# SECTION 2 DATA SCIENCE LAB

**Structure** **Page Nos.**



## 2.0 INTRODUCTION

This is the era of Information and Communication Technology. With the enhancement of information and communication technology tools, world wide web, mobile technologies, IoT devices etc. led to creation of very large amount of heterogenous data. Today, you are surrounded by a vast reservoir of data, which is required to be processed and analysed. R programming is one of the popular tools for performing statistical analysis of data. Please note that you may be using R to analyse the structured data most often, however, R programming can also be used to process unstructured data too. As far as size of data that can be processed at a time using R is concerned, R is claimed to process about 1 million records of data with ease. However, several applications claim to use even bigger data sets using R. Thus, R is truly a powerful programming language for data analysis. Also, R has a rich syntax for graphical presentation of the data and results.

R programming language was designed at the University of Auckland by R Ihaka and R Gentleman and is available as an open-source programming language. In this lab session, you may use open-source license RStudio Desktop version. This section introduces you to using RStudio desktop for R programming. You may refer to https://www.r-project.org/ about R project and https://cran.r-project.org/manuals.html R documentation. You may also refer to https://www.rstudio.com /products/rstudio/ website for RStudio.

## 2.1 OBJECTIVES

After going through this section and performing all the practical exercises you shouldbe able to:

- Install the open-source R programming and its environment;
- Install open-source tool of R Programming;
- Perform analysis using simple data using R;
- Perform complex analysis of data using R.

## 2.2 INSTALLING R AND RSTUDIO

First you must install R programming with its environment, which includes software features for data creation, storage, manipulation with a rich set of array-based operators, creation of graphs for data visualization and a rich set of data analysis functions. In addition, it has the constructs of programming language including conditional statements, looping constructs, functions etc. Like the present-day programming language, R also provides a rich set of ever increasing number of packages, which support many additional features. However, in order to use the features of R, you must first install it. You can use R website https://cran.r-project.org/ to download R programming, as per your operating system and install R into your system.

After you have installed R, the next task is to install the open source RStudio Desktop. For this you may visit the website https://www.rstudio.com/products/rstudio/download/#download and download the RStudio Desktop, which has open-source license. You should install this software on your computer.

## 2.3 RSTUDIO FOR DATA SCIENCE

After the latest version of R and RStudio Desktop has been installed, you are ready to use RStudio Desktop version for various purposes. In this section, we discuss how you can use RStudio for data science.

On opening the RStudio, you will get the window, as shown in Figure 1. The important tab, which you should all be aware of is the help tab, as shown in Figure 1.

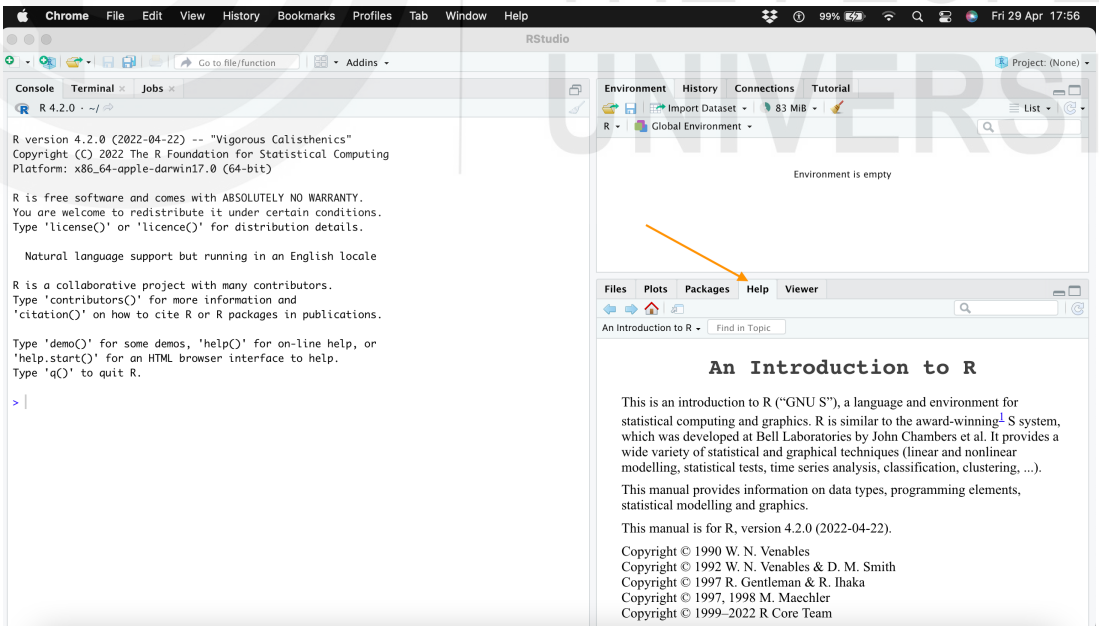



Figure 1: The RStudio Environment with focus on Help

Now, you are ready for R programming, but before that please go through the Block 4 of MCS226: Data Science and Big data, which covers some of the basic set of useful commands of R programming. You may notice that Global environment is currently empty.

A data science project involves several steps including data creation, modification, graphical data representation, exploratory data analysis and modeling. Let us now perform some basic operations using R in RStudio Desktop, with the help of a hypothetical example. Please note this example does not cover the entire gambit of data science project, which also has major steps relating to feature extraction and model optimization.

Example 1: Consider a School collecting following data of its students:

Enrolment Number, % attendance, AverageStudyHoursperweek, %of marks in final

The school believes that student's performance is highly dependent on these two factors. The hypothetical data for this was collected in an MS-Excel file. How can you use R Programming to develop this model?

Solution: For performing this analysis, we use a sample file PracticalData.xlsx, which is shown in the Figure 2.

| EnrolmentNumber | MarksPercent | PercentAttendence | WeeklyStudyHr |
|---|---|---|---|
| S20200001 | 95 | 91 | 20 |
| S20200002 | 91 | 93 | 18 |
| S20200003 | 62 | 70 | 6 |
| S20200004 | 75 | 80 | 10 |
| S20200005 | 88 | 92 | 15 |
| S20200006 | 79 | 88 | 16 |
| S20200007 | 55 | 70 | 3 |
| S20200008 | 69 | 75 | 8 |
| S20200009 | 60 | 70 | 6 |
| S20200010 | 80 | 75 | 17 |

Figure 2: Sample Data in PracticalData.xlsx File

In order to impot this data into RStudio Desktop, you may select Import Dataset option followed by "From Excel…" in the drop-down list, as shown in Figure 3.

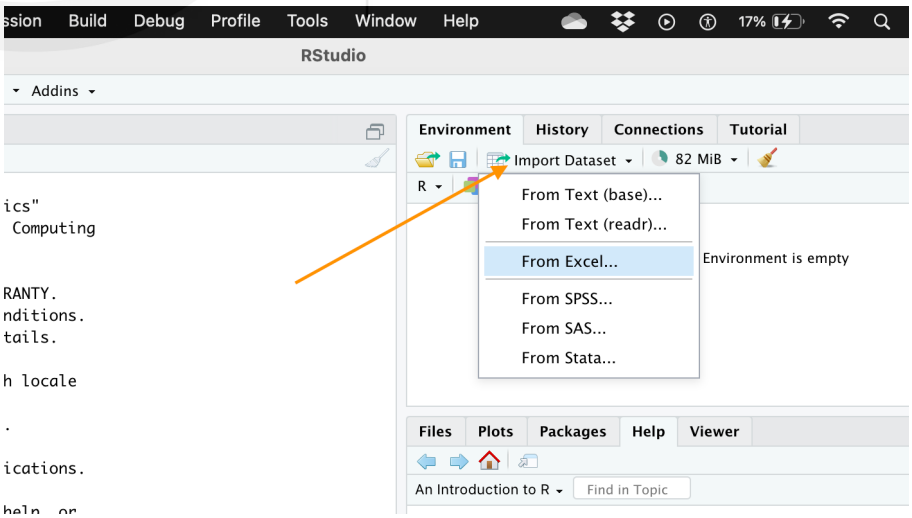



Figure 3: Selection for importing data from Excel

However, as you may be importing the data for the first time, the related package may not have been installed. Therefore, you may receive the Dialog Box, as shown in Figure 4.

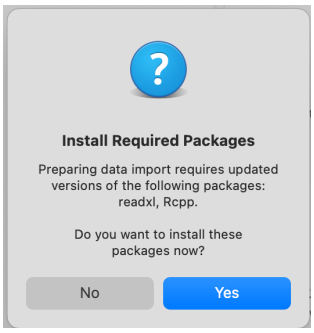



Figure 3: Dialog Box for installing packages for reading Excel file

Select Yes, the highlighted option in Figure 3. It will result in installation of the required packages. After the required package is installed, the dialog box, as shown in Figure 4, will appear for importing Excel data.

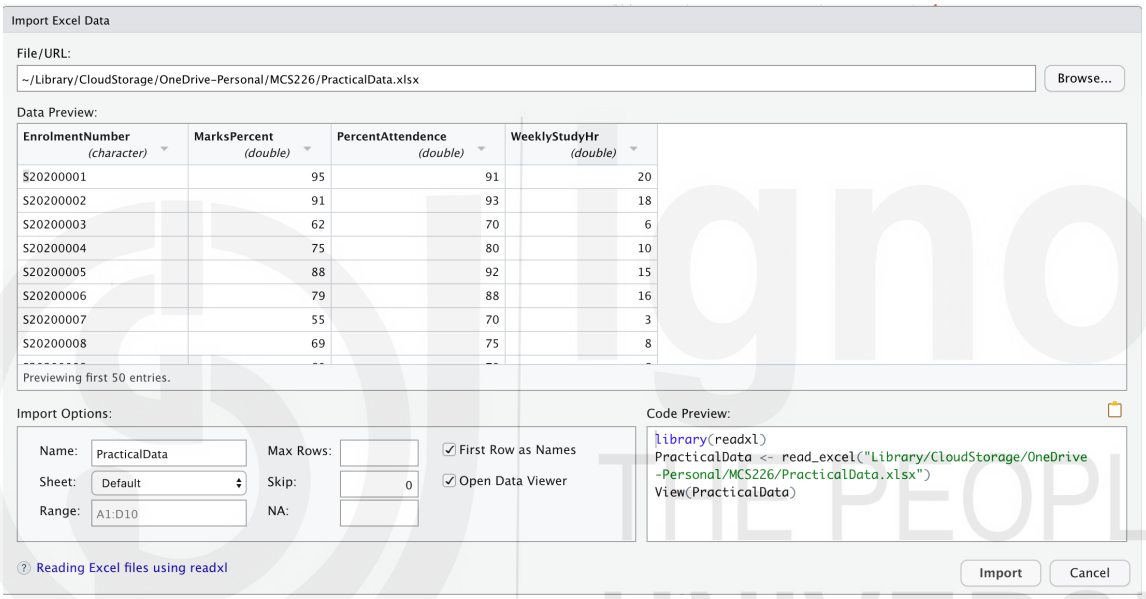



Figure 4: Importing Excel Data in RStudio Desktop

You may notice that you can see the data in Data Preview. The data type of the data is also shown. For example, MarksPercent will have double data type. In the Import option the option “First Rows as Name” is ticked. One of the important options in import options is “NA”, which relates to missing data. You may go through various options from the help files. After you press the Import button in Figure 4, the data will be imported in RStudio, as shown in Figure 5. Please notice the highlighted commands of R programming in the Console tab

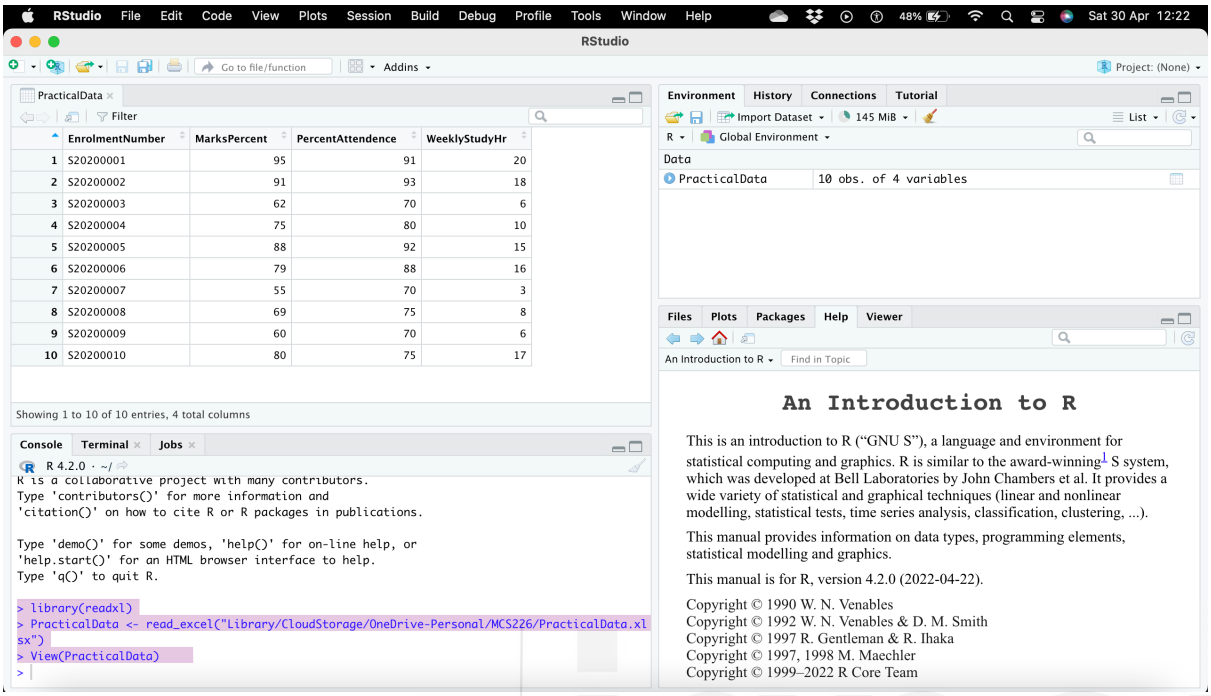



Figure 5: The R commands for importing data from excel

Now, let us do one more activity before you go to performing an analysis. Let us draw the plots.

To draw the plot it may be good idea to separate the data that you may need for plotting. This can be performed using the following R code:

```
xydata <- PracticalData[ , c('MarksPercent', 'PercentAttendence', 'WeeklyStudyHr')]
```

Next, you draw plot between attendance versus marks and study hours versus marks using the following code:

```
plot(x = xydata$PercentAttendence,y = xydata$MarksPercent,
    xlab = "Percentage of Attendence", ylab = "Final Percentage of Marks",
    xlim = c(70,100), ylim = c(50,100), main = "Attendence versus Marks Percentage"
  )
plot(x = xydata$WeeklyStudyHr,y = xydata$MarksPercent,
    xlab = "Weekly Hours of Study", ylab = "Final Percentage of Marks",
    xlim = c(1,20), ylim = c(50,100), main = "Study Hours versus Marks Percentage"
  )
```

Figure 6 shows these statements in R console.

RStudio File Edit Code View Plots Session Build Debug Profile Tools Wind

RStudio

Go to file/function | Addins

PracticalData | xydata

Filter

| | MarksPercent | PercentAttendence | WeeklyStudyHr |
|---|---|---|---|
| 1 | 95 | 91 | 20 |
| 2 | 91 | 93 | 18 |

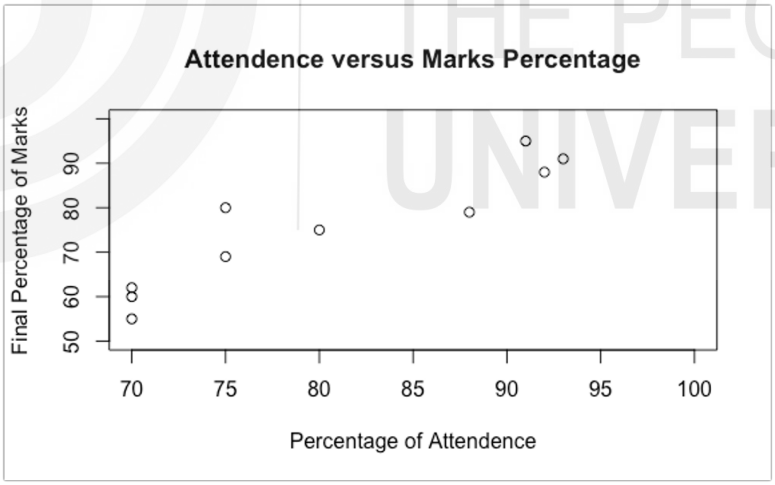
Showing 1 to 3 of 10 entries, 3 total columns

Console | Terminal | Jobs

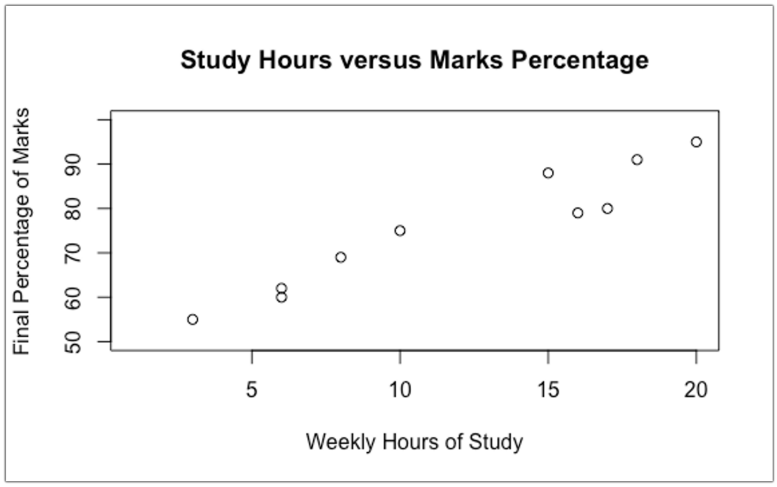
R 4.2.0 · ~/

```
>
> xydata <- PracticalData[ , c('MarksPercent', 'PercentAttendence', 'WeeklyStudyHr')]
>
>
> plot(x = xydata$PercentAttendence,y = xydata$MarksPercent,
+      xlab = "Percentage of Attendence", ylab = "Final Percentage of Marks",
+      xlim = c(70,100), ylim = c(50,100), main = "Attendence versus Marks Percentage"
+ )
>
>
> plot(x = xydata$WeeklyStudyHr,y = xydata$MarksPercent,
+      xlab = "Weekly Hours of Study", ylab = "Final Percentage of Marks",
+      xlim = c(1,20), ylim = c(50,100), main = "Study Hours versus Marks Percentage"
+ )
>
> View(xydata)
>
```

Figure 6: Extracting data and plotting Scatter plots




(a) Attendance versus Marks




(b) Hours versus Marks

Figure 7: Scatter Plots of data

Figure 7 suggests that there is possibility of relationship between marks of the students with the class attendance and hours of study. You can also plot pairs of these three variables together using the R code:

```
pairs(~ PercentAttendence+MarksPercent+WeeklyStudyHr, data = xydata,
      main = "Scatter Plot Matrix")
```

This will produce the following output:

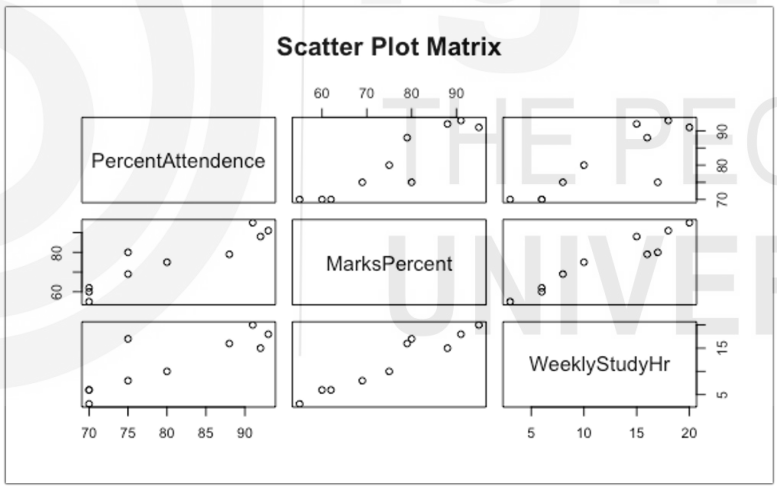



Figure 8: The Scatter plots for all the pairs

You may observe that the scatter plots of Figure 7 and Figure 8 shows possibilities of linear regression among the data. Therefore, let us use linear regression for making the model for this problem. In this case, we will use multiple linear regression, to demonstrate how multiple regression can be performed using R programming.

You need to define the response variable or dependent variable, which for the present model would be the marks percentage of the students. The two independent or exploratory variables being used for this model are attendance and study hours. Thus, the basic multiple linear regression proposed model is:

MarksPercent = a * PercentAttendence + b * WeeklyStudyHr + c

Please note that c is the intercept in the model given above. You need to determine the value of a, b and c using the multiple linear regression.

First, you create the data, as done earlier.

```
xydata <- PracticalData[ , c('MarksPercent', 'PercentAttendence', 'WeeklyStudyHr')]
summary(xydata)
```

The result of summary statistics of xydata is shown in Figure 9.

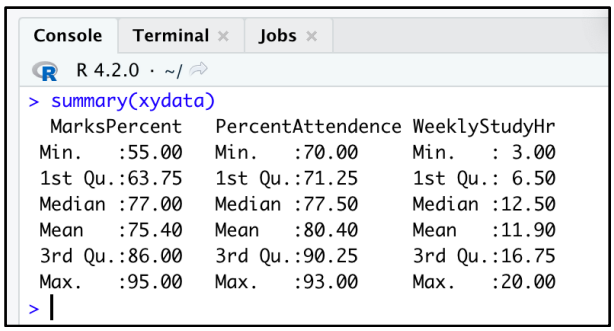



Figure 9: Statistical summary of xydata

You may notice that Figure 9 shows the summary of each of the column of xydata variable. This summary shows minimum value, the first quartile value, the median, the mean, 3rd quartile and then the maximum value of the maximum value of the data. You may please note that for all the three attributes the mean and median values are close, which represents that data may be of normal distribution.

Next, you may use lm() function of R programming (Please refer to Block 4 Course MCS226).

```
ParameterLMR = lm (MarksPercent ~ PercentAttendence+WeeklyStudyHr, data=xydata)
```

The statement given above generates the multiple regression model –

MarksPercent = a * PercentAttendence + b * WeeklyStudyHr + c

The model coefficients and other details of the model are stored in the variable ParameterLMR. To display the summary of this model, you use the summary function to display this variable, as shown in Figure 10.

```
> ParameterLMR = lm (MarksPercent ~ PercentAttendence+WeeklyStudyHr, data=xydata)
> summary(ParameterLMR)

Call:
lm(formula = MarksPercent ~ PercentAttendence + WeeklyStudyHr,
    data = xydata)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4962 -0.9043  0.4864  1.9889  2.5773

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        14.3005    12.7533   1.121  0.29914
PercentAttendence   0.5450     0.1955   2.788  0.02699 *
WeeklyStudyHr       1.4522     0.3165   4.589  0.00252 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.081 on 7 degrees of freedom
Multiple R-squared:  0.9608,    Adjusted R-squared:  0.9496
F-statistic: 85.73 on 2 and 7 DF,  p-value: 1.195e-05
```

Figure 10: Multiple Regression Result

The output first shows the value of residual, which measures the distance or difference between the actual value, also called observed value, and the predicted value by the regression model. The residuals are represented as 5-point summary. You can observe the minimum difference being about -6.5% and maximum being 2.58%. Next please observe the critical t-values and p-values (Pr) for all

the coefficients. It may be noted that the degree of freedom of this regression is 7 (also shown in the Figure 10). You may compute the critical t-values for this degree of freedom. Please note that the regression model is:

MarksPercent = 0.5450 * PercentAttendence + 1.4522 * WeeklyStudyHr + 14.3005

Next, please note that the probability (Pr) or p-value of Intercept is 0.29914 (refer to Figure 10), which is greater than 0.05, implying that the data is not providing statistically significant evidence against the null hypothesis, as shown in Figure 10. Please note that the null hypothesis for intercept is that intercept should be 0. However, the other two coefficients have the p-value as 0.027 and .0025, which are statistically significant and provides evidence that there is a significant relationship between the response variables and explanatory variables. The adjusted r-squared variable, which is normally used instead of r-squared value in case of multiple regression, shows that about 94.96% of variation of response variable can be explained by the exploratory or independent variables. However, it is not a measure of goodness of regression model. The p-value associated with the F-statistics for this regression model rejects the null hypothesis that all the coefficients in this regression model should be zero. However, due to a very high level of standard error in the intercept value leads us to situation, which requires you to explore better models.

You may try a linear regression between MarksPercent (response or dependent variable) and WeeklyStudyHr (explanatory or independent variable). The regression model for this would be:

MarksPercent = a * WeeklyStudyHr + b

You may use the following R code for performing the regression:

```
SingR = lm (MarksPercent ~ WeeklyStudyHr, data=xydata)
summary(SingR)
```

The result of these two operations would be:

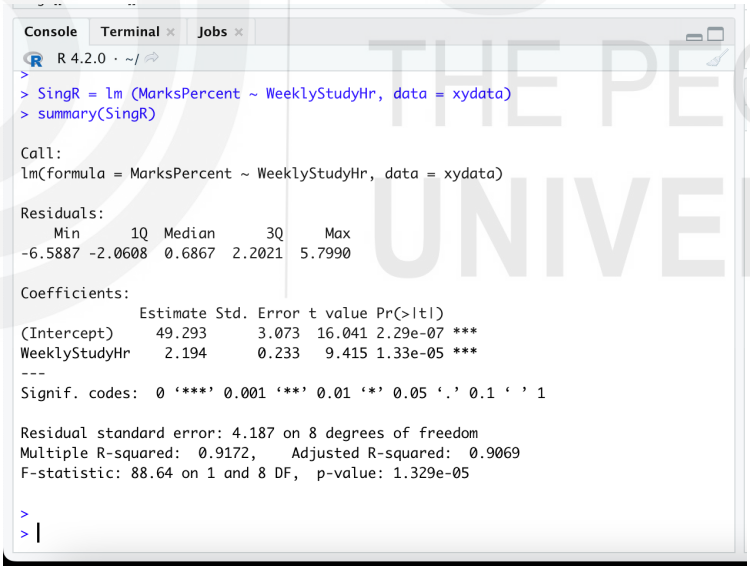



Figure 11: Result of Regression

You may observe the Pr values for both Intercept and explanatory variable is very low. In addition, the p-value for F-statistics is also very low. Thus, this regression model can be used to predict student marks. The model is represented by following linear regression equation:

MarksPercent = 2.194 * WeeklyStudyHr + 49.293

In adiition, you may also draw various plots related to regression using R. The following function call will show you various plots related to regression:

```
plot(SingR)
```

Some of these plots are shown in Figure 12.

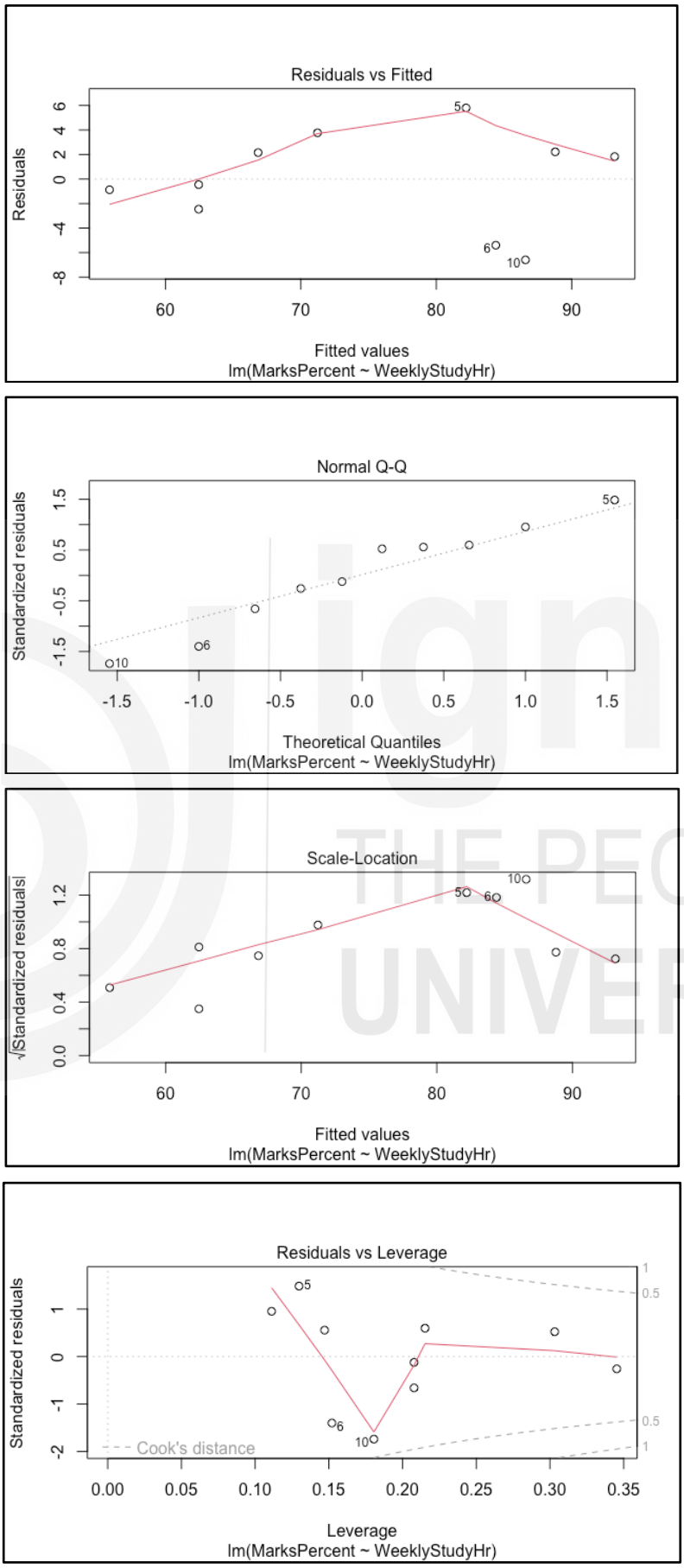



Figure 12: The plots of Linear regression

The plots in Figure 12 mainly represent characteristics of residuals in the fitted model. You may refer to the further readings for more details on these plots. However, you may notice that data items 6 and 10 are the ones, where you have maximum standard error. You may observe that both these students are putting large number of study hours, yet not able to score marks as high as the students who are putting almost similar number of hours of study.

In general, a data science project would require you to create, clean, prepare data, followed by exploratory data analysis, which will give you insights of data. This will lead you to final analysis of data and reporting your results.

In the next section, we present a list of problems, which you should attempt during your laboratory sessions.

## 2.4 SESSION WISE EXERCISES

In the previous section of this Unit, you have gone through the process of using RStudio for a simple regression analysis problem. RStudio is an ideal product for doing data science projects. As stated earlier, R programming can process a large number of data records. This practical session is aimed at introducing you to R programming and its use in data science. It does not attempt to make you an expert data scientist. You may go through contents on the web and start working on commercial projects to become an expert data scientist. The session wise list for various practical sessions for this section are given below:

**Session 1 to Session 4: Simple Exercises of using R.**

Question 1:

Create a vector of size 10, having the values 5,7,9,11,13,13,11,9,7,5. Compute the sum, mean, highest and lowest of these values. Compute the length of this vector? Find the variance and standard deviation for the data of this vector, using the formula for variance and standard deviation. Compare these values by computing the variance and standard deviation using R function. Sort this array values in decreasing order.

Question 2:

Create a vector of first 50 even numbers, starting from 2. Also create a vector having values 30 down to 1, as 30, 29, …,1

Question 3:

Create a vector of size 10 with $5^{th}$ and $7^{th}$ values as missing (store these values as NA). Use the “is.na()” to find locations of missing data.

Question 4:

Create a vector of characters of size 5, consisting of values: “This” “is” “a” “character” “vector”. Find the index of value “is” in the vector using which() or match().

Question 5:

It is always good to store numerical values rather than textual data. However, while input or output the textual values are easier to understand. An example, for this is as follows in R:

```
> Fivepointscale=c(1:5)
> names(Fivepointscale) = c("Not Satisfactory", "Satisfactory", "Fair", "Good", "Very Good")
> Feedback = Fivepointscale[c("Good", "Satisfactory")]
```

Create a 7-point scale of information input and use this scale to input feedback of 5 students about a question like “Feedback of experience of using an application (Bad, Somewhat bad,

not good, ok, good, very good, excellent)". Find the average of the feedback.

Question 6:

Create two strings and concatenate them.

Question 7:

Create a long string of words separated by punctuation marks. Replace all the punctuation marks in the string using gsub("[[:punct:]]", "", stringName) function. Find the number of words in the string without punctuation marks. Find the number of distinct words and its count, if possible.

Question 8:

Store content in external files for the following types of data in R:

(i) Vectors (ii) Lists (iii) Arrays (iv) Data frames (v) Factors

Read those contents into R. Perform operations link sorting on vector data, finding the length of lists and adding data items in list, accessing different elements of array and comparing it to other values, accessing different components of data frames and factors.

Question 9:

Create two matrices of 5*5 size using R, add, subtract and multiply these two matrices.

Question 10:

Perform transpose of a matrix.

Question 11:

Find the inverse of a matrix.

Question 12:

Create a list of a factor. Find the occurrences of each factor in the list.

Question 13:

Write function to find the largest and smallest values in a 3-dimensional array of size 3*3*3. You should use parameter passing.

Question 14:

Find the eigen values and eigen vectors of a symmetric matrix.

Question 15:

Create a table of showing the States of 20 students, assume these students stay in 5 different states. Now create a factor of these states and then compute the frequency of each factor (Hint: You may use factor() and tapply() functions)

Question 16:

Consider a state wise list of income of few persons. Use factor function to create a frequency division of income into 5 factor classes e.g. 10000-50000; 50000-100000 etc.

Question 17:

Explore different functions in R about strings, arrays, vectors, factors. You may also explore different methods of plotting the data.

**Session 5 and Session 6: Analysis Using R**

Question 18:

For the question 16, create another factor of State of these people, say these persons are residing in Just three states. Create a two-way frequency table of Income factor and State. Make suitable assumptions.

Question 19:

Find the details of all the vectors and other variables. Also find the data type of all variables. (Hint: use summary() and typeof(), you can also use stem().)

Question 20:

A class has a student strength of 50 students. The marks obtained (out of 100) by the students of the class are as per the binomial distribution. You should create the sample data of marks for the 50 students using binomial distribution. Convert these marks to grades as follows:

| | |
|---|---|
| <40 | D |
| =>40 but < 60 | C |
| =>60 but < 80 | B |
| => 80 | A |

Also, create random data for seriousness towards studies having the categories: Very Serious, Serious, Not Serious. Use chi-square testing to determine, if there is a relation between the seriousness towards learning to Grades of student, as per your data. Show and explain the results.

Question 21:

The marks of a class of 50 students are recorded as the final percentage of marks. Assuming that the percentage data is normally distributed. In addition, gender data is also stored. Create the data for the class and draw side by side box plot of Girls and Boys marks. Explain the output of the boxplots.

Question 22:

Install the inbuilt datasets available with R programming using

```
> install.packages("datasets.load")
```

Display the dataset airquality using the command

```
> datasets::airquality
```

Study the variables and observations of the data set. Remove all the observations of NA using any method and then draw a scatter plot between Ozone and Solar Radiation variables. Perform a linear regression analysis by between Ozone and Solar Radiation. Explain the selection of independent and dependent variables. Also, explain the output of the regression.

Question 23:

Using the airquality dataset, as given above, perform multiple regression of Ozone, Solar.R, Wind and Temp variables. Which of them has been selected as dependent variable and why? Explain your results.

Question24:

Study the data set of iris from the given dataset. Use first 100 records of this data set into a separately data frame. Create a logistic regression model first using just one variable (say using Sepal.Length) to answer the question “If the Species is Setosa or not?”. Test your model. Explain your results.

Question 25:

Use the dataset airquality, plot the date/month against the temperature. Also plot the moving average at a length of 3. Compare the two results.

**Session 7 and Session 8: More Analysis Using R**

Question 26:

Use the sample fictional data set given on the website:
https://www.kaggle.com/datasets/carlolepelaars/toy-dataset

Study the data set and read only 10,000 rows and all the columns of the dataset. The dataset has six variables - Number (row number), City, Gender, Age, Income and Illness. Make decision tree for determining the Illness. You must first present the summary of the dataset being used. Explain the resultant tree. Can the quality of decisions made by decision tree be enhanced? Justify your answer.

Question 27:

Use the same dataset as you generated above. Now use Random forest on the dataset, as given above. Explain the confusion matrix in detail.

Question 28:

Identify a dataset that has rainfall data along with various related factors like temperature, pressure, etc. Create first a decision tree to determine the possibility of rainfall or absence of it. Also, use random forest to do the same. Compare the results of decision tree and random forest.

Question 29:

Create or download sample data of customers of an e-commerce website. Consider it has factors like family income, total amount spent last month by the customer, Is subscriber of product review pages, etc. Classify the customers into the following categories:

High spenders, medium spenders, Low spenders. You may use any two classification algorithms/ techniques and compare the results of the two classifiers.

Question 30:

Assuming that the problem, as given above, does not have any categories. Perform k-mean clustering on the data with k=5.

Question 31:

Perform classification and clustering for easily available datasets.

**Session 9 and Session 10: Search and Implement online projects Using R**

For these two sessions, it is recommended that you participate in any two online projects of data science that are using R as programming language. Some of the suggested areas are sentiment analysis, fraud detection in Banks, data analysis of traffic, cab movements etc.

## 2.5 SUMMARY

This section has broadly covered different aspects of R programming. This section includes a brief discussion on installation of R programming and RStudio desktop both of which are open-source software. An example of use of R programming is also given. This example, has the component of reading data from a data file, plotting the data using R functions and performing the regression analysis, first as a multiple regression and later simple linear regression to explain various aspect of R programming that can be used to perform data analysis.

You should try to solve not only the problems given in these exercises, but also many other problems during your practical sessions. You should consult online help/reference manuals of R programming,

to enhance your skills in R programming. Remember, the best way to learn R programming is by trying different problems related to data science domains using R.

Finally, please keep in mind that this section will NOT make you an expert R Programmer and you may require to deal with some commercial project for that purpose. Such projects would cover the entire set of activities related to a data science project. In such projects, you should use the information and skills obtained from MCS-226 and this lab course.