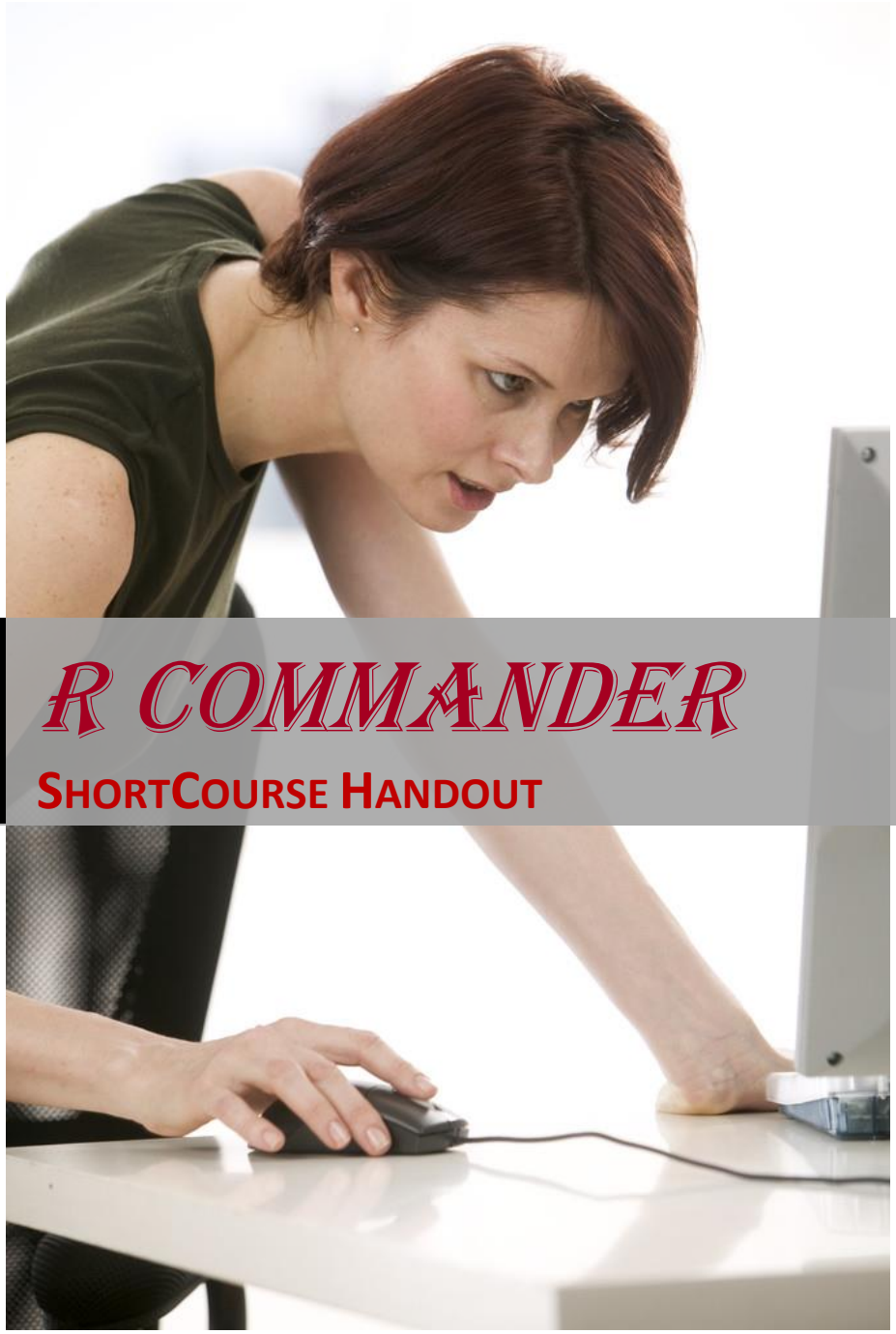


TECHNOLOGY SUPPORT

R COMMANDER

SHORTCOURSE HANDOUT

TEXAS TECH UNIVERSITY
Technology Support

Texas Tech University | Heide Mansouri

Table of Contents

R Commander for Statistical Analysis

ShortCourse Handout

Introduction

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and Mac OS, and is available from http://www.r-project.org/ . R is a command-driven system, and new users often find learning R challenging. **R Commander**, however, developed by Professor John Fox http://socserv.mcmaster.ca/jfox/Books/RCommander, is a point-and-click interface to R, which allows users to focus on statistical methods rather than remembering R commands. It provides menus for many analytic and graphical methods and shows you the R commands that it enters, making it easy to learn the commands as you use. This document describes the use of R Commander under Windows Operating System. The most recent version of R available at the time of this writing was **R version 3.3.2** (released on 10/31/2016) and **R-Commander version 2.3-2** (released on 1/2/2017).

Credits: This document is adapted mostly from the following resources:

- **Getting Started With the R Commander**, http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/Getting-Started-with-the-Rcmdr.pdf, accessed on 1/2/2017
- **R Commander an introduction**, https://cran.r-project.org/doc/contrib/Karp-Rcommander-intro2.pdf , accessed on 1/10/2017

- **R with Rcmdr: BASIC INSTRUCTIONS** http://users.monash.edu.au/~murray/stats/BIO3011/Rmanual_paper.pdf accessed on 1/12/2017
- **R in Action**, 2nd Edition, 2015, Safari book @ http://library.ttu.edu/
- **R Commander** https://www.youtube.com/user/ramstatvid

Course Objectives

After completing this ShortCourse, you should be able to:

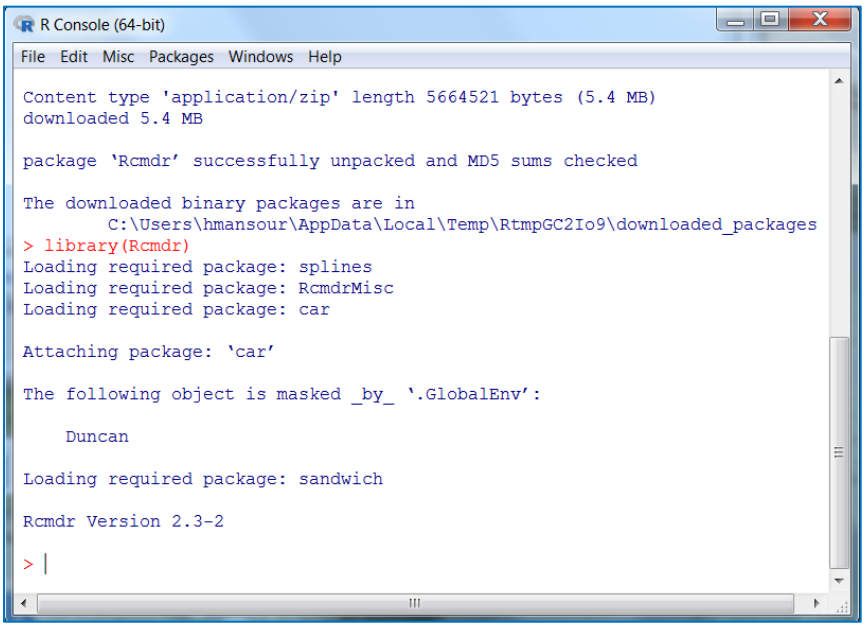
- Load Data in R Commander
- Create Graphs
- Perform One-Way ANOVA
- Create Contingency Tables
- Read Data from R Packages
- Perform t-Tests
- Create Summary Statistics
- Perform Linear Regression Analysis
- Save and Printing Data

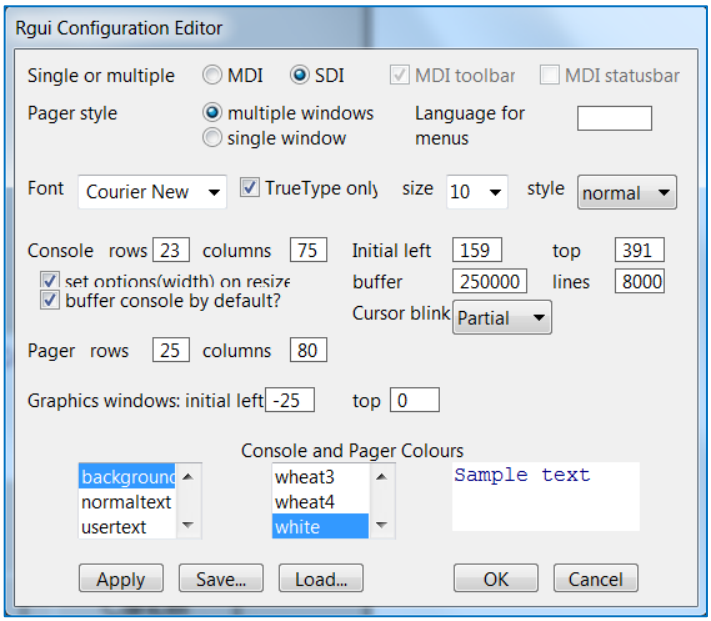
Downloading and Installing R and R Commander (Rcmdr)

To download R, navigate your internet browser to the R home page: http://www.r-project.org/, and choose your preferred **CRAN** (Comprehensive R Archive Network) **mirror**, near your physical location from https://cran.r-project.org/mirrors.html . You must always choose a CRAN mirror site whenever you download anything into R. This applies to your first download of R program, but also after you have R running and you later want to add additional R Packages.

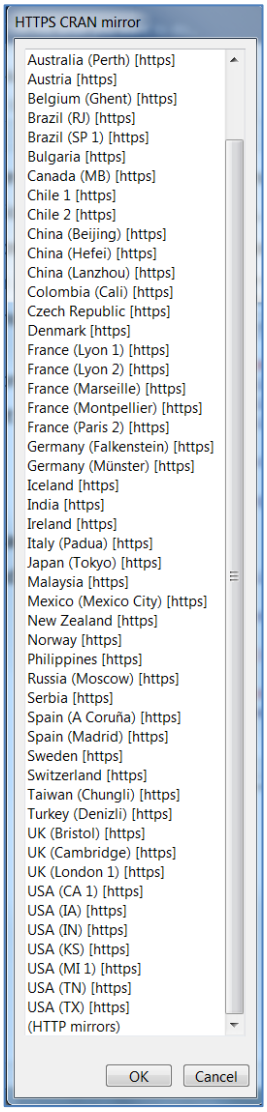
- For Windows Operating System, select **base.**
- Follow the steps for downloading and installing executable files on your computer.
- Once the executable file **Rgui.exe** is downloaded on to your computer, double-click on it (or it may run automatically), and a **Setup Wizard** will appear, after you have chosen a language.
- When you run the **Setup Wizard**, you will click a series of buttons and eventually come to Startup option. Click **Yes** (customized startup), radio button.

- In the **Display Mode** option, choose the **SDI** option. To function properly under Windows, R Commander requires the single-document interface (SDI) to R. That is, R's default is **MDI**, meaning that there is one big window, but it is much easier to use R when graphics and the help menu pop up in separate windows (as they do for the SDI choice).
- If you want to further customize R by changing font sizes, font colors, etc., navigate in the **R Console**, to **EDIT** > **GUI PREFERENCES**. You will see a window, called the **Rgui Configuration Editor**. You might also want to change the default colors of the text commands here as well.





Loading Packages and R Commander

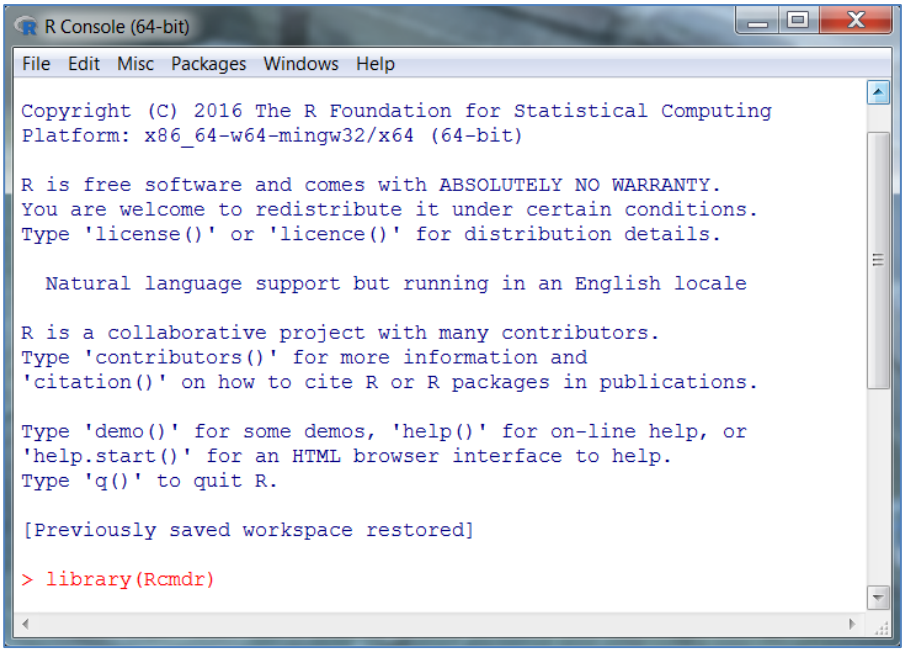


In the **R Console**, (R User Interface) navigate to the menu **Packages** -> **Install Packages**.
You will need to select your **CRAN mirror site** (a distribution site for software) in order to continue with the installation. Once you have made your choice for a CRAN mirror site, a list of packages will pop up. Scroll down and choose **Rcmdr**. Press OK and the package will load.

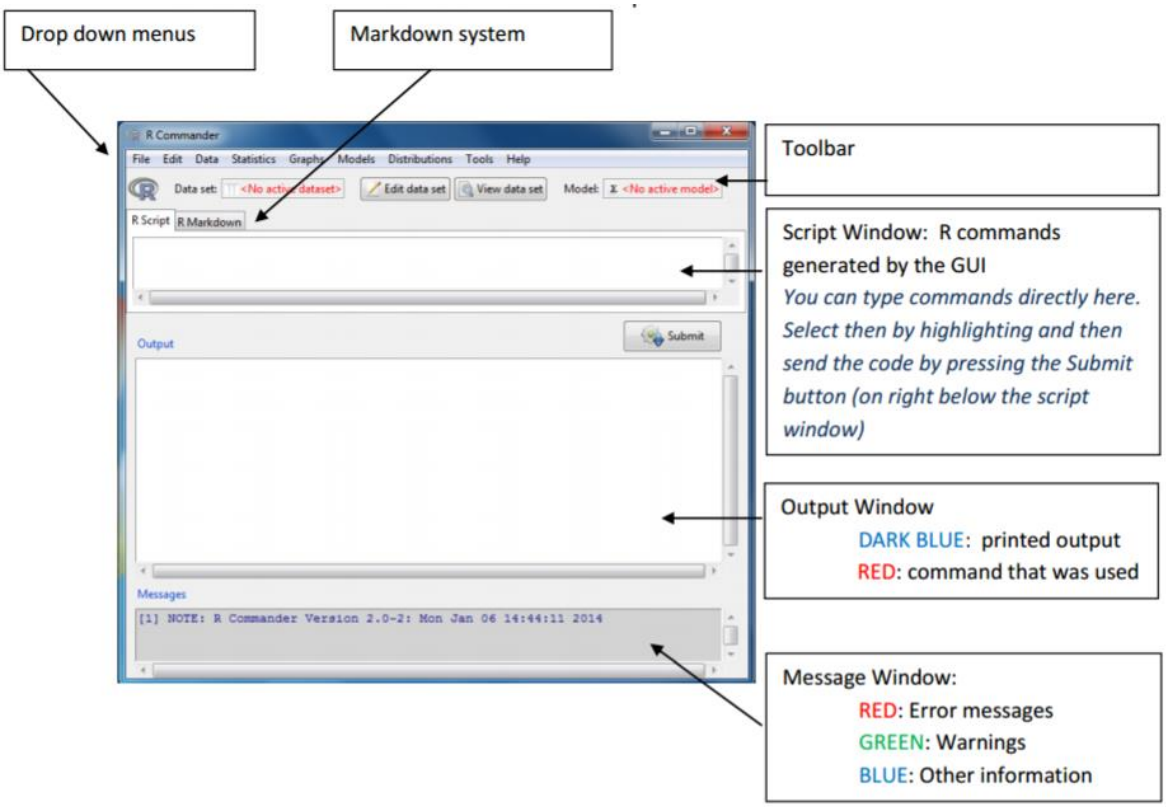
After you downloaded the R Commander, start it by typing the following line in the R Console prompt ">"):

library(Rcmdr)

Note: This should be typed exactly as given above, that is the letter R of R commander must be capitalized, and the rest of the letters (cmdr) must be lower-case.



The first time you try to open R Commander, it will tell you that you need to download some additional files; just follow the directions for this, and install from the CRAN site, not a local directory. Once those files are downloaded, use the **library()** command again (or press the "**Up Arrow**" from your keyboard, until you come back to your previous command), and **R Commander** will open.



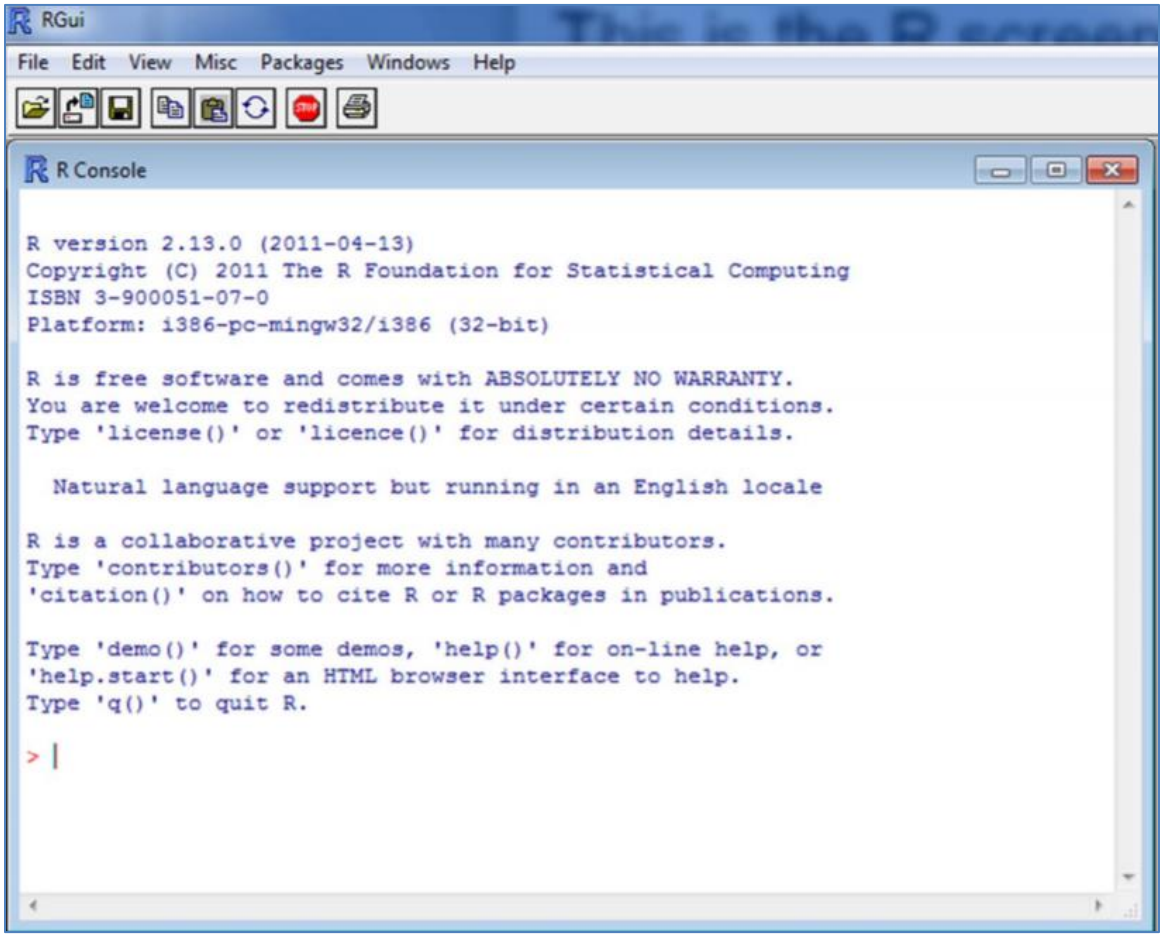
To get a description of the version of R and its attached packages used in the current session, we can use the sessionInfo function **sessionInfo()**.

Type this function in the R Script window; select it (using your computer mouse), and then press the **Submit** button.

Note: Graphs will appear in a separate Graphics Device Window. Only the most recent graph will appear. You can use page up and page down keys (⬆⬇) from your keyboard to recall previous graphs.

If you accidently close the Rcmdr window, type **Commander()** in the **R Console** and click OK.

This is the R screen in Multiple-Document Interface (MDI)…



R Packages

R Packages are collections of **R functions**, **data**, and **compiled code** in a well-defined format. R comes with a standard set of packages. They provide a wide range of functions and datasets that are available by default. Other packages are available for download and installation.

***To use packages in R, you must first install them*.** Then, if you need a particular package for your current R session, you must **load** it into the R environment using the library or required function (**library(Rcmdr)**, or **Commander()**).

R Commander Drop Down Menu Items:

Drop down Menu item	
File	Menu items for loading and saving script files; for saving output and the R workspace; and for exiting.
Edit	Menu items (Cut, Copy, Paste, etc.) for editing the contents of the script and output windows. Right clicking in the script or output window also brings up an edit "context" menu
Data	Submenus containing menu items for reading and manipulating data.
Statistics	Submenus containing menu items for a variety of basic statistical analyses.
Graphs	Menu items for creating simple statistical graphs.
Models	Menu items and submenus for obtaining numerical summaries, confidence intervals, hypothesis tests, diagnostics, and graphs for a statistical model, and for adding diagnostic quantities, such as residuals, to the data set. Distributions Probabilities, quantiles, and graphs of standard statistical distributions (to be used, for example, as a substitute for statistical tables).
Distributions	Probabilities, quantiles, sampling and graphs of standard statistical distributions
Tools	Menu items for loading R packages unrelated to the Rcmdr package (e.g., to access data saved in another package), and for setting some options.
Help	Menu items to obtain information about the R Commander (including an introductory manual derived from this paper). As well, each R Commander dialog box has a Help button.

More elements included in R Commander Interface

- Below the menus is a toolbar with a row of buttons
    - The leftmost (flat) button shows the name of the **active data set**. Initially there is no active data set. If you press this button, you will be able to choose among data sets currently in memory. Most of the menus and dialogs in R Commander reference the active data set. (The File, Edit, and Distributions menus are exceptions.)
- Two buttons allow you to open a data editor to modify the active data set or a viewer to examine it. The data set viewer can remain open while other operations are performed.
- A flat button indicates the name of the active statistical model – either a linear model (such as a linear-regression model) or a generalized linear model. Initially there is no active model. If there is more than one model in memory, you can choose among them by pressing the button.
- A **Submit** button, at the far right of the toolbar, allows you to execute commands from the script window.

- Immediately below the toolbar is the log/script window, a large scrollable text window. Commands generated by the GUI are copied into this window. You can edit the text in the log/script window or even type your own R commands into the window. Pressing the **Submit** button (or, alternatively, the key combination "**Ctrl + r**", for "**run**") causes the line containing the cursor to be submitted (or resubmitted) for execution. If several lines are selected (e.g., by left clicking and dragging the mouse over them), then pressing Submit will cause all of them to be executed. Commands entered into the log-script window can extend over more than one line, but if they do, lines after the first must be indented with one or more spaces or tabs.
- At the bottom is a large scrollable and editable text window for output. Commands echoed to this window appear in red, output in dark blue (as in the **R Console**). Once you have loaded the Rcmdr package, you can minimize the R Console. The R Commander window can also be resized or maximized in the normal manner. If you make the R Commander window wider, then you may wish to reset the width of printed output from R via the **File** > **Reset output width** . . . menu.

Updating Packages

An installation of R can include a number of separate packages and plugins and it is important that these are kept **up-to-date**. This is achieved by **update.packages()** command, issued in the **R-console** or the **Rcmdr Script Window**. This command will compare all packages and Rcmdr Plugins that have been installed on your computer with those that are available on CRAN. The user is then given the option to update any packages where updates are available.

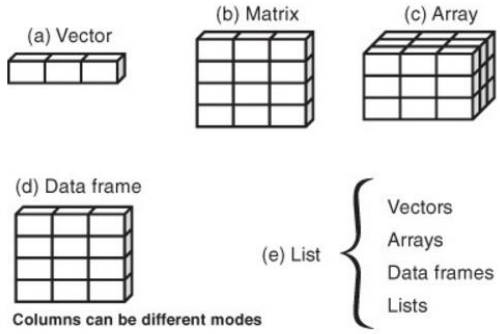
A Few Basics

- SAS and SPSS both use one main data structure, the data set. Instead, R has many different data structures. The one that is most like a data set is called a **data frame**.
- SAS and SPSS data sets are as a rectangle with variables in the columns and records in the rows. SAS calls these records **observations** and SPSS calls them **cases**. R documentation uses variables and columns interchangeably. It usually refers to observations or cases as **rows**.
- R **data frames** have a formal place for an **ID variable** called **row labels**.

- SAS and SPSS users typically have an ID variable containing an observation/case number or perhaps a subject's name. However, this variable is like any other unless you run a procedure that identifies observations.
- You can use R the same way, but procedures that identify observations may do so automatically if you set your ID variable to be official row labels. Also, when you do that, the variable's original name (ID, Subject, SSN, etc.) vanishes.
- Another data structure R uses frequently is the **vector**. A vector is a one-dimensional collection of numbers (numeric vector) or character values (character vector).
- Variable names in R can be any length consisting of **letters**, **numbers** or the **period** "." and **should begin with a letter**.
- Unlike SAS, the period has no meaning in the name of a dataset. However, **avoid the use of the period**.
- **R is case sensitive**. Some add-on packages, tweak names like the capitalized "Save" to represent a compatible, but enhanced, version of a built-in function like the lowercased "save".

Data Types

- **Datasets –** A dataset is usually a rectangular array of data with rows representing **observations** and columns representing **variables**. In an applied statistics, we will usually use numeric, character, or logical data types. However, data in any of these types are stored in R as one of the following data types.
- **Vectors** - These are sets of items of the same type and have a length attribute (one dimension).
- **Lists (collections of objects) -** Lists are similar to vectors, but may have items of different types. For example, a list may contain both numerical and character values.
- **Matrices -** Matrices contain items of one type arranged in rows and columns. Thus, there are two dimension attributes.
- **Data Frames (similar to datasets) -** These are similar to matrices except that they may contain more than one data type. Most data sets used in applied statistics courses will be of this data type.



Loading Data in R Commander

There are built-in data sets in R, or you can import data from an external file or webpage. You can also manually enter data into R.

Options to load a built-in data set:

- **Data** -> **Data in packages** -> **Read data set from an attached package…**
- **Data** -> **Import data** ->**From text file, clipboard, or URL…**
- **Enter a name**.
- Choose either "**Local file system**" or "**Internet URL**" for "**Location of Data File**".
- Choose "**Commas**" for "**Field Separator**" (for .csv files).
- Click **OK**.
- After loading in a data set, click the "**View data set**" button to make sure the data were read in correctly.
- To switch between different data sets within the same workspace, click on the button next to "Data set:" (which has the name of the current data set).

Importing Data

- **You need to get your data into R Commander before you can do any analyses.** There are several options for data import, including:
    - Directly typing in data into the built-in data editor
    - Using a dataset built in to a R package
    - Importing data from a text file

  - Importing data from an EXCEL spreadsheet
- Then to import your file:
  - Specify a name for data set
    - Valid names begin with an upper or lower-case **letter**, **period**, **underscore**, and **digits** – **Do not** include any **embedded blank** spaces in a data set name.
    - ***REMEMBER***, **R is case sensitive**
  - Specify whether the first line of the file contain variable names
  - Specify the field separator (Space, Tab, Comma, etc.)

Data files

It is possible to have multiple data sets open at any time. Each data set must have a unique name.

Importing a Text File Format

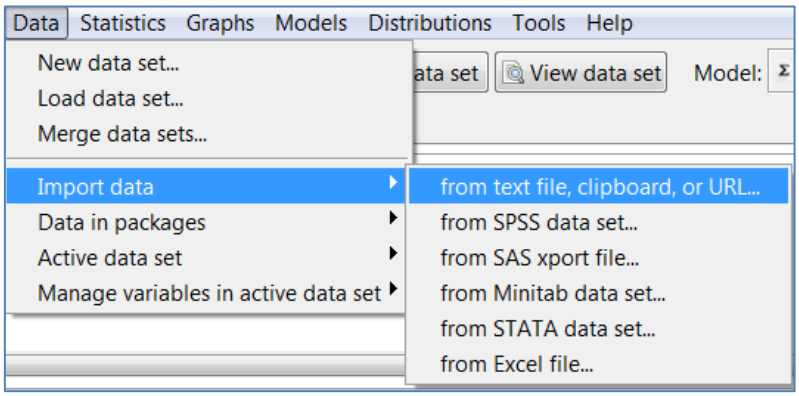
Suppose that we have the following data, on annual rainfall in inches, for various cities throughout the world.

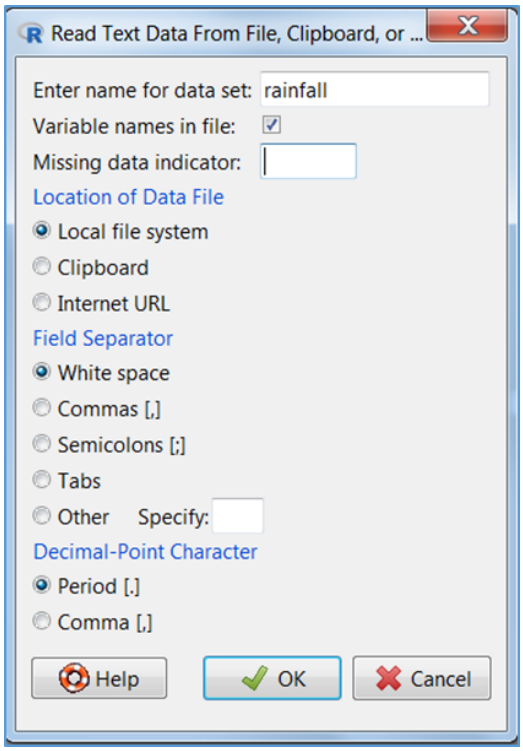
City	Rainfall
Algiers	30
Lagos	72
Athens	16
La Paz	23
Beirut	35
Lima	2
Berlin	23
London	23
Bogota	42
Madrid	17
Bombay	71
Moscow	25
Cairo	1
Oslo	27
Dublin	30
Paris	22
Geneva	34
Rome	30
Havana	48
Vienna	26

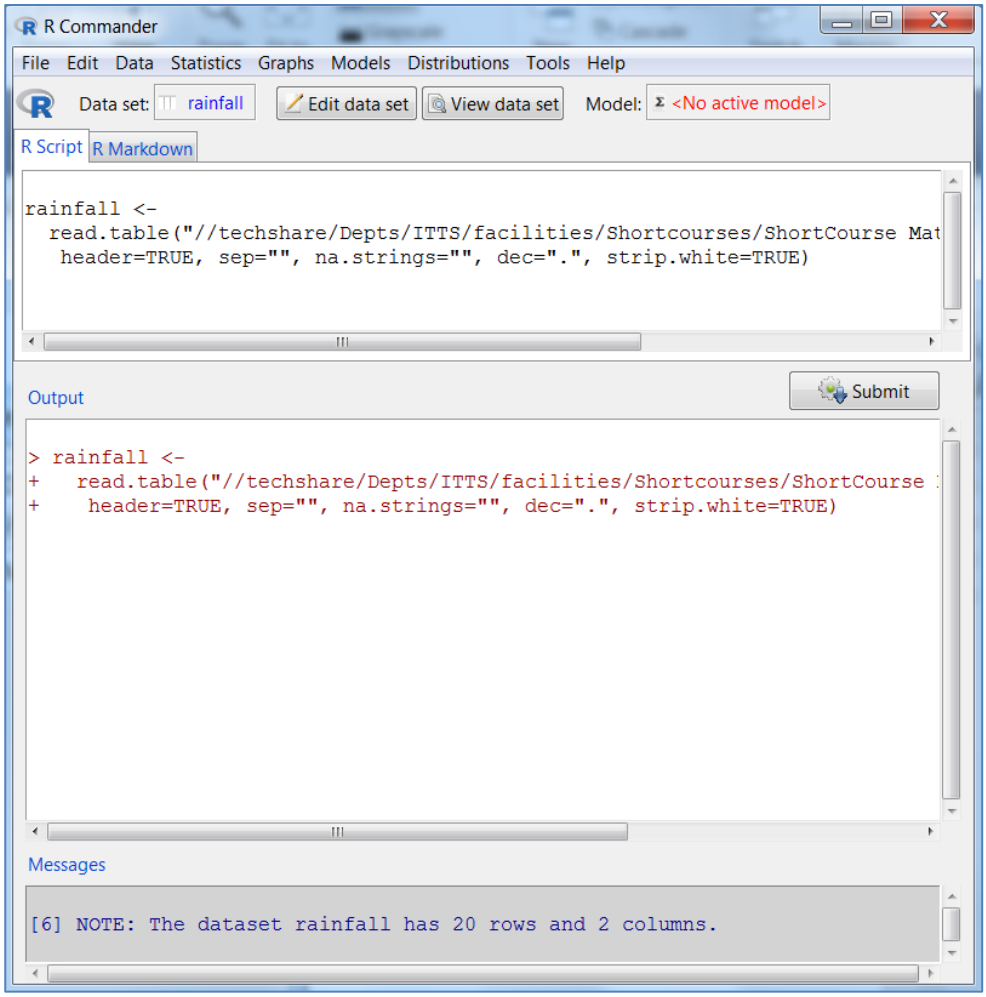
The standard file format for statistical data is that each column is a **variable** and each row is an **observation**. Here the variable is **rainfall** and the observations are the **cities**. ***Remember***, when creating this file, to remove any spaces from the

name of the cities (in La Paz, for example), and type spaces or tabs to separate columns.

- To read this file into **R-Commander** select **Data** -> **Import data** -> **From text file**
- Type **Rainfall** in **Enter Name for dataset:** box
- Delete "NA" in the **Missing data indicator** box
- Specify the **Field Separator** (White space) **-> c**lick **OK.**
- Navigate and find your file in your computer -> click **Open.**
- Click the **View data set** button, to view your dataset.





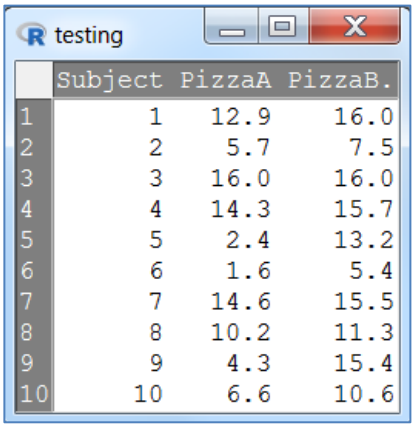


rainfall

	City	Rainfall
1	Algiers	30
2	Lagos	72
3	Athens	16
4	LaPaz	23
5	Beirut	35
6	Lima	2
7	Berlin	23
8	London	23
9	Bogota	42
10	Madrid	17
11	Bombay	71
12	Moscow	25
13	Cairo	1
14	Olso	27
15	Dublin	30
16	Paris	22
17	Geneva	34
18	Rome	30
19	Havana	48
20	Vienna	26

Importing an Excel File

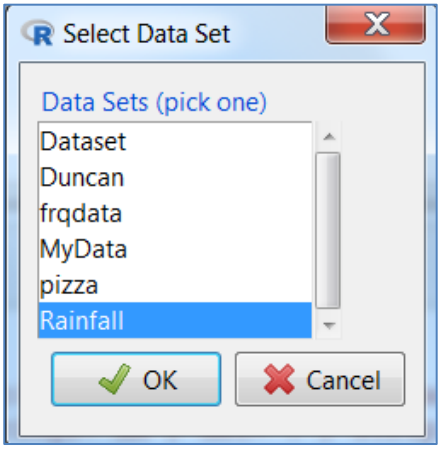
- Prepare your data set.
  - The top row of the dataset should consist of the names of the variables in the data.
  - Each variable name should consist only of **letters** and **numbers**; **no spaces** or **other symbols**.
  - Each row should be the **observations** for each variable.
  - If there is data missing, use "**NA**" for that observation; missing data will cause R to load the data set incorrectly.
- **Data** -> **Import data** -> **From Excel file**
- Type **Pizza** in **Enter Name for dataset:** box -> click **OK**
- Navigate and find your file in your computer -> click **Open**
- Click the **View data set** button, to view your data set.



	Subject	PizzaA	PizzaB.
1	1	12.9	16.0
2	2	5.7	7.5
3	3	16.0	16.0
4	4	14.3	15.7
5	5	2.4	13.2
6	6	1.6	5.4
7	7	14.6	15.5
8	8	10.2	11.3
9	9	4.3	15.4
10	10	6.6	10.6

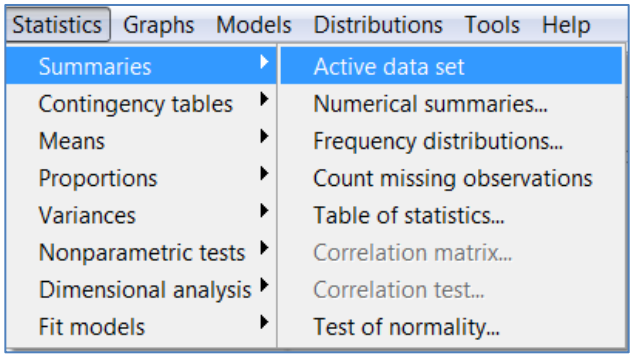
Switching between different loaded data sets

- Click on the **Data set:** display panel in the R Commander window
- The **Select Data Set** dialog box will be displayed
- Select the required data set
- Click OK.



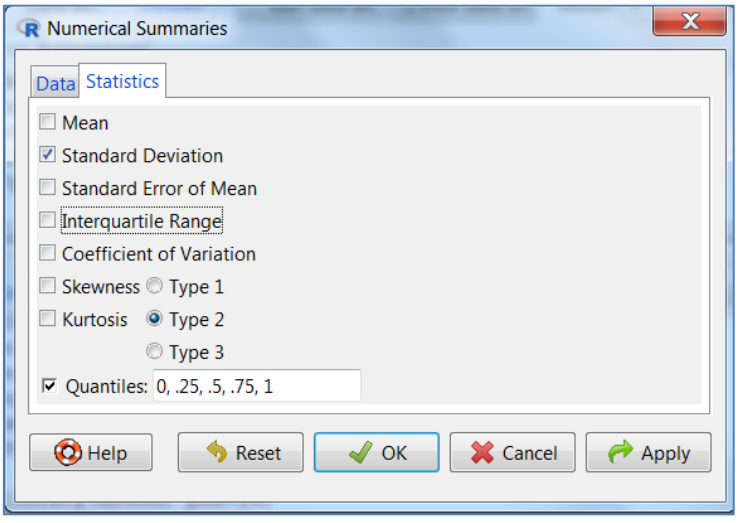
To obtain Summary Statistics, using the Rainfall data set

- From the **Data set:** box, select the Rainfall data set to make it active data set.
- From the menu, select **Statistics** -> **Summaries** -> **Active Dataset**. (Look at some of the other options available along the way.)




```
> summary(Rainfall)
      City        X             Rainfall
 Algiers: 1   Mode:logical   Min.   : 1.00
 Athens : 1   NA's:20        1st Qu.:22.75
 Beirut : 1                  Median :26.50
 Berlin : 1                  Mean   :29.85
 Bogota : 1                  3rd Qu.:34.25
 Bombay : 1                  Max.   :72.00
 (Other):14
```

- If you to edit your data set, click on the "**Edit data set**" tab below the main menu bar. Double-click on a cell to select it for editing.
- Now select **Statistics** -> **Summaries** -> **Numerical summaries**
- On the Statistics Tab, **uncheck** all the boxes except **Standard Deviation** (because we already have all the other statistics).
- Click OK.



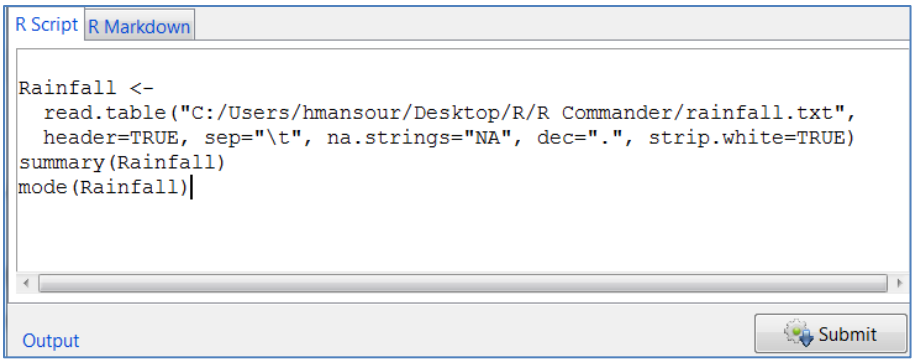
```
> numSummary(Dataset[,"Rainfall"], statistics=c("sd", "quantiles"),
+   quantiles=c(0,.25,.5,.75,1))
       sd 0%   25%  50%   75% 100%  n
 18.07084  1 22.75 26.5 34.25   72 20
```

Understanding the output:

output	What is it?
mean	Measure of central tendency
sd	Standard deviation - a measure of variability in the data
cv Coefficient of variance	The coefficient of variation (CV) is a normalized measure of variance. It is calculated as the ratio of the standard deviation to the mean. It can be compared across variables as the variability is now on a standardised scale.
Skewness	Skewness is a measure of symmetry. The output can be positive or negative. A negative value indicates negative skew indicates meaning that the *tail* on the left side of the distribution is *longer* than the right side and the bulk of the values lie to the right of the mean. A positive value indicates positive skew indicates that the *tail* on the right side is *longer* than the left side and the bulk of the values lie to the left of the mean. A zero value indicates that the values are relatively evenly distributed on both sides of the mean
kurtosis	Kurtosis is a measure of whether the data are peaked or flat relative to a normal distribution. A standard normal distribution has a kurtosis of zero. A positive kurtosis indicates a "peaked" distribution and negative kurtosis indicates a "flat" distribution.
n	Number of readings
NA	Number of missing values
0%	Minimum value
25%	The value below which 25 percent of the observations may be found.
50%	The value below which 50 percent of the observations may be found.
75%	The value below which 75 percent of the observations may be found.
100%	Maximum value

To try the R command line

In the **R Script window** type ***mode(Rainfall)*** , select this line, and then click on the **Submit** button that is between the top and middle window.



You will get something similar to this:

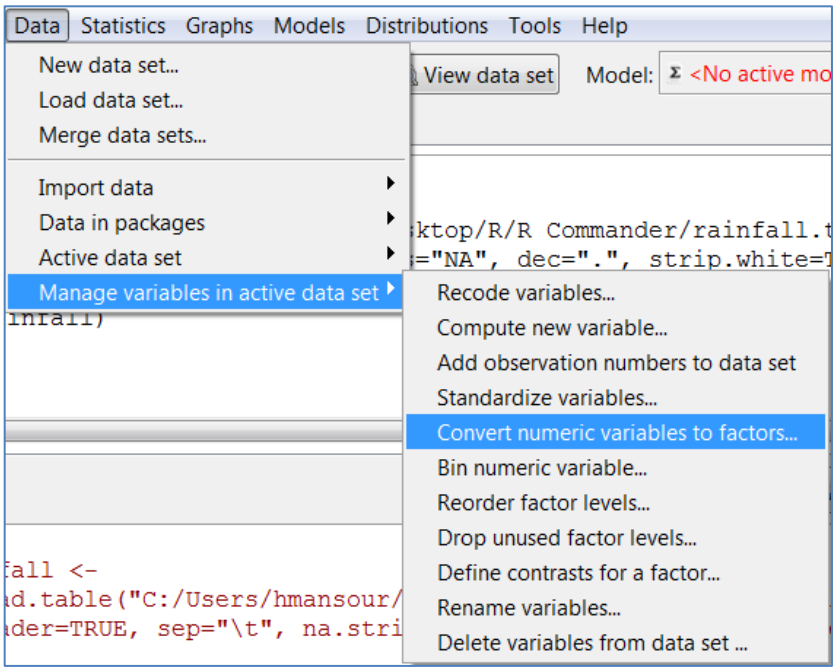
```
> mode(Rainfall)
[1] "list"
```

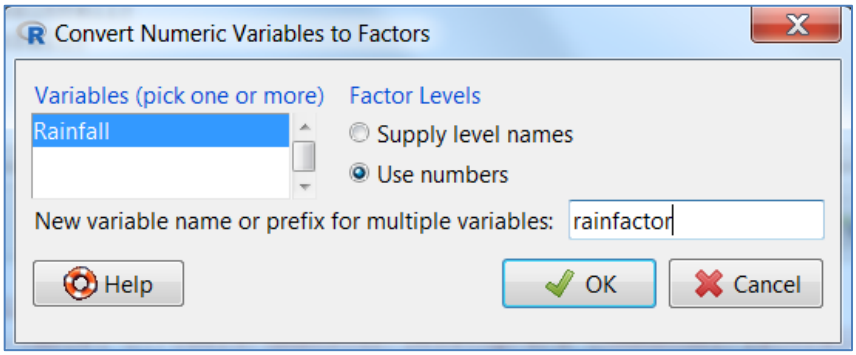
Here R is telling us that numerical data is what is stored (as a list) in Rainfall file. If you want to work with frequencies (the **mode** is the most frequent value).

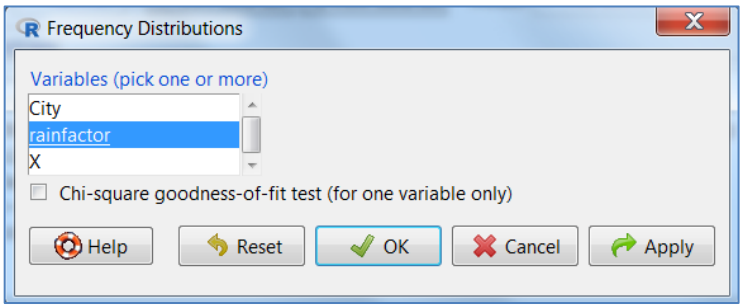
Categorical Data

With categorical data, you must tell R Commander to treat the data as categorical data.

- Select **Data** -> **Manage variables in active data set** -> **Convert numerical variables to factors**
- Check "**use numbers**" and in the text box **type a name for the new variable**, *rainfactor* for example.
- Select **Statistics** -> **Summaries** -> **Frequency distributions** (an option that was not available before).
- Select **rainfactor** -> click **OK**.








```
counts:
rainfactor
 1  2 16 17 22 23 25 26 27 30 34 35 42 48 71 72
 1  1  1  1  1  3  1  1  1  3  1  1  1  1  1  1

percentages:
rainfactor
 1  2 16 17 22 23 25 26 27 30 34 35 42 48 71 72
 5  5  5  5  5 15  5  5  5 15  5  5  5  5  5  5
```

The first table tells us 23 and 30 are tied for mode with three occurrences each.

To get this information in an easier to read format:

- Select **Statistics** -> **Summaries** -> **Active Dataset**

```
> summary(Rainfall)
     City       X             Rainfall       rainfactor
 Algiers: 1   Mode:logical   Min.   : 1.00   23     : 3
 Athens : 1   NA's:20        1st Qu.:22.75   30     : 3
 Beirut : 1                  Median :26.50   1      : 1
 Berlin : 1                  Mean   :29.85   2      : 1
 Bogota : 1                  3rd Qu.:34.25   16     : 1
 Bombay : 1                  Max.   :72.00   17     : 1
 (Other):14                                  (Other):10
```

Rainfall

	City	X	Rainfall	rainfactor
1	Algiers	NA	30	30
2	Lagos	NA	72	72
3	Athens	NA	16	16
4	LaPaz	NA	23	23
5	Beirut	NA	35	35
6	Lima	NA	2	2
7	Berlin	NA	23	23
8	London	NA	23	23
9	Bogota	NA	42	42
10	Madrid	NA	17	17
11	Bombay	NA	71	71
12	Moscow	NA	25	25
13	Cairo	NA	1	1
14	Oslo	NA	27	27
15	Dublin	NA	30	30
16	Paris	NA	22	22
17	Geneva	NA	34	34
18	Rome	NA	30	30
19	Havana	NA	48	48
20	Vienna	NA	26	26

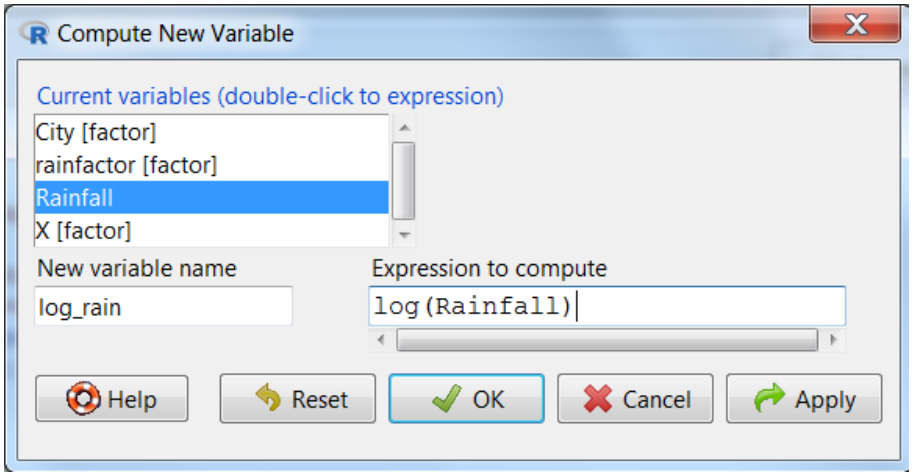
To Create Graphs

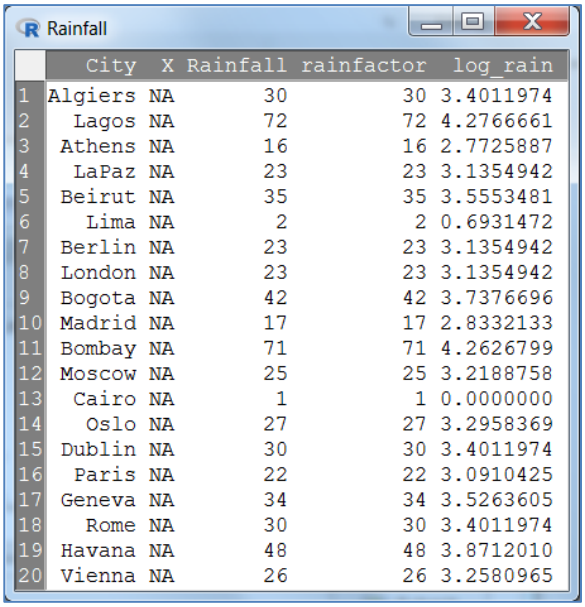
- From the main menu select **Graphs** -> **Stem and leaf display ...**
- On the **Options** tab, uncheck "**Trim outliers**" and "**Show depths**".
- Click **OK**.

```
leaf unit: 1
           n: 20
  0 | 12
  1 | 67
  2 | 2333567
  3 | 00045
  4 | 28
  5 |
  6 |
  7 | 12
```

Data Transformation

- For data such as this, which appears skewed toward high values in the stem-and leaf plot, a **transformation** is often appropriate.
- Select **Data** -> **Manage variables in active dataset** -> **Compute new variable...**
- In the right text box for **Expression to compute**, enter **log(Rainfall)** to compute the natural logarithm of the Rainfall data.
- In the left text box you can give the new variable a better name if you like; "**log_rain**", for example.
- Click OK.





	City	X	Rainfall	rainfactor	log_rain
1	Algiers	NA	30	30	3.4011974
2	Lagos	NA	72	72	4.2766661
3	Athens	NA	16	16	2.7725887
4	LaPaz	NA	23	23	3.1354942
5	Beirut	NA	35	35	3.5553481
6	Lima	NA	2	2	0.6931472
7	Berlin	NA	23	23	3.1354942
8	London	NA	23	23	3.1354942
9	Bogota	NA	42	42	3.7376696
10	Madrid	NA	17	17	2.8332133
11	Bombay	NA	71	71	4.2626799
12	Moscow	NA	25	25	3.2188758
13	Cairo	NA	1	1	0.0000000
14	Oslo	NA	27	27	3.2958369
15	Dublin	NA	30	30	3.4011974
16	Paris	NA	22	22	3.0910425
17	Geneva	NA	34	34	3.5263605
18	Rome	NA	30	30	3.4011974
19	Havana	NA	48	48	3.8712010
20	Vienna	NA	26	26	3.2580965

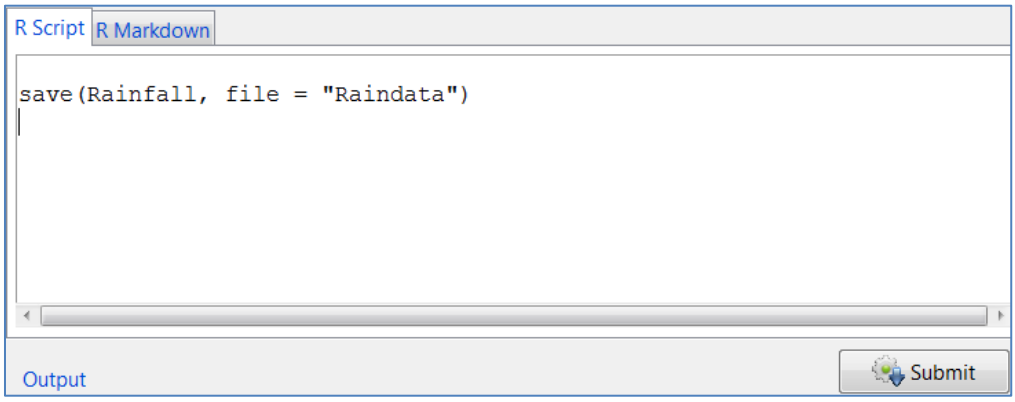
Save Your Data!

When you exit R you will be asked if you want to save your workspace. Say "yes", if you wish datasets you created will be available next time you start R.

You can use the command line

save(Rainfall, file = "Raindata")

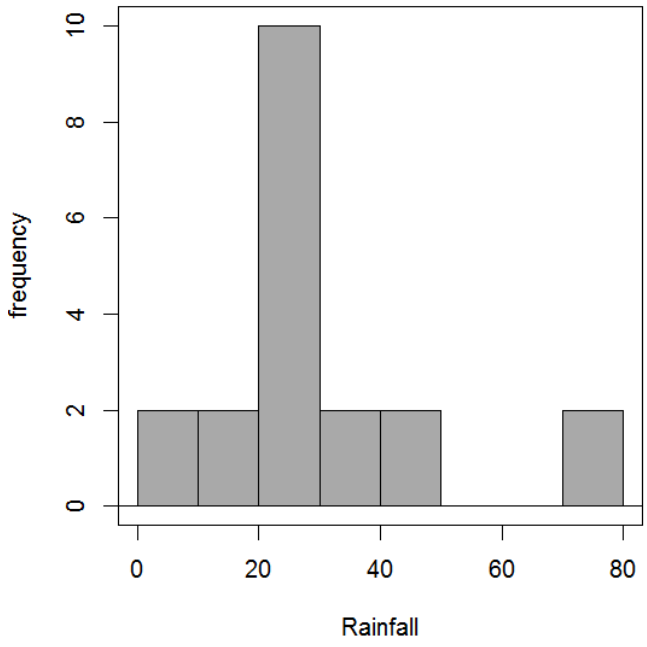
Type this command in the **R Script Window**, select it, and then press the **Submit** button.



Note that this process just saves your data sets. If you want text output, and graphics saved, you must handle each one separately.

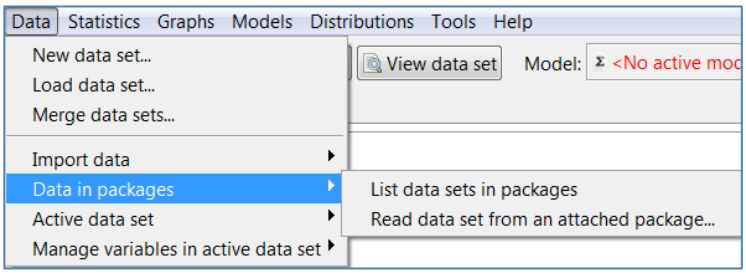
Creating Histogram, using the Rainfall data set

- Select **Graphs** > **Histogram** from the main menu.



Reading Data from a Package

- Many R packages include datasets.
- Datasets in packages can be listed in a pop-up window via Data -> Data in packages -> List data sets in packages, and can be read into the
- Commander via Data -> Data in packages -> Read data set from an attached package.
- Double-clicking on the name of a package displays its data sets in the right list box; and double-clicking on a data set name copies the name to the data-set entry field in the dialog.
- Pressing a letter key in the Data set list box will scroll to the next data set whose name begins with that letter.
- You can access additional R packages that are installed in your package library by Tools -> Load packages.



t-Tests

A t-test is an analysis of two populations means through the use of statistical examination; a t-test with two samples is commonly used with small sample sizes, testing the difference between the samples when the variances of two normal distributions are not known.

To conduct a test with three or more variables, an analysis of variance (**ANOVA**) must be used.

Types of t-tests are:

- **One sample t-Test**
- **Two-sample t-Tests (independent samples t-Test)**
- **Paired t-Test**

Summary Statistics

To load the **MASS package**, and then using the crabs dataset in that package:

- **Tools** -> **Load Packages** -> **MASS -> OK.**
- **Data** -> **Data in packages** -> **Read data from package**.
- Double-click the **MASS** package -> Scroll through data sets, and select **crabs** -> click the Help on data set -> close the Browser.
- Click OK.
- Statistics menu -> Summaries -> Active data set

```
> data(crabs, package="MASS")

> summary(crabs)
 sp      sex         index           FL              RW              CL              CW
 B:100   F:100   Min.   : 1.0   Min.   : 7.20   Min.   : 6.50   Min.   :14.70   Min.   :17.10
 O:100   M:100   1st Qu.:13.0   1st Qu.:12.90   1st Qu.:11.00   1st Qu.:27.27   1st Qu.:31.50
                 Median :25.5   Median :15.55   Median :12.80   Median :32.10   Median :36.80
                 Mean   :25.5   Mean   :15.58   Mean   :12.74   Mean   :32.11   Mean   :36.41
                 3rd Qu.:38.0   3rd Qu.:18.05   3rd Qu.:14.30   3rd Qu.:37.23   3rd Qu.:42.00
                 Max.   :50.0   Max.   :23.10   Max.   :20.20   Max.   :47.60   Max.   :54.60
       BD
 Min.   : 6.10
 1st Qu.:11.40
 Median :13.90
 Mean   :14.03
 3rd Qu.:16.60
 Max.   :21.60

> help("crabs", package="MASS")
```

Crabs Dataset Descriptions:

The crabs data set has 200 rows and 8 columns, describing 5 measurements on 50 crabs each of two color forms and both sexes. This data set contains the following columns:

- Sp - species - "B" or "O" for blue or orange.
- Sex - as it says.
- Index - index 1:50 within each of the four groups.
- FL - **frontal lobe** size (mm).
- RW - rear width (mm).
- CL - carapace length (mm).
- CW - carapace width (mm).
- BD - body depth (mm)

One Sample t-Test

To test whether the crabs population average for frontal lobe size (FL) is different than 15 mm, we perform two sided One-sample t-Test:

- **Statistics** -> **Means** -> **Single-sample t-Test …**
- Select the **FL** variable, and type 15 in the "Null hypothesis mu=" box -> click OK.
- Notice that there are some evidence that the crabs population front lobe size is different than 15 millimeter.

```
        One Sample t-test

data:  FL
t = 2.3588, df = 199, p-value = 0.0193
alternative hypothesis: true mean is not equal to 15
95 percent confidence interval:
 15.09562 16.07038
sample estimates:
mean of x
   15.583
```

Two Sample t-Test (Independent Samples t-Test)

To test Whether there is a difference between male and female crabs population average for frontal lobe size (FL), we perform Independent Sample t-Test:

- **Statistics** -> **Means** -> **Independent Samples t-Test**
- Select **sex** as a **Group**, and **FL** as a **Response** variable -> click OK.

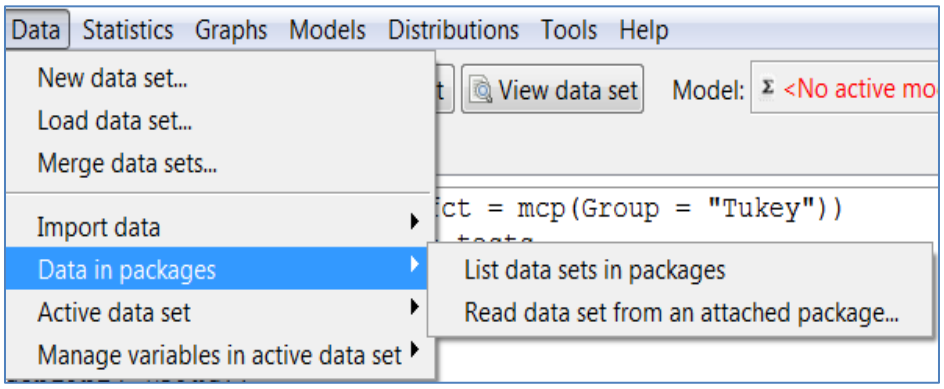
```
        Welch Two Sample t-test

data:  FL by sex
t = -0.60998, df = 197.91, p-value = 0.5426
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.2783394  0.6743394
sample estimates:
mean in group F mean in group M
         15.432          15.734
```

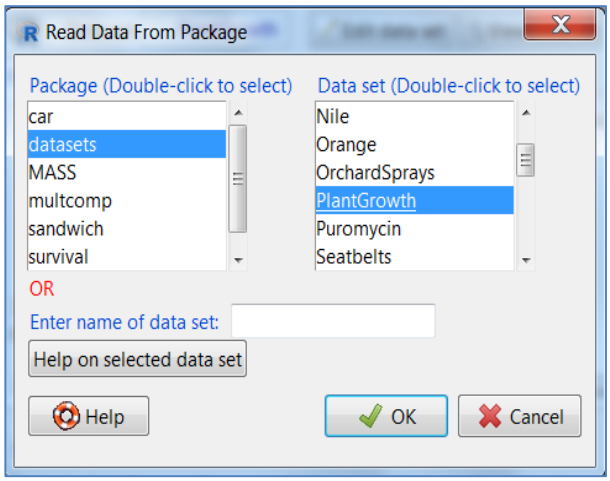
One-Way ANOVA

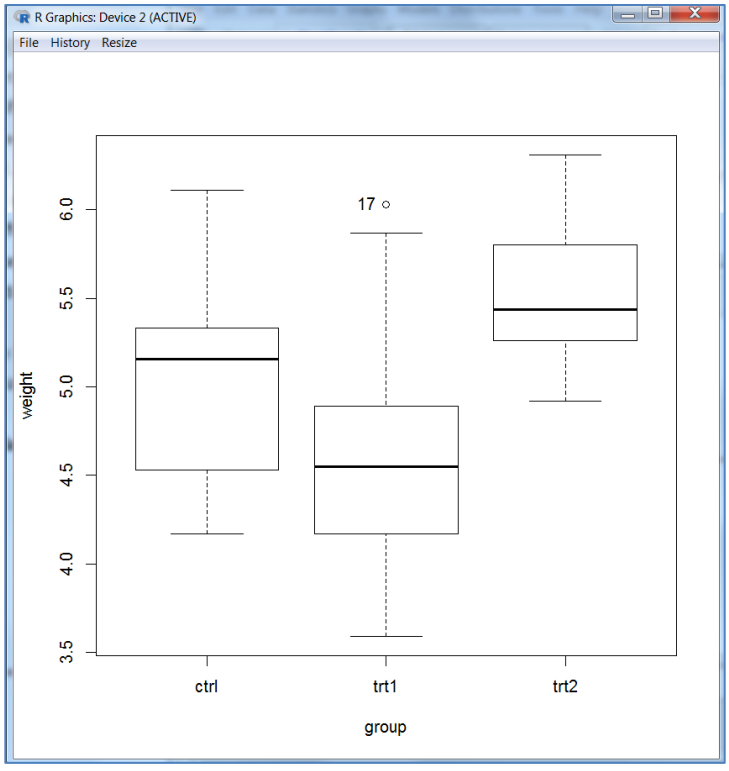
The one-way analysis of variance (**ANOVA**) is used to determine whether there are any statistically significant differences between the means of three or more independent (unrelated) groups. Analysis of variance compares the variance between the different groups (believed to be due to the grouping variable) with the variability within each of the groups (believed to be due to chance). The **R Commander** uses **Tukey post-hoc** comparison procedure.

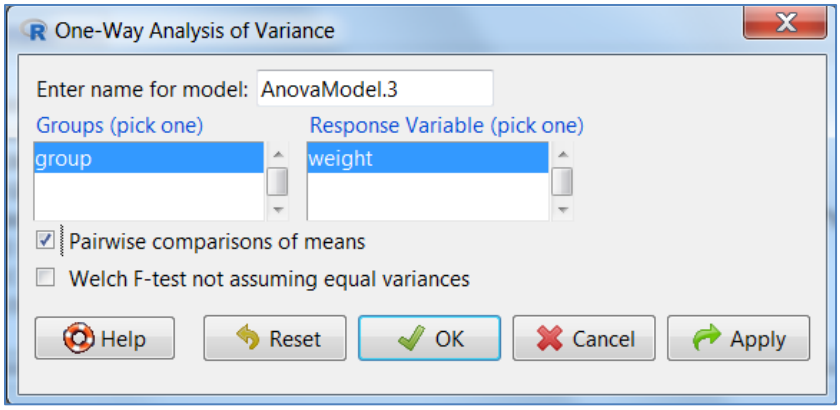
Let's use one of the datasets included in the Base System of the R Commander:

- From **Data** menu, select **Data in Packages** -> Read data set from an attached package …
- From the **Packages** list, double-click the **datasets**, and from the **Data set** list, select **PlantGrowth**
- Enter a name of data set: PlantGrowth -> click **OK**.
- From the **Graphs** menu, select **Box Plot …**-> click the **plot by groups …**-> click **OK** -> click **OK**, again.
- Now to compare means of these three groups, from **Statistics** menu, select **Means** -> **One-Way ANOVA… ->** Type **AnovaModel.3**, for the name of the model -> **c**heck the **Pairwise comparisons of means** -> Click OK.


Data
Statistics
Graphs
Models
Distributions
Tools
Help
New data set...
Load data set...
Merge data sets...
Import data
Data in packages
Active data set
Manage variables in active data set
View data set
Model:
<No active mo
ct = mcp(Group = "Tukey"))
List data sets in packages
Read data set from an attached package...


Read Data From Package
Package (Double-click to select)
Data set (Double-click to select)
car
datasets
MASS
multcomp
sandwich
survival
Nile
Orange
OrchardSprays
PlantGrowth
Puromycin
Seatbelts
OR
Enter name of data set:
Help on selected data set
Help
OK
Cancel


R Graphics: Device 2 (ACTIVE)
File
History
Resize
weight
6.0
5.5
5.0
4.5
4.0
3.5
17
ctrl
trt1
trt2
group



Here are the Outputs:

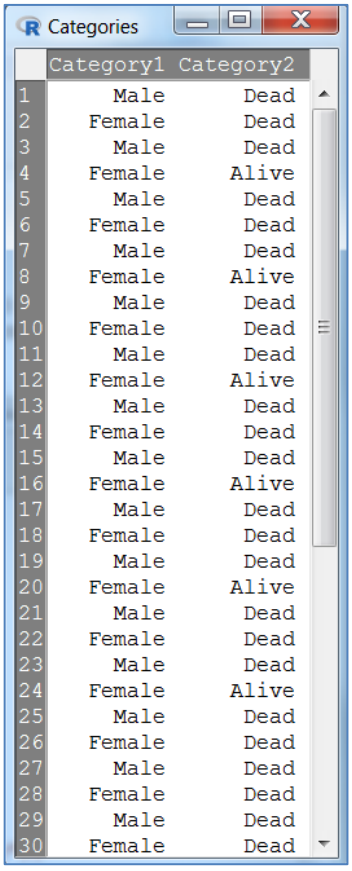
```
> AnovaModel.3 <- aov(weight ~ group, data=PlantGrowth)

> summary(AnovaModel.3)
            Df Sum Sq Mean Sq F value Pr(>F)
group        2  3.766  1.8832   4.846 0.0159 *
Residuals   27 10.492  0.3886
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> with(PlantGrowth, numSummary(weight, groups=group, statistics=c("mean",
+   "sd")))
      mean        sd data:n
ctrl 5.032 0.5830914     10
trt1 4.661 0.7936757     10
trt2 5.526 0.4425733     10
```

```
         Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: aov(formula = weight ~ group, data = PlantGrowth)

Linear Hypotheses:
                 Estimate Std. Error t value Pr(>|t|)
trt1 - ctrl == 0  -0.3710     0.2788  -1.331    0.391
trt2 - ctrl == 0   0.4940     0.2788   1.772    0.198
trt2 - trt1 == 0   0.8650     0.2788   3.103    0.012 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

```
         Simultaneous Confidence Intervals

Multiple Comparisons of Means: Tukey Contrasts


Fit: aov(formula = weight ~ group, data = PlantGrowth)

Quantile = 2.4798
95% family-wise confidence level


Linear Hypotheses:
                 Estimate lwr     upr
trt1 - ctrl == 0 -0.3710  -1.0623  0.3203
trt2 - ctrl == 0  0.4940  -0.1973  1.1853
trt2 - trt1 == 0  0.8650   0.1737  1.5563

ctrl trt1 trt2
"ab"  "a"  "b"
```

Contingency Tables, using the Categories dataset

- **Data -> Import data -> From Excel file**
- Type **Categories** in **Enter Name for dataset:** box -> click **OK**
- Navigate and find your file in your computer -> click Open
- Click the **View data set** button, to view your data set.
- Select the **Statistics** menu -> **Contingency Tables…** -> **two-way table…**
- Specify the row and column variables
- On **Statistics** Tab, select the **Chi-square test of independence ->** Click **OK**. A table of observed values, the Pearson's Chi-square test output will appear in the output window.



	Category1	Category2
1	Male	Dead
2	Female	Dead
3	Male	Dead
4	Female	Alive
5	Male	Dead
6	Female	Dead
7	Male	Dead
8	Female	Alive
9	Male	Dead
10	Female	Dead
11	Male	Dead
12	Female	Alive
13	Male	Dead
14	Female	Dead
15	Male	Dead
16	Female	Alive
17	Male	Dead
18	Female	Dead
19	Male	Dead
20	Female	Alive
21	Male	Dead
22	Female	Dead
23	Male	Dead
24	Female	Alive
25	Male	Dead
26	Female	Dead
27	Male	Dead
28	Female	Dead
29	Male	Dead
30	Female	Dead

```
Frequency table:
         Category2
Category1 Alive Dead
   Female     6   13
   Male       5   25

        Pearson's Chi-squared test

data:  .Table
X-squared = 1.4859, df = 1, p-value = 0.2229
```

Linear Regression Analysis

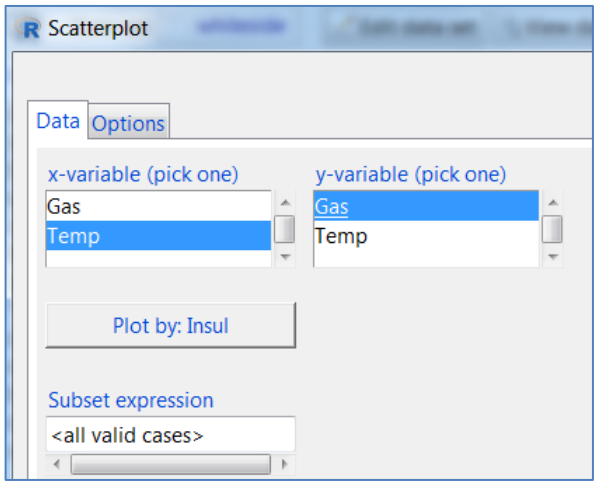
To assess the effect of the insulation on gas consumption, we will be using the **whiteside dataset**, of weekly gas consumption and average external temperature at a house included in **MASS** Package.

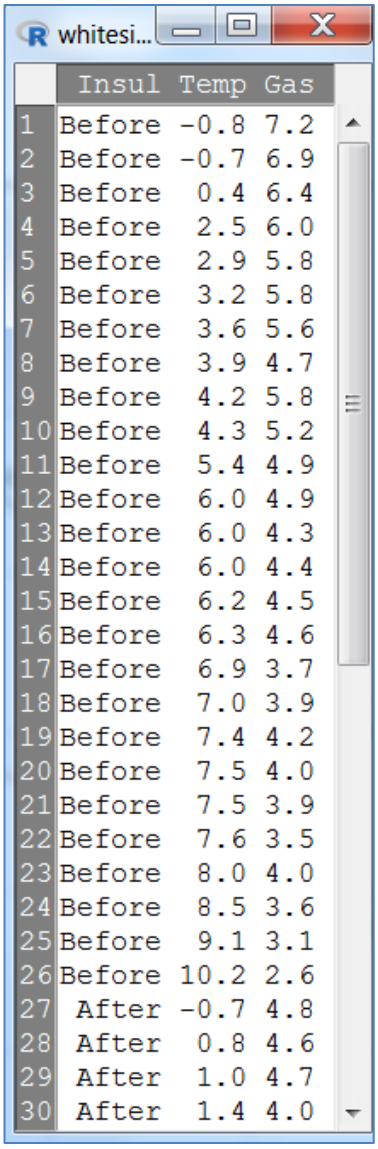
To load the **MASS package**, and then using the **Whiteside** dataset in that package:

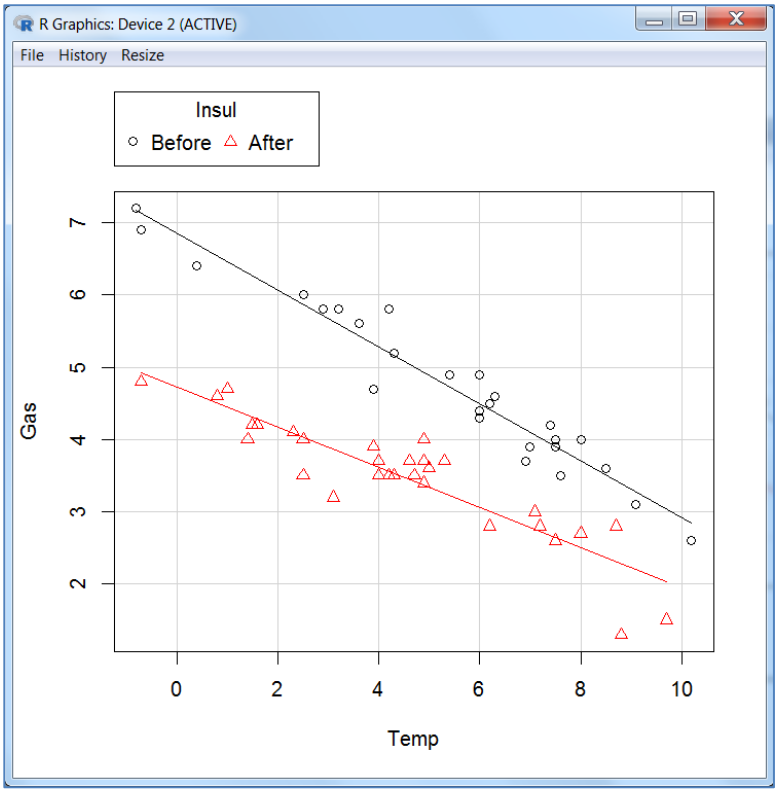
- **Data** -> **Data in packages** -> **Read Data From Package**
- Double-click on **MASS** -> Click on **Whiteside** -> click **OK** -> OK.
- **Whiteside**. The whiteside data set has 56 rows and 3 columns, and includes one **categorical** data type, and two **continuous** data type.
    - **Insul** - A factor, before or after insulation.
    - **Temp** - The average outside temperature in degrees Celsius.
    - **Gas** - The weekly gas consumption in 1000s of cubic feet.

To explore whiteside dataset

- From the **Graphs** menu, select **Scatterplot …**
- Select **Temp** as x-variable, and **Gas** as y-variable.
- **Plot by:** Insul (is selected) - OK
- On the **Options** Tab, check the "**least-squares line**" box.
- Click **OK**.

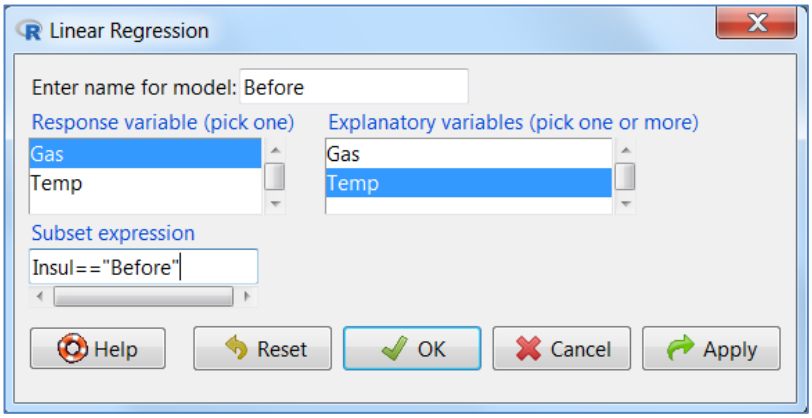






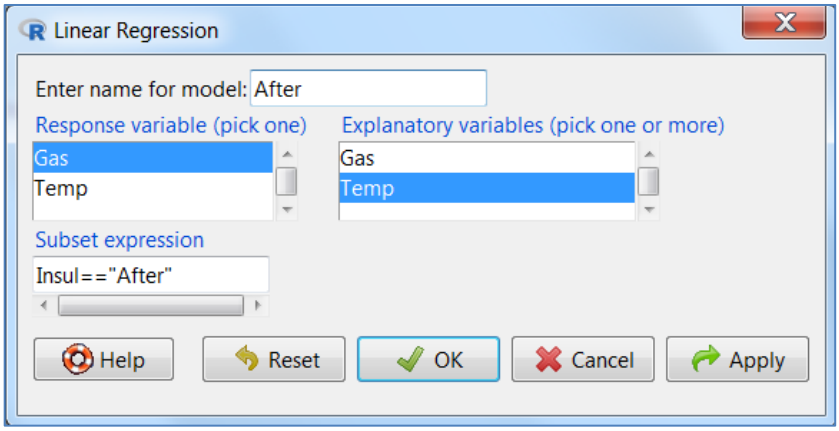
Linear Regressions for Subset of data

- **Statistics** -> **Fit Model…** -> **Linear Regression**
- Enter a name for model: **Before**
- Select **Gas** as Response variable, and **Temp** as Explanatory variable
- In the Subset expression box, type **Insul=="Before"** -> click **OK**.




```
> Before <- lm(Gas~Temp, data=whiteside, subset=Insul=="Before")

> summary(Before)

Call:
lm(formula = Gas ~ Temp, data = whiteside, subset = Insul ==
    "Before")

Residuals:
     Min       1Q   Median       3Q      Max
-0.62020 -0.19947  0.06068  0.16770  0.59778

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.85383    0.11842   57.88   <2e-16 ***
Temp        -0.39324    0.01959  -20.08   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2813 on 24 degrees of freedom
Multiple R-squared:  0.9438, Adjusted R-squared:  0.9415
F-statistic: 403.1 on 1 and 24 DF,  p-value: < 2.2e-16
```

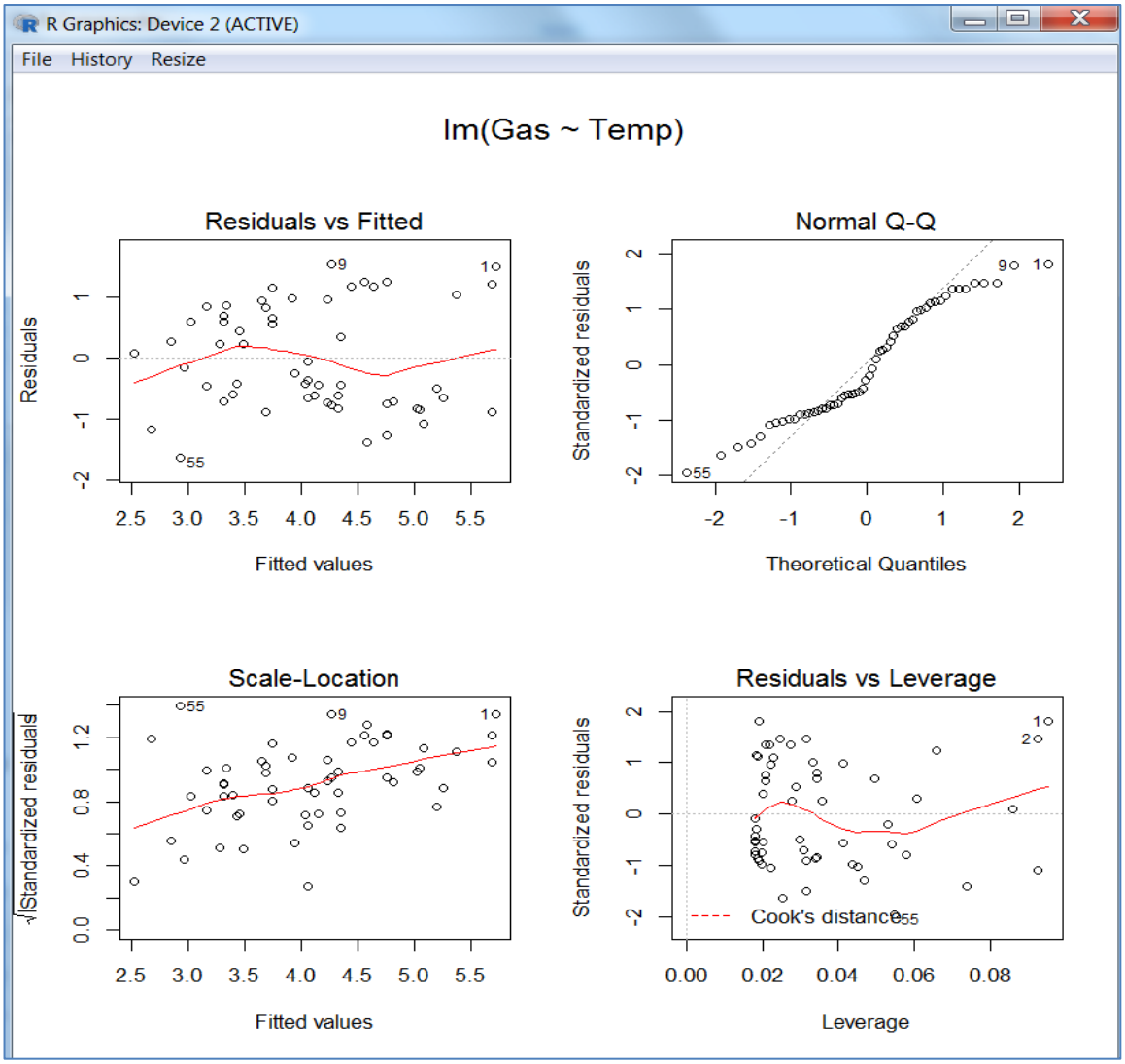
- Again, from **Statistics** menu-> **Fit Model…** -> **Linear Regression**
- Enter a name for model: **After**
- Select **Gas** as Response variable, and **Temp** as Explanatory variable
- In the Subset expression box, type **Insul==”After”** -> click **OK**.




```
> After <- lm(Gas~Temp, data=whiteside, subset=Insul=="After")

> summary(After)

Call:
lm(formula = Gas ~ Temp, data = whiteside, subset = Insul ==
    "After")

Residuals:
     Min       1Q   Median       3Q      Max
-0.97802 -0.11082  0.02672  0.25294  0.63803

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.72385    0.12974   36.41  < 2e-16 ***
Temp        -0.27793    0.02518  -11.04 1.05e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3548 on 28 degrees of freedom
Multiple R-squared:  0.8131, Adjusted R-squared:  0.8064
F-statistic: 121.8 on 1 and 28 DF,  p-value: 1.046e-11
```

Diagnostics for Linear Models

- Make sure that graph window is closed, and then from **Models** menu -> **Graphs** -> **Basic diagnostic plots**.



Saving and Printing Data

- R Commander offers you several menu options for saving and printing data.
- Select to save the active dataset as a ***.rda** data file.
- You can also select the menu option **Data**-> **Active dataset** ->**Export active dataset** to allow you to save the active dataset as a text file (commonly called cvs file).
- You can save text output directly from the **File** menu in the R Commander; likewise you can save or print a graph from the **File** menu in an R Graphics Device window.

Terminating the R Session

There are several ways to terminate your session.

- Select **File** -> **Exit** from **R Commander** window . You will be asked to confirm, and then asked whether you want to save the contents of the R Script, Output, and R Markdown windows.
- **File** -> **Exit** from the **R Console**. You will be asked whether you want to save the R workspace (i.e., the data that R keeps in memory); you would normally answer **No**.

Where to Get Help

If you need help from me, please e-mail heide.mansouri@ttu.edu. Alternatively, call 834-2935 to make an appointment.

Please e-mail your comments or suggestions to heide.mansouri@ttu.edu