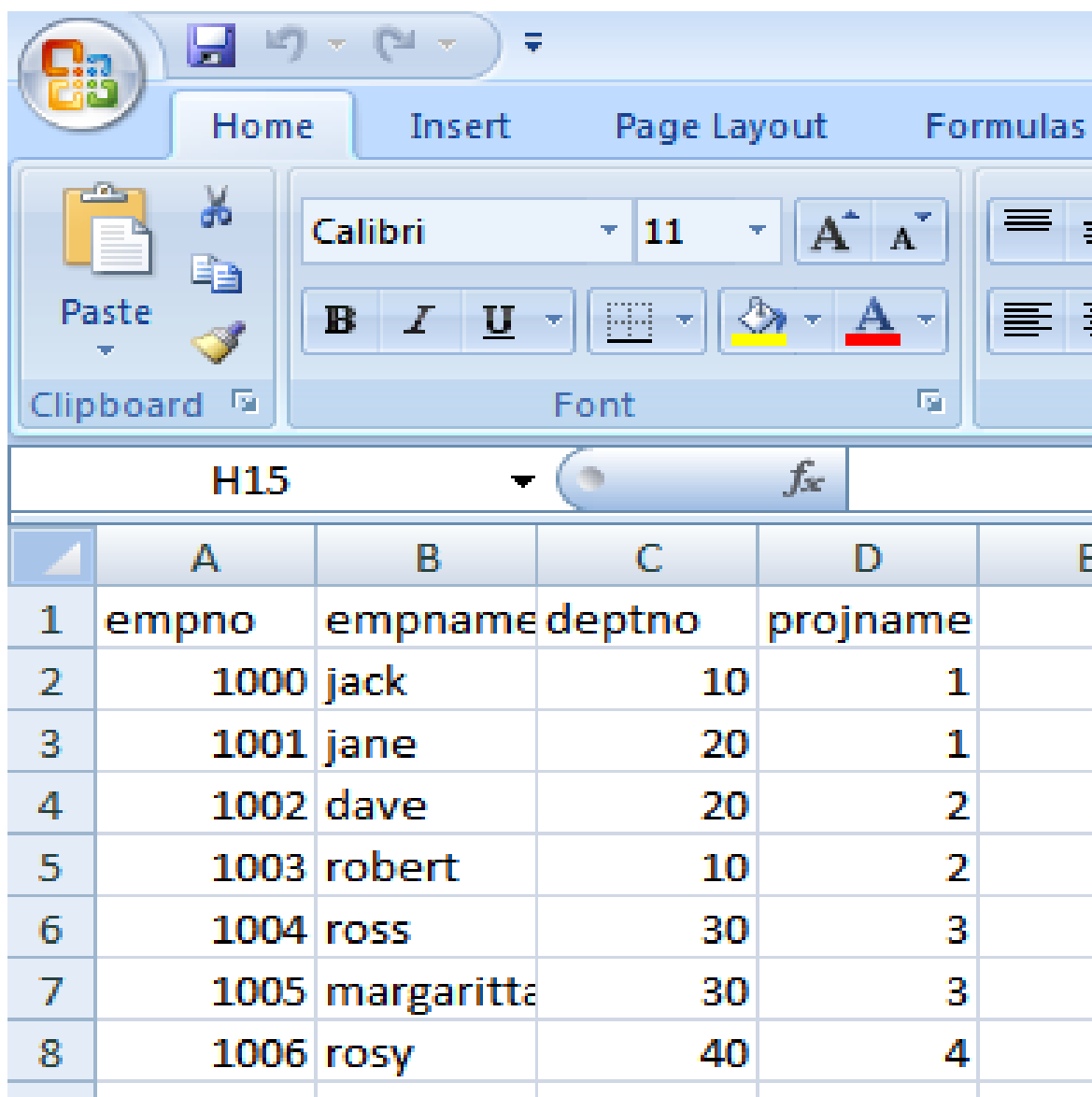


EXP 3 - READING AND WRITING DIFFERENT TYPES OF DATASETS

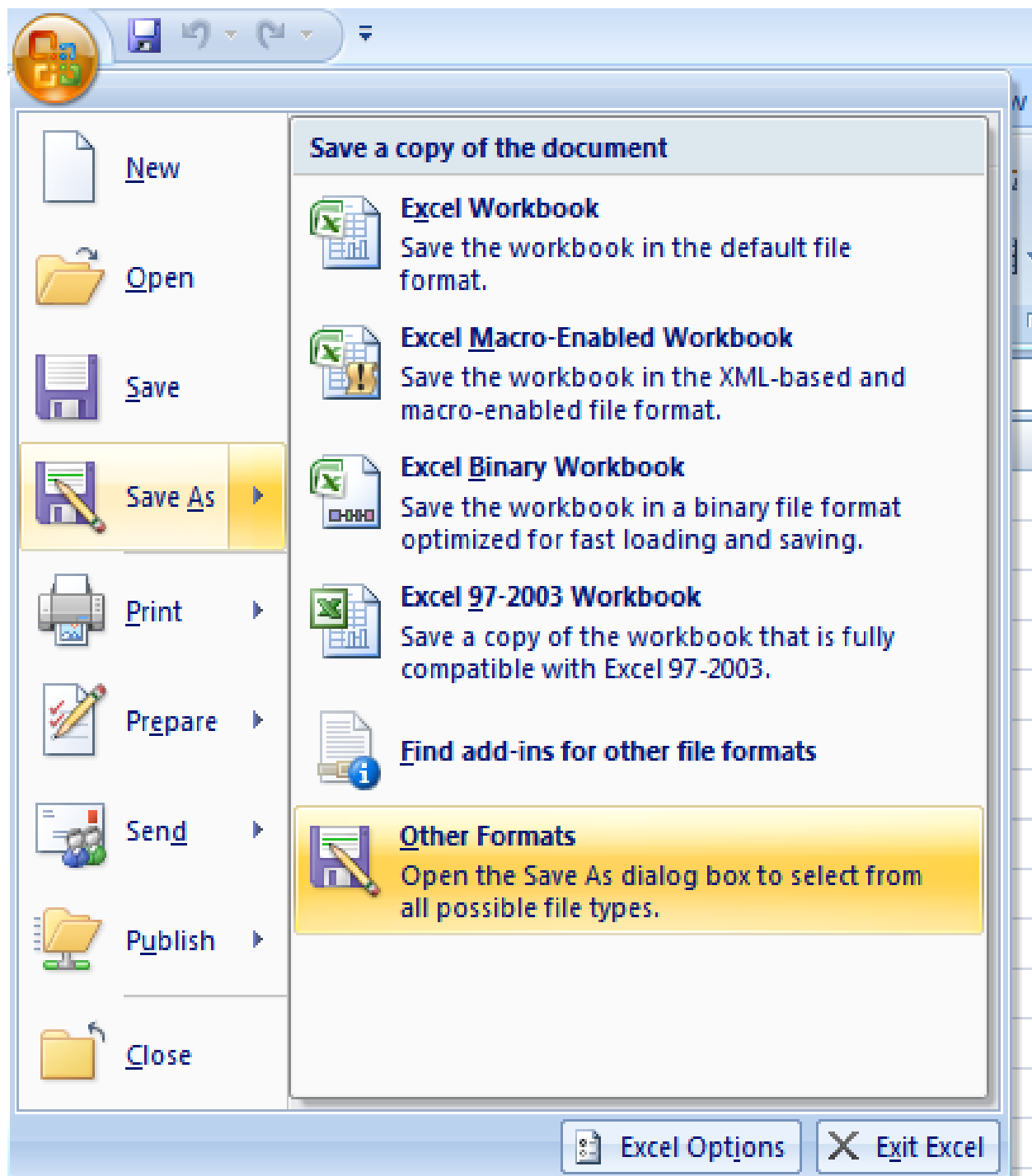
a. Reading different types of data sets (.txt, .csv) from web and disk and writing in file in specific disk location.

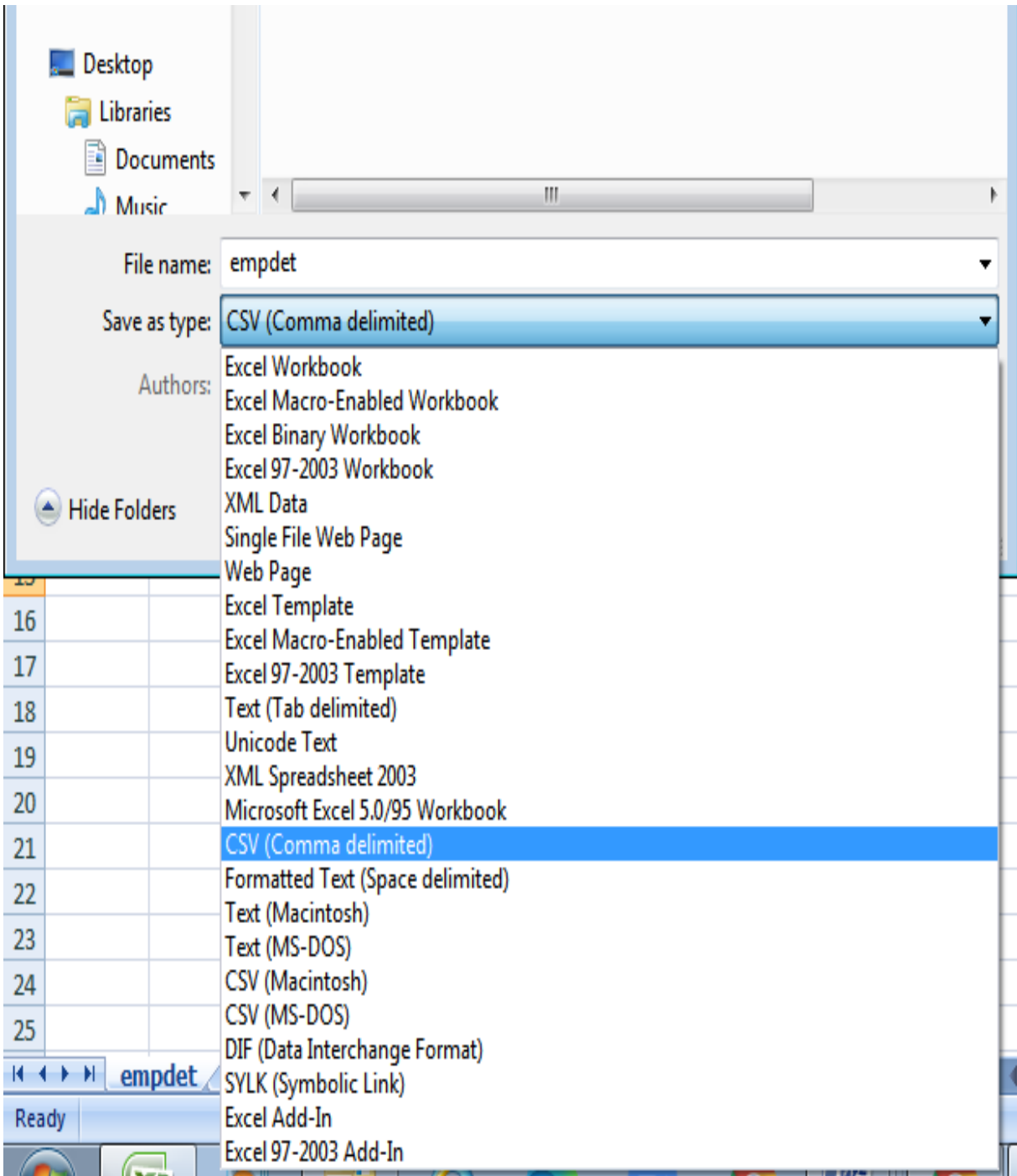
1. OPEN EXCEL TO CREATE .CSV AND .TXT FILES CALLED empdet.csv and empdet.txt and save them in D:

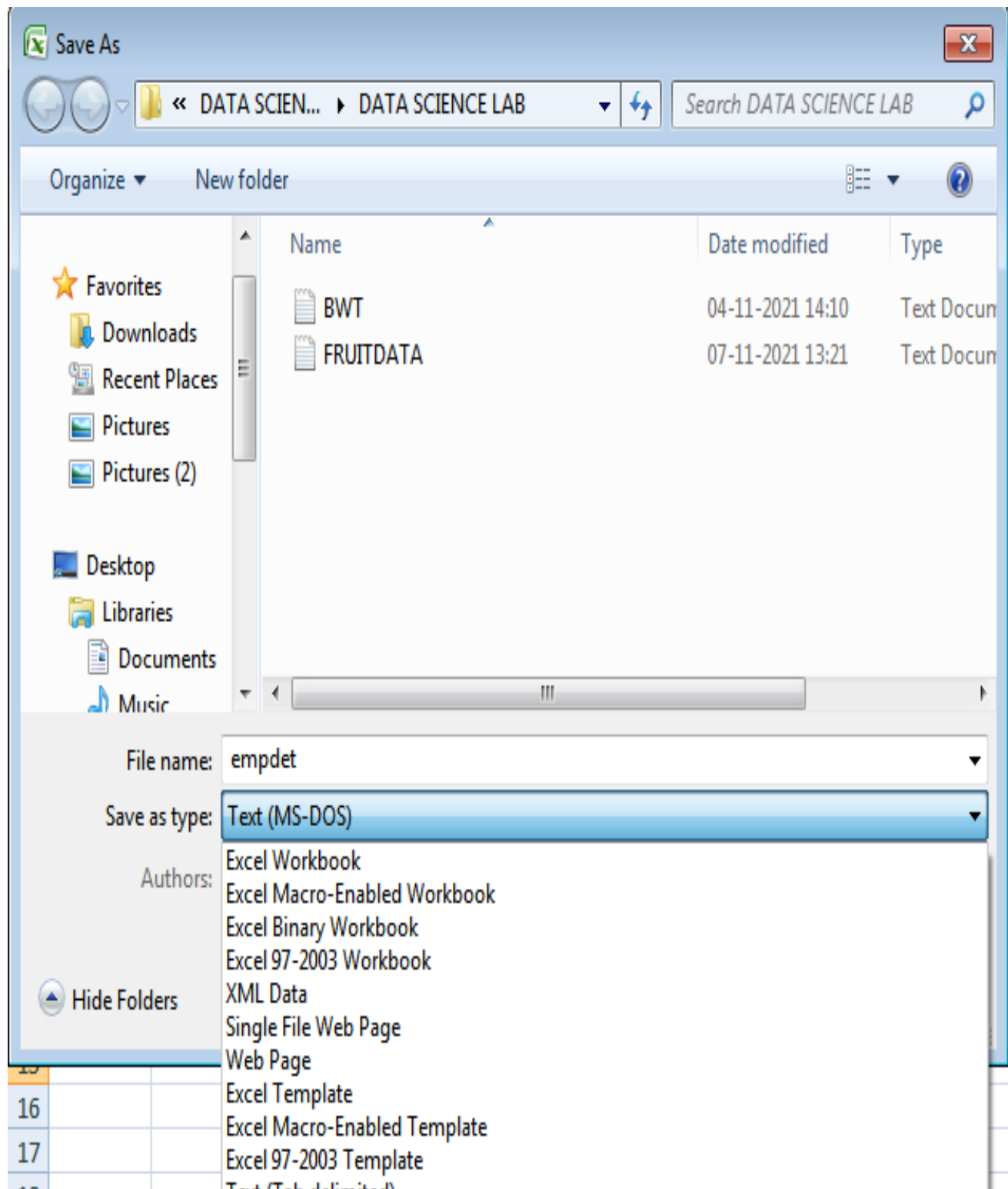


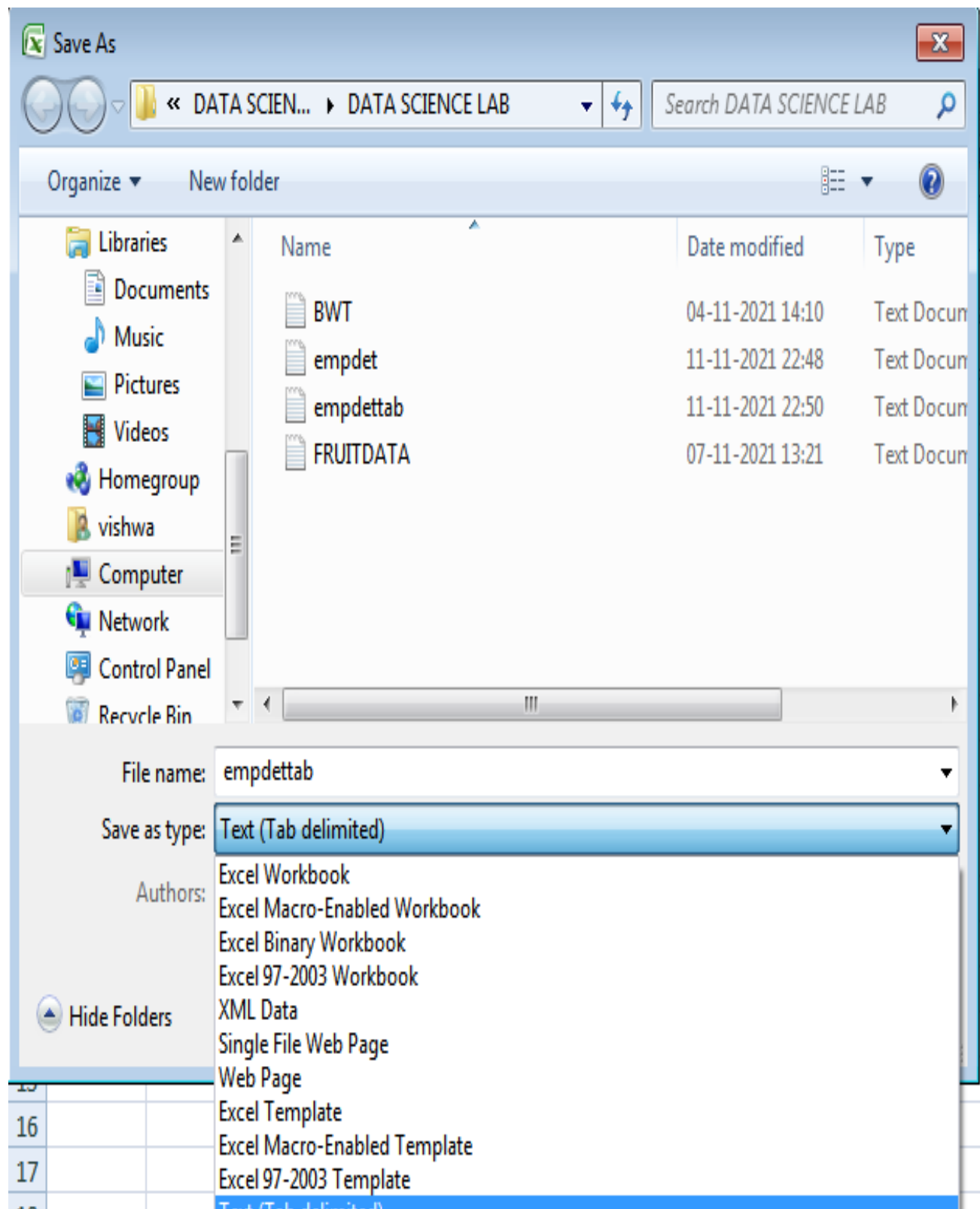
The screenshot shows the Microsoft Excel 2007 interface. The 'Home' tab is selected on the ribbon. The 'Font' group shows 'Calibri' font, size '11', with bold, italic, and underline buttons. The 'Clipboard' group shows the 'Paste' button. The 'Font' group also includes color selection buttons for text and background. The 'Paragraph' group shows bullet point and numbering options. The 'Formulas' group shows the 'fx' button. The worksheet area displays a table with 5 columns (A-E) and 8 rows (1-8). The table contains employee data.

	A	B	C	D	E
1	empno	empname	deptno	projname	
2	1000	jack	10	1	
3	1001	jane	20	1	
4	1002	dave	20	2	
5	1003	robert	10	2	
6	1004	ross	30	3	
7	1005	margaritta	30	3	
8	1006	rosy	40	4	









Installing and loading readr

```
# Installing
install.packages("readr")
# Loading
library("readr")
```

The **readr** package contains functions for reading i) delimited files, ii) lines and iii) the whole file.

Functions for reading delimited files: txt | csv

The function **read_delim()**[in **readr** package] is a general function to import a data table into R. Depending on the format of your file, you can also use:

- **read_csv()**: to read a comma (",") separated values
- **read_csv2()**: to read a semicolon (";") separated values
- **read_tsv()**: to read a tab separated ("\t") values

The simplified format of these functions are, as follow:

```
# General function
read_delim(file, delim, col_names = TRUE)
# Read comma (",") separated values
read_csv(file, col_names = TRUE)
# Read semicolon (";") separated values
# (this is common in European countries)
read_csv2(file, col_names = TRUE)
```

```
# Read tab separated values  
read_tsv(file, col_names = TRUE)
```

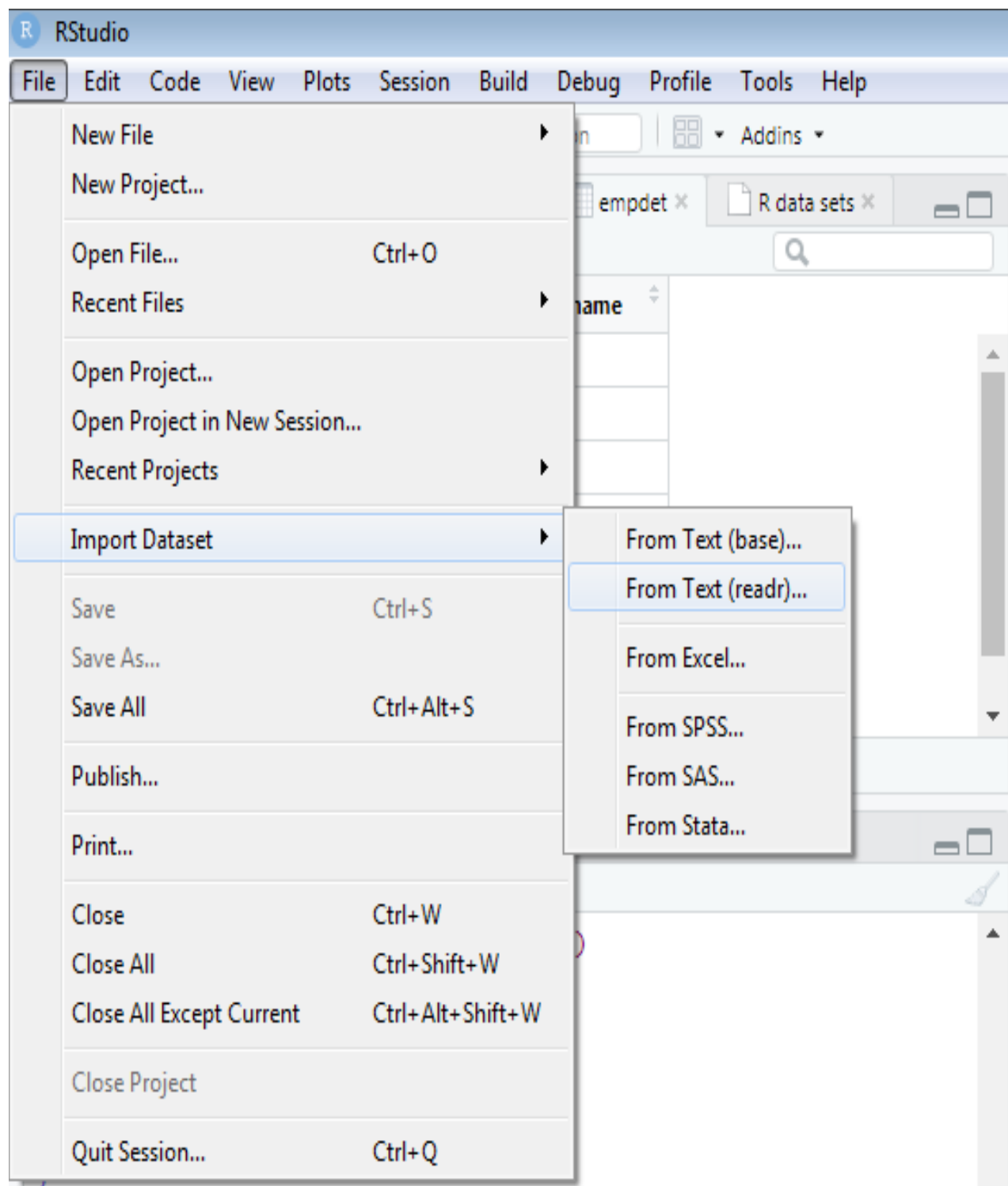
```
> mydata<-read.csv("D:/empdet.csv")  
> mydata
```

	empno	empname	deptno	projname
1	1000	jack	10	1
2	1001	jane	20	1
3	1002	dave	20	2
4	1003	robert	10	2
5	1004	ross	30	3
6	1005	margaritta	30	3
7	1006	rosy	40	4

```
Console Terminal x Jobs x
~/
> mydata<-read_tsv("D:/empdet.txt")
Parsed with column specification:
cols(
  empno = col_double(),
  empname = col_character(),
  deptno = col_double(),
  projname = col_double()
)
> mydata
# A tibble: 7 x 4
  empno empname      deptno projname
  <dbl> <chr>      <dbl>    <dbl>
1  1000 jack         10         1
2  1001 jane         20         1
3  1002 dave         20         2
4  1003 robert        10         2
5  1004 ross          30         3
6  1005 margaritta    30         3
7  1006 rosy         40         4
> |
```



```
Console Terminal x Jobs x
~/
> mydata<-read_tsv("D:/empdettab.txt")
Parsed with column specification:
cols(
  empno = col_double(),
  empname = col_character(),
  deptno = col_double(),
  projname = col_double()
)
> mydata
# A tibble: 7 x 4
  empno empname      deptno projname
  <dbl> <chr>         <dbl>   <dbl>
1  1000 jack          10         1
2  1001 jane          20         1
3  1002 dave          20         2
4  1003 robert         10         2
5  1004 ross           30         3
6  1005 margaritta    30         3
7  1006 rosy          40         4
> |
```



Import Text Data

File/URL:

Browse...

Data Preview:

Import Options:

Name: dataset

Skip: 0

☒ First Row as Names

☒ Trim Spaces

☒ Open Data Viewer

Delimiter: Comma

Quotes: Default

Locale: Configure...

Escape: None

Comment: Default

NA: Default

Code Preview:

```
library(readr)
dataset <- readr::read_csv(
  "data.csv",
  na = "NA"
)
view(dataset)
```

Import Text Data

File/URL:

D:/empdet.csv

Browse...

Data Preview:

empno (double) ▾	empname (character) ▾	deptno (double) ▾	projname (double) ▾
1000	jack	10	1
1001	jane	20	1
1002	dave	20	2
1003	robert	10	2
1004	ross	30	3
1005	margaritta	30	3
1006	rosy	40	4

Previewing first 50 entries.

Import Options:

Name: empdet

☒ First Row as Names

Delimiter: Comma ▾

Escape: None ▾

Skip: 0

☒ Trim Spaces

Quotes: Default ▾

Comment: Default ▾

☒ Open Data Viewer

Locale: Configure...

NA: Default ▾

Code Preview:



```
library(readr)
empdet <- read_csv(
  "D:/empdet.csv"
)
view(empdet)
```

```

> library(readr)
> empdet <- read_csv("D:/empdet.csv")
Parsed with column specification:
cols(
  empno = col_double(),
  empname = col_character(),
  deptno = col_double(),
  projname = col_double()
)
> view(empdet)
>

```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

+ Go to file/function Addins

Untitled1* x TESTR.R x Untitled2* x empdet x R data sets x

Filter

	empno	empname	deptno	projname
1	1000	jack	10	1
2	1001	jane	20	1
3	1002	dave	20	2
4	1003	robert	10	2
5	1004	ross	30	3
6	1005	margaritta	30	3
7	1006	rosy	40	4

Showing 1 to 7 of 7 entries, 4 total columns

Reading a file from internet

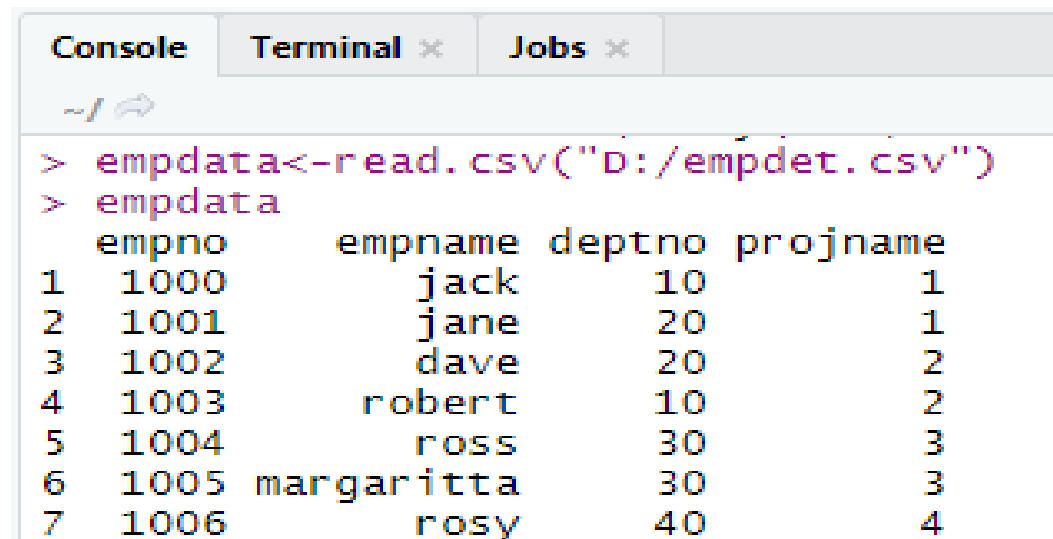
It's possible to use the functions `read_delim()`, `read_csv()` and `read_tsv()` to import files from the web.

```
my_data<-  
read_tsv("http://www.sthda.com/upload/boxplot_format.txt")
```

```
head(my_data)
```

	Nom	variable	Group
1	IND1	10	A
2	IND2	7	A
3	IND3	20	A
4	IND4	14	A
5	IND5	14	A
6	IND6	12	A

TO WRITE DATA INTO A FILE:



The screenshot shows a terminal window with tabs for 'Console', 'Terminal', and 'Jobs'. The 'Console' tab is active, displaying the following R code and its output:

```
> empdata<-read.csv("D:/empdet.csv")  
> empdata
```

	empno	empname	deptno	projname
1	1000	jack	10	1
2	1001	jane	20	1
3	1002	dave	20	2
4	1003	robert	10	2
5	1004	ross	30	3
6	1005	margaritta	30	3
7	1006	rosy	40	4

```

argument 'subset' is missing, with no default
> subempdata<-subset(empdata,empdata$deptno>10)
> subempdata
  empno      empname deptno projname
2  1001         jane     20         1
3  1002         dave     20         2
5  1004         ross     30         3
6  1005 margaritta     30         3
7  1006         rosy     40         4
> write.csv(subempdata,"empoutput.csv")
> empoutput
Error: object 'empoutput' not found
> newempdata<-read.csv("empoutput.csv")
> newempdata
  X empno      empname deptno projname
1 2  1001         jane     20         1
2 3  1002         dave     20         2
3 5  1004         ross     30         3
4 6  1005 margaritta     30         3
5 7  1006         rosy     40         4
> |

```

b. Reading Excel data sheet in R.

```

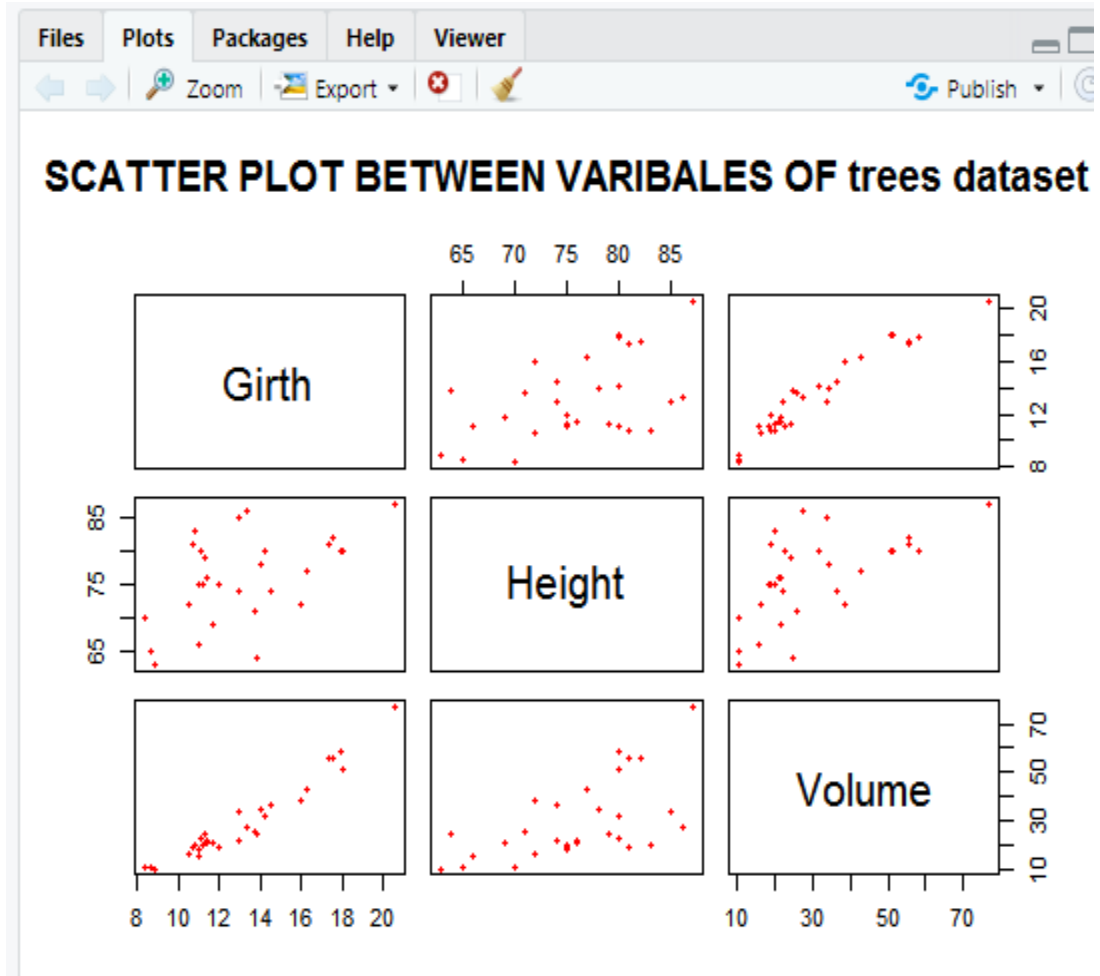
> library(readxl)
> empdet <- read_excel("D:/empdet.xls")
> View(empdet)
>

```

EXP 4 – VISUALIZATIONS

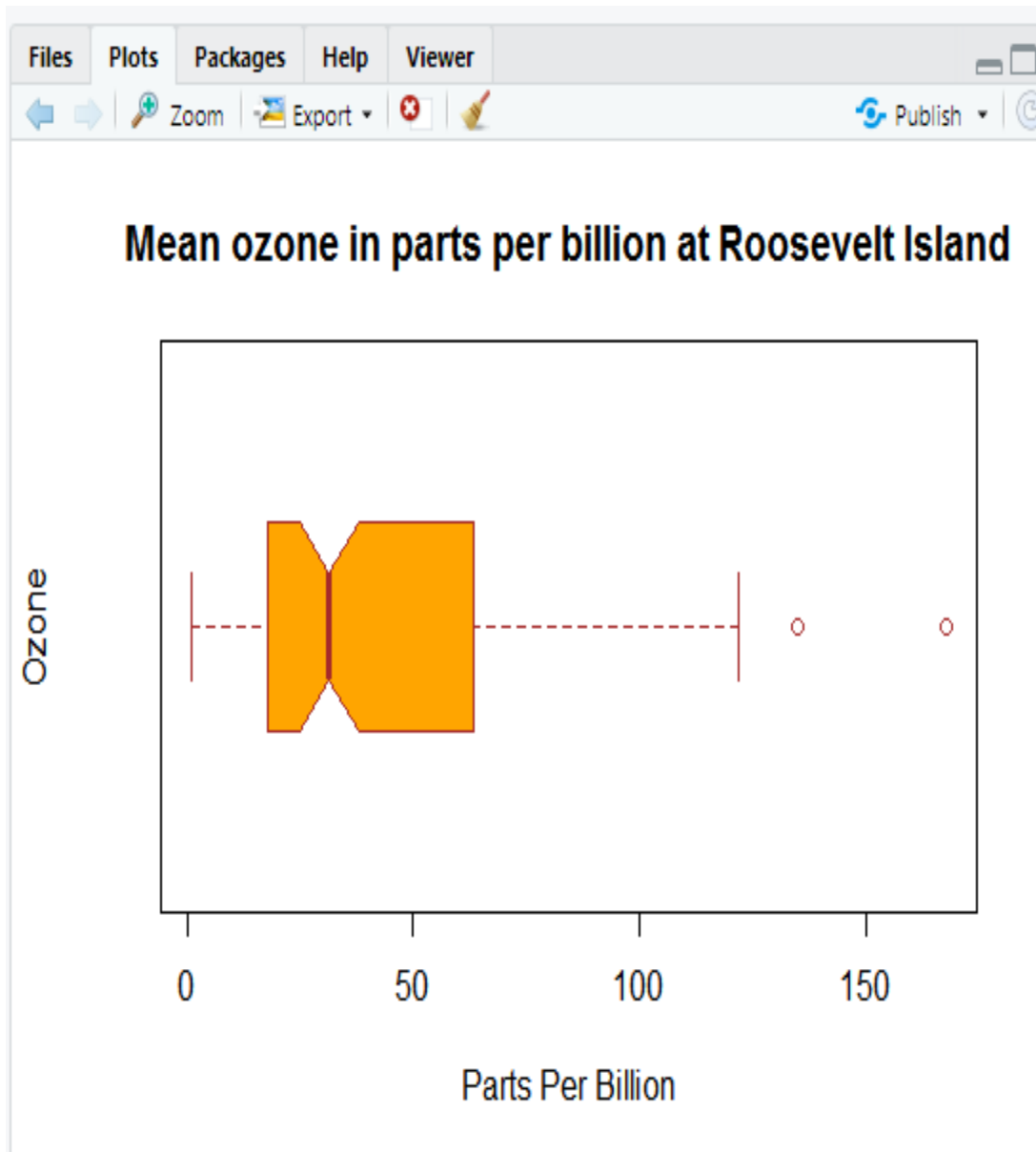
- a. Find the data distributions using box and scatter plot.

```
> trees
   Girth Height Volume
1    8.3    70   10.3
2    8.6    65   10.3
3    8.8    63   10.2
4   10.5    72   16.4
5   10.7    81   18.8
6   10.8    83   19.7
7   11.0    66   15.6
8   11.0    75   18.2
9   11.1    80   22.6
10  11.2    75   19.9
11  11.3    79   24.2
12  11.4    76   21.0
13  11.4    76   21.4
14  11.7    69   21.3
15  12.0    75   19.1
16  12.9    74   22.2
17  12.9    85   33.8
18  13.3    86   27.4
```

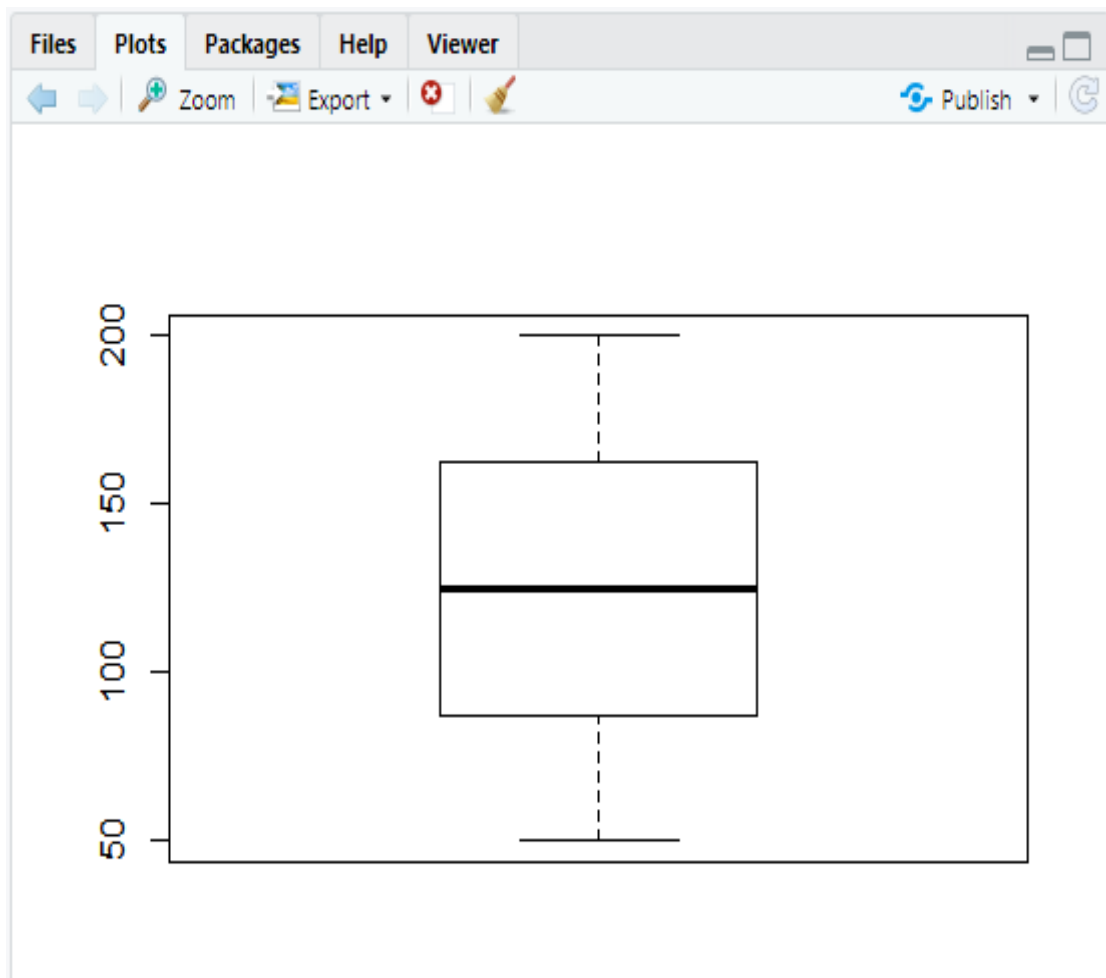
pch=plot character ,defines the point symbols in the functions plot() and lines(). It contains numeric values ranges from 0 to 25 or character symbols ("+", "-", "....") SPECIFYING POINT SYMBOLS OR SHAPES.

```
Console Terminal x Jobs x
~/ ↩
>
> str(airquality)
'data.frame':  153 obs. of  6 variables:
 $ Ozone   : int  41 36 12 18 NA 28 23 19 8 NA ...
 $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194
 ...
 $ wind    : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 2
0.1 8.6 ...
 $ Temp    : int  67 72 74 62 56 66 65 59 61 69 ...
 $ Month   : int  5 5 5 5 5 5 5 5 5 5 ...
 $ Day     : int  1 2 3 4 5 6 7 8 9 10 ...
> boxplot(airquality$wind)
> boxplot(airquality$Ozone,
+         main = "Mean ozone in parts per billion at
Roosevelt Island",
+         xlab = "Parts Per Billion",
+         ylab = "Ozone",
+         col = "orange",
+         border = "brown",
+         horizontal = TRUE,
+         notch = TRUE
+ )
>
```



a. Find the outliers using plot.

```
v=c(50,75,100,125,150,175,200)  
boxplot(v)
```

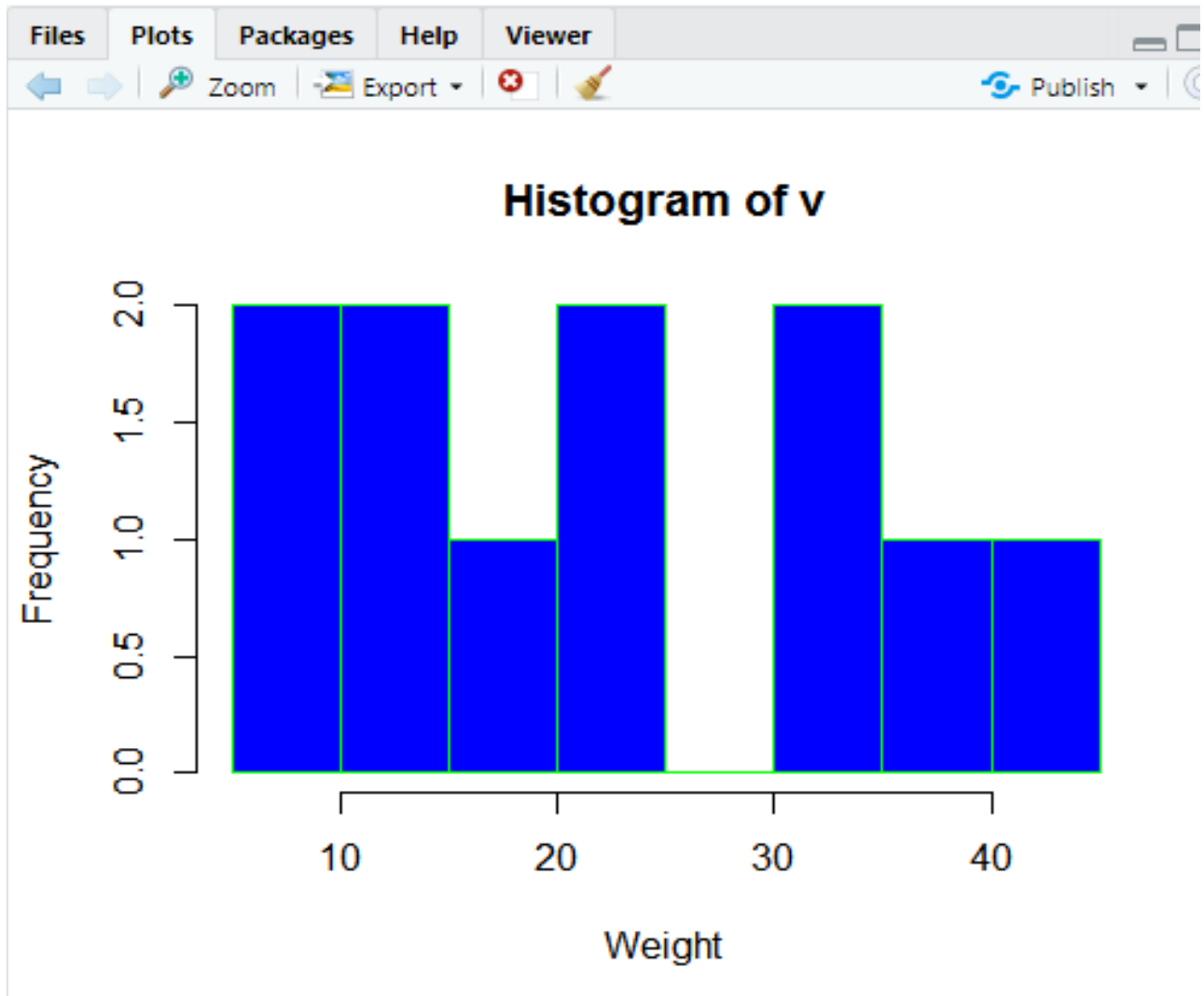


a. **Plot the histogram, bar chart and pie chart on sample data.**

Histogram

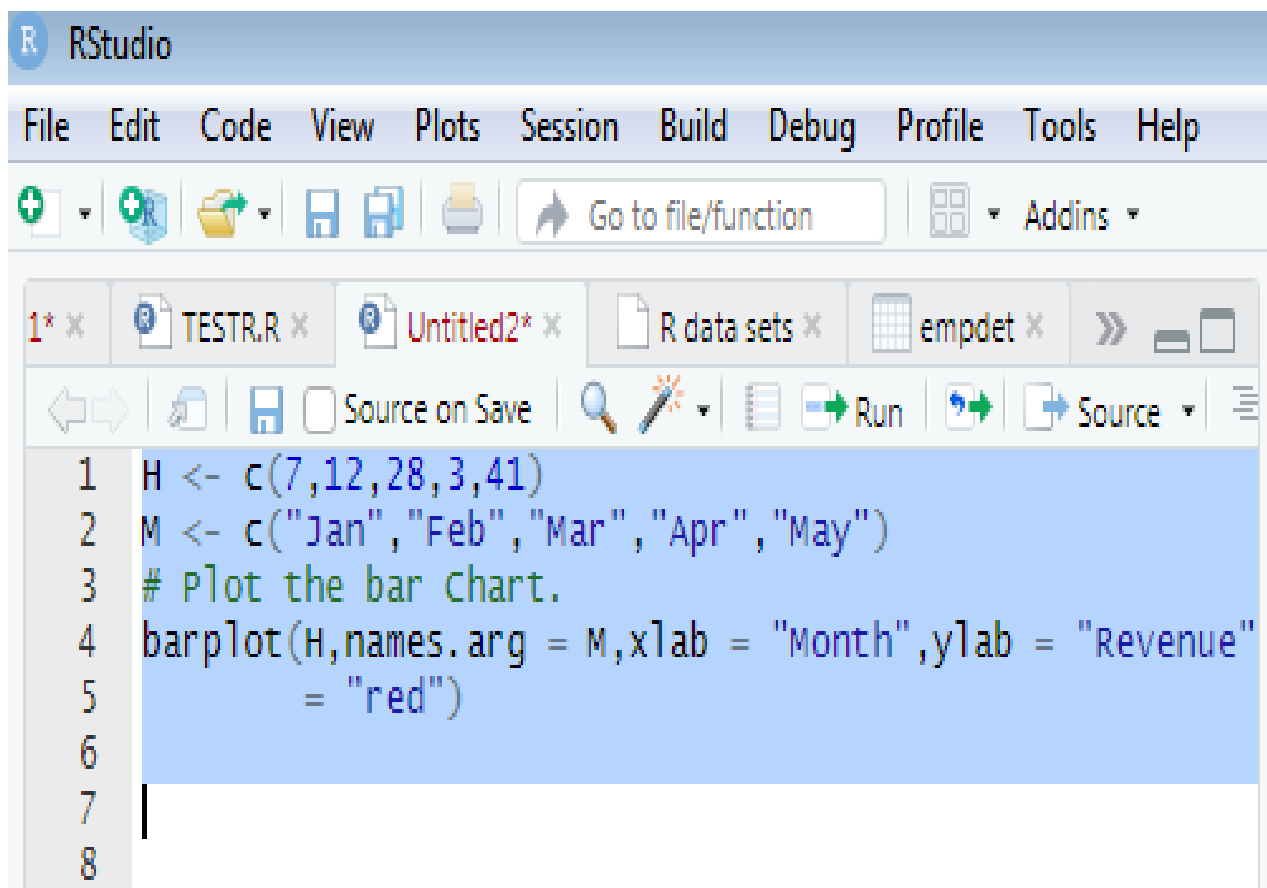
```
library(graphics)
v <- c(9,13,21,8,36,22,12,41,31,33,19)

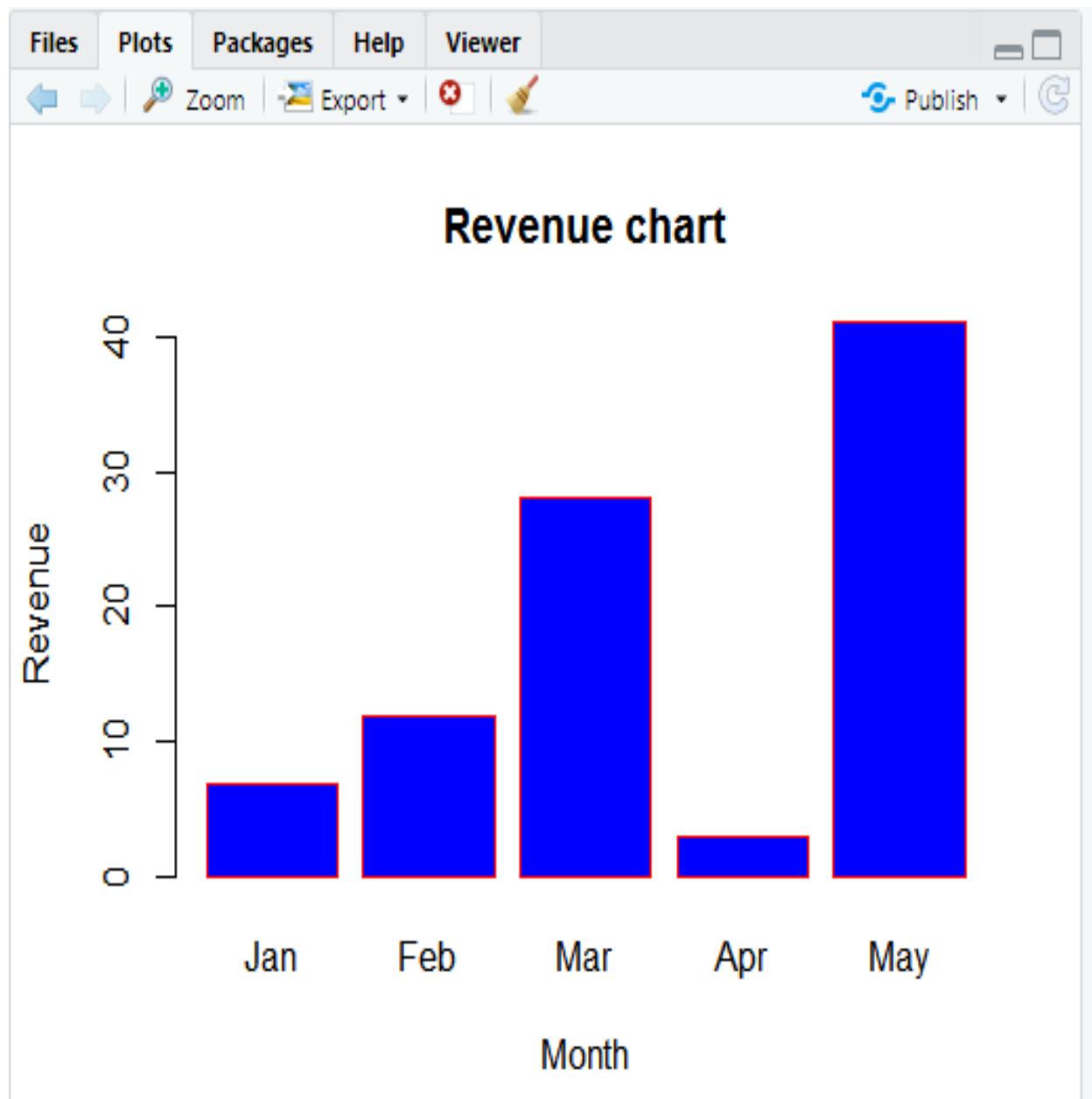
# Create the histogram.
hist(v,xlab = "Weight",col = "blue",border = "green")
```



Bar chart

```
library(graphics)
H <- c(7,12,28,3,41)
M <-
c("Jan","Feb","Mar"
,"Apr","May")
# Plot the bar
Chart.
barplot(H,names.arg = M,xlab = "Month",ylab = "Revenue",col =
"blue",main = "Revenue chart",border
= "red")
```





Pie Chart

```
library(graphics)
```

```
x <- c(21, 62, 10, 53)
```

```
labels<- c("London", "NewYork",  
"Singapore", "Mumbai")
```

```
# Plot the Pie chart.
```

```
pie(x,labels)
```

```
> x <- c(21, 62, 10, 53)
```

```
Warning messages:
```

```
1: package 'RMySQL' was built under R version 3.6.1
```

```
2: package 'DBI' was built under R version 3.6.1
```

```
3: package 'arules' was built under R version 3.6.1
```

```
>
```

```
> labels<- c("London", "NewYork", "Singapore", "Mumbai")
```

```
>
```

```
> # Plot the Pie chart.
```

```
>
```

```
> pie(x,labels)
```

```
> |
```