Name: Shaikh Inamul Hasan

Roll No: 100

Lab 4: Introduction to R programming and Data acquisition

1. Install package Rcmdr and load in the R studio

Code & Output:

```
> install.packages("Rcmdr")

The downloaded binary packages are in
        C:\Users\Inam\AppData\Local\Temp\Rtmp2Vhjsn\downloaded_packages
```

2. Exercise on workspace

a. Set the current working directory

Code:

```
> setwd('C:\ADMS')
```

b. print the current working directory

Code & Output:

```
> getwd()
[1] "C:/ADMS"
```

c. lists files in working directory

Code & Output:

```
> list.files()
[1] "Customer.accdb"          "Emplyee.xls"             "Pentaho Assignment.docx"
[4] "Staff.accdb"
```

3. Exercise on R data types

a. Use R to calculate the following:

i. 31 * 78 ii. 697 / 41

Code & Output:

```
> 31 * 78
[1] 2418
> 697 / 41
[1] 17
```

b. Assign the value of 39 to x, 22 to y and Make variable z the value of x – y and display value of z on console.

Code & Output:

```
> x <- 39
> y <- 22
> z <- x - y
> cat("The value of z is:", z, "\n")
The value of z is: 17
```

c. Checking the type of variable z

Code & Output:

```
> variable_type <- class(z)
> cat("The type of z is:", variable_type, "\n")
The type of z is: numeric
```

d. Check data type of z whether it is integer or not

Code & Output:

```
> is_integer <- is.integer(z)
> if (is_integer) {
+   cat("z is of integer type.\n")
+ } else {
+   cat("z is not of integer type.\n")
+ }
z is not of integer type.
```

e. Calculate the following quantities:

i. The sum of 100.1, 234.9 and 12.01.

ii. The square root of 256

iii. Calculate the square root of 2345.

Code & Output:

```
> sum(100.1,234.9,12.01)
[1] 347.01
> sqrt(256)
[1] 16
> sqrt(2345)
[1] 48.4252
```

4. Exercise on Vector

a. Create a vector x using : operator and display the value of dim(x) and length(x)

Code & Output:

```
> x <- 1:5
> print(dim(x))
NULL
> print(length(x))
[1] 5
```

b. Consider two vectors, x, y x=c(4,6,5,7,10,9,4,15) y=c(0,10,1,8,2,3,4,1) What is the value of: x*y and x+y

Code & Output:

```
> x <- c(4, 6, 5, 7, 10, 9, 4, 15)
> y <- c(0, 10, 1, 8, 2, 3, 4, 1)
> cat(x*y)
0 60 5 56 20 27 16 15
> cat(x+y)
4 16 6 15 12 12 8 16
```

c. Consider two vectors, a, b a=c(1,5,4,3,6) b=c(3,5,2,1,9) What is the value of: a<=b

Code & Output:

```
> a <- c(1, 5, 4, 3, 6)
> b <- c(3, 5, 2, 1, 9)
> cat(a <= b)
TRUE TRUE FALSE FALSE TRUE
```

d. If x=c ('blue', 'red', 'green', 'yellow') check for the datatype of vector x and check whether it is character datatype or not.

Code & Output:

```
> x <- c('blue', 'red', 'green', 'yellow')
> cat(class(x))
character
> cat(is.character(x))
TRUE
```

e. Consider two vectors, a, b a=c(10,2,4,15) b=c(3,12,4,11) What is the value of: rbind(a,b) and cbind(a,b).

Code & Output:

```
> a <- c(10, 2, 4, 15)
> b <- c(3, 12, 4, 11)
> rbind_result <- rbind(a, b)
> cbind_result <- cbind(a, b)
> print(rbind_result)
  [,1] [,2] [,3] [,4]
a   10    2    4   15
b    3   12    4   11
> print(cbind_result)
      a  b
[1,] 10  3
[2,]  2 12
[3,]  4  4
[4,] 15 11
```

f. The numbers below are the first ten days of rainfall amounts in 1996. Read them in to a vector using the c() function 0.1, 0.6, 33.8, 1.9, 9.6, 4.3, 33.7, 0.3, 0.0, 0.1. What was the mean rainfall, how about the standard deviation?

Code & Output:

```
> rainfall <- c(0.1, 0.6, 33.8, 1.9, 9.6, 4.3, 33.7, 0.3, 0.0, 0.1)
> cat(mean(rainfall))
8.44
> cat(sd(rainfall))
13.66473
```

g. The weights of five people before and after a diet programme are given in the table. Read the `before' and `after' values into two different vectors called before and after.

Before 78 72 78 79 105

After 67 65 79 70 93

Use R to evaluate the amount of weight lost for each participant. What is the average amount of weight lost?

Code & Output:

```
> before <- c(78, 72, 78, 79, 105)
> after <- c(67, 65, 79, 70, 93)
> weight_lost <- before - after
> average_weight_lost <- mean(weight_lost)
> cat(weight_lost)
11 7 -1 9 12
> cat(average_weight_lost)
7.6
```

5. Exercise on Matrix a. Construct the following matrix A and check dimension and attribute of this created matrix :

1	3
2	4

Code & Output:

```
> A <- matrix(c(1, 2, 3, 4), nrow = 2)
> cat(dim(A))
2 2
> cat(str(attributes(A)))
List of 1
 $ dim: int [1:2] 2 2
```

b. Construct another matrix B

12	32
12	24

Code:

```
> B <- matrix(c(12, 12, 32, 24), nrow = 2)
```

c. Calculate the following from above matrices A and B

i. A+B ii. A-B iii. A%*%B iv. A*B

Code & Output:

```
> A <- matrix(c(1, 2, 3, 4), nrow = 2)
> B <- matrix(c(12, 12, 32, 24), nrow = 2)
> cat( A + B)
13 14 35 28
> cat(A - B)
-11 -10 -29 -20
> cat(A %*% B)
48 72 104 160
> cat(A * B)
12 24 96 96
```

d. Calculate the transpose and determinant of matrix A

Code & Output:

```
> A <- matrix(c(1, 2, 3, 4), nrow = 2)
> print(t(A))
     [,1] [,2]
[1,]    1    2
[2,]    3    4
> cat( det(A))
-2
```

e. Use cbind() to add column values (5,6) and rbind() to add row values (3,1) to matrix A and observe the result

Code & Output:

```
> A <- matrix(c(1, 2, 3, 4), nrow = 2)
> print(cbind(A, c(5, 6)))
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> print(rbind(A, c(3, 1)))
     [,1] [,2]
[1,]    1    3
[2,]    2    4
[3,]    3    1
```

6. Exercise on Factor

a. 1 If x = c(1, 2, 3, 3, 5, 3, 2, 4, NA), what are the levels of factor(x)?

Code & Output:

```
> x <- c(1, 2, 3, 3, 5, 3, 2, 4, NA)
> factor_x <- factor(x, levels = c(1, 2, 3, 4, 5, NA))
> factor_levels <- levels(factor_x)
> cat("Levels of factor(x):", factor_levels, "\n")
Levels of factor(x): 1 2 3 4 5
```

b. If z <- c("p", "a" , "g", "t", "b"), then What is the R expression will re place the third element in z with "b".

Code & Output:

```
> z <- c("p", "a", "g", "t", "b")
> z[3] <- "b"
> z
[1] "p" "a" "b" "t" "b"
```

c. If z <- factor(c("p", "q", "p", "r", "q")) and levels of z are "p", "q" ,"r", write an R expression that will change the level "p" to "w" so that z is equal to: "w", "q" , "w", "r" , "q". levels(z)[1]<-'w'

Code & Output:

```
> z <- factor(c("p", "q", "p", "r", "q"), levels = c("p", "q", "r"))
> levels(z)[1] <- "w"
> z
[1] w q w r q
Levels: w q r
```

d. If: s1 <- factor(sample(letters, size=5, replace=TRUE)) and s2 <- factor(sample(letters, size=5, replace=TRUE)), write an R expression that will concatenate s1 and s2 in a single factor with 10 elements. factor(c(levels(s1)[s1], levels(s2)[s2]))

Code & Output:

```
> s1 <- factor(sample(letters, size = 5, replace = TRUE))
> s2 <- factor(sample(letters, size = 5, replace = TRUE))
> concatenated_factor <- factor(c(levels(s1)[s1], levels(s2)[s2]))
> print(concatenated_factor)
 [1] u m t j w v q s u c
Levels: c j m q s t u v w
```

7. Exercise on DataFrame

a. Create the following "Student" data frame

	Age	Height	Weight	Sex
Alex	25	177	57	F
Lilly	31	163	69	F
Mark	23	190	83	M
Oliver	52	179	75	M
Martha	76	163	70	F
Lucas	49	183	83	M
Caroline	26	164	53	F

Code & Output:

```
> Student <- data.frame(
+   Name = c("Alex", "Lilly", "Mark", "Oliver", "Martha", "Lucas", "Caroline"),
+   Age = c(25, 31, 23, 52, 76, 49, 26),
+   Height = c(177, 163, 190, 179, 163, 183, 164),
+   Weight = c(57, 69, 83, 75, 70, 83, 53),
+   Sex = c("F", "F", "M", "M", "F", "M", "F")
+ )
> Student
      Name Age Height Weight Sex
1     Alex  25    177     57   F
2    Lilly  31    163     69   F
3     Mark  23    190     83   M
4   Oliver  52    179     75   M
5   Martha  76    163     70   F
6    Lucas  49    183     83   M
7 Caroline  26    164     53   F
```

b. Create this data frame (make sure you import the variable working as character and not factor).

Code & Output:

```
> data <- data.frame(
+   Name = c("Alex", "Lilly", "Mark", "Oliver", "Martha", "Lucas", "Caroline"),
+   Working = c("Yes", "No", "No", "Yes", "Yes", "No", "Yes"),
+   stringsAsFactors = FALSE
+ )
> data
      Name Working
1     Alex     Yes
2    Lilly      No
3     Mark      No
4   Oliver     Yes
5   Martha     Yes
6    Lucas      No
7 Caroline     Yes
```

c. Add this data frame column-wise to the previous one table.

i. How many rows and columns does the new data frame have?

ii. What class of data is in each column?

iii. Access only Height column and display it

iv. Display total rows and cols in table

Code & Output:

```
> Student <- data.frame(
+   Name = c("Alex", "Lilly", "Mark", "Oliver", "Martha", "Lucas", "Caroline"),
+   Age = c(25, 31, 23, 52, 76, 49, 26),
+   Height = c(177, 163, 190, 179, 163, 183, 164),
+   Weight = c(57, 69, 83, 75, 70, 83, 53),
+   Sex = c("F", "F", "M", "M", "F", "M", "F"),
+   stringsAsFactors = FALSE
+ )
> data <- data.frame(
+   Name = c("Alex", "Lilly", "Mark", "Oliver", "Martha", "Lucas", "Caroline"),
+   Working = c("Yes", "No", "No", "Yes", "Yes", "No", "Yes"),
+   stringsAsFactors = FALSE
+ )
> combined_data <- cbind(Student, data)
> dimensions <- dim(combined_data)
> data_classes <- sapply(combined_data, class)
> height_column <- combined_data$Height
> total_rows <- nrow(combined_data)
> total_cols <- ncol(combined_data)
> cat("i. Number of rows and columns in the new data frame:", dimensions[1], "rows and", dimensions
[2], "columns\n")
i. Number of rows and columns in the new data frame: 7 rows and 7 columns
> cat("ii. Class of data in each column:", data_classes, "\n")
ii. Class of data in each column: character numeric numeric numeric character character character
> cat("iii. Height column:\n")
iii. Height column:
> print(height_column)
[1] 177 163 190 179 163 183 164
> cat("iv. Total rows in the table:", total_rows, "and total columns:", total_cols, "\n")
iv. Total rows in the table: 7 and total columns: 7
```

d. Create empty dataframe having columns Name and Age with data type character and numeric respectively.

Code & Output:

```
> empty_data_frame <- data.frame(
+   Name = character(0),
+   Age = numeric(0)
+ )
> empty_data_frame
[1] Name Age
<0 rows> (or 0-length row.names)
```

8. Create above Student table using table command.

Code & Output:

```
> Student <- data.frame(
+   Name = c("Alex", "Lilly", "Mark", "Oliver", "Martha", "Lucas", "Caroline"),
+   Age = c(25, 31, 23, 52, 76, 49, 26),
+   Height = c(177, 163, 190, 179, 163, 183, 164),
+   Weight = c(57, 69, 83, 75, 70, 83, 53),
+   Sex = c("F", "F", "M", "M", "F", "M", "F"),
+   stringsAsFactors = FALSE
+ )
> student_table <- table(Student)
> student_table
, , Height = 163, Weight = 53, Sex = F

          Age
Name       23 25 26 31 49 52 76
  Alex      0  0  0  0  0  0  0
  Caroline  0  0  0  0  0  0  0
  Lilly     0  0  0  0  0  0  0
  Lucas     0  0  0  0  0  0  0
  Mark      0  0  0  0  0  0  0
  Martha    0  0  0  0  0  0  0
  Oliver    0  0  0  0  0  0  0

, , Height = 164, Weight = 53, Sex = F

          Age
Name       23 25 26 31 49 52 76
  Alex      0  0  0  0  0  0  0
  Caroline  0  0  1  0  0  0  0
  Lilly     0  0  0  0  0  0  0
  Lucas     0  0  0  0  0  0  0
  Mark      0  0  0  0  0  0  0
  Martha    0  0  0  0  0  0  0
  Oliver    0  0  0  0  0  0  0
```

9. Exercise on List

a. If: p <- c(2,7,8), q <- c("A", "B", "C") and x <- list(p, q), then what is the value of x[2]?

Code & Output:

```
> p <- c(2, 7, 8)
> q <- c("A", "B", "C")
> x <- list(p, q)
> second_element <- x[[2]]
> print(second_element)
[1] "A" "B" "C"
```

b. If Newlist <- list(a=1:10, b="Good morning", c="Hi"), write an R statement that will add 1 to each element of the first vector in Newlist.

Code & Output:

```
> Newlist <- list(a=1:10, b="Good morning", c="Hi")
> Newlist$a <- Newlist$a + 1
> print(Newlist)
$a
 [1]  2  3  4  5  6  7  8  9 10 11

$b
[1] "Good morning"

$c
[1] "Hi"
```

10. Exercise on Reading and writing .csv file

a. Load comm separated .csv file in R studio and check dimensions of loaded data

Code & Output:

```
> data <- read.csv("user_csv.csv")
> sedata_dimensions <- dim(data)
> cat("Dimensions of loaded data:", data_dimensions[1], "rows and", data_dimensions[2], "columns
\n")
Dimensions of loaded data: 7 rows and 2 columns
```

b. Check first few rows of the inserted data

Code & Output:

```
> print(head(data))
  userid date_start day_of_week_start hour_start min_start time_status  date_end
1   5967  1/10/2011                 2         10        56           M 1/10/2011
2   5967  1/11/2011                 3          6        24           M 1/11/2011
3   5967  1/11/2011                 3         10        56           M 1/11/2011
4   5967  1/11/2011                 3         12         8           N 1/11/2011
5   5967  1/11/2011                 3         16        49           A 1/11/2011
6   5967  1/12/2011                 4          1         0           M 1/12/2011
  day_of_week_end hour_end min_end working_day time_diff placeid_char
1               2       11      59           1        63           P1
2               3       10      49           1       265           P2
3               3       11      59           1        63           P1
4               3       16      35           1       267           P2
5               3       24       0           1       431           P1
6               4        6      14           1       314           P1
```

c. Check last few rows of the inserted data

Code & Output:

```
> print(tail(data))
    userid date_start day_of_week_start hour_start min_start time_status  date_end
97    5967  3/25/2011                 6         17         3           A 3/25/2011
98    5967  3/25/2011                 6         20        38           N 3/25/2011
99    5967  3/25/2011                 6         23         8           N 3/25/2011
100   5967  3/26/2011                 7          1         0           M 3/26/2011
101   5967  3/26/2011                 7         16         0           A 3/26/2011
102   5967  3/26/2011                 7         16        16           A 3/26/2011
    day_of_week_end hour_end min_end working_day time_diff placeid_char
97                6       20      33           1       210          P96
98                6       22      51           1       133         P154
99                6       24       0           1        52           P3
100               7       14      33           0       813           P3
101               7       16      15           0        15           P1
102               7       17       0           0        44           P4
```

d. create data frame and save data in test.csv file

Code & Output:

```
> my_data <- data.frame(
+   Name = c("Alice", "Bob", "Charlie", "David"),
+   Age = c(25, 30, 22, 28),
+   Score = c(95, 88, 75, 92)
+ )
> write.csv(my_data, file = "test.csv", row.names = FALSE)
> cat("Data has been saved to 'test.csv'\n")
Data has been saved to 'test.csv'
> data<-read.csv("test.csv")
> print(data)
     Name Age Score
1   Alice  25    95
2     Bob  30    88
3 Charlie  22    75
4   David  28    92
```

11. Exercise on Reading and writing excel file library("readxl")

a. Load .xlsx file in R studio data frame

Code & Output:

```
> library(readxl)
> data <- read_excel("Employee.xlsx")
> data
# A tibble: 5 × 8
     ID `First Name` `Last Name` Street       City    State      ZipCode   Age
  <dbl> <chr>        <chr>       <chr>        <chr>   <chr>        <dbl> <dbl>
1     1 Mary         Joe         Thakur Marg  Mumbai  Maharastra  400101     5
2     2 Ram          Singh       LT Marg      Baroda  Gujrat      400206    15
3     3 Akshay       Kumar       SV Road      Lucknow UP          400207    25
4     4 Viraj        Gupta       Linking Road Jaipur  Rajasthan   400203    11
5     5 Samay        Khurana     WE highway   Nagpur  Maharastra  400102    18
```

b. Display all columns and selected rows

Code & Output:

```
> print(data[2:5, ])
# A tibble: 4 × 8
     ID `First Name` `Last Name` Street       City    State      ZipCode   Age
  <dbl> <chr>        <chr>       <chr>        <chr>   <chr>        <dbl> <dbl>
1     2 Ram          Singh       LT Marg      Baroda  Gujrat      400206    15
2     3 Akshay       Kumar       SV Road      Lucknow UP          400207    25
3     4 Viraj        Gupta       Linking Road Jaipur  Rajasthan   400203    11
4     5 Samay        Khurana     WE highway   Nagpur  Maharastra  400102    18
```

c. Reading and writing data suing ReadXL and WriteXL command

Code & Output:

```
> library(readxl)
> data <- read_excel("Employee.xlsx")
> library(writexl)
> my_data <- data.frame(
+   Name = c("Alice", "Bob", "Charlie"),
+   Age = c(25, 30, 22),
+   Score = c(95, 88, 75)
+ )
> write_xlsx(my_data, path = "output.xlsx")
> new_data <- read_excel("output.xlsx")
> print(new_data)
# A tibble: 3 × 3
  Name      Age Score
  <chr>   <dbl> <dbl>
1 Alice      25    95
2 Bob        30    88
3 Charlie    22    75
```