Introduction to

By

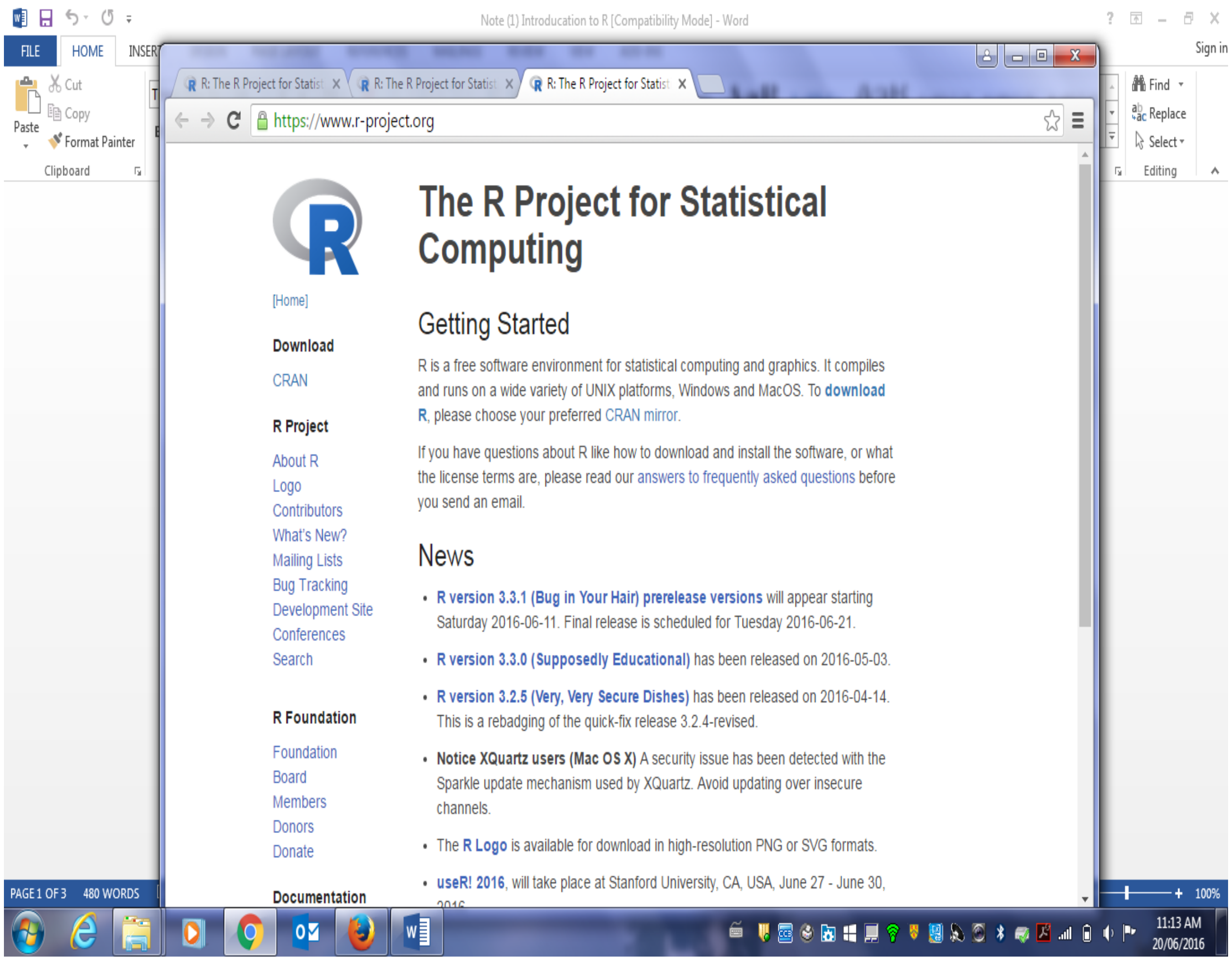
Dina Dawoud

and

Nagham Mohammad

What is R?

According to their site http://www.r-project.org/ :



The R Project for Statistical Computing

[Home]

Download

CRAN

R Project

About R
Logo
Contributors
What's New?
Mailing Lists
Bug Tracking
Development Site
Conferences
Search

R Foundation

Foundation
Board
Members
Donors
Donate

Documentation

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To download R, please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News

- R version 3.3.1 (Bug in Your Hair) prerelease versions will appear starting Saturday 2016-06-11. Final release is scheduled for Tuesday 2016-06-21.
- R version 3.3.0 (Supposedly Educational) has been released on 2016-05-03.
- R version 3.2.5 (Very, Very Secure Dishes) has been released on 2016-04-14. This is a rebadging of the quick-fix release 3.2.4-revised.
- Notice XQuartz users (Mac OS X) A security issue has been detected with the Sparkle update mechanism used by XQuartz. Avoid updating over insecure channels.
- The R Logo is available for download in high-resolution PNG or SVG formats.
- useR! 2016, will take place at Stanford University, CA, USA, June 27 - June 30, 2016.

"R is a language and environment for statistical computing and graphics."

"R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering ...) and graphical techniques, and is highly extensible."

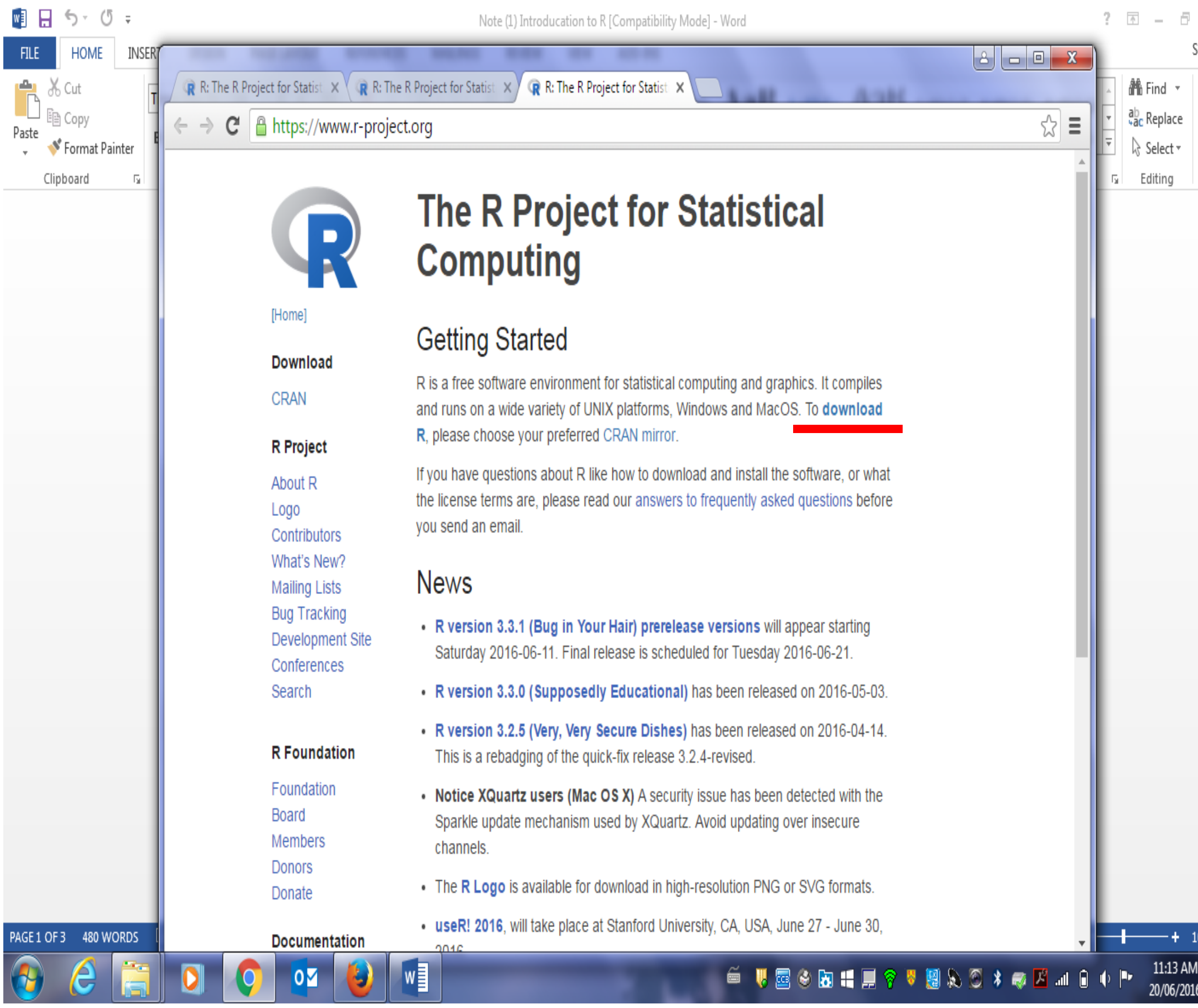
R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the *R Development Core Team*.

SCOPING:
function (x)
function (y)
x + y
Robert Gentleman
Ross Ihaka
Фото: http://www.nytimes.com/2009/01/07/technology/business-computing/07program.html

Obtaining a copy of the R applications

R is free software - see the R site above for the terms of use. It runs on a wide variety of platforms including UNIX, Windows and MacOS.

Download a copy of this application from their site: http://www.r-project.org/



https://www.r-project.org

The R Project for Statistical Computing

[Home]

Download

CRAN

R Project

About R
Logo
Contributors
What's New?
Mailing Lists
Bug Tracking
Development Site
Conferences
Search

R Foundation

Foundation
Board
Members
Donors
Donate

Documentation

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News

- **R version 3.3.1 (Bug in Your Hair) prerelease versions** will appear starting Saturday 2016-06-11. Final release is scheduled for Tuesday 2016-06-21.
- **R version 3.3.0 (Supposedly Educational)** has been released on 2016-05-03.
- **R version 3.2.5 (Very, Very Secure Dishes)** has been released on 2016-04-14. This is a rebadging of the quick-fix release 3.2.4-revised.
- **Notice XQuartz users (Mac OS X)** A security issue has been detected with the Sparkle update mechanism used by XQuartz. Avoid updating over insecure channels.
- The **R Logo** is available for download in high-resolution PNG or SVG formats.
- **useR! 2016**, will take place at Stanford University, CA, USA, June 27 - June 30, 2016.

R

Organization Web Site:	http://www.r-project.org/
Any Special Technical Requirements:	This is an open source product available for a variety of computing platforms. R for Mac OS X FAQ R for Windows FAQ
Available online:	

Helpful Resources

The following references would be a good step towards building a solid foundation in using R.

- The *R Project Homepage*
- *An Introduction to R* - written by W. N. Venables, D. M. Smith and the R Development Core Team
- *Short Reference Card* - written by Tom Short, EPRI PEAC
- *R Seek* helps you find the R function you require

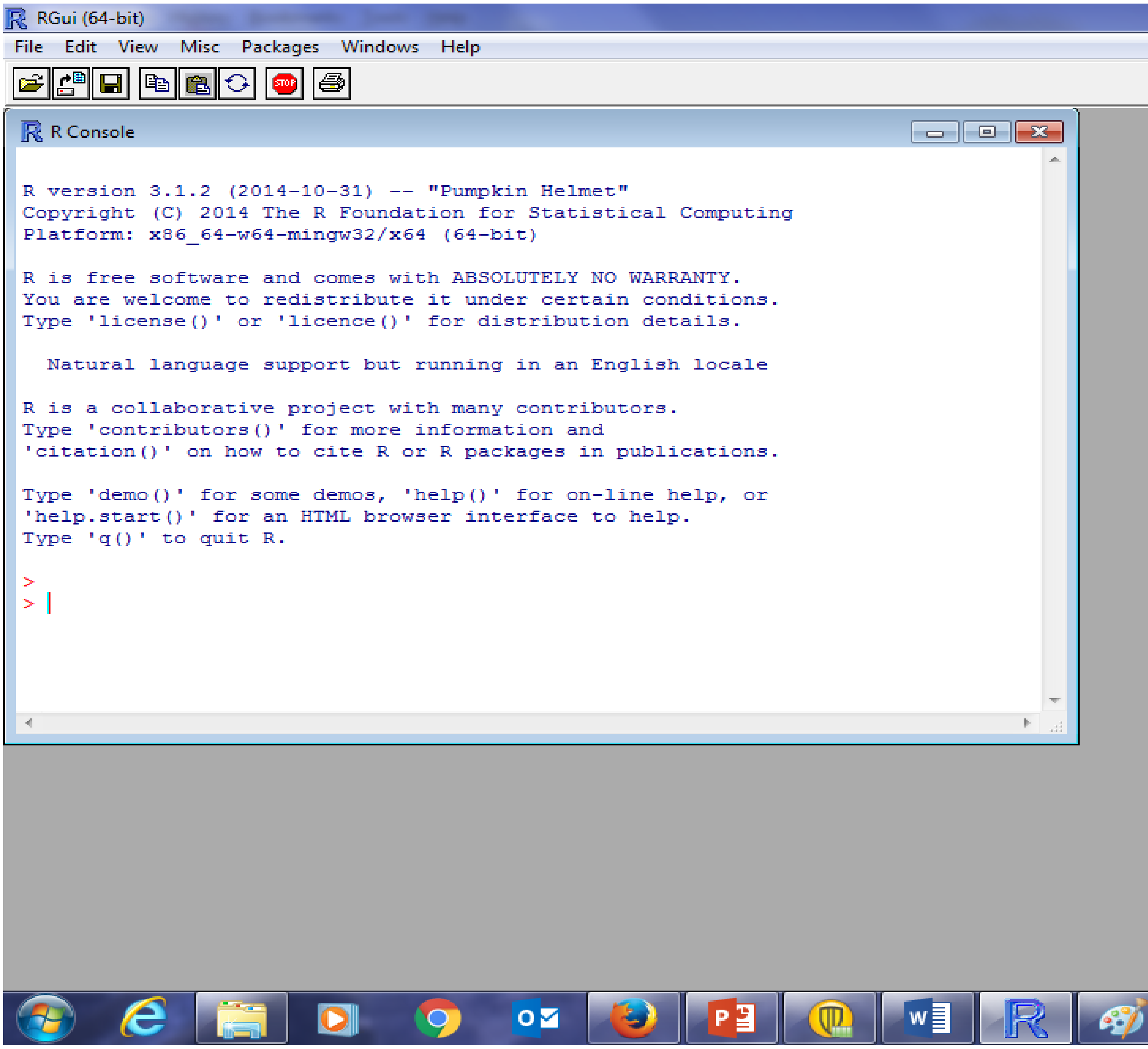
Book

Modern Applied Statistics with S. Fourth Edition by W. N. Venables and B. D. Ripley

Starting and Quitting R.

- Double clicking on the R icon starts the program.

- The first thing that will happen is that R will open the *console*, into which the user can type commands.



```
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-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.

  Natural language support but running in an English locale

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 greater-than sign (>) is the prompt symbol. When this appears, you can begin typing commands.

- R can be used as a calculator.

- We can type simple arithmetical expressions at the > prompt:

```
15 + 10
```

- Upon pressing the Enter key, the result 25 appears, prefixed by the number 1 in square brackets:

```
[1] 25
```

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

Notes

1) Help

If you need help understanding a command or its syntax type either (**?) command**, or (**help) command** and R will display the help available on this topic.

Examples

```
?mean
```

```
help(var)
```

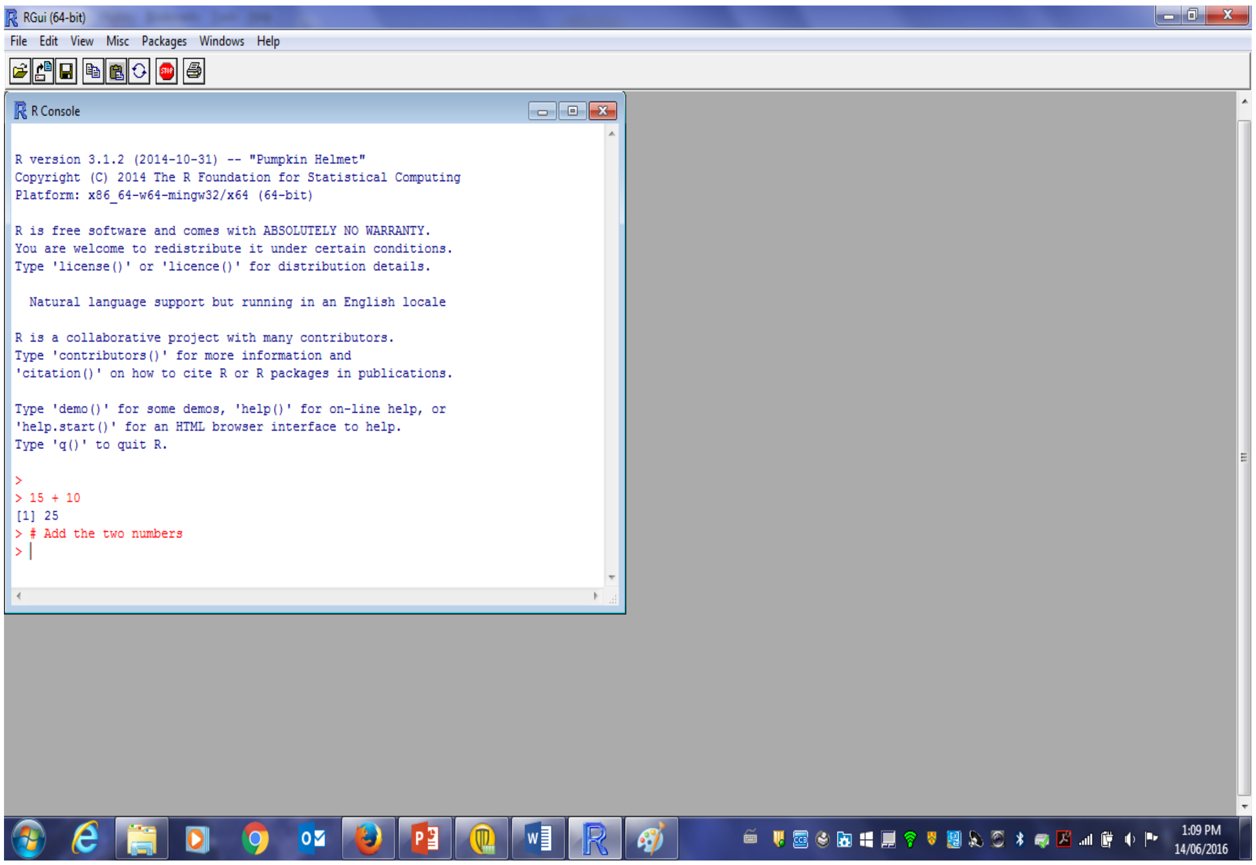
2) Certain functions in R may NOT run on all platforms (e.g., Windows, MAC, Linux, etc...) the same way.

Comments

Words after a pound sign (#) are considered comments and will not be executed by R.

Examples

```
 2 * 5   # "*" is the symbol for multiplication.
[1] 10

 3 - 10
[1] -7

12 / 4
[1] 3

3^4
[1] 81
```



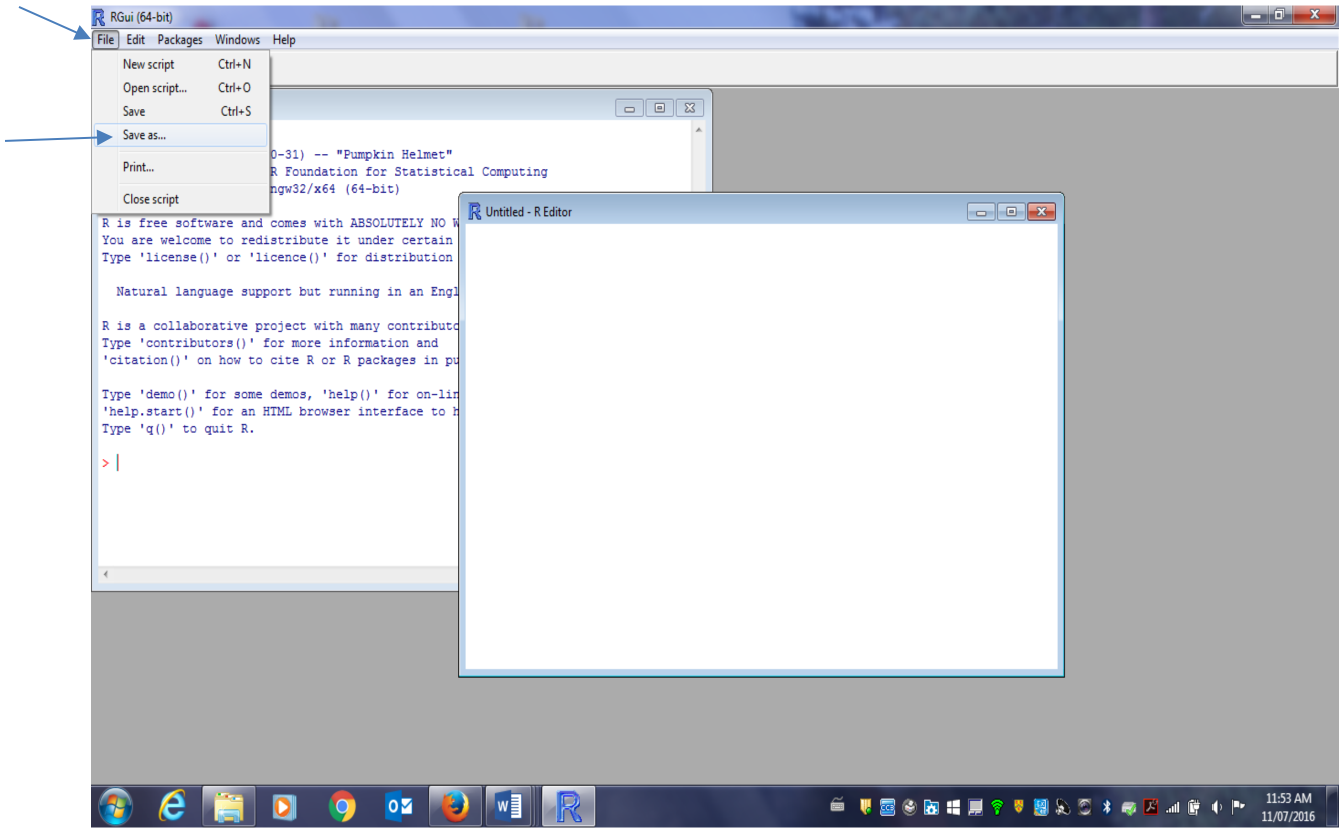
```
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

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.

  Natural language support but running in an English locale

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.

>
> 15 + 10
[1] 25
> # Add the two numbers
> |
```

- To quit your R session, type
  q( )

If you then hit the Enter key, you will be asked whether to save an image of the current workspace, or not, or to cancel. The workspace image contains a record of the computations you've done, and may contain some saved results. Hitting the Cancel option allows you to continue your current R session. We rarely save the current workspace image, but occasionally find it convenient to do so.

Recording your work

Rather than saving the workspace, we prefer to save the commands we entered. In Windows, the easiest way to do this is from the File menu. Open new script. At the end of a session, save the final script for a permanent record of your work.



```
0-31) -- "Pumpkin Helmet"
R Foundation for Statistical Computing
ngw32/x64 (64-bit)

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

  Natural language support but running in an Engl

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

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

> |
```

Named storage

R has a workspace known as the *global environment* that can be used to store the results of calculations, and many other types of objects.

Example

- Suppose we would like to store the result of the calculation 5^5 for future use.
- We will assign this value to an object called X. To this, we type

```
X <- 5^5
```

- We tell R to make the assignment using an arrow that points to the left, created with the less-than sign (<) and the hyphen (-).

- R also supports using the equals sign (=) in place of the arrow in most circumstances, but we recommend using the arrow, as it makes clear that we are requesting an *action* (i.e. an assignment), rather than stating a *relation* (i.e. that X is equal to 5^5).
- Note that when we hit Enter, nothing appears on the screen except a new prompt: R has done what we asked, and is waiting for us to ask for something else.
- We can see the results of this assignment by typing the name of our new object at the prompt:

```
  X
[1] 3125
```

Installing R packages

Part of the reason R has become so popular is the vast array of packages available at the cran and bioconductor repositories.

Go to the packages menu item ...Click ... go to install package(s)...You will see a list of CRAN mirror where you can download ... Click on the mirror close to you (Canada ON).. . R will take a couple of moments and then will provide you with all the available packages...Click on the one you need or a list of them by shift and click on others items.

If you want to choose more than one and they are not on order

Click on Control and hold it until you choose all of the packages you need.

Then click on ok

If you need more information about the packages in R go to the CRAN home page

http://cran.r-project.org/

To the left hand side go down to the link of packages click on that and it will take you to the contributed packages in R

Available Packages

Currently, the CRAN package repository features 7415 available packages.

Table of available packages, sorted by date of publication

Table of available packages, sorted by name

You will see all the available packages in R the same ones at the packages menu box. Or, you can install packages using the command

```
install.packages("ggplot2")

library(ggplot2)
```

Functions

Most of the work in R is done through *functions*.

Example (1)

Consider the standard formula for the sample mean of a sample $x_1 . . . x_n$ is

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

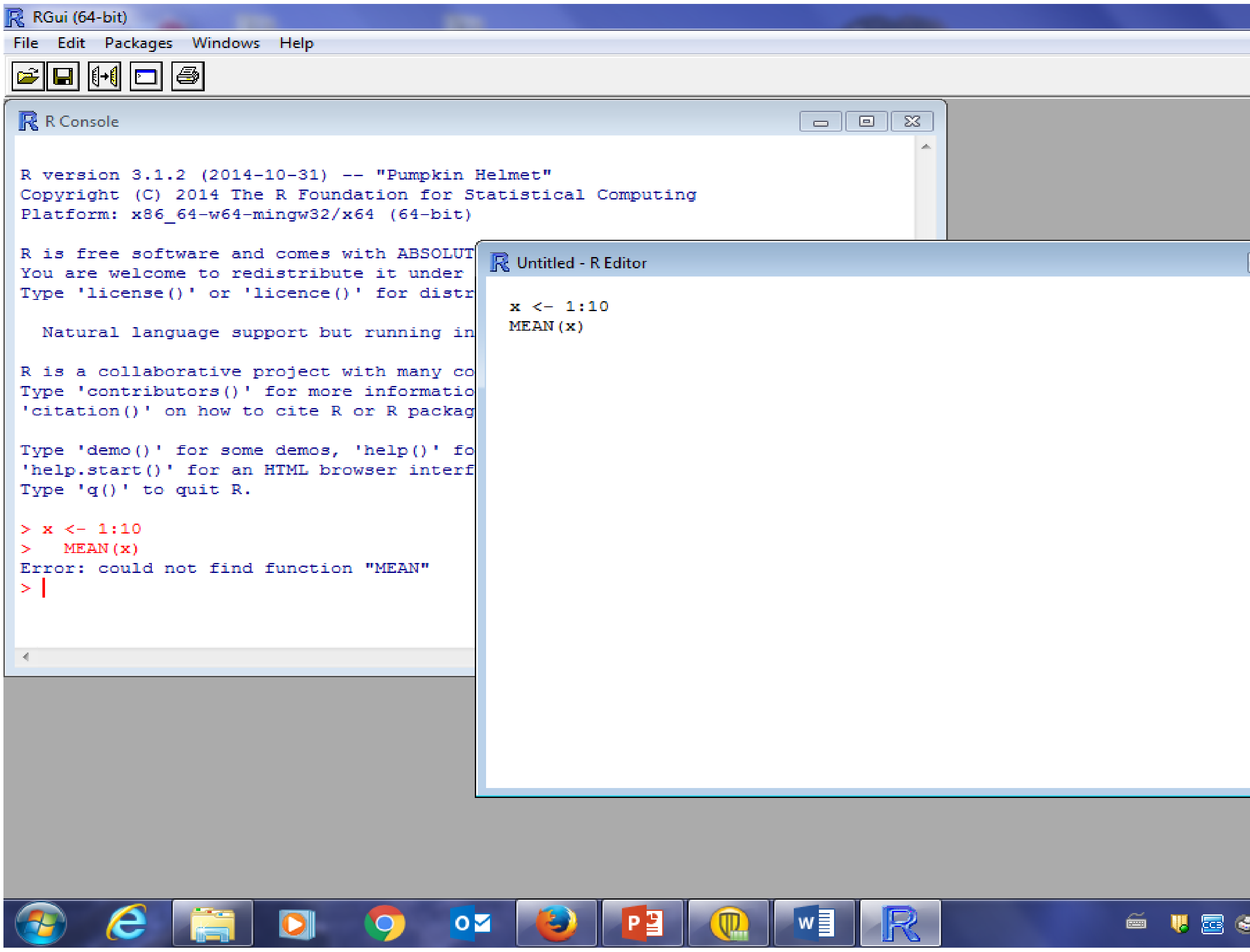
In R, mean (). For example:

```
  x <- 1:11

 mean(x)
[1] 6
```

R is case-sensitive

See what happens if you type

```
x <- 1:10
 MEAN(x)
Error: could not find function "MEAN"
```



```
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUT
You are welcome to redistribute it under
Type 'license()' or 'licence()' for distr

  Natural language support but running in

R is a collaborative project with many co
Type 'contributors()' for more informatio
'citation()' on how to cite R or R packag

Type 'demo()' for some demos, 'help()' fo
'help.start()' for an HTML browser interf
Type 'q()' to quit R.

> x <- 1:10
>   MEAN(x)
Error: could not find function "MEAN"
> |
```

```
x <- 1:10
MEAN(x)
```

Or

```
 mean(x)
[1] 5.5
```

Now try

```
 MEAN <- mean
 MEAN(x)
[1] 5.5
```

The function mean () is built in to R. R considers MEAN to be a different function, because it is case-sensitive: m is a different letter than M.

Vectors

- A numeric vector is a list of numbers. The c( ) function is used to collect things together into a vector. We can type

```
 c(0, 7, 8)
[1] 0 7 8
```

- We can assign this to a named object:

```
 x <- c(0, 7, 8) # now x is a 3-element vector
```

To see the contents of x, simply type

```
 x
[1] 0 7 8
```

- Symbol can be used to create sequences of increasing (or decreasing) values.

Example

```
 numbers2to30 <- 2:30
 numbers2to30

 numbers2to30
[1] 2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[25] 26 27 28 29 30


 x <- 1:11
 x

[1]  1  2  3  4  5  6  7  8  9 10 11
```

- Vectors can be joined together (i.e. concatenated) with the c function.

Example

```
 newvector <- c(numbers2to30, x)
 newvector

[1]  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[25] 26 27 28 29 30 1  2  3  4  5  6  7  8  9 10 11
```

Extracting elements from vectors

- A nicer way to display the 22nd element of newvector is to use square brackets to extract just that element:

```
  newvector[22]
[1] 23
```

- To get the fourth through eighth elements

```
  newvector[4:8]
[1] 5 6 7 8 9
```

- Negative indices can be used to avoid certain elements. For example,

we can select all but the second element of newvector as follows:

```
  newvector [-2]

 [1]  2  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
[25] 27 28 29 30  1  2  3  4  5  6  7  8  9 10 11
```

- The third through eleventh elements of newvector can be avoided as follows:

```
 newvector [-(3:11)]

[1]  2  3 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30  1  2  3  4
[25]  5  6  7  8  9 10 11
```

- Do not mix positive and negative indices. To see what happens, consider

```
 newvector [c(-2, 3)]

Error in newvector[c(-2, 3)] :
```

The problem is that it is not clear what is to be extracted: do we want the third element of newvector before or after removing the second one?

Vector Arithmetic

- Arithmetic can be done on R vectors. For example, we can multiply all elements of x by 3:

```
 x * 3
[1]  3  6  9 12 15 18 21 24 27 30 33
```

Note that the computation is performed element wise.

- Addition, subtraction and division by a constant have the same kind of effect. For example,

```
 y <- x - 5
 y
[1] -4 -3 -2 -1  0  1  2  3  4  5  6
```

Simple Patterned Vectors

- Patterned vectors can be produced using the seq( )function as well as the rep( )function.

Example, the sequence of odd numbers less than or equal to 17 can be obtained using

```
seq(1, 17, by=2)
[1]  1  3  5  7  9 11 13 15 17
```

Notice the use of by=2 here. The seq()function has several *optional parameters*, including one named by. If by is not specified, the default value of 1 will be used.

- Repeated patterns are obtained using rep(). Consider the following examples:

```
 rep(9, 12) # repeat the value 9, 12 times
[1] 9 9 9 9 9 9 9 9 9 9 9 9
```

```
 rep(seq(2, 20, by=2), 2) # repeat the pattern 2 4 ... twice

 [1]  2  4  6  8 10 12 14 16 18 20  2  4  6  8 10 12 14 16 18 20
```

Data frames (data.frame)

Data sets frequently consist of more than one column of data, where each column represents measurements of a single variable. Each row usually represents a single observation. This format is referred to as *case-by-variable* format.

Example
The following data set consists of four observations on the three variables x, y, z

x	y	z
61	13	4
175	21	18
111	24	14
124	23	18

```
x <- c( 61,175,111,124)
y <- c( 13,21,24,23)
z <- c( 4,18,14,18)
Mydata  <-  data.frame( x,y,z)
Mydata
```

You can see the names of the objects in your data.frame using the names()function,

```
names( Mydata)
```

You can extract parts of your data.frame

```
Mydata$x
Mydata$y
Mydata$z
```

Example

```
 colors <- c("red", "yellow", "blue")
 numbers <- c(1, 2, 3)
  Data1 <- data.frame(colors, numbers)
 Data1

  colors numbers
1    red       1
2 yellow       2
3   blue       3
```

Practice Question

Using rep() and seq()as needed, create the vectors

(a) 0 0 0 0 0 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4

and

(b) 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

Solution

```
(a) X <-  seq(0, 4, by=1)        ## or you can   X<-  c( 0, seq(1:4))
rep( X, each=5)

[1] 0 0 0 0 0 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4

(b) rep( seq(1:5), 4)

 [1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

Practice Problem

Consider the pressure data frame in R. There are two columns: temperature and pressure.

To work with the data set **pressure** in R

Step (1)

Type **pressure** in R, you suppose to get the data set, if not probably you need to install some packages and R will instruct you to their names.

packages ----- Install Package(s)--- Canada(ON)---- the name of the package

Step(2)

library(the name of the package you install)

Step(3)

Now type **pressure** again you should get the data

Step(4)

To obtain a specific column in pressure, use:

x1<- pressure$temperature

x2<-pressure$pressure

Note : Before working with the data do the following

Add two numbers to each observation of the variable that you will work with based on your student number; so if my student number is (117893) I will add the number 93.

Example

Suppose the first observation (data point) in your variable is 7 then it will be (7+93=100) and the same for all the other n remaining observations.

Note: Do not do that by your hand let R do that for you, use

```
x1<- x1+93
x2<-x2+93.
```

(a) Find the mean, median, variance, and standard deviation of the temperature variable.

(b) Find the mean, median, variance, and standard deviation of the pressure variable.

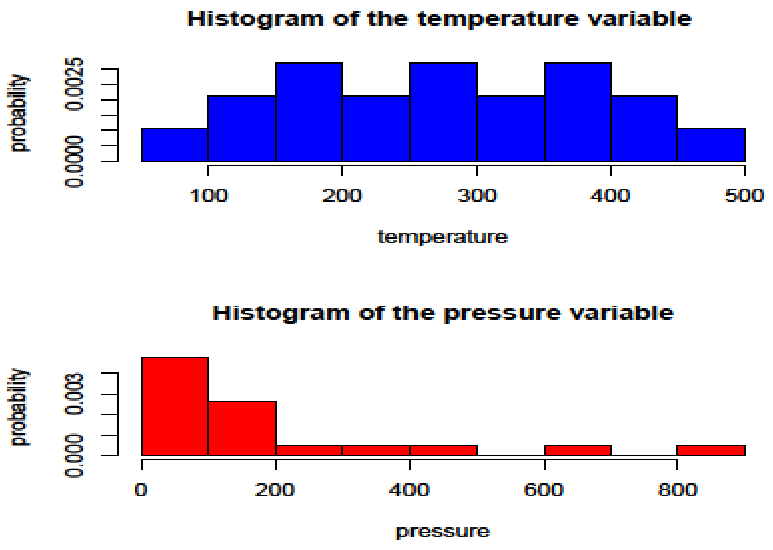
(c) Plot the probability histogram for the temperature and the pressure using the par function to display them in a 2 × 1 layout on the graphics page. Repeat once again using a 1 × 2 layout with different colors, use

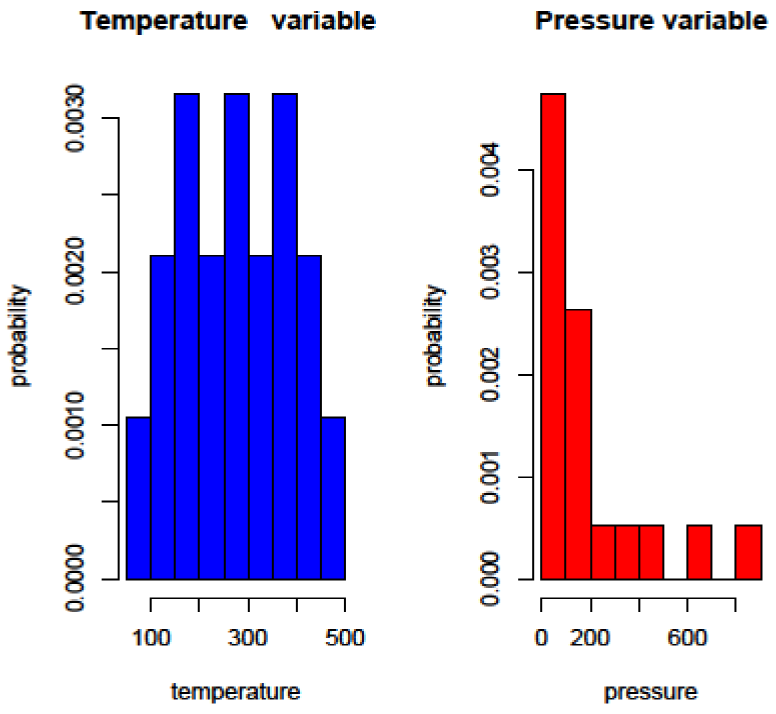
```
par(mfrow=c(2,1))
par(mfrow=c(1,2))
```

Add a suitable title and labels to the graphs.

Solution

```
pressure
  x1 <-  pressure$temperature
  x2 <-  pressure$pressure
  x1 <-  x1+93 #temperature
  x2 <-  x2+93 #pressure
  mean(x1) # [1] 273
  median(x1) #273
  sd(x1) #[1] 112.5463


  par(mfrow=c(2,1))
  hist(x1,col="blue",probability = TRUE,ylab="probability"
,xlab="temperature",main="Histogram of the temperature variable ")
  hist(x2,col="red",probability = TRUE,ylab="probability" ,
xlab="pressure",main="Histogram of the pressure variable")
  par(mfrow=c(1,2))
  hist(x1,col="blue",probability = TRUE,ylab="probability",
xlab="temperature",main="Temperature   variable  ")
  hist(x2,col="red",probability = TRUE,ylab="probability",
xlab="pressure",main="Pressure variable")
```


Histogram of the temperature variable
probability
0.0000
0.0025
100
200
300
400
500
temperature
Histogram of the pressure variable
probability
0.000
0.003
0
200
400
600
800
pressure


Temperature variable
probability
0.0000
0.0010
0.0020
0.0030
100
300
500
temperature
Pressure variable
probability
0.000
0.001
0.002
0.003
0.004
0
200
600
pressure