Tutorial Using R (and RStudio)

1 INTRODUCTION TO STATISTICS

Statistics is the science of learning from data. It is the fields that deal with all aspect of data including the planning of data collection, data preparation, analysis and interpretation to make credible understanding or meaning from data in making more effective decisions. In general, there are two types of statistics:

- **Descriptive statistics**: statistics that summarize, organize and make sense of observations. In other word, descriptive statistics is concerned to describe certain features of a variable.
- **Inferential statistics**: statistics use data gathered from a sample to make inferences about the larger population from which the sample was drawn.

1.1 Basic terms

- **Population** consists of all items or elements of interest for a particular decision or investigation. Example: all married staff over the age of 25 in UTeM.
- **Sample** is a certain number of elements that have been chosen from a population. Sample is a subset of population. Example: a list of married staffs over the age of 25 in the Registrar's Office would be a sample from the population of all married staff over the age of 25 in the UTeM.
- **Variable** is a characteristic of the individual within the sample or population.
- **Observation/Measurement** is the value of a variable for an element.
- **Data set** is a collection of values of one or more variables.
- **Raw data** is data recorded in the sequence in which they are collected and before they are processed or ranked.
- **Outliers / Extreme Values** are values that are very small or very large relative to the majority of the values in a data set.

1.2 Variable

In statistics, each variable is required to be defined and categorized with the scales of measurement. Each scale of measurement has certain properties which in turn determine the appropriateness for use of certain statistical analyses. The four scales of measurement are nominal, ordinal, interval, and ratio.

- Nominal (Qualitative)

Categorical data and numbers that are simply used as identifiers or names represent a nominal scale of measurement. Each observation belongs to some distinctively different categories. It is cannot be quantified or ranked. (i.e. 1=Married, 2=Single ; M=Married, S=Single)

- Ordinal: (Quantitative)

An ordinal scale of measurement represents an ordered series of relationships or rank order. Fundamentally, these scales do not represent a measurable quantity. For example, individuals competing in a contest may be fortunate to achieve first, second, or third place. Likert-type scales (such as "On a scale of 1 to 10, with one being no pain and ten being high pain, how much pain are you in today?") also represent ordinal data.

- Interval: (Quantitative)

A scale that represents quantity and has equal units but for which zero represents simply an additional point of measurement is an interval scale. The Fahrenheit scale is a clear example of the interval scale of measurement. Thus, 60 degree Fahrenheit or -10 degrees Fahrenheit represent interval data. Measurement of Sea Level is another example of an interval scale. With each of these scales there are direct, measurable quantities with equality of units. In addition, zero does not represent the absolute lowest value. Rather, it is point on the scale with numbers both above and below it (for example, -10degrees Fahrenheit).

- Ratio: (Quantitative)

The ratio scale of measurement is similar to the interval scale in that it also represents quantity and has equality of units. However, this scale also has an absolute zero (no numbers exist below zero). Very often, physical measures will represent ratio data (for example, height and weight). If one is measuring the length of a piece of wood in centimeters, there is quantity, equal units, and that measure cannot go below zero centimeters. A negative length is not possible.

The table can be used to conduct descriptive statistics between the four scales of measurement

Table 1.1 Descriptive statistic

	Qualitative	Quantitative	
	[Nominal]	[Ordinal]	[Interval/ratio]
Definition	Unordered Categories	Ordered Categories	Numeric Values
Examples	State Code, Gender	Satisfaction Level, Age Group	Income, Age
Central Tendency	Mode	Mode / Median	Mode / Median / Mean
Dispersion		Min / Max / IQR / Range	Min / Max / IQR / Range / Std. Dev
Graph	Pie / Bar	Pie / Bar	Histogram (with normal curve)

2 INTRODUCTION TO R

R is widely known as a free software programming language and software environment for statistical computing and graphics. It provides a wide variety of statistical analysis such as linear and nonlinear modelling, time-series analysis, classical statistical tests, and graphical techniques which is highly extensible. Likewise, R is also an integrated suite of software which facilities data manipulation, calculation and graphical display. The functionalities of R includes an effective data handling and storage facility, a suite of operators for calculations, integrated collection of intermediate tools for data analysis, graphical facilities for data analysis as well as a well-developed, simple yet effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

There are around eight packages supplied with the R. Those packages are available through the Comprehensive R Archives Network (CRAN) sites, http://cran.stat.nus.edu.sg. The functionalities are quite similar as Minitab, SPSS and other statistical software that we can find in the market. However, this R is almost similar like a Matlab which it allows us to do programming in order to do computation and generate graphical view (graph).

2.1 Motivation to Use R

R is the most preferable software because of:

- It is free
- Leading developers of statistical software are writing functions for this package. Thus competency in R means we can stay up to date with statistics.
- It covers statistical applications from the simplest to the complex and would allow us to complete all our statistical training using R.
- R covers different topics in a consistent way so that the programming we learn for say linear models will also be done the same way for non-linear models. This consistency is convenient to statistical modelling.
- R has a powerful suite of functions that allow you to use modern statistical methods. Modern statistics has simplified many problems through the use of graphics and computer intensive ideas.

- It involves the biggest concentration of statisticians worldwide. So it can be accessed to the best and most efficient methods used by R clients worldwide.

2.2 The background of R package

R is an open-source package for statistical computing. “Open source” has a number of different meanings; here the important one is that R is freely available, and its users are free to see how it is written, and to improve it. R is based on the computer language S, developed by John Chambers and others at Bell Laboratories in 1976. In 1993 Robert Gentleman and Ross Ihaka at the University of Auckland wanted to do experiment with the language, so they developed an implementation, and named it R. They made it open source in 1995, and hundreds of people around the world have contributed to its development until today.
S-PLUS is a commercial implementation of the S language. Because both R and S-PLUS are based on the S language, much of what is described in what follows will apply without change to S-PLUS. However, other statistical packages such as S-PLUS, SAS, SPSS are not free. These are the licensed software and we have to buy in the market.

2.3 Installation of R

R can be downloaded from http://www.r-project.org/ by following these steps:

- Browse http://www.r-project.org/
- Click download R
- Go to Singapore website => http://cran.stat.nus.edu.sg/. The software can be obtained from National University of Singapore
- We can download the software depends on our operating system
- Download R for Linux or Download R for (Mac) OS X or Download R for Windows
- Go to install R for the first time.
- We can read the available information and most frequent asked questions in the website.

- Click Download R-3.1.0 for Windows and save into our hard disk (will take some time as the file is big, 54MB for 32/64 bit machines).
- Run R-3.1.0 - window installation file
- Follow all instruction until finish.

2.4 Starting and Quitting R

2.4.1 Syntax of R

In Microsoft Windows, the R installer will have created a start menu item and an icon for R on our desktop. Double clicking on the R icon to start the program.

From the start menu we can go to All Programs => R => R.

First thing that will happen is that R will open the *console*, into which the user can type any commands. The greater-than sign (>) is the prompt symbol. When this appears, you can begin typing commands.

2.4.2 Simple mathematical calculation

R can be used as a calculator. For example, Simple arithmetic expression:

```
> 5 + 49
```

Upon pressing the Enter key, the result 54 appears, prefixed by the number 1 in square brackets:

```
> 5 + 49
[1] 54
```

The [1] indicates that this is the first (and in this case only) result from the command

The sequence of integers from 1 to 100 may be displayed as follows:

```
> 1:100
```

You can set the width for better view of data.

```
> options(width=40)
> 1:100
```

Multiplication symbol is "*"

```
>3*5
[1] 15
```

Comment

Sign "#" is used for any remark and comment. Everything following a "#" sign is assumed to be a ignored by R.

```
>  # type any comment and the comment is ignored by R
```

More calculation

```
>3-8
[1] -5
> 12 / 4
[1] 3
> 13 / 5
[1] 2.6
```

2.4.3 Quit From R

To quit your R session, type

```
> q()
```

Note what happens if you omit the parentheses () when attempting to quit:

```
>q
function (save = "default", status = 0, runLast = TRUE)
.Internal(quit(save, status, runLast))
<environment: namespace:base>
```

This has happened because q is a *function* that is used to tell R to quit. Typing q by itself tells R to show us the (not very pleasant-looking) contents of the function q. By typing q(), we are telling R to *call* the function q. The action of this function is to quit R. *Everything* that R does is done through calls to functions.

3 Introduction to RStudio

The RStudio IDE is a user interface for R. It's free and open source, and works on Windows, Mac, and Linux. RStudio is not a replacement for R. R must be installed in addition to RStudio. However, when both are installed on your system, only RStudio needs to be run.

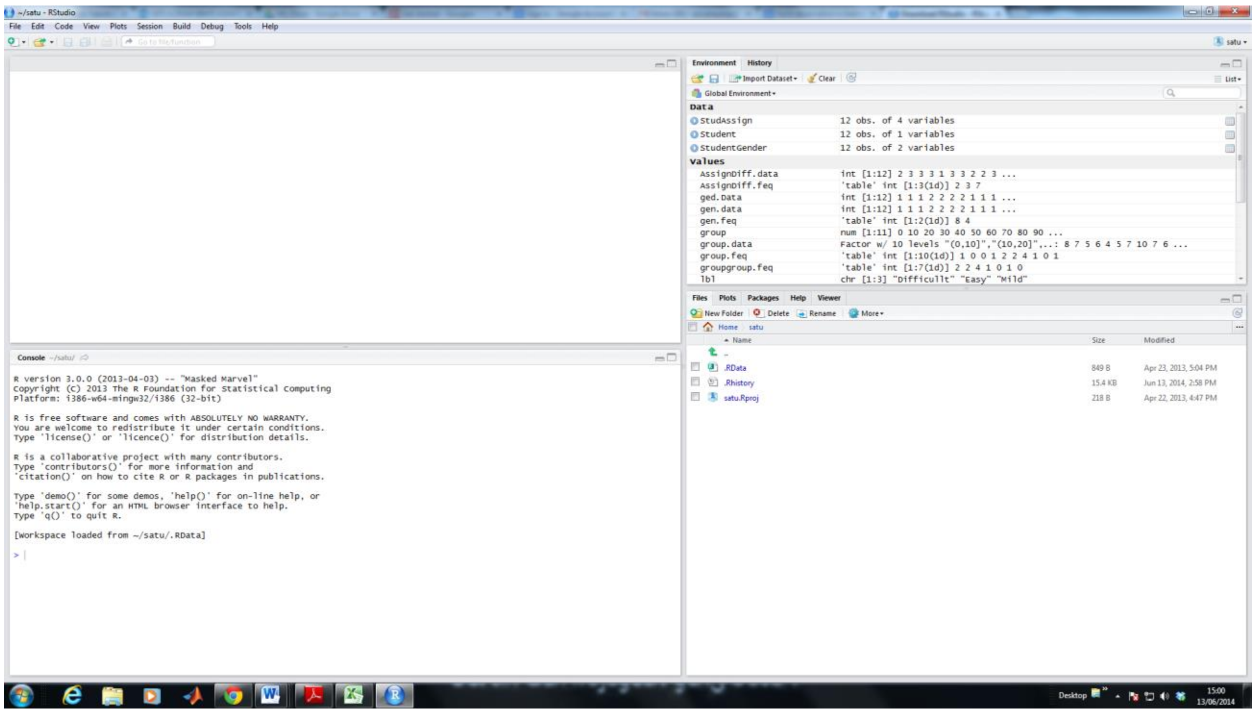
R Studio installation.

R Studio installation is straightforward as follows:

i) Go to http://www.rstudio.com/
ii) Go to Powerful IDE for R
iii) Click download now
iv) If you run R on your desktop, click

v) Download RStudio 0.98.507 - Windows XP/Vista/7/8 35.4Mb (depends on your OS)
vi) Run the installation file until finish.

Start > All programs > RStudio>RStudio


~/satu - RStudio
File Edit Code View Plots Session Build Debug Tools Help
satu
Environment History
Import Dataset
Clear
List
Global Environment
Files Plots Packages Help Viewer
New Folder Delete Rename More
Home satu
Console ~/satu/

```
Data
StudAssign        12 obs. of 4 variables
Student           12 obs. of 1 variables
StudentGender     12 obs. of 2 variables
Values
AssignDiff.data   int [1:12] 2 3 3 3 1 3 3 2 2 3 ...
AssignDiff.feq    'table' int [1:3(1d)] 2 3 7
ged.Data          int [1:12] 1 1 1 2 2 2 2 1 1 1 ...
gen.data          int [1:12] 1 1 1 2 2 2 2 1 1 1 ...
gen.feq           'table' int [1:2(1d)] 8 4
group             num [1:11] 0 10 20 30 40 50 60 70 80 90 ...
group.data        Factor w/ 10 levels "(0,10]","(10,20]",..: 8 7 5 6 4 5 7 10 7 6 ...
group.feq         'table' int [1:10(1d)] 1 0 0 1 2 2 4 1 0 1
groupgroup.feq    'table' int [1:7(1d)] 2 2 4 1 0 1 0
lbl               chr [1:3] "Difficullt" "Easy" "Mild"
```

Name	Size	Modified
.RData	849 B	Apr 23, 2013, 5:04 PM
.Rhistory	15.4 KB	Jun 13, 2014, 2:58 PM
satu.Rproj	218 B	Apr 22, 2013, 4:47 PM

```
R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/satu/.RData]

> 
```

3.1 RStudio layout

The RStudio interface consists of several main windows sitting below a top-level toolbar and menu bar. Although this placement can be adjusted, the default layout utilizes four main panels or panes in the following positions:

Position	Panels	Function
Top left	Source window	for editing files or viewing some data sets.
Bottom left	Console window	The console is where you can type commands and see output
Top right	Workspace/history window	to hold a Workspace browser and History

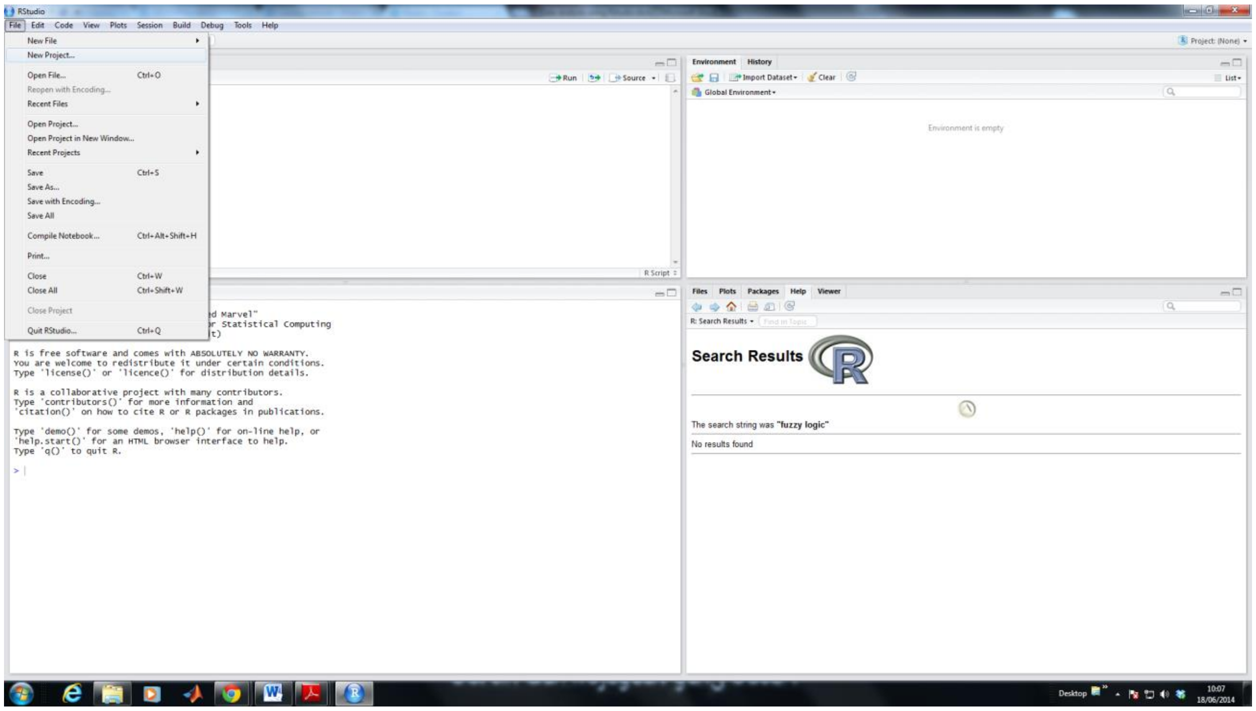
		browser. The workspace tab stores any object, value, function or anything you create during your R session. In the example below, if you click on the dotted squares you can see the data on a screen to the left. The history tab keeps a record of all previous commands. It helps when testing and running processes. Here you can either save the whole list or you can select the commands you want and send them to an R script to keep track of your work.
Bottom right	Files/plot/packages/help window	to hold tabs for interacting with the Files, Plots, Packages, and Help system components .

3.2 Changing the working directory

If you have different projects you can change the working directory for that session, see above. Or you can type:

```
# Shows the working directory (wd)

getwd()

# Changes the wd

setwd("C:/myfolder/data")
```

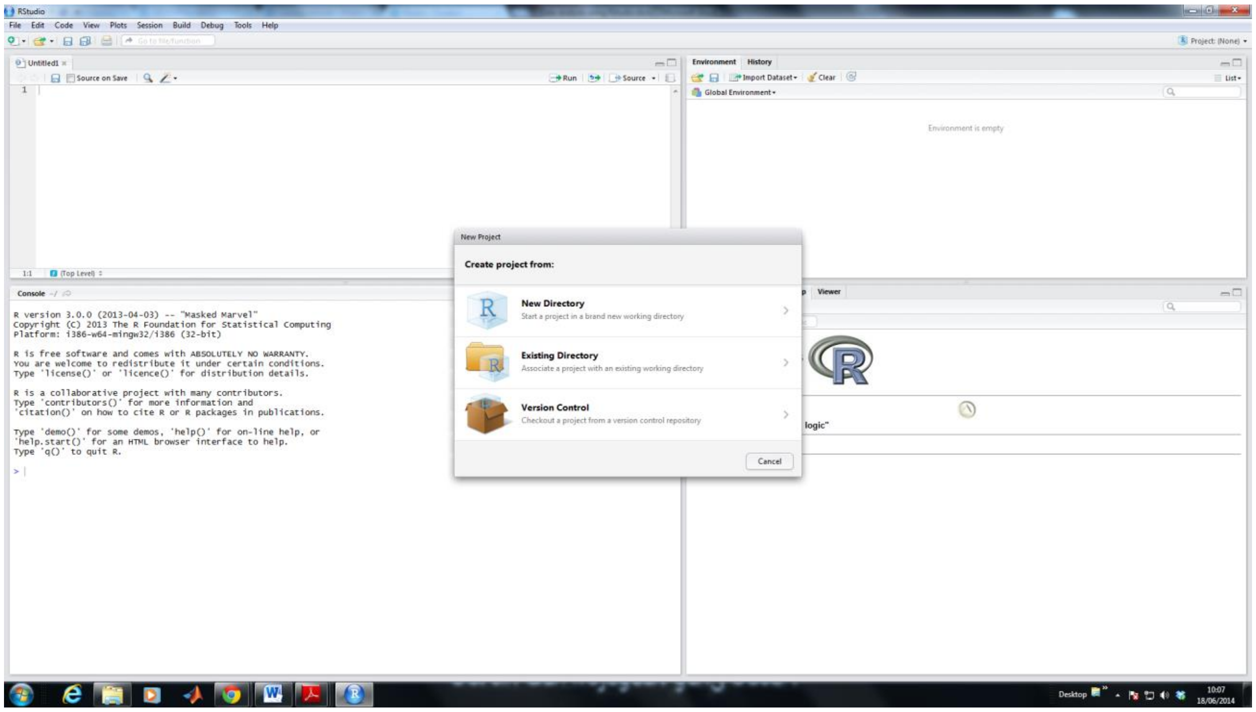
3.3 Starting new project



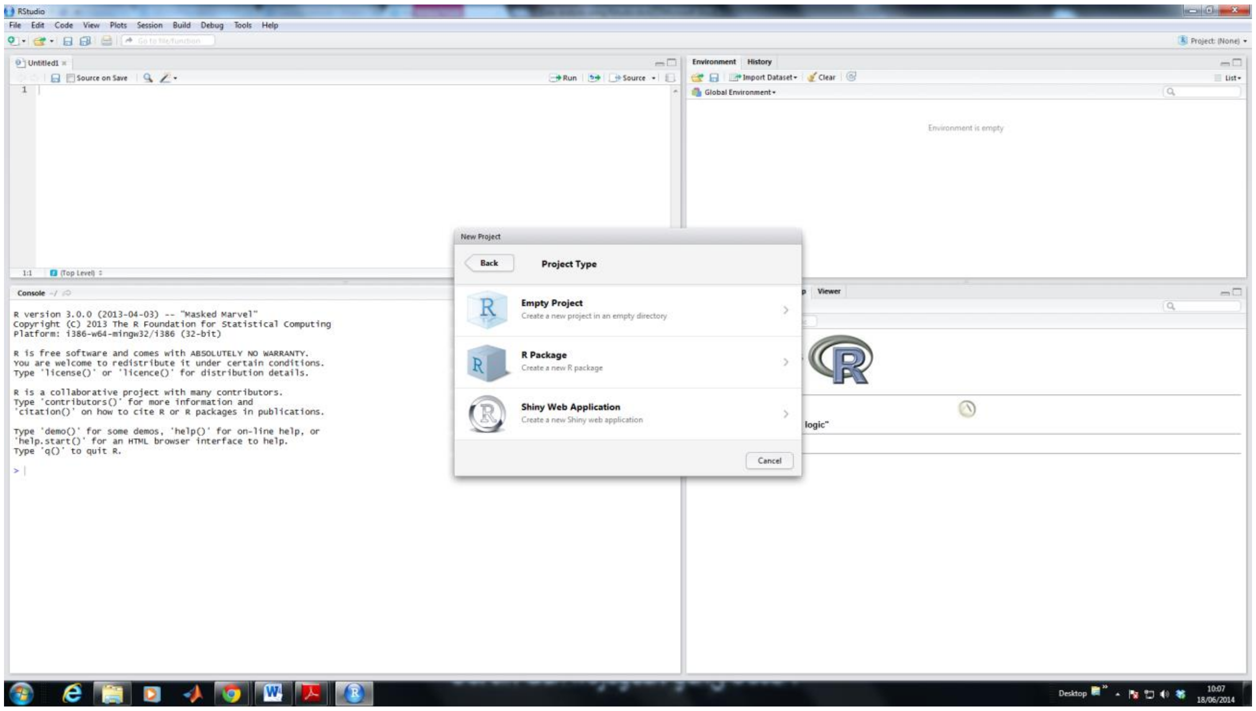
```
ed Marvel"
r Statistical Computing
t)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

The search string was "fuzzy logic"

No results found

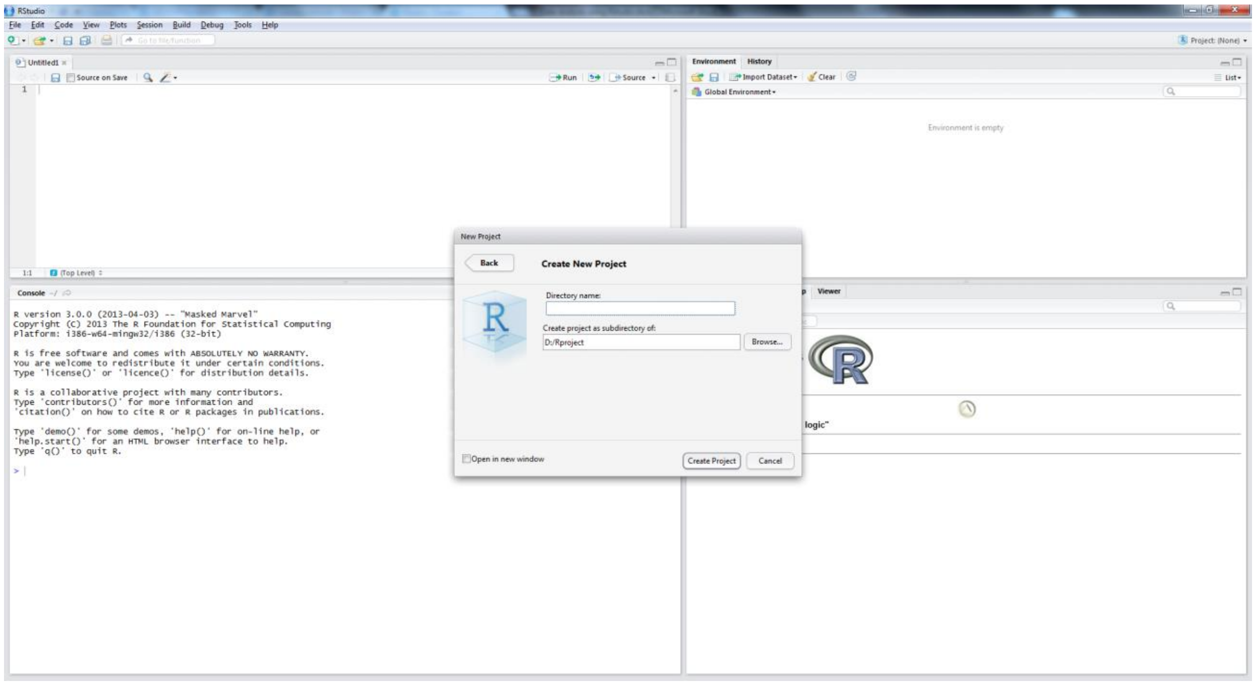


```
R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```



```
R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Click > Empty Project

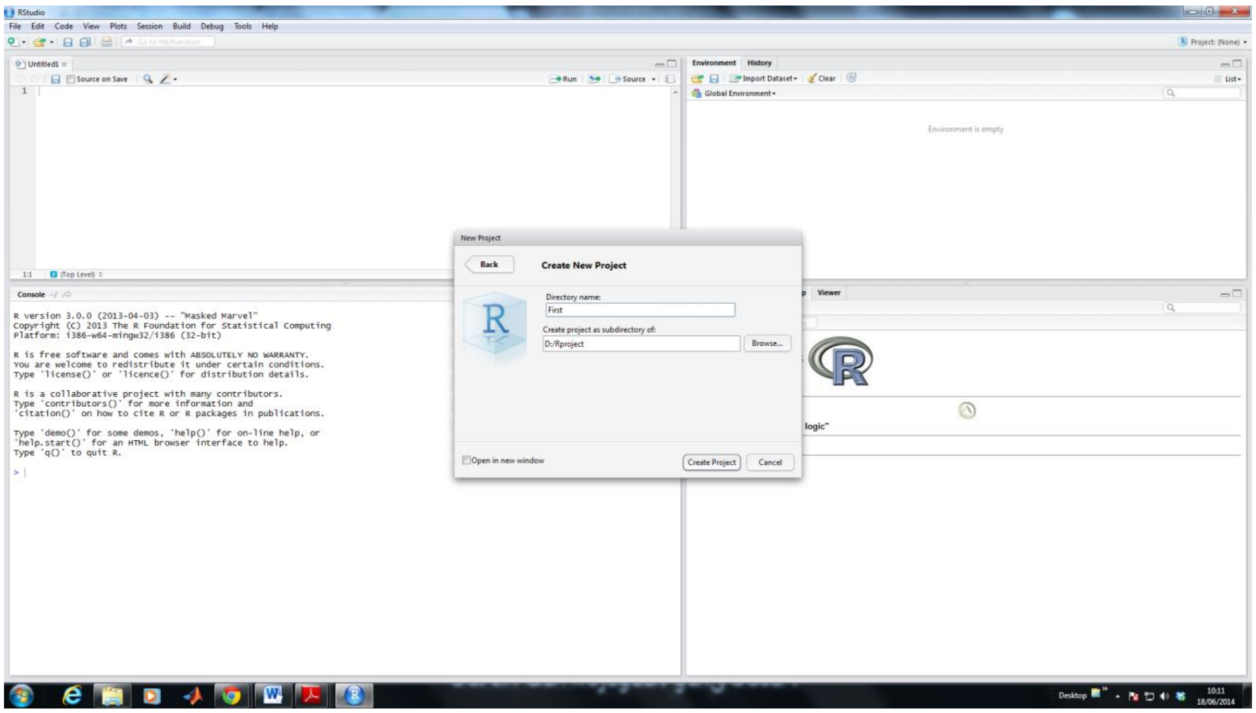


Directory name:

Create project as subdirectory of: D:/Rproject

Open in new window

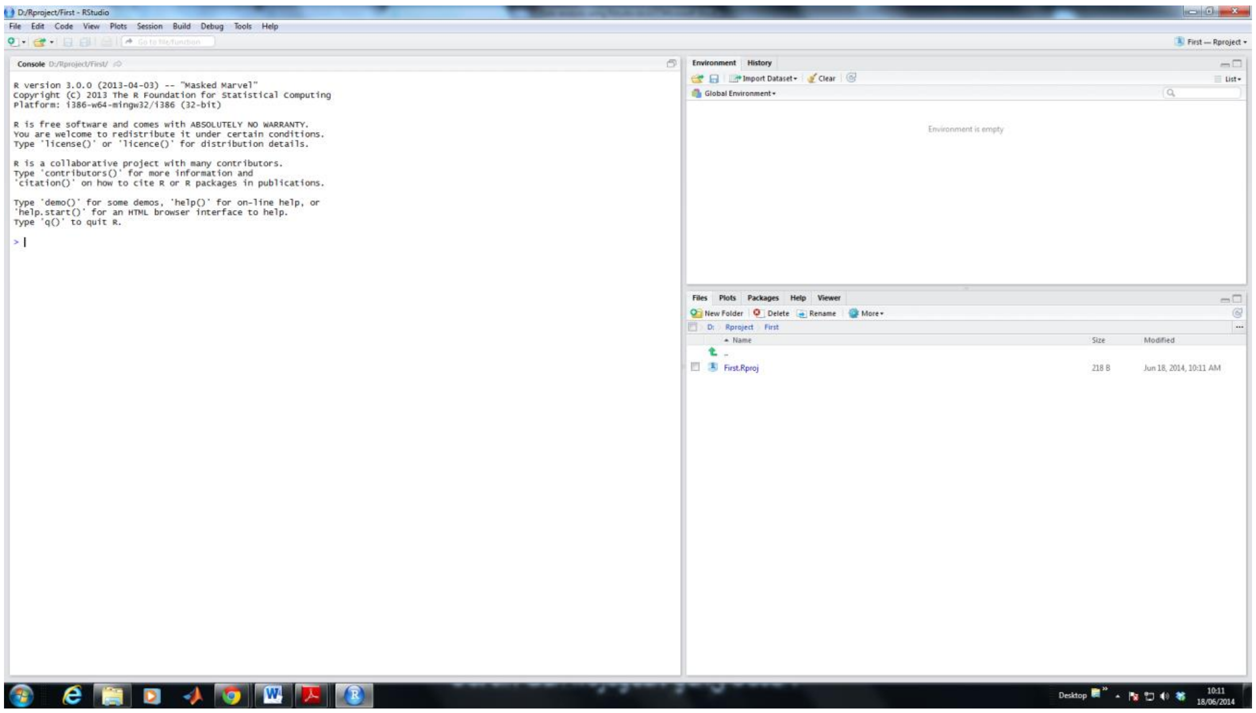
```
R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```



Directory name: First

Create project as subdirectory of: D:/Rproject

```
R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```



```
R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

Name	Size	Modified
..		
First.Rproj	218 B	Jun 18, 2014, 10:11 AM

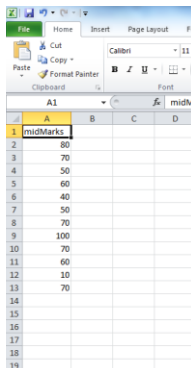
3.4 Entering Data & Analysis

The easiest way to enter data in R is to work with a text file, in which the columns are separated by tabs. There are many ways in which you can get such files. Here we simply explain how to make such a file with Excel. Open Excel and enter the data.

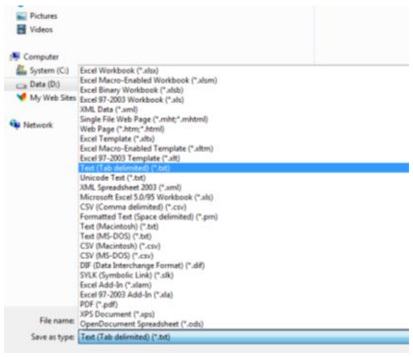
3.4.1 Quantitative data

Example 1: Student‘s marks

After enter the data in worksheet in Excel, give the file a name (e.g. student) and select the option Text (Tab delimited). Now save the file.

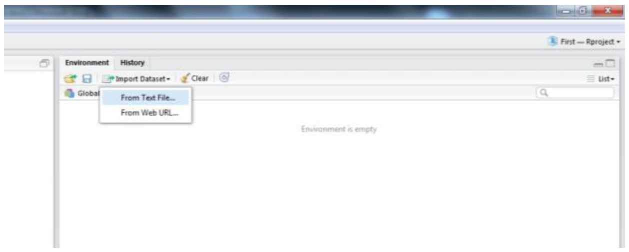


	A	B	C	D
1	midMarks			
2	80			
3	70			
4	50			
5	60			
6	40			
7	50			
8	70			
9	100			
10	70			
11	60			
12	10			
13	70			
14				
15				
16				
17				
18				
19				

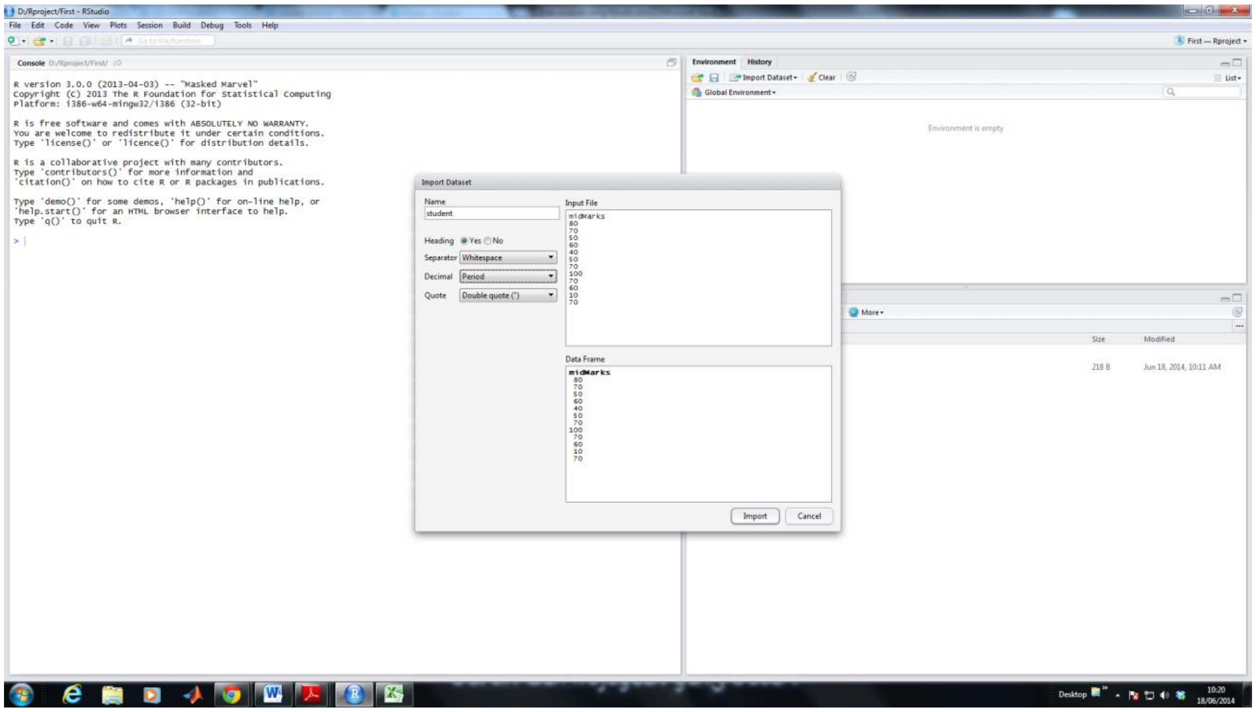

Pictures
Videos
Computer
System (C:)
Data (D:)
My Web Sites
Network
Excel Workbook (*.xlsx)
Excel Macro-Enabled Workbook (*.xlsm)
Excel Binary Workbook (*.xlsb)
Excel 97-2003 Workbook (*.xls)
XML Data (*.xml)
Single File Web Page (*.mht;*.mhtml)
Web Page (*.htm;*.html)
Excel Template (*.xltx)
Excel Macro-Enabled Template (*.xltm)
Excel 97-2003 Template (*.xlt)
Text (Tab delimited) (*.txt)
Unicode Text (*.txt)
XML Spreadsheet 2003 (*.xml)
Microsoft Excel 5.0/95 Workbook (*.xls)
CSV (Comma delimited) (*.csv)
Formatted Text (Space delimited) (*.prn)
Text (Macintosh) (*.txt)
Text (MS-DOS) (*.txt)
CSV (Macintosh) (*.csv)
CSV (MS-DOS) (*.csv)
DIF (Data Interchange Format) (*.dif)
SYLK (Symbolic Link) (*.slk)
Excel Add-In (*.xlam)
Excel 97-2003 Add-In (*.xla)
PDF (*.pdf)
XPS Document (*.xps)
OpenDocument Spreadsheet (*.ods)

File name:
Save as type: Text (Tab delimited) (*.txt)

To enter the data in RStudio you can use the command **Import Dataset** from Text File:



Find the student.txt file and open it. Import the data as shown in the following screenshot:


D:/Rproject/First - RStudio
File Edit Code View Plots Session Build Debug Tools Help
Console D:/Rproject/First/
Environment History
Import Dataset
Clear
Global Environment
Environment is empty
More
Size Modified
218 B Jun 18, 2014, 10:11 AM
Import Dataset
Import
Cancel
First — Rproject

```
R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

Name: student

Heading: [✓ Yes] No

Separator: Whitespace

Decimal: Period

Quote: Double quote (")

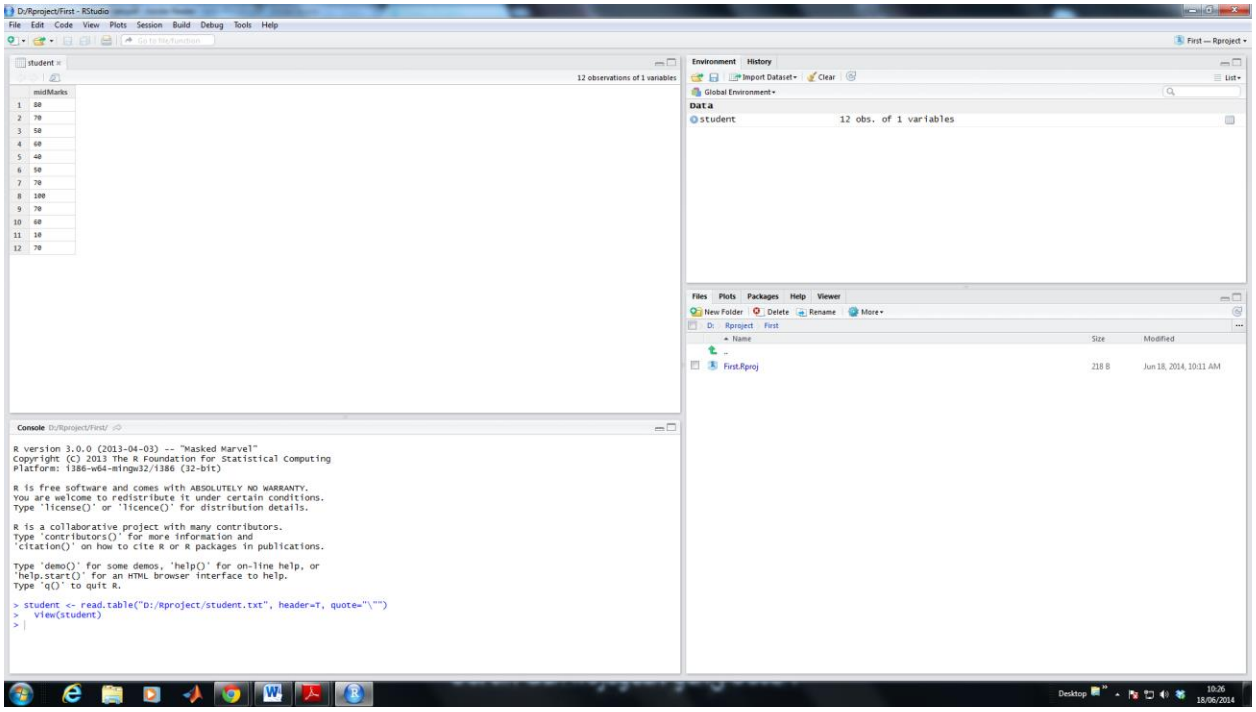
Input File

```
midMarks
80
70
50
60
40
50
70
100
70
60
10
70
```

Data Frame

```
midMarks
  80
  70
  50
  60
  40
  50
  70
 100
  70
  60
  10
  70
```

Now you have defined a dataset in R, called student, containing the variable midMarks as shown in source window (upper left panel).



12 observations of 1 variables

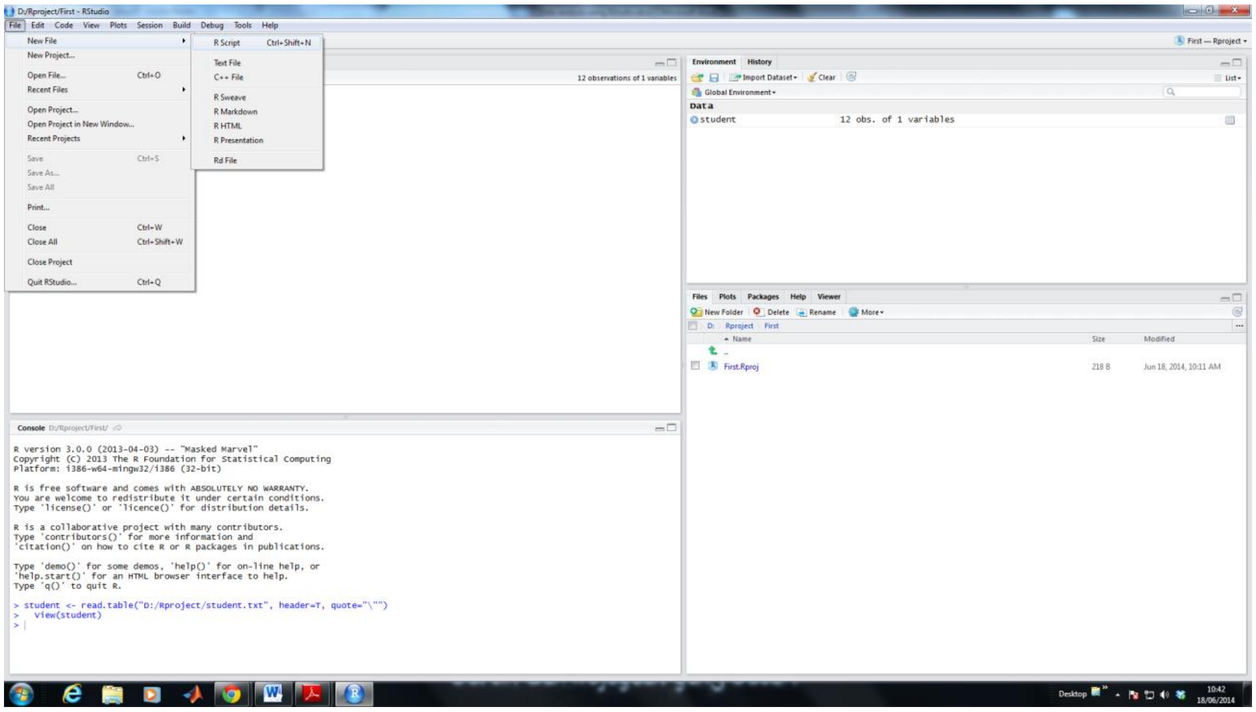
	midMarks
1	80
2	70
3	50
4	60
5	40
6	50
7	70
8	100
9	70
10	60
11	10
12	70

```
Data
student                12 obs. of 1 variables
```

```
R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> student <- read.table("D:/Rproject/student.txt", header=T, quote="\"")
>   View(student)
> |
```

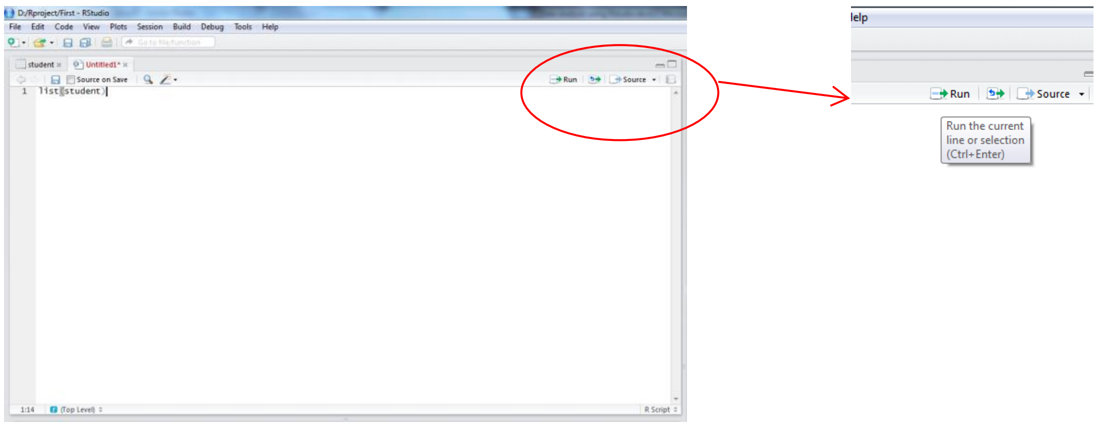
3.5 Simple Statistical Analysis

To start analyzing, create the workspace with a new Rscript file (click on the option File or create icon in the menu bar)



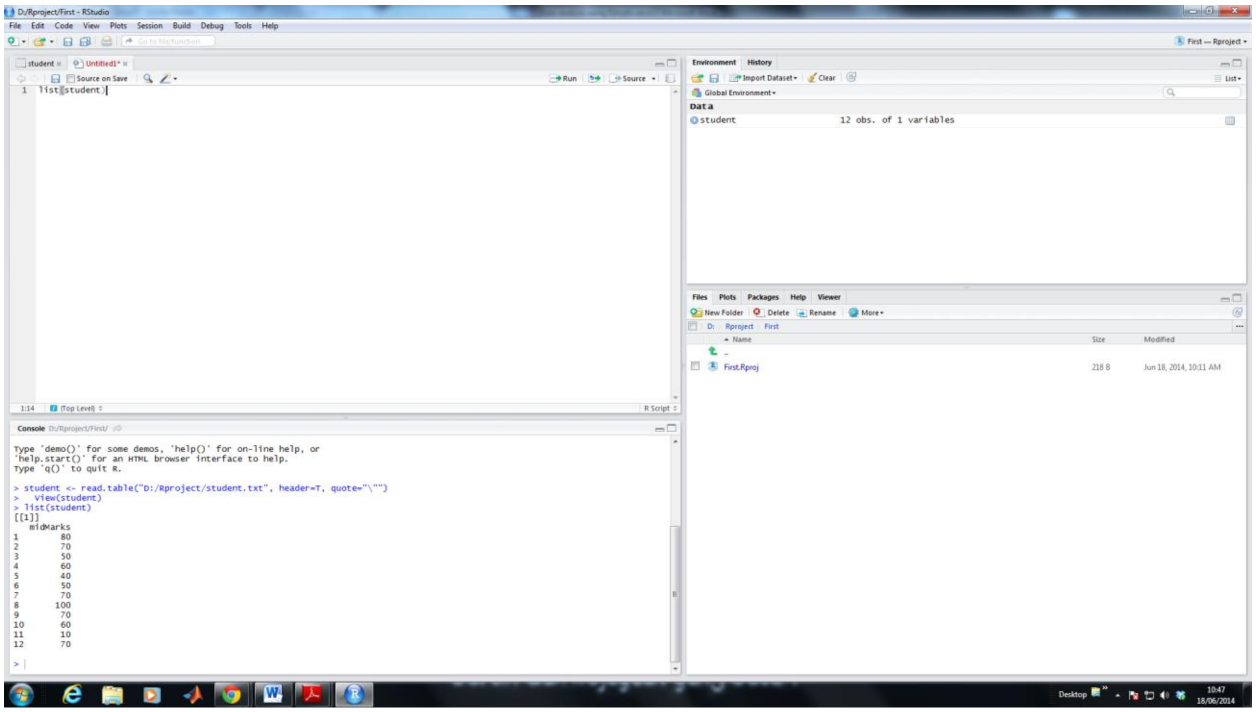
```
R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> student <- read.table("D:/Rproject/student.txt", header=T, quote="\"")
>   View(student)
> 
```

To make sure you have entered your data correctly, type the command `list (student)`, activate it (by going over it with the mouse, while you keep the left button pressed and click on Run Line(s).



```
list(student)
```

This should show you the data of the `midmarks` in the lower left panel.



```
list(student)
```

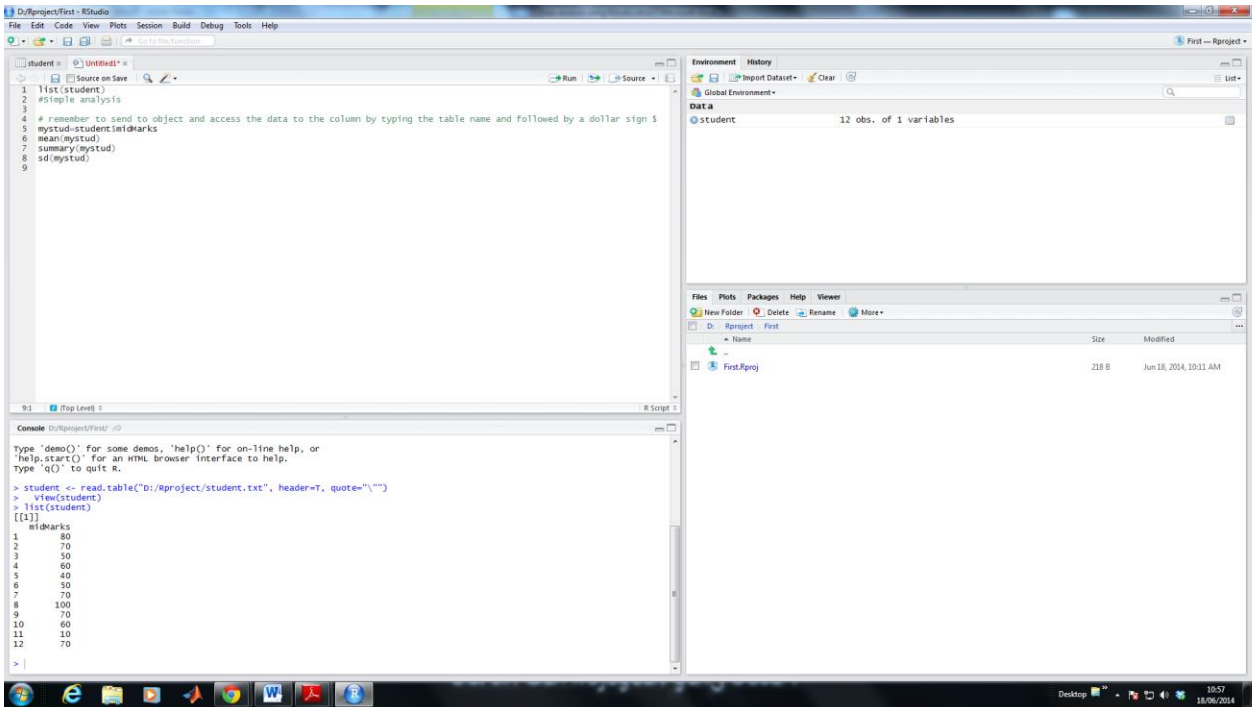
```
student              12 obs. of 1 variables
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> student <- read.table("D:/Rproject/student.txt", header=T, quote="\"")
>   View(student)
> list(student)
[[1]]
   midMarks
1        80
2        70
3        50
4        60
5        40
6        50
7        70
8       100
9        70
10       60
11       10
12       70

> |
```

3.5.1 Running simple analysis

Two of the most common statistics are the mean and standard deviation:

Mean : $$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$ Variance : $$s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2$$ Standard Deviation : = square root Variance $\sqrt{S^2}$	`#Simple statistic`
	
	`## remember to send to object and access the data to the column by typing the table name and followed by a dollar sign $`
	
	`Mystud=Student$Mid_marks`
	`mean(mystud)`
	`summary(mystud)`
	`sd(mystud)`


D:/Rproject/First - RStudio
File Edit Code View Plots Session Build Debug Tools Help
student
Untitled1*
Source on Save
Run
Source
Environment History
Import Dataset
Clear
Global Environment
Files Plots Packages Help Viewer
New Folder Delete Rename More
D: Rproject First
Name Size Modified
First.Rproj 218 B Jun 18, 2014, 10:11 AM
(Top Level)
R Script
Console D:/Rproject/First/
First — Rproject
Desktop
10:57
18/06/2014

```
list(student)
#Simple analysis

# remember to send to object and access the data to the column by typing the table name and followed by a dollar sign $
mystud=student$midMarks
mean(mystud)
summary(mystud)
sd(mystud)
```

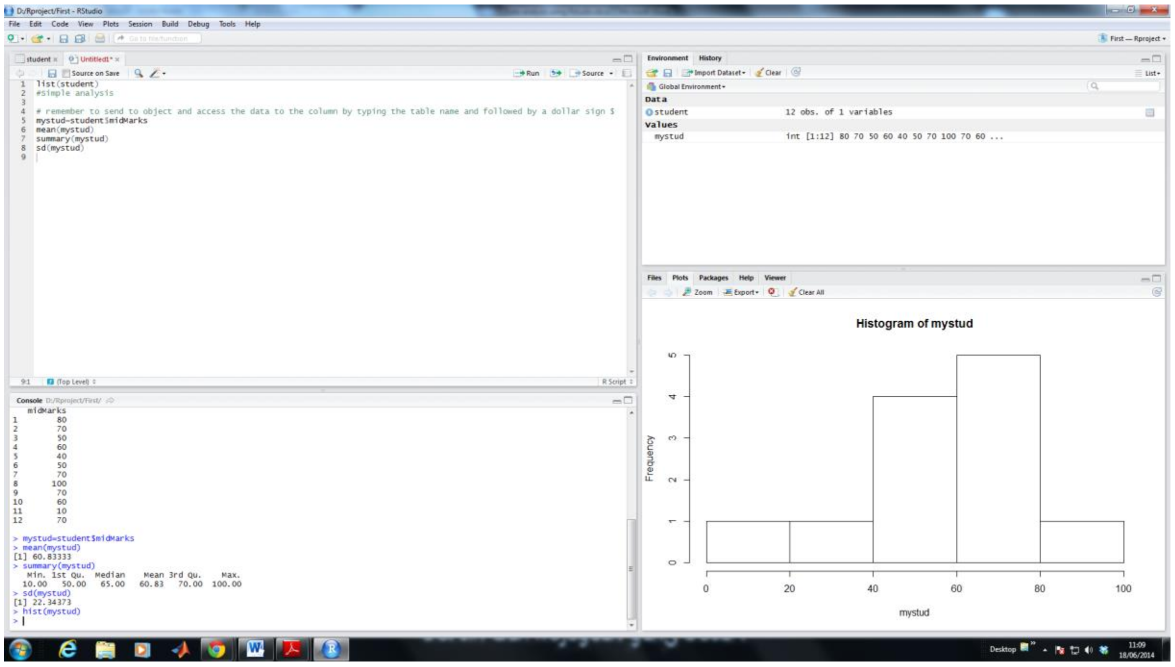
```
Data
student                12 obs. of 1 variables
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> student <- read.table("D:/Rproject/student.txt", header=T, quote="\"")
>   View(student)
> list(student)
[[1]]
   midMarks
1        80
2        70
3        50
4        60
5        40
6        50
7        70
8       100
9        70
10       60
11       10
12       70

> 
```

3.5.2 Creating the graph

Histogram

```
> hist(mystud)
```



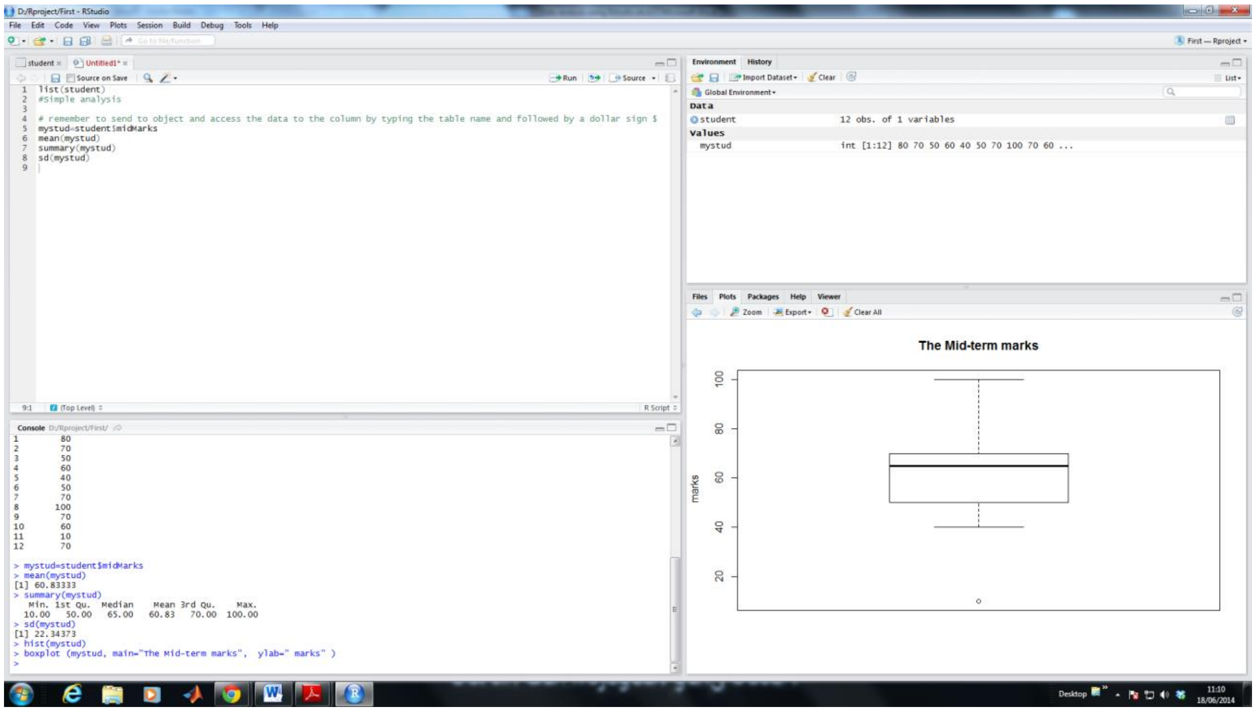
```
list(student)
#Simple analysis

# remember to send to object and access the data to the column by typing the table name and followed by a dollar sign $
mystud=student$midMarks
mean(mystud)
summary(mystud)
sd(mystud)
```

```
Data
student                12 obs. of 1 variables
Values
  mystud               int [1:12] 80 70 50 60 40 50 70 100 70 60 ...
```

```
   midMarks
1        80
2        70
3        50
4        60
5        40
6        50
7        70
8       100
9        70
10       60
11       10
12       70

> mystud=student$midMarks
> mean(mystud)
[1] 60.83333
> summary(mystud)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  10.00   50.00   65.00   60.83   70.00  100.00 
> sd(mystud)
[1] 22.34373
> hist(mystud)
> |
```

Boxplot

```
>boxplot (mystud, main="The Mid-term marks",  ylab=" marks" )
```



```
list(student)
#Simple analysis

# remember to send to object and access the data to the column by typing the table name and followed by a dollar sign $
mystud=student$midMarks
mean(mystud)
summary(mystud)
sd(mystud)
```

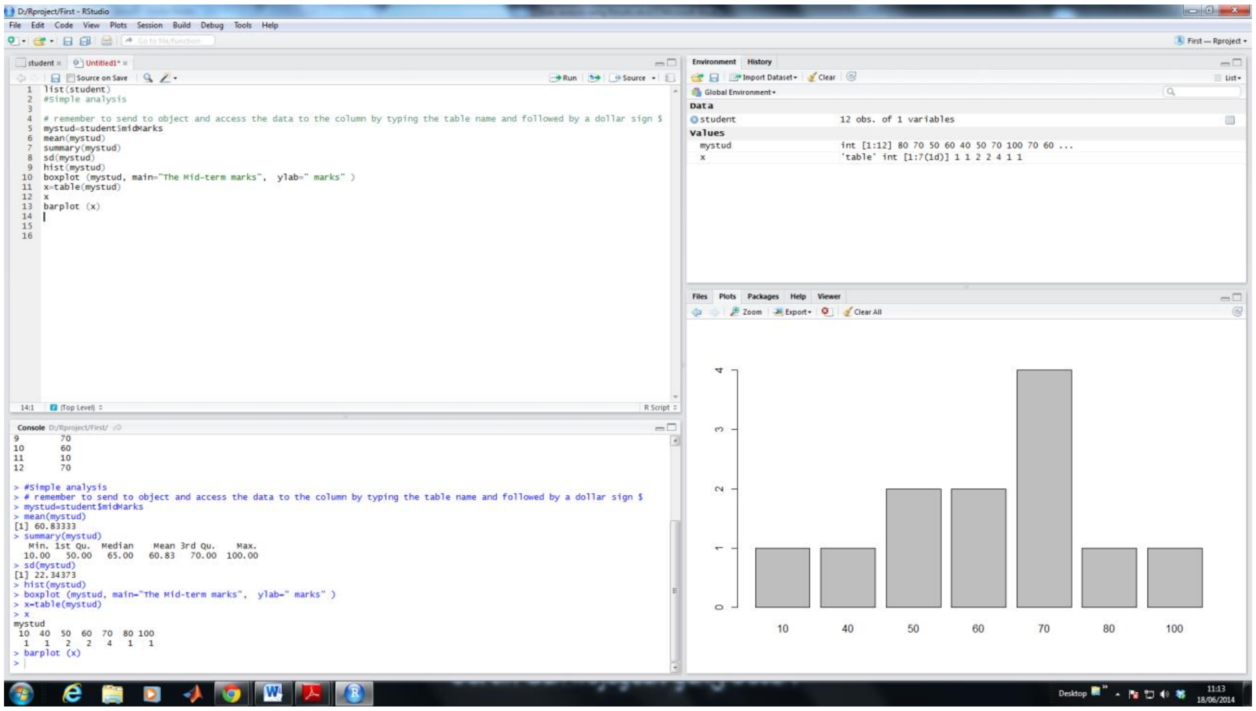
```
Data
student              12 obs. of 1 variables
Values
  mystud             int [1:12] 80 70 50 60 40 50 70 100 70 60 ...
```

```
1        80
2        70
3        50
4        60
5        40
6        50
7        70
8       100
9        70
10       60
11       10
12       70

> mystud=student$midMarks
> mean(mystud)
[1] 60.83333
> summary(mystud)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  10.00   50.00   65.00   60.83   70.00  100.00
> sd(mystud)
[1] 22.34373
> hist(mystud)
> boxplot (mystud, main="The Mid-term marks",  ylab=" marks" )
>
```

4 DESCRIPTIVE STATISTICS

4.1 Frequency

```
#simple frequency

x=table(mystud)
x
barplot (x)
```



```
list(student)
#Simple analysis

# remember to send to object and access the data to the column by typing the table name and followed by a dollar sign $
mystud=student$midMarks
mean(mystud)
summary(mystud)
sd(mystud)
hist(mystud)
boxplot (mystud, main="The Mid-term marks",  ylab=" marks" )
x=table(mystud)
x
barplot (x)
```

```
Data
student                    12 obs. of 1 variables
Values
  mystud                   int [1:12] 80 70 50 60 40 50 70 100 70 60 ...
  x                        'table' int [1:7(1d)] 1 1 2 2 4 1 1
```

```
9         70
10        60
11        10
12        70

> #Simple analysis
> # remember to send to object and access the data to the column by typing the table name and followed by a dollar sign $
> mystud=student$midMarks
> mean(mystud)
[1] 60.83333
> summary(mystud)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  10.00   50.00   65.00   60.83   70.00  100.00
> sd(mystud)
[1] 22.34373
> hist(mystud)
> boxplot (mystud, main="The Mid-term marks",  ylab=" marks" )
> x=table(mystud)
> x
mystud
 10  40  50  60  70  80 100
  1   1   2   2   4   1   1
> barplot (x)
>
```

4.2 More Frequency

```
group=seq(0,100, by=20)
group.data=cut(student$midMarks,group,right=FALSE)
group.data
group.feq=table(group.data)
list(group.feq)
cbind(group.feq)

barplot(group.feq)
barplot(group.feq, col=c("lightblue", "mistyrose",
  "lightcyan","lavender", "cornsilk"), )
barplot(group.feq, col=c("lightblue", "mistyrose",
  "lightcyan","lavender", "cornsilk"), main= list("Carry mark
  BACS1111", font=2))
```

Making a frequency distribution table at first sight looks more complicated in R than in other programs. The reason for this, however, is that you use the same commands for the simplest table as for the more complicated. So, the commands are slightly harder to start with, but remain the same to make all types of tables, included grouped frequency distribution tables.

• First, you have to define the spans you are interested in. Suppose we are interested in all spans from 0 to 100. This is defined by the command:

```
group=seq(0,100, by=10)
```

- This commands defines the lower values of the intervals as the sequence of values from 0 to 100 with a step function of 10 (i.e., it will go from 0 to 10, to 20, etc.). If we define by=20, then the lower values of the spans will go from 0 to 20, to 40, and so on, which means that we will have a grouped frequency distribution table.

- Next we segment the continuum of values in midMarks according to the spans we defined above. We do this with the following command:

```
    group.data=cut(student$midMarks,group,right=FALSE)
```

- This commands cuts the values of the variable midMarks data from the dataset student into parts starting with the lower value defined in span, and ending with the upper value immediately below the lower value of the next span, and puts it into a new variable group,data. The following parts are important:

```
    group.data
```

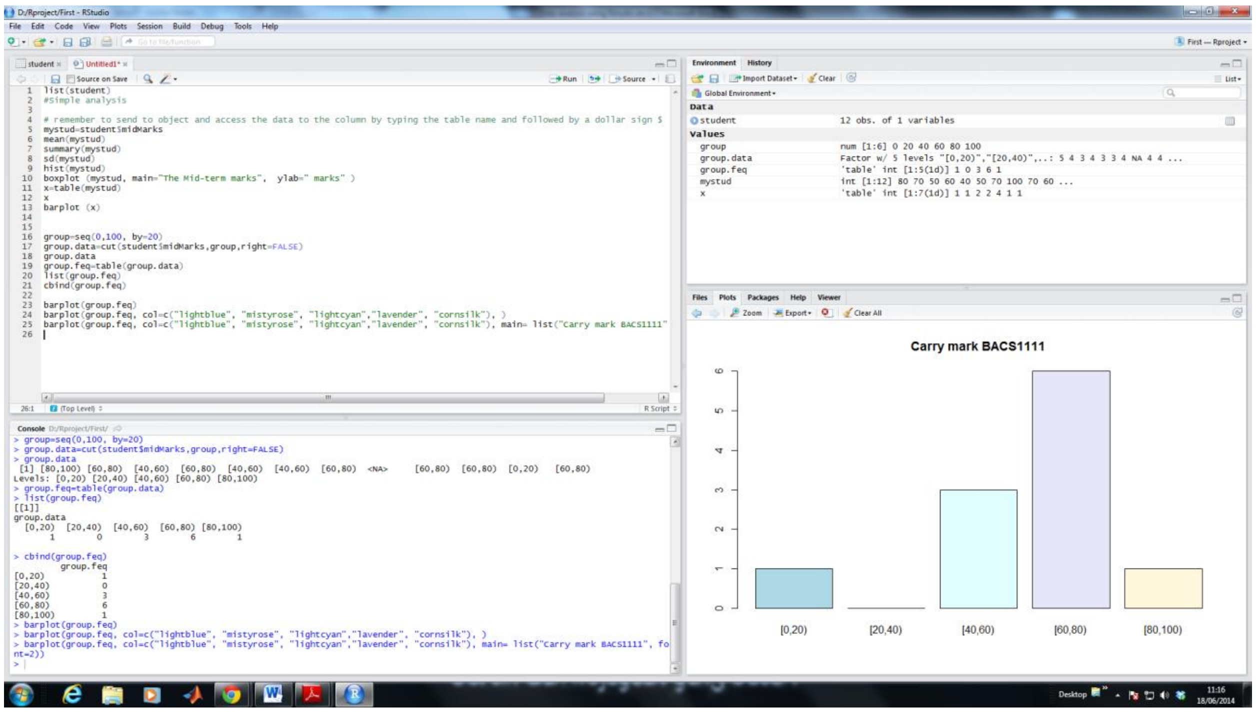
- This is the variable group.data from the dataset student. Variables from a dataset are defined as dataset$variable (i.e., with a $ sign between the name of the dataset and the name of the variable).

- Span refers to the lower values of the intervals we defined above.

- right=FALSE makes that the lower value of the next interval will be excluded from the present interval. If we had not done this, the interval 20- 40 would exclude the number 20 and include the number 40. Now the interval goes from 20.0 to 39.9999….

- To calculate the frequencies of the intervals (Midterm group) we use the command:

group.feq=table(group.data)



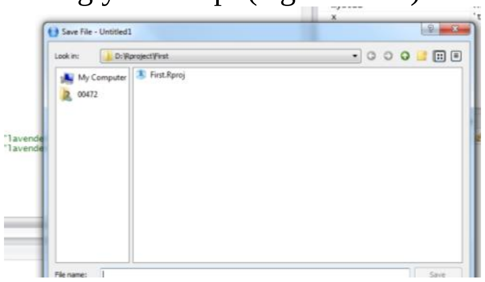
```
list(student)
#Simple analysis

# remember to send to object and access the data to the column by typing the table name and followed by a dollar sign $
mystud=student$midMarks
mean(mystud)
summary(mystud)
sd(mystud)
hist(mystud)
boxplot (mystud, main="The Mid-term marks",  ylab=" marks" )
x=table(mystud)
x
barplot (x)


group=seq(0,100, by=20)
group.data=cut(student$midMarks,group,right=FALSE)
group.data
group.feq=table(group.data)
list(group.feq)
cbind(group.feq)

barplot(group.feq)
barplot(group.feq, col=c("lightblue", "mistyrose", "lightcyan","lavender", "cornsilk"), )
barplot(group.feq, col=c("lightblue", "mistyrose", "lightcyan","lavender", "cornsilk"), main= list("Carry mark BACS1111"
```

```
> group=seq(0,100, by=20)
> group.data=cut(student$midMarks,group,right=FALSE)
> group.data
 [1] [80,100) [60,80)  [40,60)  [60,80)  [40,60)  [40,60)  [60,80)  <NA>     [60,80)  [60,80)  [0,20)   [60,80)
Levels: [0,20) [20,40) [40,60) [60,80) [80,100)
> group.feq=table(group.data)
> list(group.feq)
[[1]]
group.data
  [0,20)  [20,40)  [40,60)  [60,80) [80,100)
       1        0        3        6        1

> cbind(group.feq)
         group.feq
[0,20)           1
[20,40)          0
[40,60)          3
[60,80)          6
[80,100)         1
> barplot(group.feq)
> barplot(group.feq, col=c("lightblue", "mistyrose", "lightcyan","lavender", "cornsilk"), )
> barplot(group.feq, col=c("lightblue", "mistyrose", "lightcyan","lavender", "cornsilk"), main= list("Carry mark BACS1111", fo
nt=2))
> 
```

```
Data
student       12 obs. of 1 variables
Values
  group       num [1:6] 0 20 40 60 80 100
  group.data  Factor w/ 5 levels "[0,20)","[20,40)",..: 5 4 3 4 3 3 4 NA 4 4 ...
  group.feq   'table' int [1:5(1d)] 1 0 3 6 1
  mystud      int [1:12] 80 70 50 60 40 50 70 100 70 60 ...
  x           'table' int [1:7(1d)] 1 1 2 2 4 1 1
```

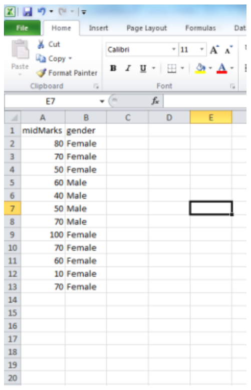
Try change the code > group.data=cut(student$midMarks,group,right=TRUE) and see the difference.

Saving your script (eg. Scale.R)

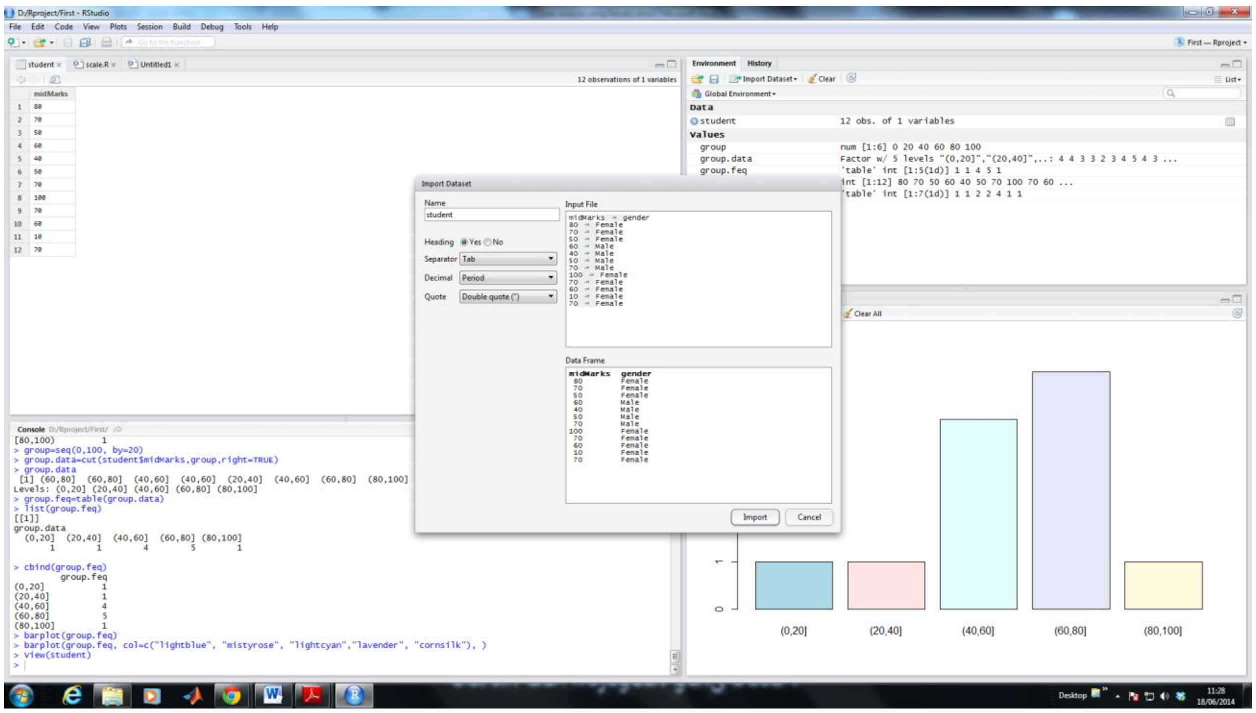


4.3 Quantitative Variable

4.3.1 Entering data



	A	B
1	midMarks	gender
2	80	Female
3	70	Female
4	50	Female
5	60	Male
6	40	Male
7	50	Male
8	70	Male
9	100	Female
10	70	Female
11	60	Female
12	10	Female
13	70	Female



Import Dataset

Name: student

Heading: (● Yes) ○ No

Separator: Tab

Decimal: Period

Quote: Double quote (")

Input File

```
midMarks → gender
80 → Female
70 → Female
50 → Female
60 → Male
40 → Male
50 → Male
70 → Male
100 → Female
70 → Female
60 → Female
10 → Female
70 → Female
```

Data Frame

```
midMarks  gender
  80      Female
  70      Female
  50      Female
  60      Male
  40      Male
  50      Male
  70      Male
 100      Female
  70      Female
  60      Female
  10      Female
  70      Female
```

Environment

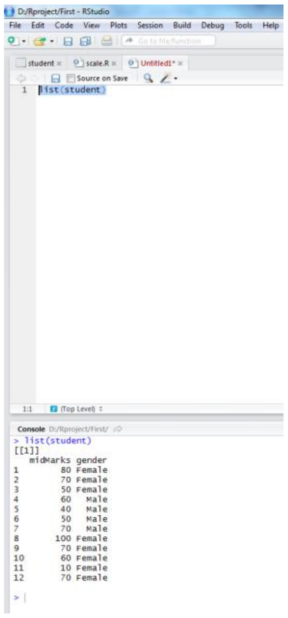
```
Data
student        12 obs. of 1 variables
Values
group          num [1:6] 0 20 40 60 80 100
group.data     Factor w/ 5 levels "(0,20]","(20,40]",..: 4 4 3 3 2 3 4 5 4 3 ...
group.feq      'table' int [1:5(1d)] 1 1 4 5 1
               int [1:12] 80 70 50 60 40 50 70 100 70 60 ...
               'table' int [1:7(1d)] 1 1 2 2 4 1 1
```

Console

```
[80,100)          1
> group=seq(0,100, by=20)
> group.data=cut(student$midMarks,group,right=TRUE)
> group.data
 [1] (60,80]  (60,80]  (40,60]  (40,60]  (20,40]  (40,60]  (60,80]  (80,100]
Levels: (0,20] (20,40] (40,60] (60,80] (80,100]
> group.feq=table(group.data)
> list(group.feq)
[[1]]
group.data
  (0,20]  (20,40]  (40,60]  (60,80] (80,100]
       1        1        4        5        1

> cbind(group.feq)
         group.feq
(0,20]           1
(20,40]          1
(40,60]          4
(60,80]          5
(80,100]         1
> barplot(group.feq)
> barplot(group.feq, col=c("lightblue", "mistyrose", "lightcyan","lavender", "cornsilk"), )
> View(student)
>
```

Write command:

```
list(student)
```

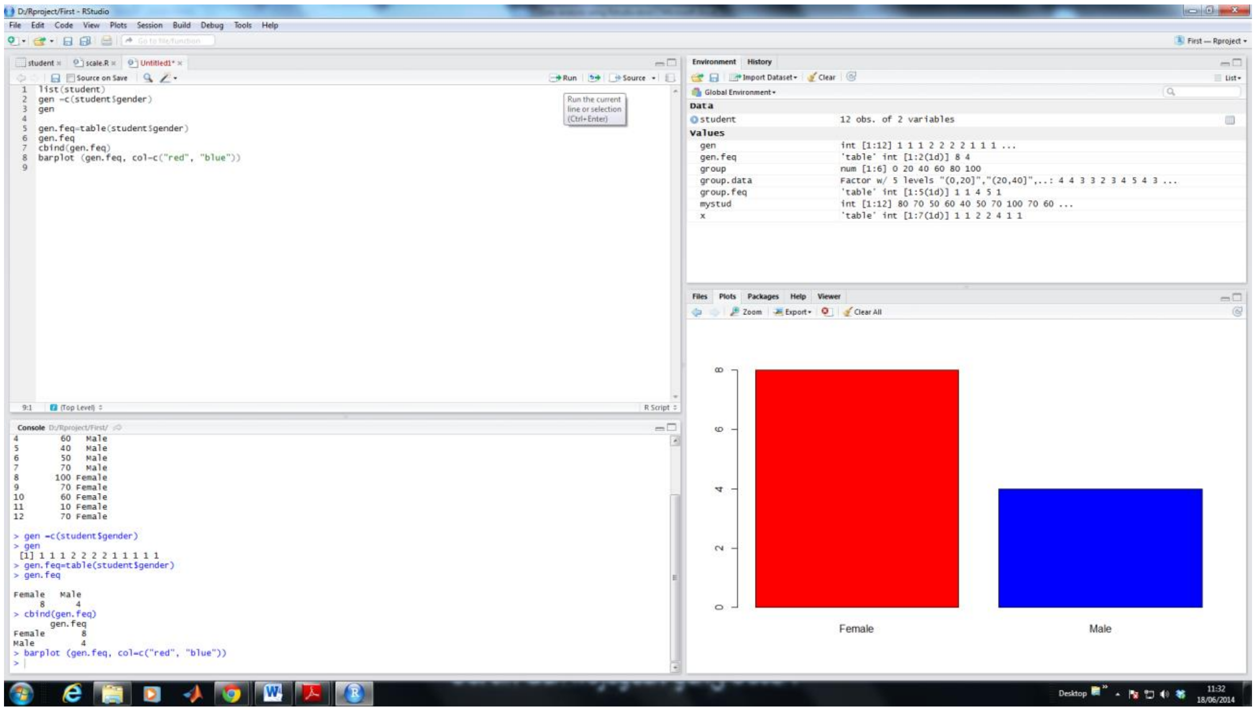

D:/Rproject/First - RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
student
scale.R
Untitled1*
Source on Save
1:1
(Top Level)
Console D:/Rproject/First/

```
list(student)
```

```
> list(student)
[[1]]
   midMarks gender
1        80 Female
2        70 Female
3        50 Female
4        60   Male
5        40   Male
6        50   Male
7        70   Male
8       100 Female
9        70 Female
10       60 Female
11       10 Female
12       70 Female

> 
```

4.3.2 Descriptive Statistics-frequency

```
##access to the column gender
gen =c(student$gender)
gen

gen.feq=table(student$gender)
gen.feq
cbind(gen.feq)
barplot (gen.feq, col=c("red", "blue"))
```



```
list(student)
gen -c(student$gender)
gen

gen.feq=table(student$gender)
gen.feq
cbind(gen.feq)
barplot (gen.feq, col=c("red", "blue"))
```

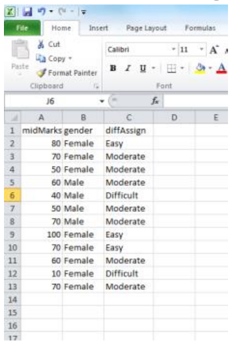
```
student      12 obs. of 2 variables
Values
gen          int [1:12] 1 1 1 2 2 2 2 1 1 1 ...
gen.feq      'table' int [1:2(1d)] 8 4
group        num [1:6] 0 20 40 60 80 100
group.data   Factor w/ 5 levels "(0,20]","(20,40]",..: 4 4 3 3 2 3 4 5 4 3 ...
group.feq    'table' int [1:5(1d)] 1 1 4 5 1
mystud       int [1:12] 80 70 50 60 40 50 70 100 70 60 ...
x            'table' int [1:7(1d)] 1 1 2 2 4 1 1
```

```
4         60   Male
5         40   Male
6         50   Male
7         70   Male
8        100 Female
9         70 Female
10        60 Female
11        10 Female
12        70 Female

> gen -c(student$gender)
> gen
 [1] 1 1 1 2 2 2 2 1 1 1 1 1
> gen.feq=table(student$gender)
> gen.feq

Female   Male
     8      4
> cbind(gen.feq)
       gen.feq
Female       8
Male         4
> barplot (gen.feq, col=c("red", "blue"))
>
```

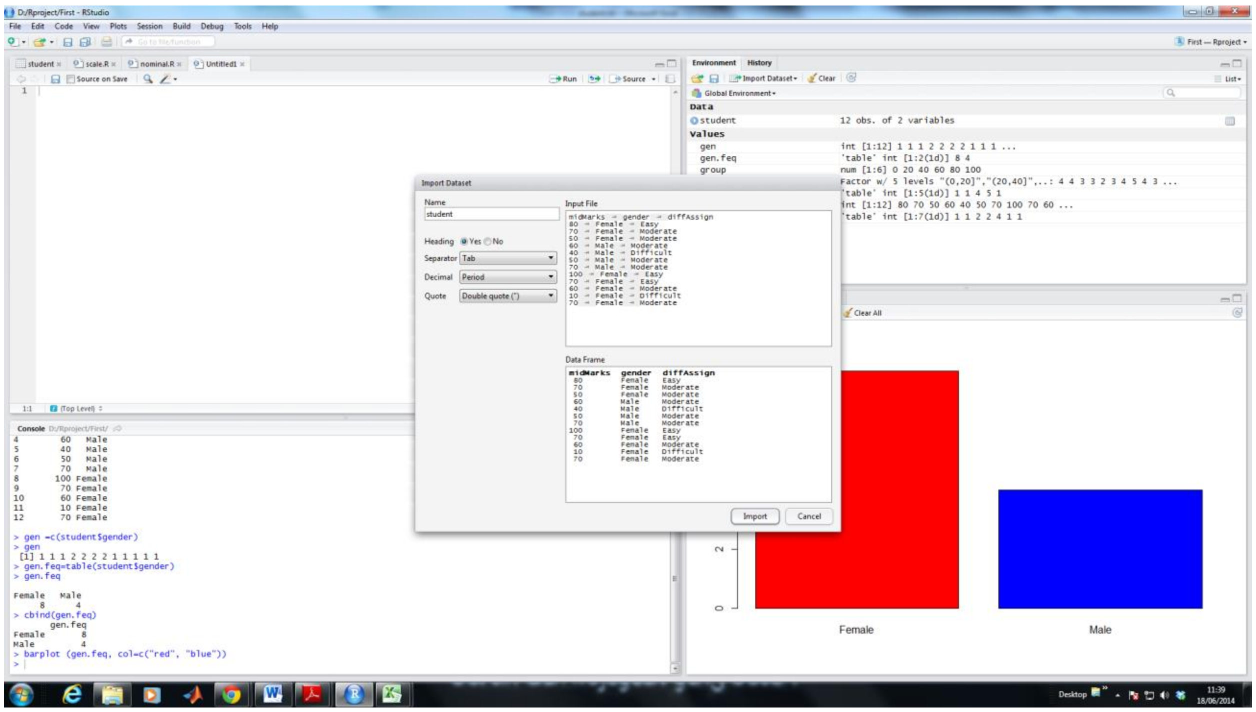
D:/Rproject/First - RStudio
File Edit Code View Plots Session Build Debug Tools Help
Run the current line or selection (Ctrl+Enter)
Environment History
Files Plots Packages Help Viewer
0
2
4
6
8
Female
Male

4.4 Quantitative variable (Ordinal)

4.4.1 Entering



	A	B	C	D	E
1	midMarks	gender	diffAssign		
2	80	Female	Easy		
3	70	Female	Moderate		
4	50	Female	Moderate		
5	60	Male	Moderate		
6	40	Male	Difficult		
7	50	Male	Moderate		
8	70	Male	Moderate		
9	100	Female	Easy		
10	70	Female	Easy		
11	60	Female	Moderate		
12	10	Female	Difficult		
13	70	Female	Moderate		
14					
15					
16					



```
Data
student         12 obs. of 2 variables
Values
gen             int [1:12] 1 1 1 2 2 2 2 1 1 1 ...
gen.feq         'table' int [1:2(1d)] 8 4
group           num [1:6] 0 20 40 60 80 100
                Factor w/ 5 levels "(0,20]","(20,40]",..: 4 4 3 3 2 3 4 5 4 3 ...
                'table' int [1:5(1d)] 1 1 4 5 1
                int [1:12] 80 70 50 60 40 50 70 100 70 60 ...
                'table' int [1:7(1d)] 1 1 2 2 4 1 1
```

Name: student
Heading: [✓ Yes] No
Separator: Tab
Decimal: Period
Quote: Double quote (")

Input File

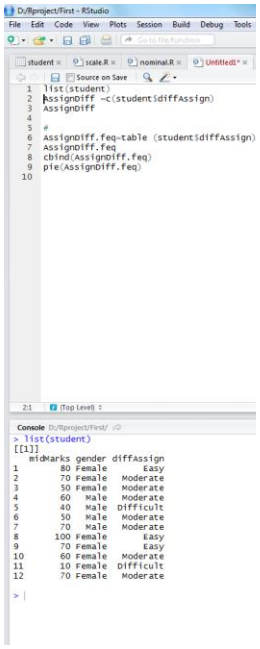
```
midMarks → gender → diffAssign
80 → Female → Easy
70 → Female → Moderate
50 → Female → Moderate
60 → Male → Moderate
40 → Male → Difficult
50 → Male → Moderate
70 → Male → Moderate
100 → Female → Easy
70 → Female → Easy
60 → Female → Moderate
10 → Female → Difficult
70 → Female → Moderate
```

Data Frame

```
midMarks  gender  diffAssign
 80       Female  Easy
 70       Female  Moderate
 50       Female  Moderate
 60       Male    Moderate
 40       Male    Difficult
 50       Male    Moderate
 70       Male    Moderate
100       Female  Easy
 70       Female  Easy
 60       Female  Moderate
 10       Female  Difficult
 70       Female  Moderate
```

Console D:/Rproject/First/

```
4          60   Male
5          40   Male
6          50   Male
7          70   Male
8         100 Female
9          70 Female
10         60 Female
11         10 Female
12         70 Female

> gen -c(student$gender)
> gen
 [1] 1 1 1 2 2 2 2 1 1 1 1 1
> gen.feq=table(student$gender)
> gen.feq

Female   Male
     8      4
> cbind(gen.feq)
       gen.feq
Female       8
Male         4
> barplot (gen.feq, col=c("red", "blue"))
> 
```


D:/Rproject/First - RStudio
File Edit Code View Plots Session Build Debug Tools
student scale.R nominal.R Untitled1*
Source on Save
2:1 (Top Level)

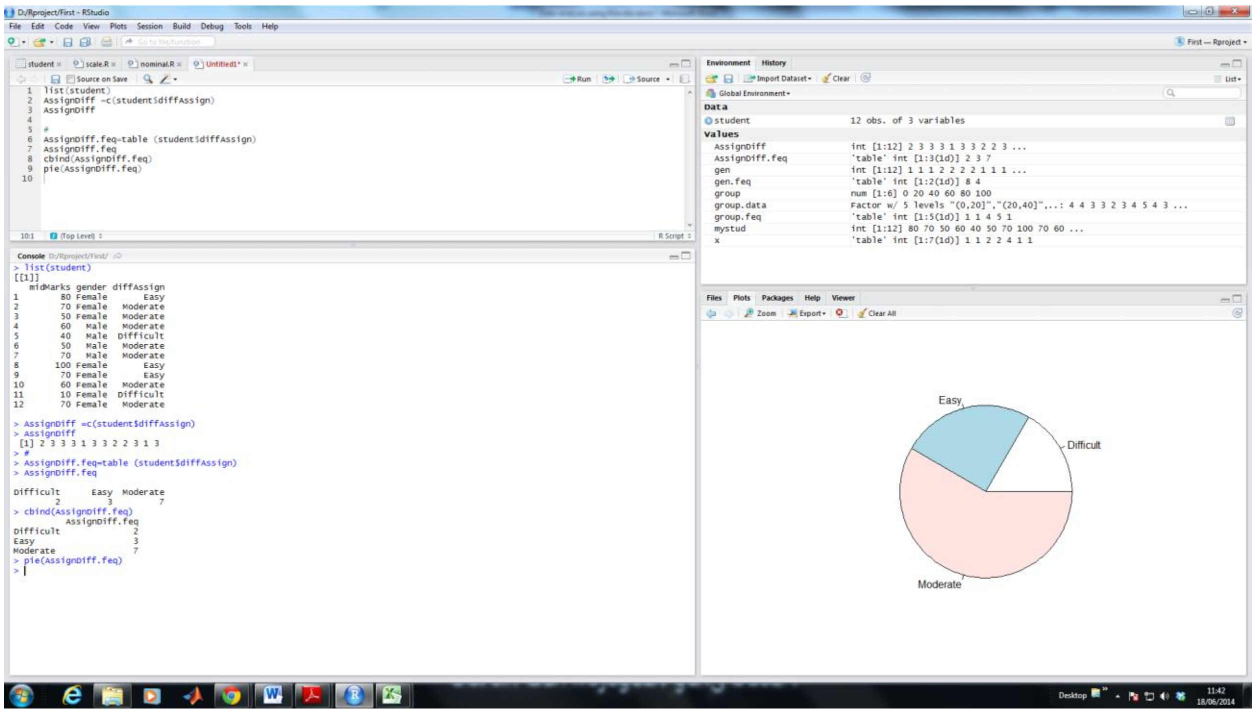
```
list(student)
AssignDiff -c(student$diffAssign)
AssignDiff

#
AssignDiff.feq-table (student$diffAssign)
AssignDiff.feq
cbind(AssignDiff.feq)
pie(AssignDiff.feq)
```

Console D:/Rproject/First/

```
> list(student)
[[1]]
   midMarks gender diffAssign
1        80 Female       Easy
2        70 Female   Moderate
3        50 Female   Moderate
4        60   Male   Moderate
5        40   Male  Difficult
6        50   Male   Moderate
7        70   Male   Moderate
8       100 Female       Easy
9        70 Female       Easy
10       60 Female   Moderate
11       10 Female  Difficult
12       70 Female   Moderate

> 
```

4.4.2 Descriptive statistics-frequency

```
List(student)
AssignDiff =c(student$diffAssign)
AssignDiff

#
AssignDiff.feq=table (student$diffAssign)
AssignDiff.feq
cbind(AssignDiff.feq)
pie(AssignDiff.feq)
```



```
list(student)
AssignDiff =c(student$diffAssign)
AssignDiff

#
AssignDiff.feq=table (student$diffAssign)
AssignDiff.feq
cbind(AssignDiff.feq)
pie(AssignDiff.feq)
```

```
> list(student)
[[1]]
   midMarks gender diffAssign
1        80 Female       Easy
2        70 Female   Moderate
3        50 Female   Moderate
4        60   Male   Moderate
5        40   Male  Difficult
6        50   Male   Moderate
7        70   Male   Moderate
8       100 Female       Easy
9        70 Female       Easy
10       60 Female   Moderate
11       10 Female  Difficult
12       70 Female   Moderate

> AssignDiff =c(student$diffAssign)
> AssignDiff
 [1] 2 3 3 3 1 3 3 2 2 3 1 3
> #
> AssignDiff.feq=table (student$diffAssign)
> AssignDiff.feq

Difficult      Easy  Moderate
        2         3         7
> cbind(AssignDiff.feq)
          AssignDiff.feq
Difficult              2
Easy                   3
Moderate               7
> pie(AssignDiff.feq)
> |
```

```
Data
student              12 obs. of 3 variables
Values
  AssignDiff         int [1:12] 2 3 3 3 1 3 3 2 2 3 ...
  AssignDiff.feq     'table' int [1:3(1d)] 2 3 7
  gen                int [1:12] 1 1 1 2 2 2 2 1 1 1 ...
  gen.feq            'table' int [1:2(1d)] 8 4
  group              num [1:6] 0 20 40 60 80 100
  group.data         Factor w/ 5 levels "(0,20]","(20,40]",..: 4 4 3 3 2 3 4 5 4 3 ...
  group.feq          'table' int [1:5(1d)] 1 1 4 5 1
  mystud             int [1:12] 80 70 50 60 40 50 70 100 70 60 ...
  x                  'table' int [1:7(1d)] 1 1 2 2 4 1 1
```

D:/Rproject/First - RStudio
File Edit Code View Plots Session Build Debug Tools Help
student scale.R nominal.R Untitled1*
Source on Save
Run Source
10:1 (Top Level) R Script
Console D:/Rproject/First/
Environment History
Import Dataset Clear List
Global Environment
First — Rproject
Files Plots Packages Help Viewer
Zoom Export Clear All
Easy
Difficult
Moderate
Desktop
11:42
18/06/2014

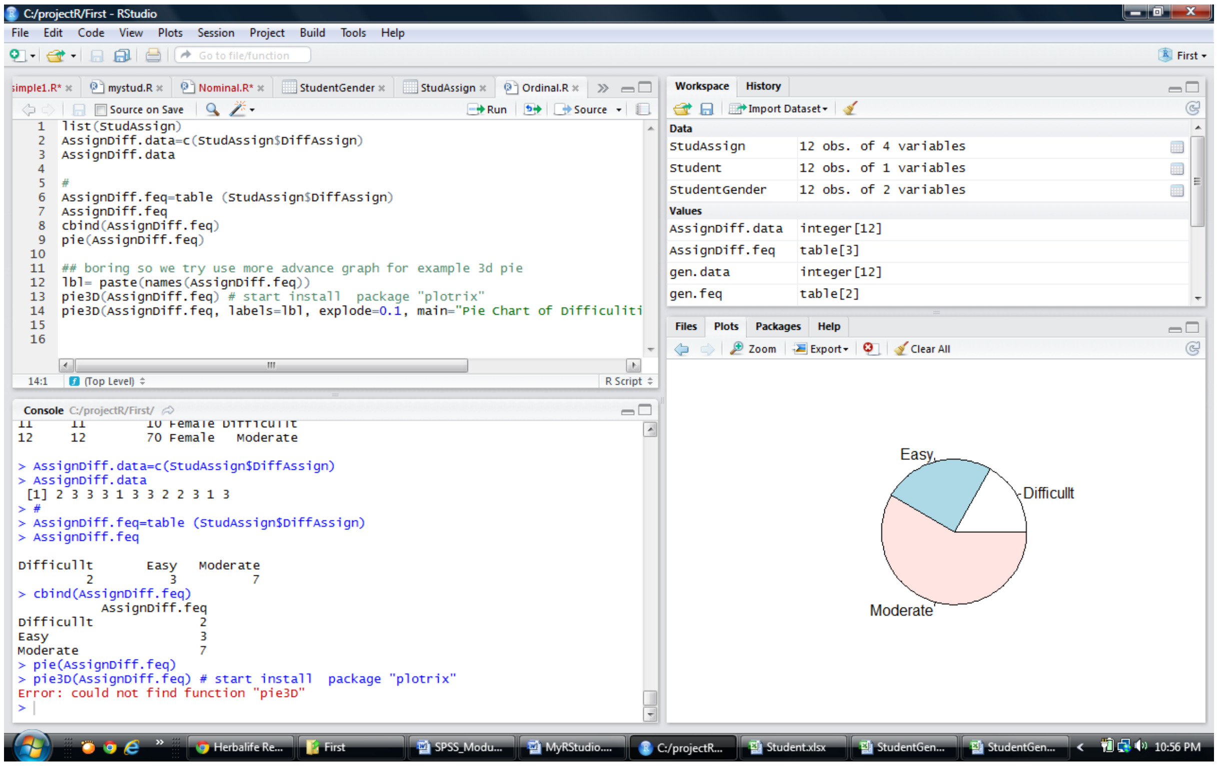
More advance figure or graph –3D graph .

Try write > `pie3D(AssignDiff.feq)`

What will you get? `Error: could not find function "pie3D"`

4.5 Install the package

In order to have 3D pie, you need to install the package "`plotrix`"

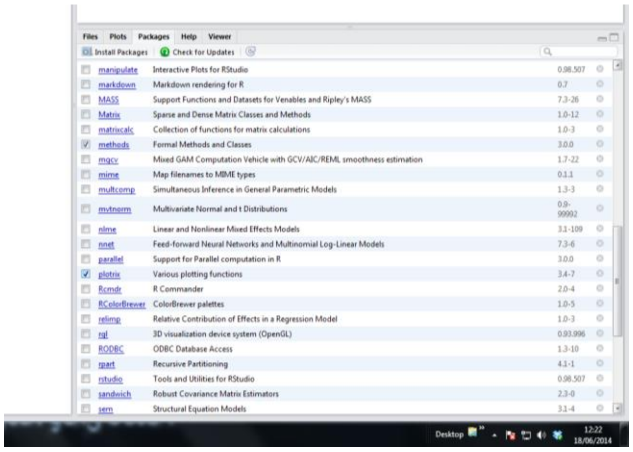


```
list(StudAssign)
AssignDiff.data=c(StudAssign$DiffAssign)
AssignDiff.data

#
AssignDiff.feq=table (StudAssign$DiffAssign)
AssignDiff.feq
cbind(AssignDiff.feq)
pie(AssignDiff.feq)

## boring so we try use more advance graph for example 3d pie
lbl= paste(names(AssignDiff.feq))
pie3D(AssignDiff.feq) # start install  package "plotrix"
pie3D(AssignDiff.feq, labels=lbl, explode=0.1, main="Pie Chart of Difficuliti
```

Data	
StudAssign	12 obs. of 4 variables
Student	12 obs. of 1 variables
StudentGender	12 obs. of 2 variables
Values	
AssignDiff.data	integer[12]
AssignDiff.feq	table[3]
gen.data	integer[12]
gen.feq	table[2]

```
11     11        10 Female  Difficult
12     12        70 Female   Moderate

> AssignDiff.data=c(StudAssign$DiffAssign)
> AssignDiff.data
 [1] 2 3 3 3 1 3 3 2 2 3 1 3
> #
> AssignDiff.feq=table (StudAssign$DiffAssign)
> AssignDiff.feq

Difficullt      Easy   Moderate
         2         3          7
> cbind(AssignDiff.feq)
           AssignDiff.feq
Difficullt              2
Easy                    3
Moderate                7
> pie(AssignDiff.feq)
> pie3D(AssignDiff.feq) # start install  package "plotrix"
Error: could not find function "pie3D"
> |
```

Click on Packages tab and checked the `plotrix` library



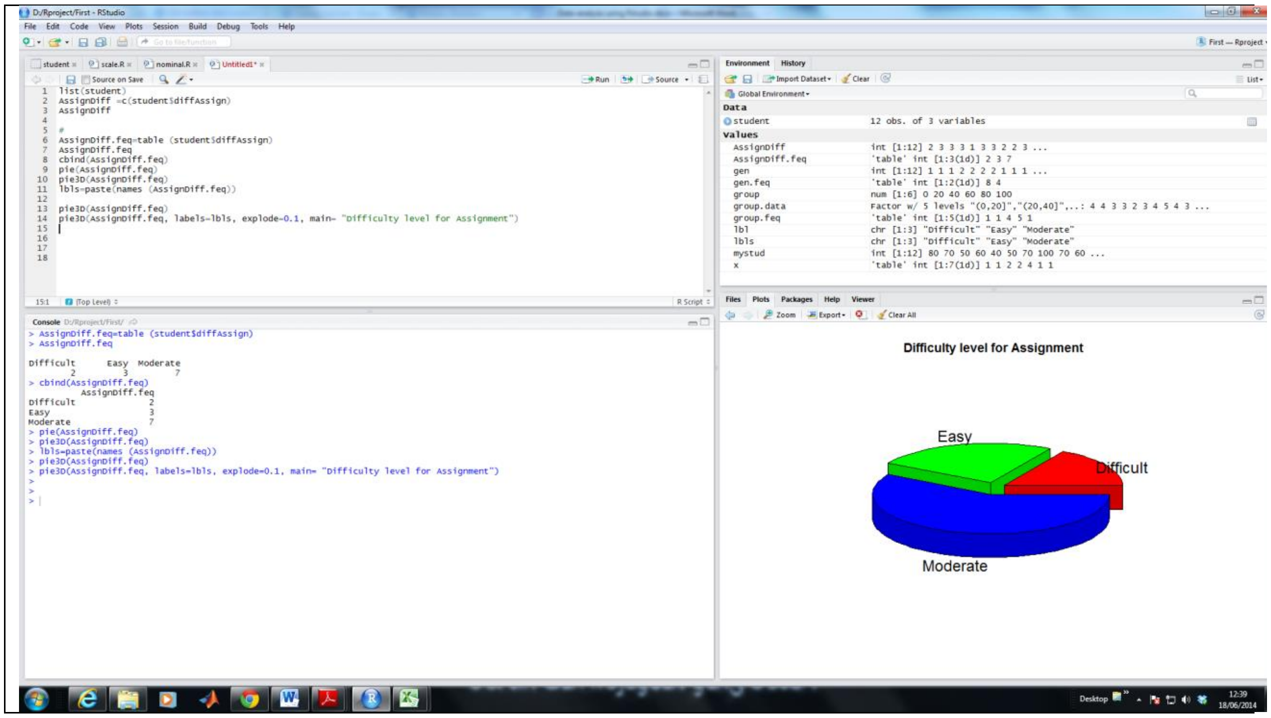
☐	manipulate	Interactive Plots for RStudio	0.98.507
☐	markdown	Markdown rendering for R	0.7
☐	MASS	Support Functions and Datasets for Venables and Ripley's MASS	7.3-26
☐	Matrix	Sparse and Dense Matrix Classes and Methods	1.0-12
☐	matrixcalc	Collection of functions for matrix calculations	1.0-3
✓	methods	Formal Methods and Classes	3.0.0
☐	mgcv	Mixed GAM Computation Vehicle with GCV/AIC/REML smoothness estimation	1.7-22
☐	mime	Map filenames to MIME types	0.1.1
☐	multcomp	Simultaneous Inference in General Parametric Models	1.3-3
☐	mvtnorm	Multivariate Normal and t Distributions	0.9-99992
☐	nlme	Linear and Nonlinear Mixed Effects Models	3.1-109
☐	nnet	Feed-forward Neural Networks and Multinomial Log-Linear Models	7.3-6
☐	parallel	Support for Parallel computation in R	3.0.0
✓	plotrix	Various plotting functions	3.4-7
☐	Rcmdr	R Commander	2.0-4
☐	RColorBrewer	ColorBrewer palettes	1.0-5
☐	relimp	Relative Contribution of Effects in a Regression Model	1.0-3
☐	rgl	3D visualization device system (OpenGL)	0.93.996
☐	RODBC	ODBC Database Access	1.3-10
☐	rpart	Recursive Partitioning	4.1-1
☐	rstudio	Tools and Utilities for RStudio	0.98.507
☐	sandwich	Robust Covariance Matrix Estimators	2.3-0
☐	sem	Structural Equation Models	3.1-4

The code `> library("plotrix", lib.loc="C:/Program Files/R/R-3.0.0/library")`

will appear in console window that indicate function Plotrix in your RStudio.

Write the following code :

```
pie3D(AssignDiff.feq)
lbls=paste(names (AssignDiff.feq))
pie3D(AssignDiff.feq)
pie3D(AssignDiff.feq, labels=lbls, explode=0.1, main= "Difficulty
  level for Assignment")
```



```
list(student)
AssignDiff =c(student$diffAssign)
AssignDiff

#
AssignDiff.feq=table (student$diffAssign)
AssignDiff.feq
cbind(AssignDiff.feq)
pie(AssignDiff.feq)
pie3D(AssignDiff.feq)
lbls=paste(names (AssignDiff.feq))

pie3D(AssignDiff.feq)
pie3D(AssignDiff.feq, labels=lbls, explode=0.1, main= "Difficulty level for Assignment")
```

```
> AssignDiff.feq=table (student$diffAssign)
> AssignDiff.feq

Difficult      Easy  Moderate 
        2         3         7 
> cbind(AssignDiff.feq)
          AssignDiff.feq
Difficult              2
Easy                   3
Moderate               7
> pie(AssignDiff.feq)
> pie3D(AssignDiff.feq)
> lbls=paste(names (AssignDiff.feq))
> pie3D(AssignDiff.feq)
> pie3D(AssignDiff.feq, labels=lbls, explode=0.1, main= "Difficulty level for Assignment")
>
>
> 
```

```
Data
student          12 obs. of 3 variables
Values
AssignDiff       int [1:12] 2 3 3 3 1 3 3 2 2 3 ...
AssignDiff.feq   'table' int [1:3(1d)] 2 3 7
gen              int [1:12] 1 1 1 2 2 2 2 1 1 1 ...
gen.feq          'table' int [1:2(1d)] 8 4
group            num [1:6] 0 20 40 60 80 100
group.data       Factor w/ 5 levels "(0,20]","(20,40]",..: 4 4 3 3 2 3 4 5 4 3 ...
group.feq        'table' int [1:5(1d)] 1 1 4 5 1
lbl              chr [1:3] "Difficult" "Easy" "Moderate"
lbls             chr [1:3] "Difficult" "Easy" "Moderate"
mystud           int [1:12] 80 70 50 60 40 50 70 100 70 60 ...
x                'table' int [1:7(1d)] 1 1 2 2 4 1 1
```

D:/Rproject/First - RStudio
File Edit Code View Plots Session Build Debug Tools Help
student scale.R nominal.R Untitled1*
Source on Save
Run Source
Console D:/Rproject/First/
Environment History
Import Dataset Clear
Global Environment
Files Plots Packages Help Viewer
Zoom Export Clear All
Difficulty level for Assignment
Easy
Difficult
Moderate

5 INFERENTIAL ANALYSIS

5.1 Introduction

In terms of inferential statistics R has many varieties of functions for doing statistical analyses including the parametric and non-parametric t-test, analysis of variance, t-test, correlation and linear regression.

i. Compare mean using T-test (Parametric)

Some of them are self-understanding, and really easy to use. This is the case for example to perform a t-test to compare one average value with zero or two average values. The function `t.test()` do this. With one argument, the function tests if the average value of the variable differs from zero. With two arguments it simply compares the two average values, e.g. `t.test(midMarks,finalMarks)`.

ii. Compare mean Using Wilcoxon rank test (non-parametric)

The function `wilcox.test()`performs wilcoxon rank test, which assumes that the means of two unnormally distributed datasets are equal.

iii. Binomial test

The function `binom.test()` performs an exact test of a simple null hypothesis about the probability of success in a percentage. For example, among 125 insects counted, 45 are females, and we want to see whether this differs from 50% females. For this, `binom.test(45,125,.5)` give the corresponding results (the answer is yes, with p= 0.002226).

iv. Kruskal-Wallis rank sum test

The function `kruskal.test()`performs a Kruskal-Wallis rank sum test of the null that the location parameters of the distribution of x are the same in each group (sample). The alternative is that they differ in at least one. It can take a single argument that is either a table with several columns corresponding to variables, or else a `list()` function regrouping all independent variables in one object as shown below:

```
## Hollander & Wolfe (1973), 116.
## Mucociliary efficiency from the rate of removal of dust in normal
##  subjects, subjects with obstructive airway disease, and subjects
##  with asbestosis.
x <- c(2.9, 3.0, 2.5, 2.6, 3.2) # normal subjects
y <- c(3.8, 2.7, 4.0, 2.4)      # with obstructive airway disease
z <- c(2.8, 3.4, 3.7, 2.2, 2.0) # with asbestosis
kruskal.test(list(x, y, z))
## Equivalently,
x <- c(x, y, z)
g <- factor(rep(1:3, c(5, 4, 5)),
            labels = c("Normal subjects",
                       "Subjects with obstructive airway disease",
                       "Subjects with asbestosis"))
kruskal.test(x, g)

## Formula interface.
require(graphics)
boxplot(Ozone ~ Month, data = airquality)
kruskal.test(Ozone ~ Month, data = airquality)
```

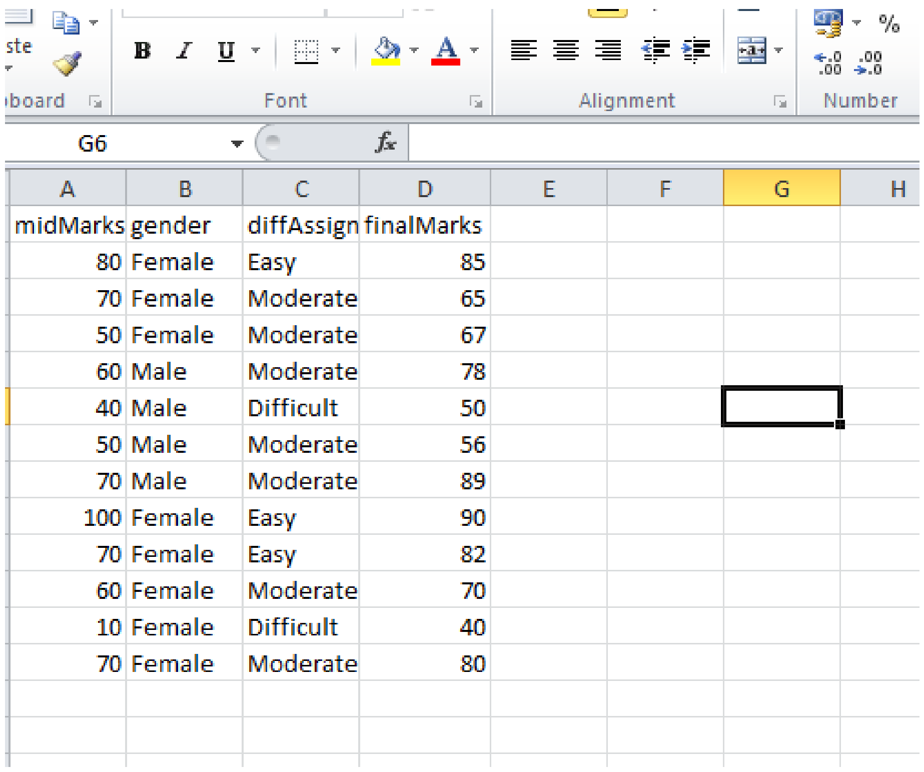
v. Correlation

The function `cor.test()` tests the correlation or relationship between two vectors, e.g.,

`cor.test(midMarks,finalMarks).`

Overall, there are several other functions in R, covering (almost) all standard statistical methods. See http://www.statmethods.net/stats/ttest.html

5.2 Entering Data



A	B	C	D
midMarks	gender	diffAssign	finalMarks
80	Female	Easy	85
70	Female	Moderate	65
50	Female	Moderate	67
60	Male	Moderate	78
40	Male	Difficult	50
50	Male	Moderate	56
70	Male	Moderate	89
100	Female	Easy	90
70	Female	Easy	82
60	Female	Moderate	70
10	Female	Difficult	40
70	Female	Moderate	80

5.3 Compare mean analysis

5.3.1 T-test

```
list(student)
midterm.data<-c(student$midMarks)
midterm.data
final.data<-c(student$finalMarks)
final.data
t.test(midterm.data,final.data) # independent t-test
t.test(midterm.data,final.data, paired=TRUE) #paired sample t-test
t.test(final.data, mu=75) #Ho:mu=75 #One sample t-test.
```

5.3.2 Non-parametric t-test: Wilcoxon rank test

If we are assuming the data to have not normal distribution or small sample size. In our case n<20, so we can proceed with the non-parametric t-test or **Wilcoxon rank test .**

```
wilcox.test(midterm.data,final.data, paired=TRUE)
```

5.4 Exercises

Exercise 1: A company has proposed a new packaging training and wants to test the effectiveness of the training by comparing the average times of 10 workers in the packaging process. The table below shows the time in minute before and after training for each worker.

Before new training	12.9, 13.5, 12.8, 15.6, 17.2, 19.2, 12.6, 15.3, 14.4, 11.3
After new taraining	12.7, 13.6, 12.0, 15.2, 16.8, 20.0, 12.0, 15.9, 16.0, 11.1

Answer:

In this case we have two sets of paired samples, since the measurements were made on the same worker before and after the training. To see if there was an improvement, deterioration, or if the means of times have remained substantially the same (hypothesis H0), we need to make a Student's t-test for paired samples, proceeding in this way:

```
a = c(12.9, 13.5, 12.8, 15.6, 17.2, 19.2, 12.6, 15.3, 14.4, 11.3)
b = c(12.7, 13.6, 12.0, 15.2, 16.8, 20.0, 12.0, 15.9, 16.0, 11.1)

t.test(a,b, paired=TRUE)

    Paired t-test
```

```
data: a and b
t = -0.2133, df = 9, p-value = 0.8358
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
    -0.5802549 0.4802549
sample estimates:
mean of the differences
    -0.05
```

The p-value is greater than 0.05, then we can accept the hypothesis H_0 of equality of the averages. In conclusion, the new training has not made any significant improvement (or deterioration) to the team of workers.

Exercise 2:

An investigator believes that caffeine facilitates performance on a simple spelling test. Two groups of subjects are given either 200 mg of caffeine or a placebo. The data are:

Placebo	Drug
24	24
25	29
27	26
26	23
26	25
22	28
21	27
22	24
23	27
25	28
25	27
25	26

Answer:

To describe the differences between these two groups, we can use basic descriptive statistics (means and standard deviations), and graph the results. To see how likely a difference of this magnitude would happen by chance if, in fact, the two groups were sampled from the same population, we can do a t-test. Please try… ☺

Other exercises:

t-test example: http://www.r-bloggers.com/paired-students-t-test/

non parametric test :http://www.r-bloggers.com/wilcoxon-signed-rank-test/

6 REFERENCES:

For more information on RStudio, try the following resources:

- Visit the RStudio homepage, http://www.rstudio.com/ . The site includes a variety of helpful resources, including the Online learning to learn how to write the R code . The support websites are available at http://blog.rstudio.org/ and http://www.r-bloggers.com/. More detailed RStudio help can be found in *Getting Started with RStudio.*

- Other websites:

  http://www.statmethods.net/graphs/bar.html

  http://www.r-bloggers.com/

  https://sites.google.com/site/r4statistics/popularity

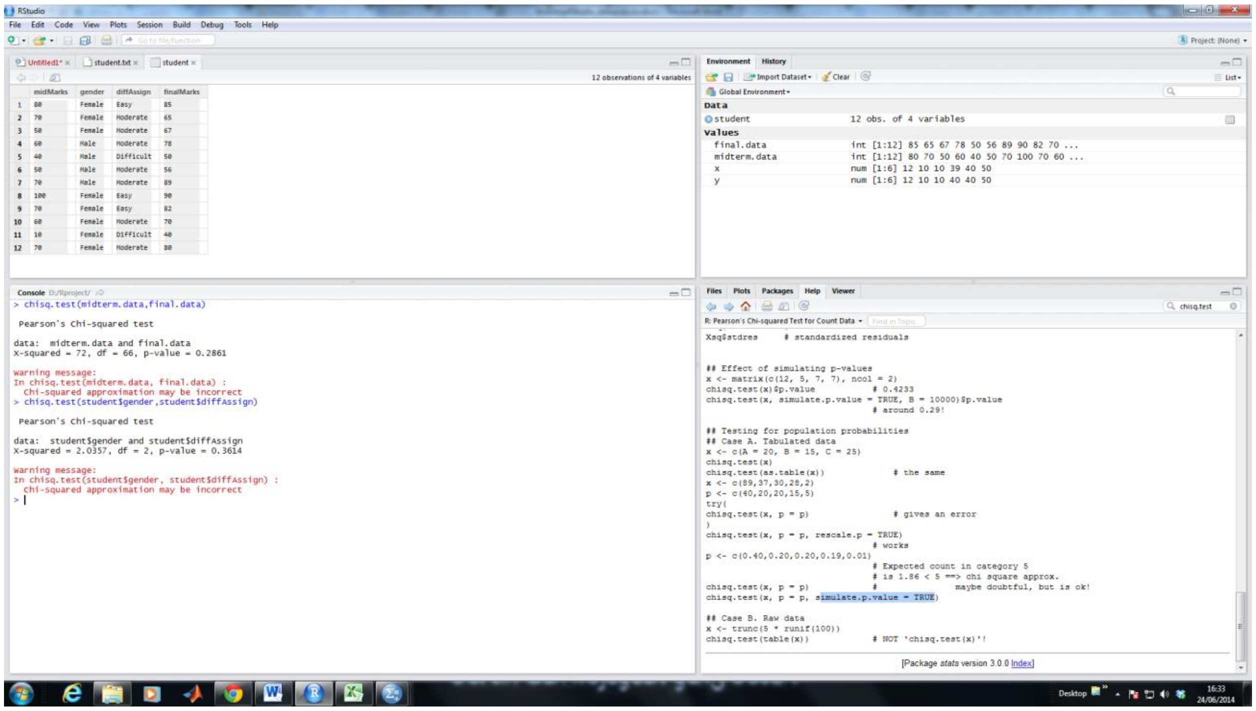
  http://en.freestatistics.info/

  http://lib.stat.cmu.edu/

  http://www.comfsm.fm/~dleeling/statistics/notes000.html

7 APPENDIX

In RStudio



RStudio
File Edit Code View Plots Session Build Debug Tools Help
Untitled1* student.txt student
12 observations of 4 variables
Environment History
Import Dataset Clear
Global Environment
Project: (None)
Console D:/Rproject/
Files Plots Packages Help Viewer
chisq.test
R: Pearson's Chi-squared Test for Count Data
Desktop
16:33
24/06/2014

	midMarks	gender	diffAssign	finalMarks
1	80	Female	Easy	85
2	70	Female	Moderate	65
3	50	Female	Moderate	67
4	60	Male	Moderate	78
5	40	Male	Difficult	50
6	50	Male	Moderate	56
7	70	Male	Moderate	89
8	100	Female	Easy	90
9	70	Female	Easy	82
10	60	Female	Moderate	70
11	10	Female	Difficult	40
12	70	Female	Moderate	80

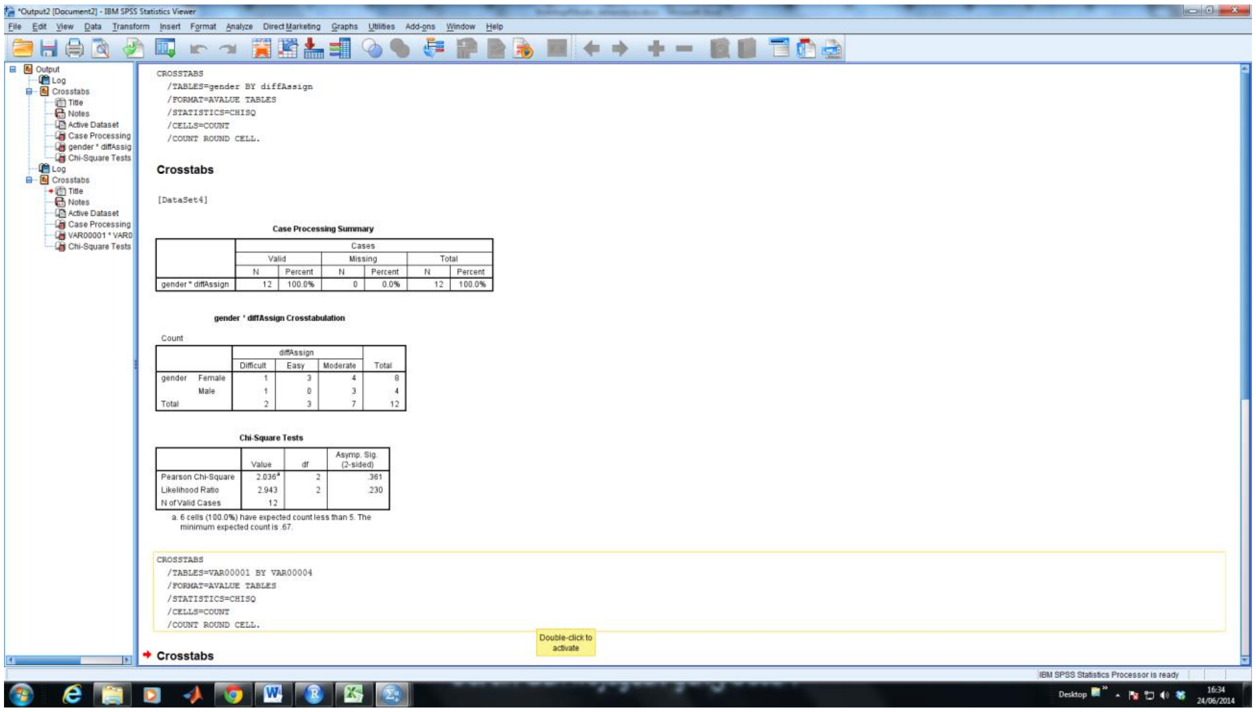
```
Data
student                 12 obs. of 4 variables
Values
  final.data            int [1:12] 85 65 67 78 50 56 89 90 82 70 ...
  midterm.data          int [1:12] 80 70 50 60 40 50 70 100 70 60 ...
  x                     num [1:6] 12 10 10 39 40 50
  y                     num [1:6] 12 10 10 40 40 50
```

```
> chisq.test(midterm.data,final.data)

 Pearson's Chi-squared test

data:  midterm.data and final.data
X-squared = 72, df = 66, p-value = 0.2861

Warning message:
In chisq.test(midterm.data, final.data) :
  Chi-squared approximation may be incorrect
> chisq.test(student$gender,student$diffAssign)

 Pearson's Chi-squared test

data:  student$gender and student$diffAssign
X-squared = 2.0357, df = 2, p-value = 0.3614

Warning message:
In chisq.test(student$gender, student$diffAssign) :
  Chi-squared approximation may be incorrect
> |
```

```
Xsq$stdres     # standardized residuals


## Effect of simulating p-values
x <- matrix(c(12, 5, 7, 7), ncol = 2)
chisq.test(x)$p.value           # 0.4233
chisq.test(x, simulate.p.value = TRUE, B = 10000)$p.value
                                # around 0.29!

## Testing for population probabilities
## Case A. Tabulated data
x <- c(A = 20, B = 15, C = 25)
chisq.test(x)
chisq.test(as.table(x))             # the same
x <- c(89,37,30,28,2)
p <- c(40,20,20,15,5)
try(
chisq.test(x, p = p)                # gives an error
)
chisq.test(x, p = p, rescale.p = TRUE)
                                # works
p <- c(0.40,0.20,0.20,0.19,0.01)
                                # Expected count in category 5
                                # is 1.86 < 5 ==> chi square approx.
chisq.test(x, p = p)            #               maybe doubtful, but is ok!
chisq.test(x, p = p, simulate.p.value = TRUE)

## Case B. Raw data
x <- trunc(5 * runif(100))
chisq.test(table(x))            # NOT 'chisq.test(x)'!
```

[Package *stats* version 3.0.0 Index]

In SPSS



```
CROSSTABS
  /TABLES=gender BY diffAssign
  /FORMAT=AVALUE TABLES
  /STATISTICS=CHISQ
  /CELLS=COUNT
  /COUNT ROUND CELL.
```

Crosstabs

[DataSet4]

Case Processing Summary

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
gender * diffAssign	12	100.0%	0	0.0%	12	100.0%

gender * diffAssign Crosstabulation

Count

		diffAssign			Total
		Difficult	Easy	Moderate	
gender	Female	1	3	4	8
	Male	1	0	3	4
Total		2	3	7	12

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	2.036[a]	2	.361
Likelihood Ratio	2.943	2	.230
N of Valid Cases	12		

a. 6 cells (100.0%) have expected count less than 5. The minimum expected count is .67.

```
CROSSTABS
  /TABLES=VAR00001 BY VAR00004
  /FORMAT=AVALUE TABLES
  /STATISTICS=CHISQ
  /CELLS=COUNT
  /COUNT ROUND CELL.
```

Crosstabs