Introduction to R

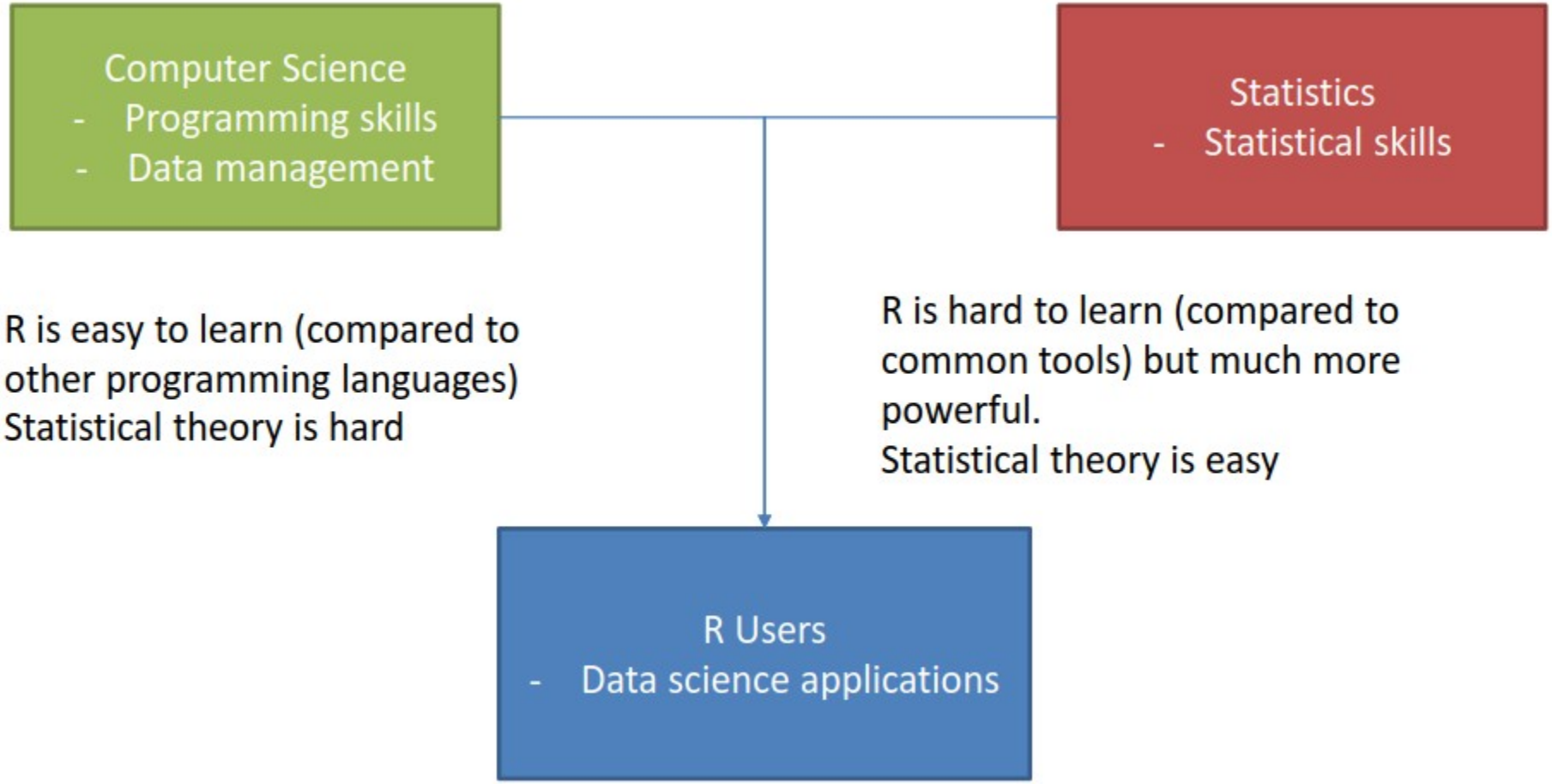
Prof Farhi Marir
Director of Big Data Analytics Research Lab,
College of Technological Innovation,
Zayed University

Lecture Content

- Introduction to R
  - GUI,
  - Data Import & Export
  - Attributes & Data Types,
  - Descriptive Statistics

Introduction to R

- R is a programming language and software framework for statistical analysis and graphics,



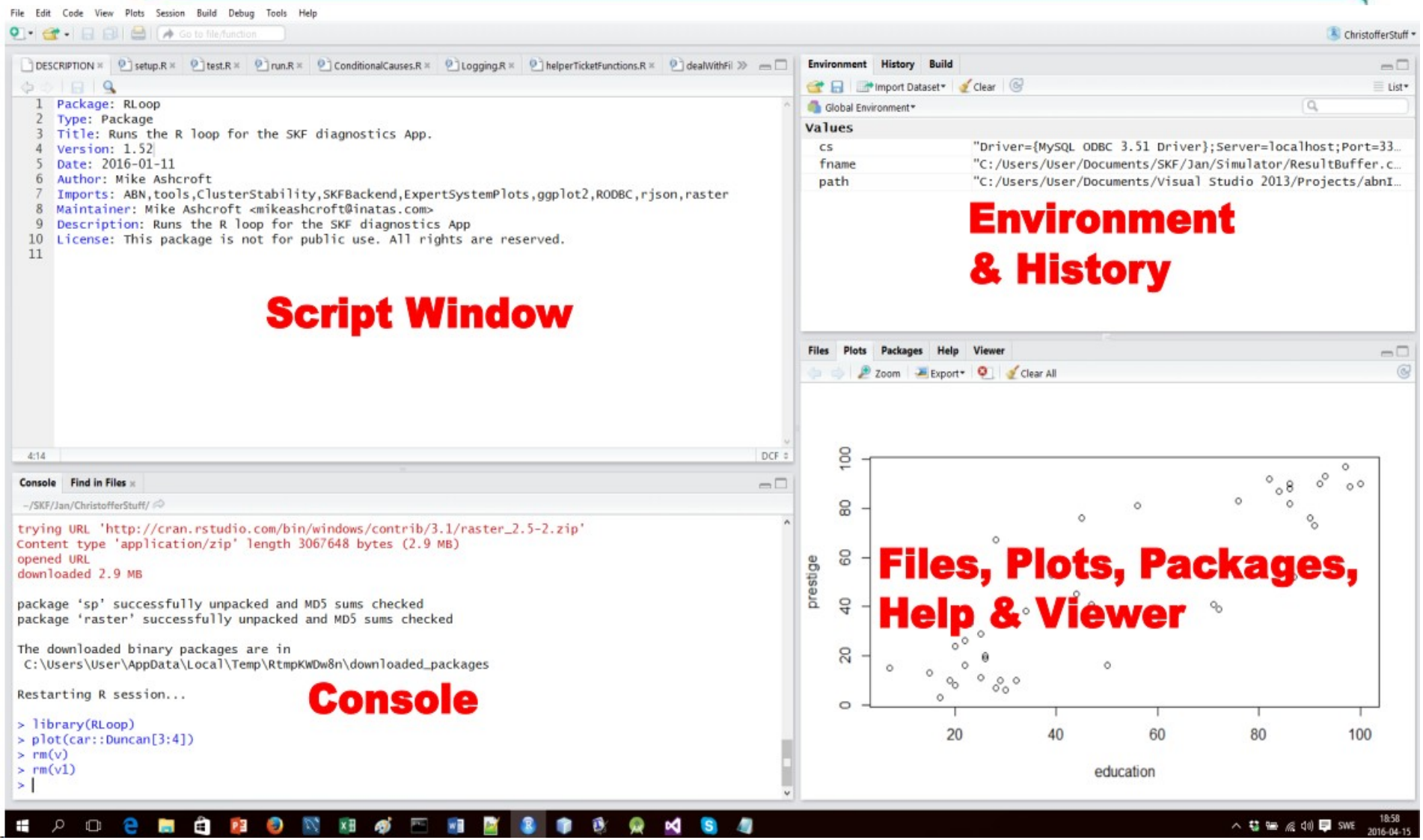
Installing R & R Studio

- You should install R & R Studio as follow:

  https://www.youtube.com/watch?v=NZxSA80lF1I

- The Website contains for R documentation is: https://cran.r-project.org/doc/manuals/r-release/R-intro.html

R Studio


File Edit Code View Plots Session Build Debug Tools Help
DESCRIPTION setup.R test.R run.R ConditionalCauses.R Logging.R helperTicketFunctions.R dealWithFil
Package: RLoop
Type: Package
Title: Runs the R loop for the SKF diagnostics App.
Version: 1.52
Date: 2016-01-11
Author: Mike Ashcroft
Imports: ABN,tools,ClusterStability,SKFBackend,ExpertSystemPlots,ggplot2,RODBC,rjson,raster
Maintainer: Mike Ashcroft <mikeashcroft@inatas.com>
Description: Runs the R loop for the SKF diagnostics App
License: This package is not for public use. All rights are reserved.
Script Window
Console Find in Files
~/SKF/Jan/ChristofferStuff/
trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.1/raster_2.5-2.zip'
Content type 'application/zip' length 3067648 bytes (2.9 MB)
opened URL
downloaded 2.9 MB
package 'sp' successfully unpacked and MD5 sums checked
package 'raster' successfully unpacked and MD5 sums checked
The downloaded binary packages are in
C:\Users\User\AppData\Local\Temp\RtmpKWDw8n\downloaded_packages
Restarting R session...
Console
> library(RLoop)
> plot(car::Duncan[3:4])
> rm(v)
> rm(v1)
Environment History Build
Import Dataset Clear
Global Environment
Values
cs "Driver={MySQL ODBC 3.51 Driver};Server=localhost;Port=33...
fname "C:/Users/User/Documents/SKF/Jan/Simulator/ResultBuffer.c...
path "C:/Users/User/Documents/Visual Studio 2013/Projects/abnI...
Environment & History
Files Plots Packages Help Viewer
Zoom Export Clear All
prestige
education
Files, Plots, Packages, Help & Viewer

IDE

Console

- Where you type commands and receive text output.

Script Window

- Script files are text files used to store scripts of R commands. Multiple can be open at once.
- Source runs an entire file.
- Run runs a highlighted selection.
- Write multiline code, including functions, in a script file and then run them from there.

IDE

Environment & History

•Environment – Display the objects (including functions) present in the environment.

•History – Display commands previously entered into the console.

Files, Plots, Packages, Help & Viewer Window

•Files – Navigate your computer's file system. Double clicking a file will open it in the script window.

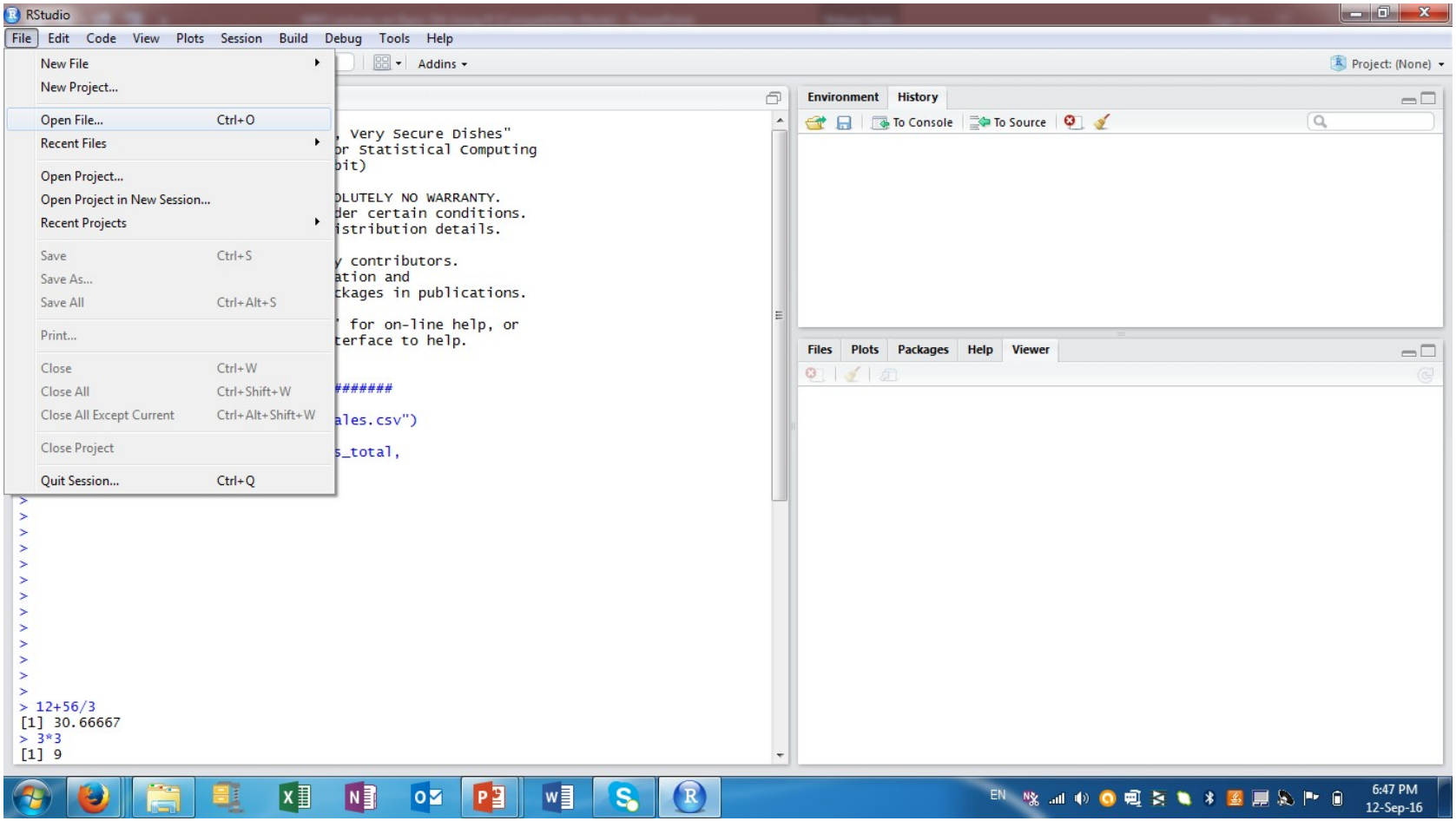
•Plots – Basic graphic output. Export graphics using the export button.

•Packages – Manage packages.

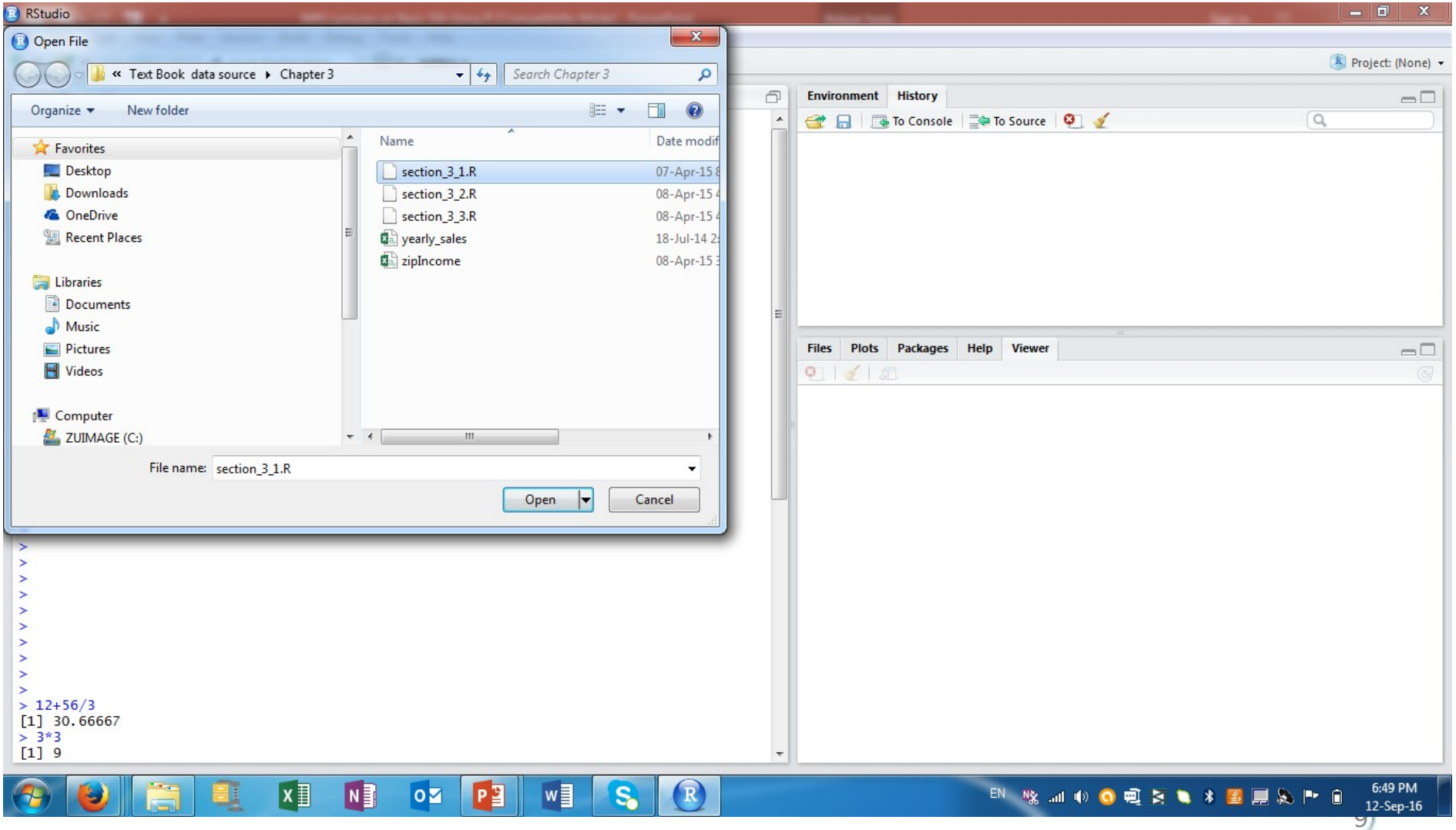
•Help – Displays help information.

•Viewer – Used to view local web content, web graphics and local web applications. We will not use it.

Open File/Project



Section 3.1 R code



Section 3.1 in Script Window

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function | Addins | Project: (None)

section_3_1.R

Source on Save | Run | Source

```
1
2  ##########################################
3  # section 3.1 Introduction to R
4  ##########################################
5
6  # import a csv file of the total annual sales for each customer
7  sales <- read.csv("c:/data/yearly_sales.csv")
8
9  # examine the imported dataset
10 head(sales)
11 summary(sales)
12
13 # plot num_of_orders vs. sales
14 plot(sales$num_of_orders,sales$sales_total,
15      main="Number of Orders vs. Sales")
16
17 # perform a statistical analysis (fit a linear regression model)
18 results <- lm(sales$sales_total ~ sales$num_of_orders)
19 results
20 summary(results)
```

1:1 (Top Level) R Script

Console ~/

```
R version 3.2.5 (2016-04-14) -- "Very, Very Secure Dishes"
Copyright (C) 2016 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.

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.
```

Environment | History

To Console | To Source

Files | Plots | Packages | Help | Viewer

EN 6:50 PM 12-Sep-16

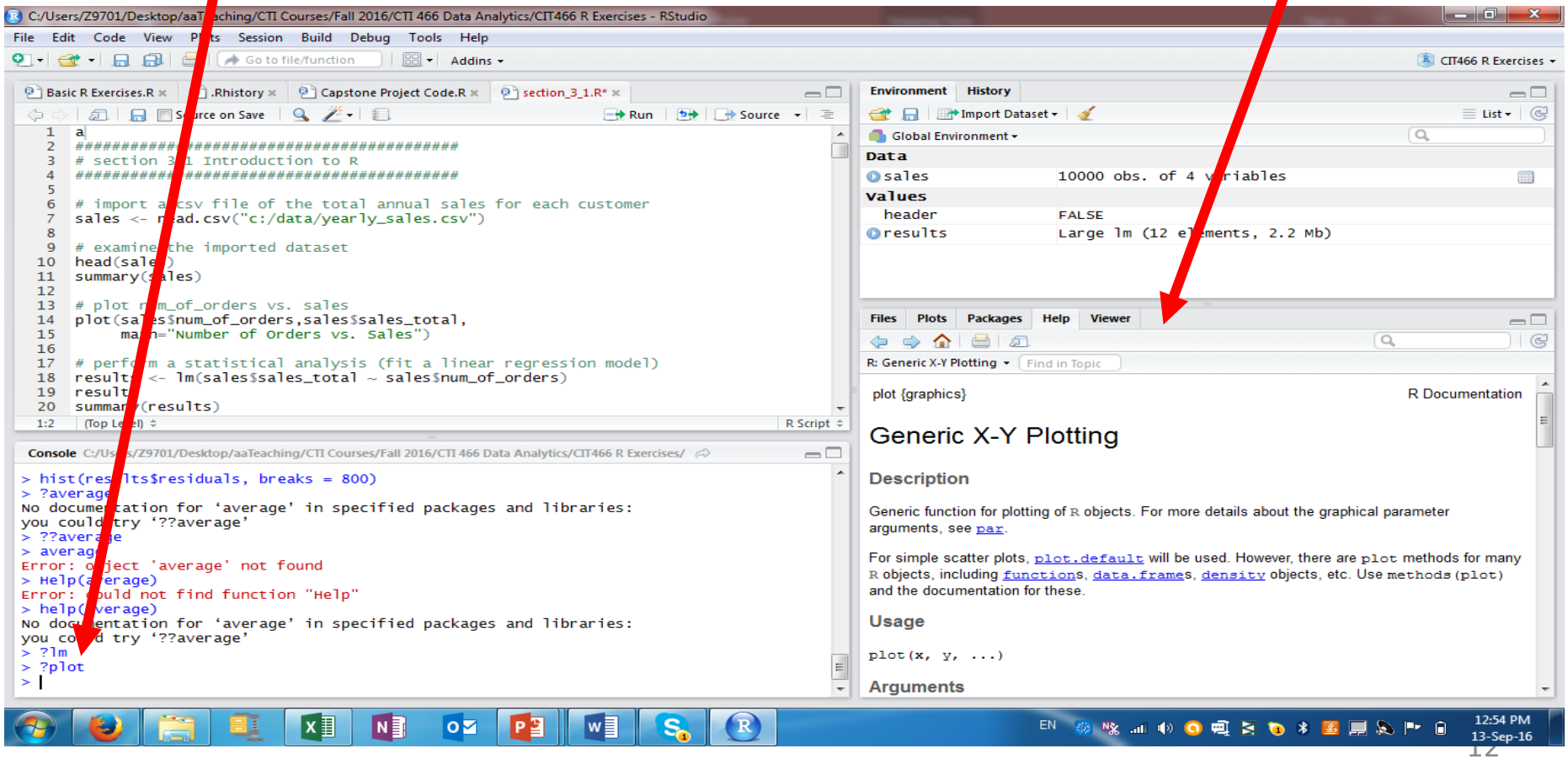
Introduction to R

```
# import a csv file of the total annual sales for each customer
sales <- read.csv("c:/data/yearly_sales.csv")

# examine the imported dataset
head(sales)
summary(sales)

# plot num_of_orders vs. sales
plot(sales$num_of_orders,sales$sales_total, main="Number of Orders vs. Sales")
```

Accessing Help in R Studio

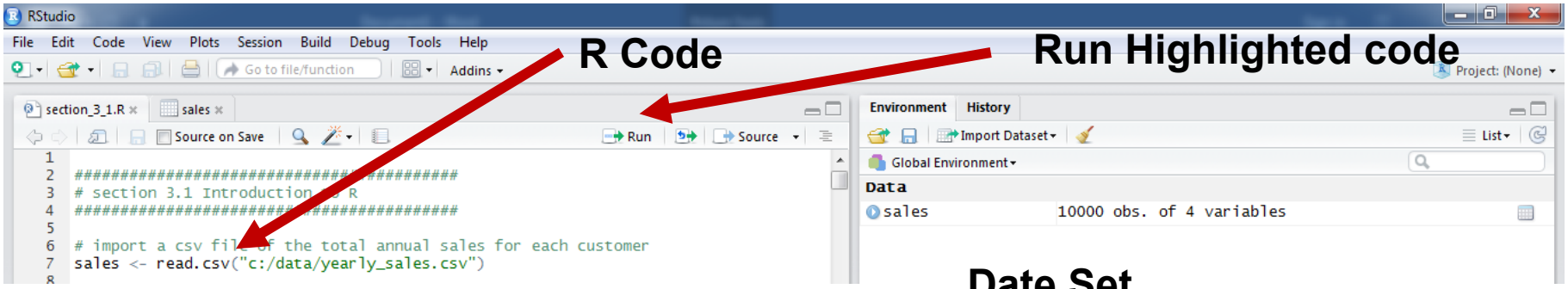
You can either use help(R function) or use ? R command/function

Below ?plot asks R to explain what Plot means and response in Help Window

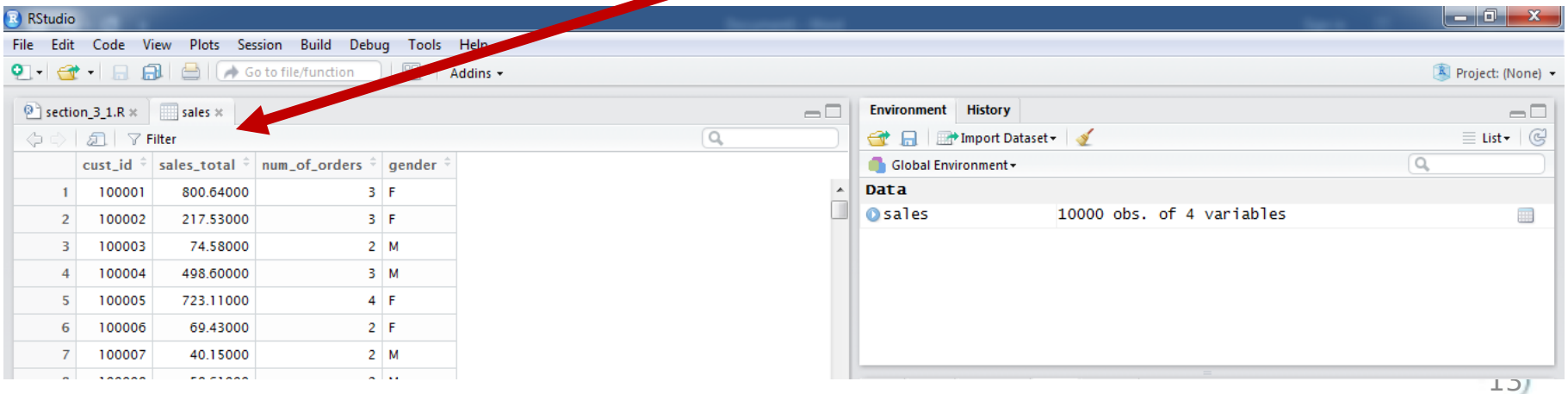


Import CSV Data Set file

sales <- read.csv("c:/data/yearly_sales.csv") means Import *yearly_sales.csv* dataset file and **(<-)** means save it into a file called *Sales*



Read-csv imports the *Yearly_sales.csv* file and save it into the file *Sales*



Head () Function

Head (Sales) function by default list the six Records of Sales as shown below

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function Addins

section_3_1.R × sales ×

Source on Save Run Source

```
1
2  ##########################################
3  # section 3.1 Introduction to R
4  ##########################################
5
6  # import a csv file of the total annual sales for each customer
7  sales <- read.csv("c:/data/yearly_sales.csv")
8
9  # examine the imported dataset
10 head(sales)
11 summary(sales)
12
13 # plot num_of_orders vs. sales
14 plot(sales$num_of_orders,sales$sales_total,
15      main="Number of Orders vs. Sales")
16
17 # perform a statistical analysis (fit a linear regression model)
18 results <- lm(sales$sales_total ~ sales$num_of_orders)
19 results
20 summary(results)
```

10:12 (Top Level) R Script

Console ~/

```
 Mean   :105001   Mean   : 249.46   Mean   : 2.428
 3rd Qu.:107500   3rd Qu.: 295.50   3rd Qu.: 3.000
 Max.   :110000   Max.   :7606.09   Max.   :22.000
> view(sales)
> help(plot)
> sales <- read.csv("c:/data/yearly_sales.csv")
> head(sales)
  cust_id sales_total num_of_orders gender
1  100001      800.64             3      F
2  100002      217.53             3      F
3  100003       74.58             2      M
4  100004      498.60             3      M
5  100005      723.11             4      F
6  100006       69.43             2      F
>
```

Summary() Function

- Summary() Function provides some descriptive statistics such as Means and Median, etc.

```r
##########################################
# section 3.1 Introduction to R
##########################################

# import a csv file of the total annual sales for each customer
sales <- read.csv("c:/data/yearly_sales.csv")

# examine the imported dataset
head(sales)
summary(sales)

# plot num_of_orders vs. sales
plot(sales$num_of_orders,sales$sales_total,
     main="Number of Orders vs. Sales")

# perform a statistical analysis (fit a linear regression model)
results <- lm(sales$sales_total ~ sales$num_of_orders)
results
summary(results)
```

11:15 (Top Level) R Script

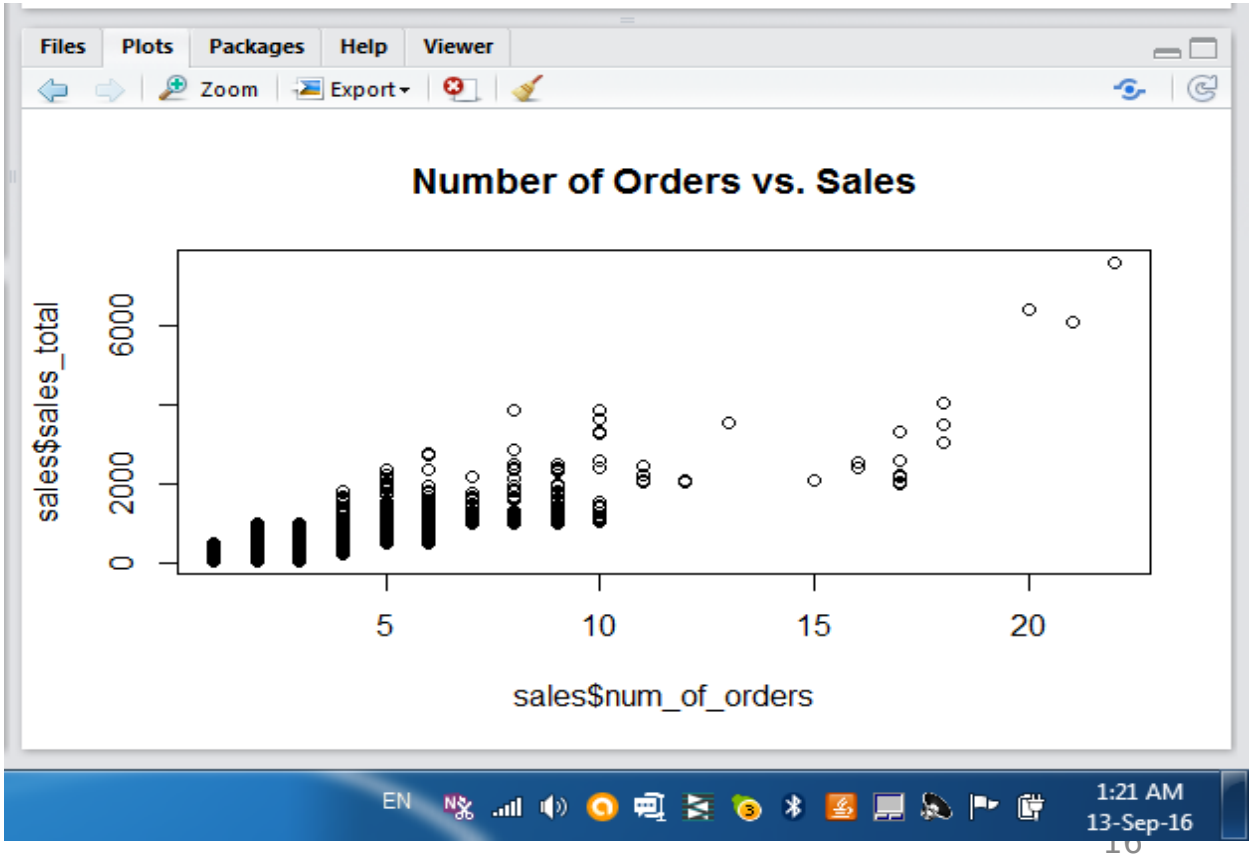
Console ~/

```
  cust_id sales_total num_of_orders gender
1  100001      800.64             3      F
2  100002      217.53             3      F
3  100003       74.58             2      M
4  100004      498.60             3      M
5  100005      723.11             4      F
6  100006       69.43             2      F
> summary(sales)
    cust_id        sales_total       num_of_orders    gender
 Min.   :100001   Min.   :  30.02   Min.   : 1.000   F:5035
 1st Qu.:102501   1st Qu.:  80.29   1st Qu.: 2.000   M:4965
 Median :105001   Median : 151.65   Median : 2.000
 Mean   :105001   Mean   : 249.46   Mean   : 2.428
 3rd Qu.:107500   3rd Qu.: 295.50   3rd Qu.: 3.000
 Max.   :110000   Max.   :7606.09   Max.   :22.000
>
```

Plot () function

- Plotting a dataset's content can provide information about the relationships between the various column,
- In this example, Plot() function generate a scatterplot of the number of orders (*Sales$sum_of_orders*) against the annual sales (*Sales$sales_toltal*)
- NB: $ selects an attribute of a table e.g. *sum_of_orders* attribute of *Sales* Table

```
# plot num_of_orders vs. sales
plot(sales$num_of_orders,sales$sales_total,main="Number of Orders vs. Sales")
```



Data Import and Export

Example of CSV files

Weights3468.csv - Notepad

File Edit Format View Help

```
"Date","Weight""Wed Jun 30 08:00:01 GMT 20
Jun 29 08:00:01 GMT 2010","180.2""Mon Jun
2010","180.2""Sun Jun 27 08:00:01 GMT 2010
Jun 26 08:00:01 GMT 2010","180.2""Fri Jun
2010","180.2""Thu Jun 24 08:00:01 GMT 2010
Jun 23 08:00:01 GMT 2010","180.2""
2010","181.4""Mon Jun 21 08:00:01
Jun 20 08:00:01 GMT 2010","181.4"
2010","181.4""Fri Jun 18 08:00:01
Jun 17 08:00:01 GMT 2010","181.4"
2010","181.4""Tue Jun 15 08:00:01
Jun 14 08:00:01 GMT 2010","181.4"
2010","181.4""Sat Jun 12 08:00:01
Jun 11 08:00:01 GMT 2010","180.0"
2010","180.0""Wed Jun 09 08:00:01
Jun 08 08:00:01 GMT 2010","180.0"
2010","180.0""Sun Jun 06 08:00:01
Jun 05 08:00:01 GMT 2010","178.2"
2010","178.2""Thu Jun 03 08:00:01
Jun 02 08:00:01 GMT 2010","178.2"
2010","178.2""Mon May 31 08:00:01
May 30 08:00:01 GMT 2010","178.2"
2010","178.2""Fri May 28 08:00:01
May 27 08:00:01 GMT 2010","178.2"
2010","178.2""Tue May 25 08:00:01
May 24 08:00:01 GMT 2010","178.2"
2010","178.2""Sat May 22 08:00:01
May 21 08:00:01 GMT 2010","178.2"
```

widgets.csv - Notepad

File Edit Format View Help

```
Widget1, blue, £10
Widget2, red, £12
Widget3, green, £14
Widget4, black, £16
Widget5, white, £18
```

Text file - Notepad

File Edit Format View Help

```
Index
One
Print Runs (x1000)
Page numbers
Orders (x1000)
 1   1   2800    22.
 2   1   2670    14.
 3   1   2800    37.
 4   1   2784    15.
 5   1   2800    38.
 6   1   2620   172.
 7   1   2620   249.
 8   1   2470    84.
 9   1   2620   242.
10   1   2475   100.
11   1   2620   114.
```

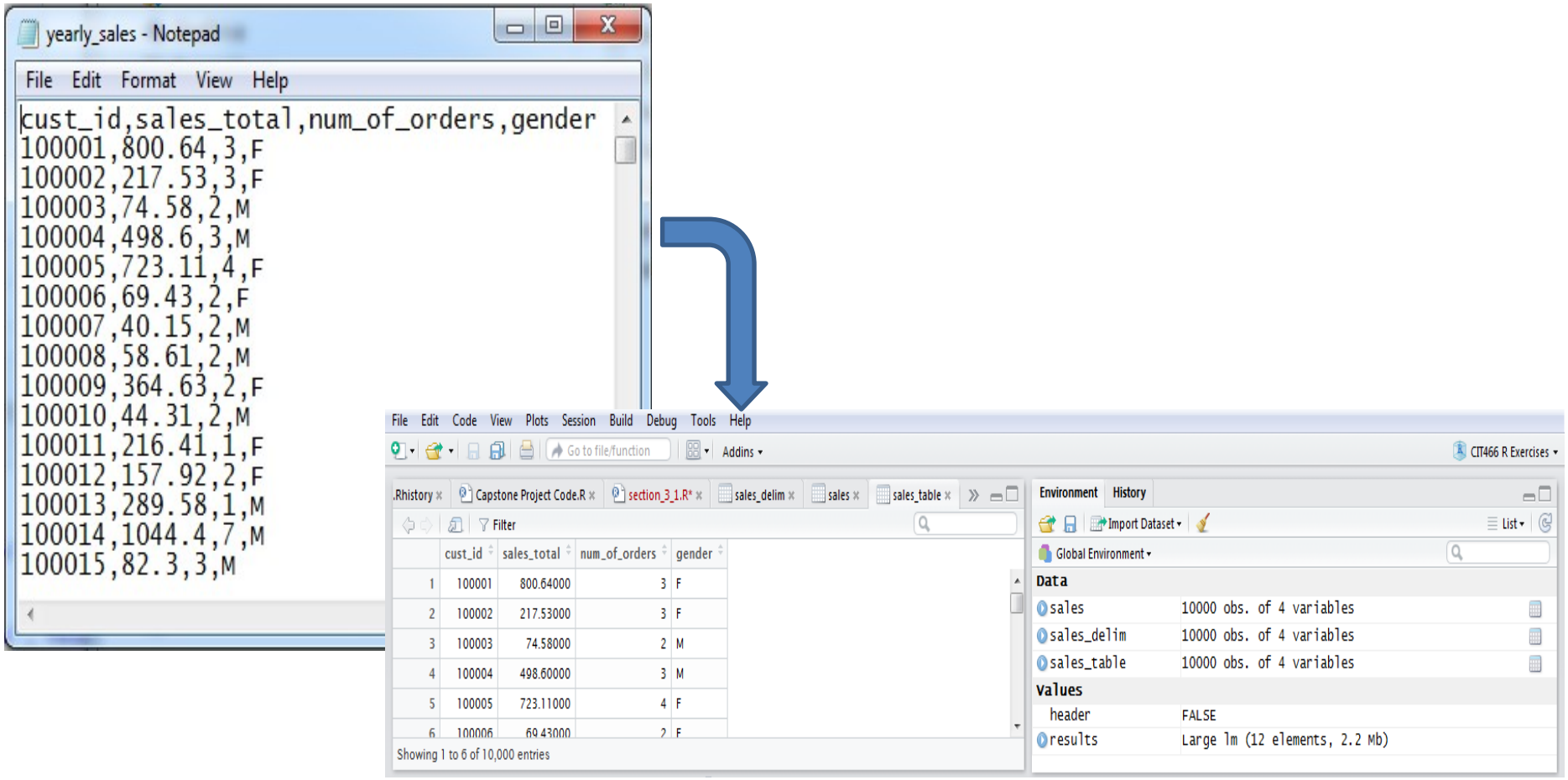
yearly_sales - Notepad

File Edit Format View Help

```
cust_id,sales_total,num_of_orders,gender
100001,800.64,3,F
100002,217.53,3,F
100003,74.58,2,M
100004,498.6,3,M
100005,723.11,4,F
100006,69.43,2,F
100007,40.15,2,M
100008,58.61,2,M
100009,364.63,2,F
100010,44.31,2,M
100011,216.41,1,F
100012,157.92,2,F
100013,289.58,1,M
100014,1044.4,7,M
100015,82.3,3,M
```

Usage of read.csv function

read.csv() converts Comma Separated Values (CSV) file into formatted Column & Row table and upload into R aeropospace as shown below



Data Import and Export

- In the annual Sales example the dataset was imported using *read.csv* as follow:
- To simplify multiple files with long path names, the *setwd()* function can be used to set the working directory for subsequent import and export as follows:

```
setwd("c:/data/")
sales <- read.csv("yearly_sales.csv")
```

- Other import function include *read.table()* and *read.delim()* *function are also used* to import CSV files like *yearly_Sales.csv* or other common files such as TXT.
- There are also two additional R functions: *read.csv2()* and *read.delim2()*

```
sales <- read.csv("c:/data/yearly_sales.csv")
```

Main Differences between R Import Functions

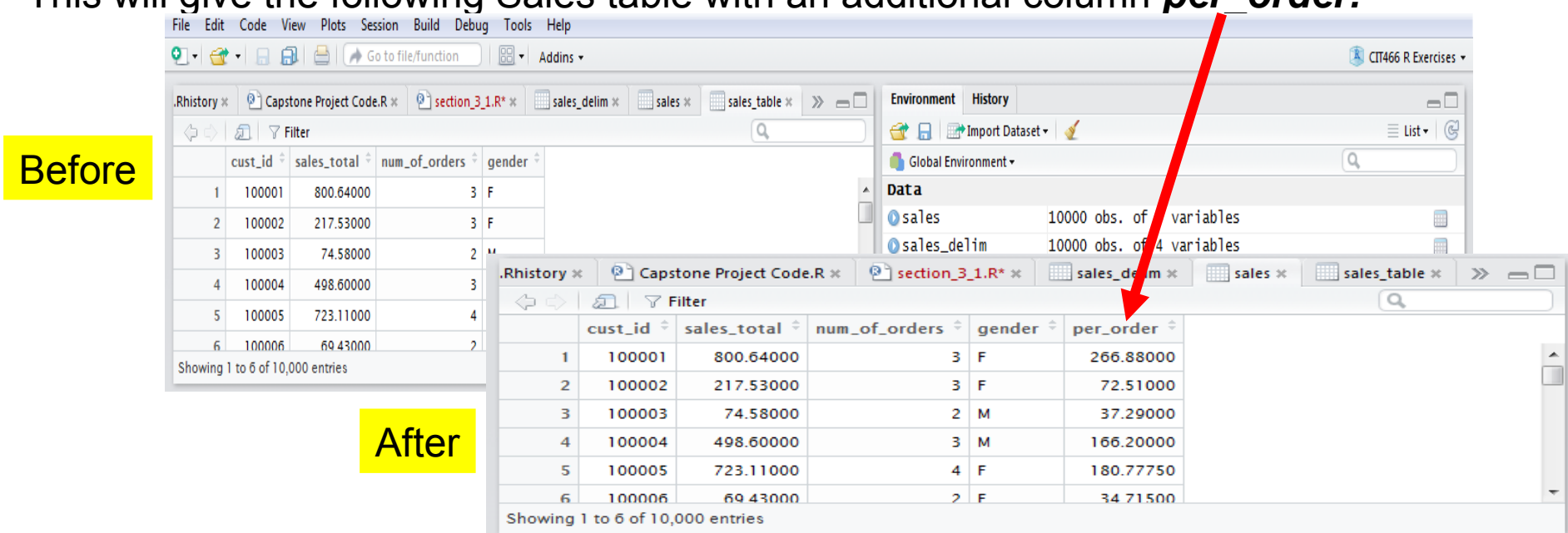
Function	Headers	Separators	Decimal Points
read.table()	FALSE	“ ”	“ ”
read.csv()	TRUE	“ ”	“ ”
read.csv2()	TRUE	“;”	“ ”
read.delim()	TRUE	“\t”	“ ”
read.delim2()	TRUE	“\t”	“ ”

R Export Functions

- The analogue R functions are *write.table(), write.csv()* and *write.csv2()* *enable exporting of R data sets to an external file*
- *Example below show making change to Sales file and exporting it*

```r
# add a column for the average sales per order
sales$per_order <- sales$sales_total/sales$num_of_orders
# export data as tab delimited without the row names
write.table(sales,"sales_modified.txt", sep="\t", row.names=FALSE)
```

This will give the following Sales table with an additional column ***per_order:***



	cust_id	sales_total	num_of_orders	gender
1	100001	800.64000	3	F
2	100002	217.53000	3	F
3	100003	74.58000	2	M
4	100004	498.60000	3	
5	100005	723.11000	4	
6	100006	69.43000	2	

	cust_id	sales_total	num_of_orders	gender	per_order
1	100001	800.64000	3	F	266.88000
2	100002	217.53000	3	F	72.51000
3	100003	74.58000	2	M	37.29000
4	100004	498.60000	3	M	166.20000
5	100005	723.11000	4	F	180.77750
6	100006	69.43000	2	F	34.71500