUT:-

```

> data<-sample(LETTERS,size = 20,replace = TRUE)
> print("printing the original data")
[1] "printing the original data"
> data
[1] "P" "T" "Y" "O" "V" "O" "R" "F" "R" "Y" "M" "N" "W" "Q"
[15] "L" "Q" "G" "C" "B" "Z"
> fac<-factor(data)
> print("printing original factors")
[1] "printing original factors"
> fac
[1] P T Y O V O R F R Y M N W Q L Q G C B Z
Levels: B C F G L M N O P Q R T V W Y Z
> print("only five levels")
[1] "only five levels"
> print(table(data[1:5]))

O P T V Y
1 1 1 1 1
> |

```

13.Iris dataset is a very famous dataset in almost all data mining, machine learning courses, and it has been an R build-in dataset. The dataset consists of 50 samples from each of three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor). Four features(variables) were measured from each sample, they are the length and the width of sepal and petal, in centimetres. Perform the following EDA steps .

- (i) Find dimension, Structure, Summary statistics, Standard Deviation of all features.
- (ii)Find mean and standard deviation of features groped by three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor)
- (iii)Find quantile value of sepal width and length
- (iv)create new data frame named iris1 which have a new column name Sepal.Length.Cate that categorizes "Sepal.Length" by quantile
- (V) Average value of numerical variabelbes by two categorical variables: Species and Sepal.Length.Cate:

PROGRAM:-

```

data<-iris
dim(data)
structure(data)
summary(data)
a<-data$Sepal.Length
b<-data$Sepal.Width
c<-data$Petal.Length
d<-data$Petal.Width
sd(a)
sd(b)
sd(c)
sd(d)
quantile(a)
quantile(b)
quantile(c)
quantile(d)

```

OUTPUT:-

```

R 4.2.2 · ~/
> data<-iris
> dim(data)
[1] 150 5
> structure(data)
      Sepal.Length Sepal.Width Petal.Length Petal.Width
1         5.1         3.5         1.4         0.2
2         4.9         3.0         1.4         0.2
3         4.7         3.2         1.3         0.2
4         4.6         3.1         1.5         0.2
5         5.0         3.6         1.4         0.2
6         5.4         3.9         1.7         0.4
7         4.6         3.4         1.4         0.3
8         5.0         3.4         1.5         0.2
9         4.4         2.9         1.4         0.2
10        4.9         3.1         1.5         0.1
11        5.4         3.7         1.5         0.2
12        4.8         3.4         1.6         0.2
13        4.8         3.0         1.4         0.1
14 virginica
149 virginica
150 virginica
> summary(data)
      Sepal.Length      Sepal.Width      Petal.Length
Min.   :4.300      Min.   :2.000      Min.   :1.000
1st Qu.:5.100      1st Qu.:2.800      1st Qu.:1.600
Median :5.800      Median :3.000      Median :4.350
Mean   :5.843      Mean   :3.057      Mean   :3.758
3rd Qu.:6.400      3rd Qu.:3.300      3rd Qu.:5.100
Max.   :7.900      Max.   :4.400      Max.   :6.900
      Petal.Width      Species
Min.   :0.100      setosa   :50
1st Qu.:0.300      versicolor:50
Median :1.300      virginica :50
Mean   :1.199
3rd Qu.:1.800
Max.   :2.500

```

```

R 4.2.2 · ~/
> b<-data$Sepal.Width
> c<-data$Petal.Length
> d<-data$Petal.Width
> sd(a)
[1] 0.8280661
> sd(b)
[1] 0.4358663
> sd(c)
[1] 1.765298
> sd(d)
[1] 0.7622377
> quantile(a)
 0%  25%  50%  75% 100%
4.3  5.1  5.8  6.4  7.9
> quantile(b)
 0%  25%  50%  75% 100%
2.0  2.8  3.0  3.3  4.4
> quantile(c)
 0%  25%  50%  75% 100%
1.00 1.60 4.35 5.10 6.90
> quantile(d)
 0%  25%  50%  75% 100%
0.1  0.3  1.3  1.8  2.5
>

```

14. Titanic Casualties – Use the standard ‘Titanic’ dataset which is part of R Base to answer the following questions.

- (i). Use an appropriate apply function to get the sum of males vs females aboard.
- (ii). Get a table with the sum of survivors vs sex.

(iii). Get a table with the sum of passengers by sex vs age

PROGRAM:-

```
data<-Titanic  
aa<-data$Male  
bb<-data$Female  
aa  
bb
```