NAME – ANSH GOEL

REG NO. – 20BCE1798

COURSE NAME – FOUNDATION OF DATA ANALYTICS (FDA)

COURSE CODE: 3505

DATE – 4th AUGUST, 2022

LAB-7

1. Complete all 9 programs as given in the class for functions, looping and branching statements.

Indexing and Slicing data frames

1. Read the content of 'Events.csv' in a data frame and view it.

2. Access the scores of participants in event2 using the column name.

3. Use index number to retrieve the same data.

4. Extract the score of third participant in event3.

5. Extract the scores of the first and second participant in all the events.

6. Display the names and total scores of all participants.

7. Make the column "name" as the row index of the data frame.

8. Display the names of the students participated in event3.

9. Obtain the names whose total score is above its average.

MASS package contains a data frame called 'survey' which contains the responses of 237 Statistics I students at the University of Adelaide to a number of questions.

The components of the data frame are:

· Sex The sex of the student. (Factor with levels "Male" and "Female".)

· Wr.Hnd span (distance from tip of thumb to tip of little finger of spread hand) of writing hand, in centimetres.

· NW.Hnd span of non-writing hand.

· W.Hnd writing hand of student. (Factor, with levels "Left" and "Right".)

· Fold “Fold your arms! Which is on top” (Factor, with levels "R on L", "L on R", "Neither".)

· Pulse pulse rate of student (beats per minute).

· Clap ‘Clap your hands! Which hand is on top?’ (Factor, with levels "Right", "Left", "Neither".)

· Exer how often the student exercises. (Factor, with levels "Freq" (frequently), "Some", "None".)

· Smoke how much the student smokes. (Factor, levels "Heavy", "Regul" (regularly), "Occas" (occasionally), "Never".)

· Height height of the student in centimetres. M.I whether the student expressed height in imperial (feet/inches) or metric (centimetres/metres) units. (Factor, levels "Metric", "Imperial".)

· Age age of the student in years.

Do the following:

1. Install the package MASS.
2. Import the package MASS.
3. Display the structure of the data survey.
4. Check the class and type of the data set survey in MASS.
5. Get the number of rows and columns of the survey data frame.
6. Get the dimension of the survey data frame.
7. Provide the statistical summary of the data frame.
8. Display the column names of the survey data frame
9. Retrieve the top 3 rows from the data frame.
10. Extract the bottom 2 rows from the data frame.

Code:

```
#Q1  area of circle
area=function(radius)
{
  a=pi*radius^2;

  return(a)
}

area(5)

#Q2 area of the rectangle
area=function(len,br)
{
  a=len*br
  print(a)
}

area(5,4)

#Q3 to print the numbers from 1:20 in reverse
num=function(n)
{
  for(x in n:1)
  {
    print(x)
  }
}

num(8)

#Q4 function to swap two number
sw=function(a,b)
{
  temp=a
```

```
  a=b
  b=temp
  print(a)
  print(b)
}
sw(5,20)

#Q5  write the function to print the squares in sequence
sq=function(n)
{
  for(i in 1:n)
  {
    print(i^2)
  }
}

sq(10)
#Q6  default function to do all mathematical calculation
cal=function(a,b)
{
  sum=a+b
  print(sum)
  dif=a-b
  print(dif)
  div=a/b
  print(div)
  mod=a%%b
  print(mod)
  mul=a*b
  print(mul)
}
cal(10,5)

#Q7 3x3 matrix multiplication
data <- c(1, 2, 3, 0, 1, 2, 0, 0, 1)
A <- matrix(data, nrow = 3, ncol = 3)

data <- c(0, 1, 1, 1, 0, 3, 1, 3, 3)
B <- matrix(data, nrow = 3, ncol = 3)
```

```
AB <- A %*% B

print("Matrix A")
print(A)
print("Matrix B")
print(B)
print("Matrix Multiplication Result")
print(AB)

#Q8 print the element which is in a but not in b
dif=function(a,b)
{


  print("Elements of vector a that are not in vector b are:")

  ans = setdiff(a, b)
  print(ans)
}
a = c(1, 3, 8, 29, 9, 71, 90)
b = c(17, 8, 6, 90)
dif(a,b)

#Q9 Get three vector A ,B and C
x <- c(1,2,5)
y <- c(5,1,8,9)
z<-c(1,10,8,80,16)

k=union(x,y)
k

p=intersect(x,y)
p
setdiff(x,y)

intersect(x,k)
```

```
union(x,p)
#Q2  Indexing and Slicing data frames

#1st column - name of the participants
#next 7 column contains score of 7 events
#score <=25 score = 0 or else take the exact score

#1 Read the content of 'Events.csv' in a data frame and view it.
data <- read.csv("D:/fda/Book1.csv")
data
data$event1[data$event1 <= 25] <- 0
data$event2[data$event2 <= 25] <- 0
data$event3[data$event3 <= 25] <- 0
data$event4[data$event4 <= 25] <- 0
data$event5[data$event5 <= 25] <- 0
data$event6[data$event6 <= 25] <- 0
data$event7[data$event7 <= 25] <- 0
data

#2 Access the scores of participants in event2 using the column name.
data[,"event2"]

#3 Use index number to retrieve the same data.
data[,3]

#4 Extract the score of third participant in event3.
data[3,4]

#5 Extract the scores of the first and second participant in all the events.
data[1:2, 1:8]

#6 Display the names and total scores of all participants.
data$total <- rowSums(data[2:8])
data[1:20,9]

#7 Make the column "name" as the row index of the data frame.
data$name <- seq.int(nrow(data))
data
```

```
#8 Display the names of the students participated in event3.
subset(data[1:20,1], data$event3>25)

#9 Obtain the names whose total score is above its average.
average <- sum(data[,9])/20
subset(data[1:20,1], data$total>average)

#1.  Install the package MASS.
install.packages("MASS")

#2.  Import the package MASS.
library(MASS)

#3.  Display the structure of the data survey.
str(survey)

#4.  Check the class and type of the data set survey in MASS.
print(class(survey))
print(typeof(survey))

#5.  Get the number of rows and columns of the survey data frame.
print(nrow(survey))
print(ncol(survey))

#6.  Get the dimension of the survey data frame.
print(dim(survey))

#7.  Provide the statistical summary of the data frame.
print(summary(survey))

#8.  Display the column names of the survey data frame
print(colnames(survey))

#9.  Retrieve the top 3 rows from the data frame.
print(head(survey,3))

#10.  Extract the bottom 2 rows from the data frame.
print(tail(survey,2))
```

Output:

1. ***write the function to print the squares in sequence***

```
#Q1  write the function to print the squares in sequence
sq=function(n)
{
  for(i in 1:n)
  {
    print(i^2)
  }
}
```

2.

```
> #Q2 to print the numbers from 1:20 in reverse
> num=function(n)
+ {
+   for(x in n:1)
+   {
+     print(x)
+   }
+ }
>
> num(8)
[1] 8
[1] 7
[1] 6
[1] 5
[1] 4
[1] 3
[1] 2
[1] 1
```

3.

```
> #Q3  area of circle
> area=function(radius)
+ {
+   a=pi*radius^2;
+
+   return(a)
+ }
>
> area(5)
[1] 78.53982
```

4.

```
> #Q4 area of the rectangle
> area=function(len,br)
+ {
+   a=len*br
+   print(a)
+ }
>
> area(5,4)
[1] 20
```

5.

```
> #Q5 function to swap two number
> sw=function(a,b)
+ {
+   temp=a
+   a=b
+   b=temp
+   print(a)
+   print(b)
+ }
> sw(5,20)
[1] 20
[1] 5
>
>
> sq(10)
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
[1] 36
[1] 49
[1] 64
[1] 81
[1] 100
```

6.

```
> #Q6  default function to do all mathematical calculation
> cal=function(a,b)
+ {
+   sum=a+b
+   print(sum)
+   dif=a-b
+   print(dif)
+   div=a/b
+   print(div)
+   mod=a%%b
+   print(mod)
+   mul=a*b
+   print(mul)
+ }
> cal(10,5)
[1] 15
[1] 5
[1] 2
[1] 0
[1] 50
>
```

7.

```
> #Q7 3x3 matrix multiplication
> data <- c(1, 2, 3, 0, 1, 2, 0, 0, 1)
> A <- matrix(data, nrow = 3, ncol = 3)
>
> data <- c(0, 1, 1, 1, 0, 3, 1, 3, 3)
> B <- matrix(data, nrow = 3, ncol = 3)
>
>
> AB <- A %*% B
>
> print("Matrix A")
[1] "Matrix A"
> print(A)
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    2    1    0
[3,]    3    2    1
> print("Matrix B")
[1] "Matrix B"
     [,1] [,2] [,3]
[1,]    0    1    1
[2,]    1    0    3
[3,]    1    3    3
> print("Matrix Multiplication Result")
[1] "Matrix Multiplication Result"
> print(AB)
     [,1] [,2] [,3]
[1,]    0    1    1
[2,]    1    2    5
[3,]    3    6   12
>
```

8.

```
> #Q8 print the element which is in a but not in b
> dif=function(a,b)
+ {
+
+
+   print("Elements of vector a that are not in vector b are:")
+
+   ans = setdiff(a, b)
+   print(ans)
+ }
> a = c(1, 3, 8, 29, 9, 71, 90)
> b = c(17, 8, 6, 90)
> dif(a,b)
[1] "Elements of vector a that are not in vector b are:"
[1]  1  3 29  9 71
>
```

9.

```
> #Q9 Get three vector A ,B and C
> x <- c(1,2,5)
> y <- c(5,1,8,9)
> z<-c(1,10,8,80,16)
>
> k=union(x,y)
> k
[1] 1 2 5 8 9
>
> p=intersect(x,y)
> p
[1] 1 5
> setdiff(x,y)
[1] 2
>
> intersect(x,k)
[1] 1 2 5
>
> union(x,p)
[1] 1 2 5
>
```

Que2:

```
> #Q2  Indexing and Slicing data frames
>
> #1st column - name of the participants
> #next 7 column contains score of 7 events
> #score <=25 score = 0 or else take the exact score
>
> #1 Read the content of 'Events.csv' in a data frame and view it.
> data <- read.csv("D:/fda/Book1.csv")
> data
   participants event1 event2 event3 event4 event5 event6 event7
1            p1      1      2      3      4      5      6      7
2            p2      3      4      5      6      7      8      9
3            p3      5      6      7      8      9     10     11
4            p4      7      8      9     10     11     12     13
5            p5      9     10     11     12     13     14     15
6            p6     11     12     13     14     15     16     17
7            p7     13     14     15     16     17     18     19
8            p8     15     16     17     18     19     20     21
9            p9     17     18     19     20     21     22     23
10          p10     19     20     21     22     23     24     25
11          p11     21     22     23     24     25     26     27
12          p12     23     24     25     26     27     28     29
13          p13     25     26     27     28     29     30     31
14          p14     27     28     29     30     31     32     33
15          p15     29     30     31     32     33     34     35
16          p16     31     32     33     34     35     36     37
17          p17     33     34     35     36     37     38     39
18          p18     35     36     37     38     39     40     41
19          p19     37     38     39     40     41     42     43
20          p20     39     40     41     42     43     44     45
```

```
> data$event1[data$event1 <= 25] <- 0
> data$event2[data$event2 <= 25] <- 0
> data$event3[data$event3 <= 25] <- 0
> data$event4[data$event4 <= 25] <- 0
> data$event5[data$event5 <= 25] <- 0
> data$event6[data$event6 <= 25] <- 0
> data$event7[data$event7 <= 25] <- 0
> data
   participants event1 event2 event3 event4 event5 event6 event7
1            p1      0      0      0      0      0      0      0
2            p2      0      0      0      0      0      0      0
3            p3      0      0      0      0      0      0      0
4            p4      0      0      0      0      0      0      0
5            p5      0      0      0      0      0      0      0
6            p6      0      0      0      0      0      0      0
7            p7      0      0      0      0      0      0      0
8            p8      0      0      0      0      0      0      0
9            p9      0      0      0      0      0      0      0
10          p10      0      0      0      0      0      0      0
11          p11      0      0      0      0      0     26     27
12          p12      0      0      0     26     27     28     29
13          p13      0     26     27     28     29     30     31
14          p14     27     28     29     30     31     32     33
15          p15     29     30     31     32     33     34     35
16          p16     31     32     33     34     35     36     37
17          p17     33     34     35     36     37     38     39
18          p18     35     36     37     38     39     40     41
19          p19     37     38     39     40     41     42     43
20          p20     39     40     41     42     43     44     45
```

2.

```
> #2 Access the scores of participants in event2 using the column name.
> data[,"event2"]
 [1]  0  0  0  0  0  0  0  0  0  0  0  0 26 28 30 32 34 36 38 40
>
```

3.

```
> #3 Use index number to retrieve the same data.
> data[,3]
 [1]  0  0  0  0  0  0  0  0  0  0  0  0 26 28 30 32 34 36 38 40
>
```

4.

```
> #4 Extract the score of third participant in event3.
> data[3,4]
[1] 0
```

5.

```
> #5 Extract the scores of the first and second participant in all the events.
> data[1:2, 1:8]
  participants event1 event2 event3 event4 event5 event6 event7
1           p1      0      0      0      0      0      0      0
2           p2      0      0      0      0      0      0      0
>
```

6.

```
> #6 Display the names and total scores of all participants.
> data$total <- rowSums(data[2:8])
> data[1:20,9]
 [1]   0   0   0   0   0   0   0   0   0   0  53 110 171 210 224 238 252 266 280 294
>
```

7.

```
> #7 Make the column "name" as the row index of the data frame.
> data$name <- seq.int(nrow(data))
> data
   participants event1 event2 event3 event4 event5 event6 event7 total name
1            p1      0      0      0      0      0      0      0     0    1
2            p2      0      0      0      0      0      0      0     0    2
3            p3      0      0      0      0      0      0      0     0    3
4            p4      0      0      0      0      0      0      0     0    4
5            p5      0      0      0      0      0      0      0     0    5
6            p6      0      0      0      0      0      0      0     0    6
7            p7      0      0      0      0      0      0      0     0    7
8            p8      0      0      0      0      0      0      0     0    8
9            p9      0      0      0      0      0      0      0     0    9
10          p10      0      0      0      0      0      0      0     0   10
11          p11      0      0      0      0      0     26     27    53   11
12          p12      0      0      0     26     27     28     29   110   12
13          p13      0     26     27     28     29     30     31   171   13
14          p14     27     28     29     30     31     32     33   210   14
15          p15     29     30     31     32     33     34     35   224   15
16          p16     31     32     33     34     35     36     37   238   16
17          p17     33     34     35     36     37     38     39   252   17
18          p18     35     36     37     38     39     40     41   266   18
19          p19     37     38     39     40     41     42     43   280   19
20          p20     39     40     41     42     43     44     45   294   20
>
```

8.

```
> #8 Display the names of the students participated in event3.
> subset(data[1:20,1], data$event3>25)
[1] "p13" "p14" "p15" "p16" "p17" "p18" "p19" "p20"
>
```

9.

```
> #9 Obtain the names whose total score is above its average.
> average <- sum(data[,9])/20
> subset(data[1:20,1], data$total>average)
[1] "p12" "p13" "p14" "p15" "p16" "p17" "p18" "p19" "p20"
```

1.

```
> #1.  Install the package MASS.
> install.packages("MASS")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the
ppropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/anish/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/MASS_7.3-58.1.zip'
Content type 'application/zip' length 1174504 bytes (1.1 MB)
downloaded 1.1 MB

package 'MASS' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\anish\AppData\Local\Temp\RtmpcljDEf\downloaded_packages
```

2.

```
> #2.  Import the package MASS.
> library(MASS)
>
```

3.

```
> #3.  Display the structure of the data survey.
> str(survey)
'data.frame':   237 obs. of  12 variables:
 $ Sex   : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 1 2 1 2 2 ...
 $ Wr.Hnd: num  18.5 19.5 18 18.8 20 18 17.7 17 20 18.5 ...
 $ NW.Hnd: num  18 20.5 13.3 18.9 20 17.7 17.7 17.3 19.5 18.5 ...
 $ W.Hnd : Factor w/ 2 levels "Left","Right": 2 1 2 2 2 2 2 2 2 2 ...
 $ Fold  : Factor w/ 3 levels "L on R","Neither",..: 3 3 1 3 2 1 1 3 3 3 ...
 $ Pulse : int  92 104 87 NA 35 64 83 74 72 90 ...
 $ Clap  : Factor w/ 3 levels "Left","Neither",..: 1 1 2 2 3 3 3 3 3 3 ...
 $ Exer  : Factor w/ 3 levels "Freq","None",..: 3 2 2 2 3 3 1 1 3 3 ...
 $ Smoke : Factor w/ 4 levels "Heavy","Never",..: 2 4 3 2 2 2 2 2 2 2 ...
 $ Height: num  173 178 NA 160 165 ...
 $ M.I   : Factor w/ 2 levels "Imperial","Metric": 2 1 NA 2 2 1 1 2 2 2 ...
 $ Age   : num  18.2 17.6 16.9 20.3 23.7 ...
>
```

4.

```
> #4.  Check the class and type of the data set survey in MASS.
> print(class(survey))
[1] "data.frame"
> print(typeof(survey))
[1] "list"
>
```

5.

```
> #5.  Get the number of rows and columns of the survey data frame.
> print(nrow(survey))
[1] 237
> print(ncol(survey))
[1] 12
>
```

6.

```
> #6.  Get the dimension of the survey data frame.
> print(dim(survey))
[1] 237  12
>
```

7.

```
> #7.  Provide the statistical summary of the data frame.
> print(summary(survey))
     Sex          Wr.Hnd          NW.Hnd          W.Hnd          Fold        Pulse            Clap
 Female:118   Min.   :13.00   Min.   :12.50   Left : 18   L on R : 99   Min.   : 35.00   Left   : 39
 Male  :118   1st Qu.:17.50   1st Qu.:17.50   Right:218   Neither: 18   1st Qu.: 66.00   Neither: 50
 NA's  :  1   Median :18.50   Median :18.50   NA's :  1   R on L :120   Median : 72.50   Right  :147
              Mean   :18.67   Mean   :18.58                             Mean   : 74.15   NA's   :  1
              3rd Qu.:19.80   3rd Qu.:19.73                             3rd Qu.: 80.00
              Max.   :23.20   Max.   :23.50                             Max.   :104.00
              NA's   :1       NA's   :1                                 NA's   :45
   Exer       Smoke         Height           M.I           Age
 Freq:115   Heavy: 11   Min.   :150.0   Imperial: 68   Min.   :16.75
 None: 24   Never:189   1st Qu.:165.0   Metric  :141   1st Qu.:17.67
 Some: 98   Occas: 19   Median :171.0   NA's    : 28   Median :18.58
            Regul: 17   Mean   :172.4                  Mean   :20.37
            NA's :  1   3rd Qu.:180.0                  3rd Qu.:20.17
                        Max.   :200.0                  Max.   :73.00
                        NA's   :28
>
```

8.

```
> #8.  Display the column names of the survey data frame
> print(colnames(survey))
 [1] "Sex"    "Wr.Hnd" "NW.Hnd" "W.Hnd"  "Fold"   "Pulse"  "Clap"   "Exer"   "Smoke"  "Height" "M.I"
[12] "Age"
>
```

9.

```
> #9.  Retrieve the top 3 rows from the data frame.
> print(head(survey,3))
     Sex Wr.Hnd NW.Hnd W.Hnd   Fold Pulse    Clap Exer Smoke Height      M.I    Age
1 Female   18.5   18.0 Right R on L    92    Left Some Never  173.0   Metric 18.250
2   Male   19.5   20.5  Left R on L   104    Left None Regul  177.8 Imperial 17.583
3   Male   18.0   13.3 Right L on R    87 Neither None Occas     NA     <NA> 16.917
>
```

10.

```
> #10.  Extract the bottom 2 rows from the data frame.
> print(tail(survey,2))
       Sex Wr.Hnd NW.Hnd W.Hnd   Fold Pulse  Clap Exer Smoke Height    M.I    Age
236   Male   21.0   21.5 Right R on L    90 Right Some Never  183.0 Metric 17.167
237 Female   17.6   17.3 Right R on L    85 Right Freq Never  168.5 Metric 17.750
```