

# Introduction to R & R Commander

Alexander Ploner <[alexander.ploner@ki.se](mailto:alexander.ploner@ki.se)>

2011-03-18



## Contents

<b>1</b>	<b>Getting started</b>	<b>3</b>
1.1	First steps . . . . .	3
1.2	A simple tutorial session . . . . .	6
1.3	Statistical data . . . . .	9
<b>2</b>	<b>Common tasks</b>	<b>10</b>
2.1	Enter data manually . . . . .	10
2.2	Import a text/Excel file . . . . .	10
2.3	Load plug-ins . . . . .	10
2.4	Save plots for MS Word . . . . .	10
2.5	Copy and paste with MS Office . . . . .	10
<b>A</b>	<b>Installation</b>	<b>11</b>
A.1	Install R . . . . .	11
A.2	Install Rcmdr . . . . .	12

# 1 Getting started

## 1.1 First steps

You can start R through the Start menu in the usual manner. After the start-up, the program window should look like Figure 1(a): though the usual menus (File, Edit, Help etc.) plus a few others are available, R is mostly controlled through the command line, i.e. by typing commands into the R console window shown within the main application. The commands then generate numerical output (usually also displayed in the console) and/or graphical output, which is shown in a separate graphics window.

While this is a perfectly feasible and eventually very powerful way of using R, it may require some getting used to, especially if one has little previous experience with the command line.

For starters, however, we make use of the *R Commander*, a graphical interface that allows the user to perform many common statistical tasks through a simple menu system. You can start the R Commander either through typing `library(Rcmdr)` at the prompt in the R console window (and pressing Return), or through the menu entry **Packages/Load packages** (and selection of `Rcmdr` from the list of packages, confirmed with Ok).

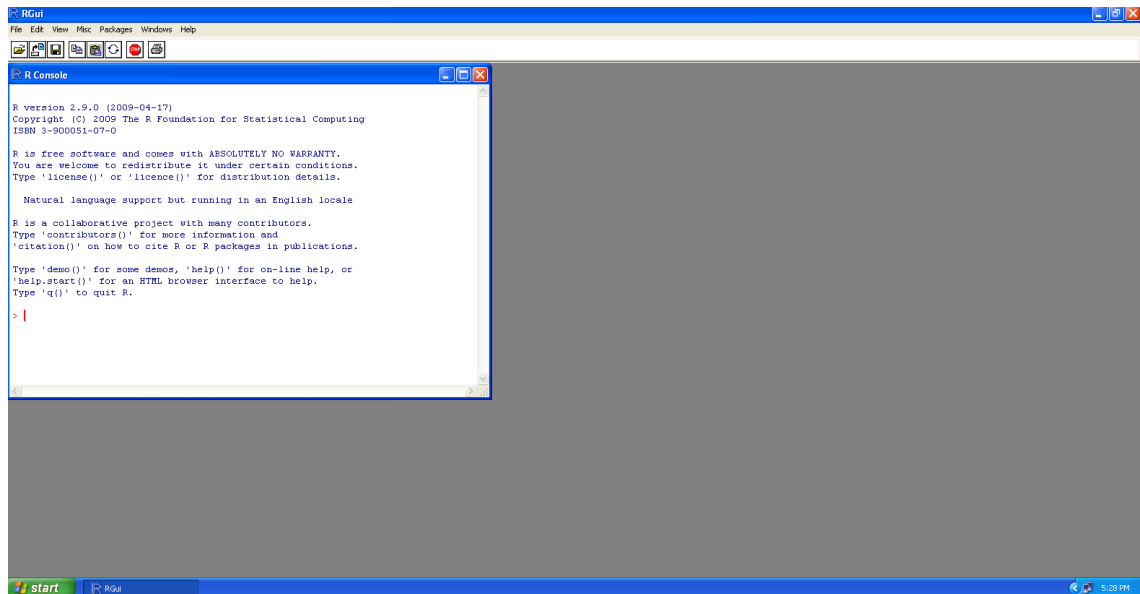
In both cases, the R Commander main window should appear within the R application, as seen in Figure 1(b). If the R Commander window is for some reason not immediately visible, it is probably just minimized in the Taskbar and can be activated through clicking on it.

Let's have a closer look at the R Commander window in Figure 2. We only give a short overview of its components in this section, more details and a demonstration follow in the next section.

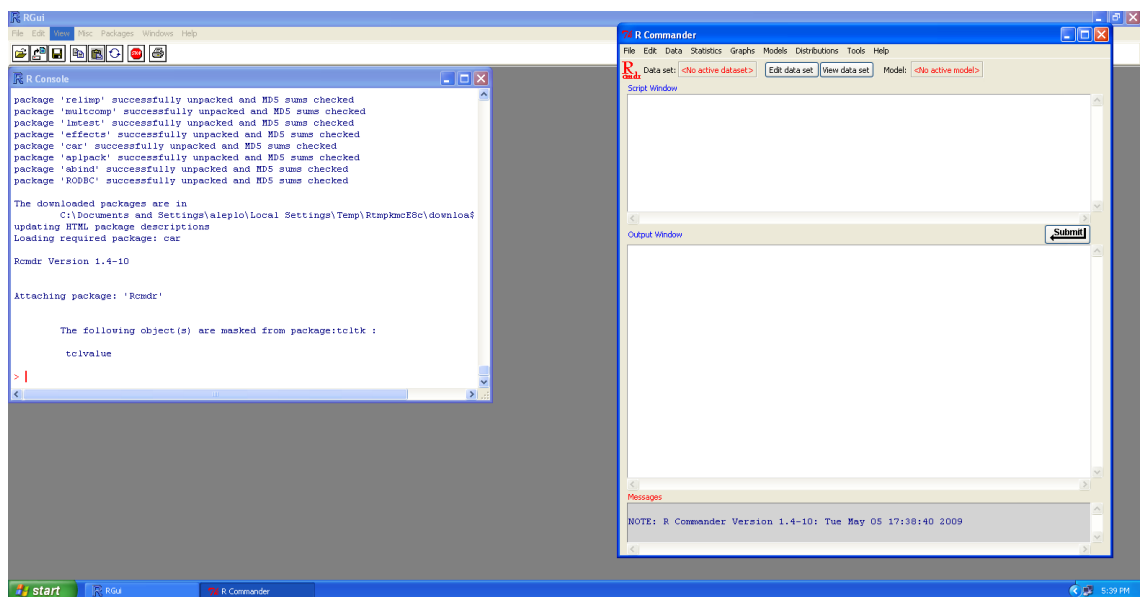
The main elements of the R Commander are (from top to bottom) menus, a toolbar, an upper text window labelled *Script Window*, a lower text window labelled *Output Window*, and a text field (in grey) labelled *Messages*.

The menus allow the user to load, edit, and save data, perform numerous types of statistical analysis, generate a variety of graphical representations, and to manipulate configurations and settings for R and the R commander. Typically, each menu or sub-menu entry starts a dialogue where the user can select the data and variables they want to process, as well as some settings. After confirmation with Ok, the R Commander takes the input from the dialogue window and generates the corresponding R command, which is then executed in the background. In that sense, the R Commander only serves as a bridge between the user and the complex R command language in the background.

- The command so generated is by default shown in the *Script Window*.
- The numerical output (if any) generated when running the command is shown in the *Output Window*.
- Any errors, warnings, or external information generated during execution of the command are shown in the *Messages* field.



(a) R window with console only, after program start



(b) R window after start of the R Commander

Figure 1: The main window of the R statistical software, before and after starting the R Commander GUI

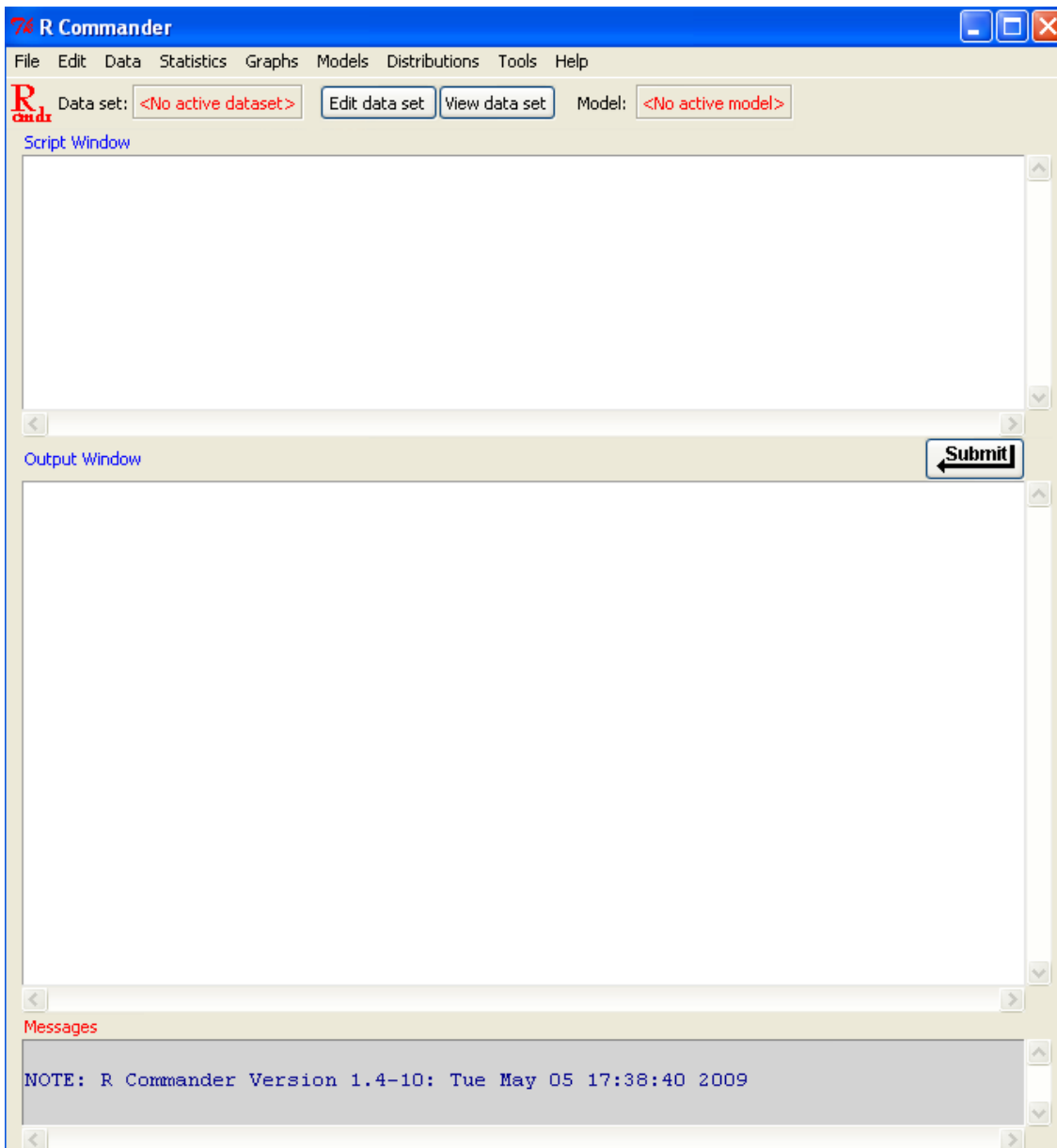
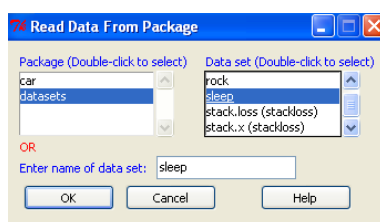
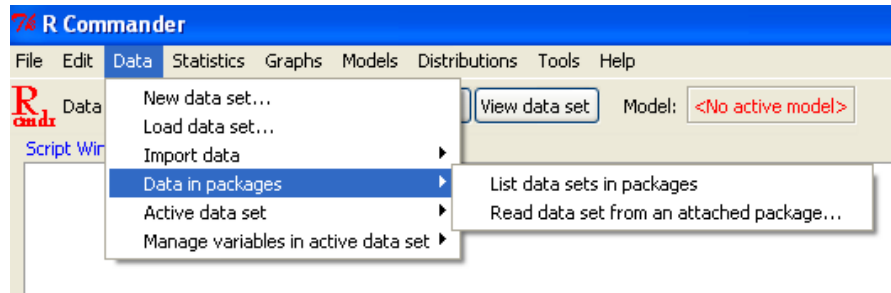


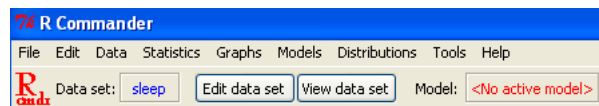
Figure 2: The R Commander window directly after start; no active data set has been set yet.

## 1.2 A simple tutorial session

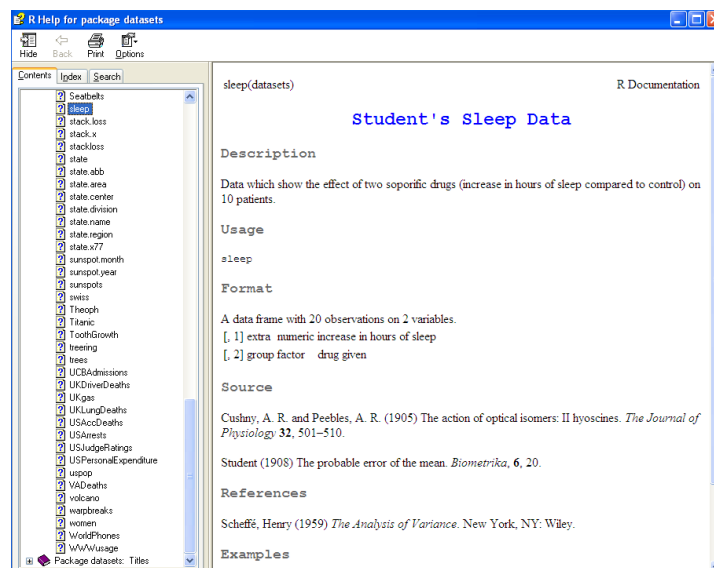
**Load a data set** There are different ways of getting data into R, as discussed in the next section. Here, we go with the easiest option and use example data that comes as part of the R installation. These can be accessed via the menu `Data/Data in packages/Read data set from an attached package`:



At start up, only two add-on packages have been attached to base R, `car` and `datasets`, each with multiple example data sets. We choose the set `sleep` from package `datasets` (left). As soon as we have confirmed this selection via `Ok`, `sleep` becomes the active data set for R Commander and is shown as such in the tool bar (below).



**Working with the active data set** For prepackaged data like this there is usually some description available, which can be accessed via menu `Help/Help on active data set`

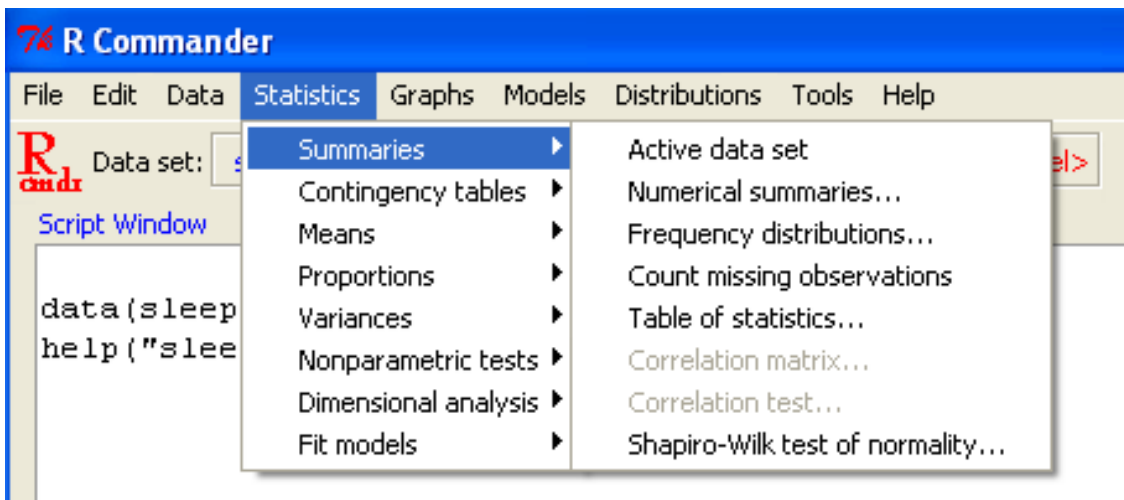


	extra	group	var3	var4
1	0.7	1		
2	-1.6	1		
3	-0.2	1		
4	-1.2	1		
5	-0.1	1		
6	3.4	1		
7	3.7	1		
8	0.8	1		
9	0	1		
10	2	1		
11	1.9	2		
12	0.8	2		
13	1.1	2		
14	0.1	2		
15	-0.1	2		
16	4.4	2		
17	5.5	2		
18	1.6	2		
19	4.6	2		
20	3.4	2		
21				

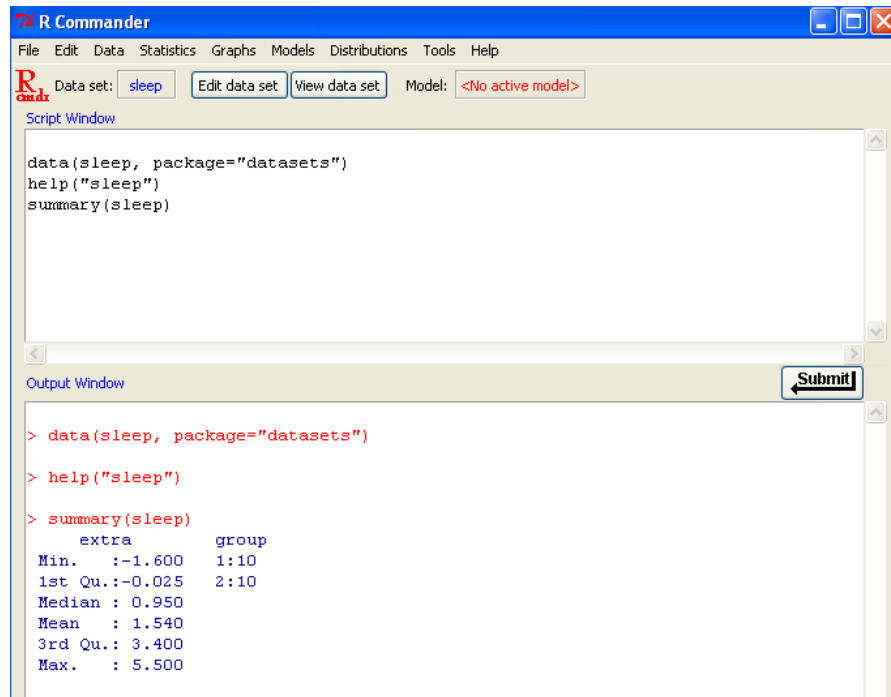
You can also edit the active data set via the **Edit data set** button in the tool bar. This opens a simple spreadsheet showing the data set as rows (subjects) and columns (variables), see left. As promised in the help screen, the data set `sleep` contains two variables, `extra` and `group`, on 20 subjects in two groups (1 and 2). Note that you cannot perform any other activity with the R Commander while the editing window is open; you have to close it to get back to the Commander menus.

The **View data set** button works exactly the same, except that the resulting spreadsheet window is read-only, so that nothing can be changed.

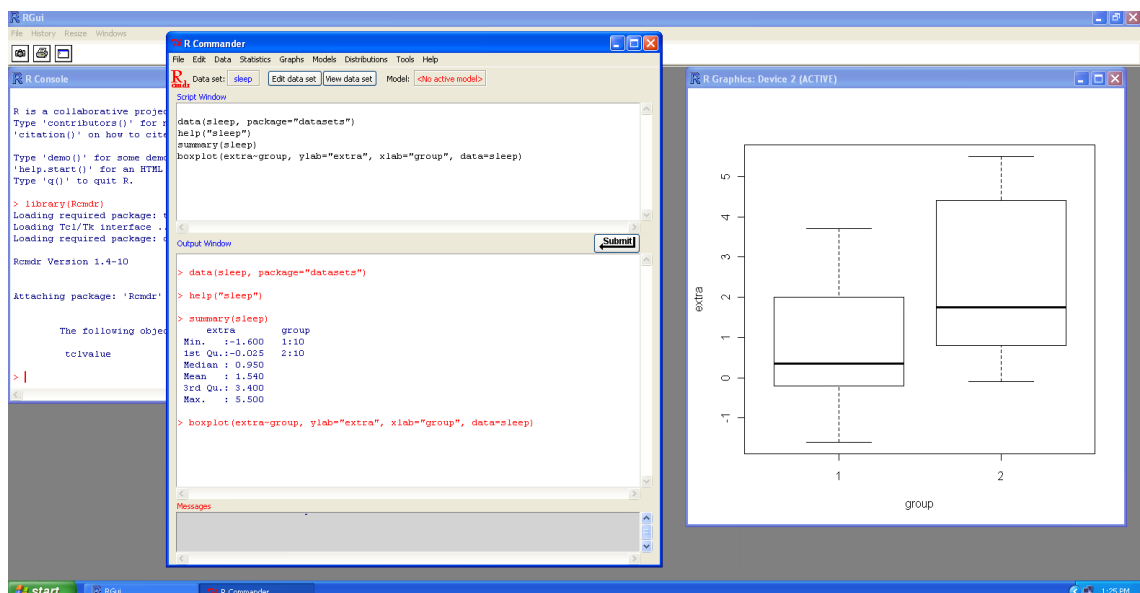
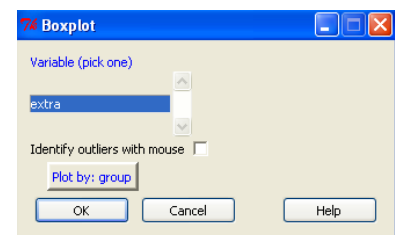
**Descriptive statistics** We want to calculate some simple descriptive statistics for the active data set. For our example, we select the menu **Statistics/Summaries/Active data set** (note how now that we have selected an active data set, the entries in the menu **Statistics** are no longer greyed-out):



This generates the output below; note how the Script Window shows the commands that we have started via the R Commander so far: `data` to load the active data set, `help` to get some extra information, and finally `summary` to calculate descriptive statistics, which are shown below in the Output Window: for the quantitative `extra` variable, we get minimum, maximum, first and third quartile and mean and median; for the qualitative variable `group`, we only get the frequency distribution of the different values.



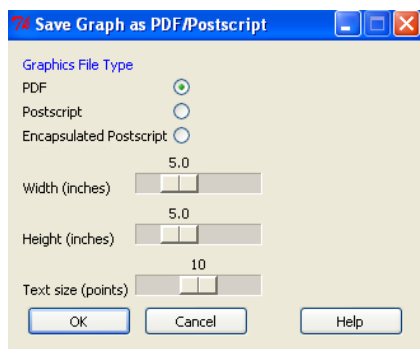
**Plot the data** We want to compare the extra sleep achieved in the two groups graphically. This can be done via two side-by-side boxplots, one for each group. We start the corresponding dialogue through the menu **Graphs/Boxplots** (right). Confirming with **Ok** shows the plot in a new graphics window as seen below. This graphics window can easily live together with an active R Commander window, so there is no need to close it like the editing window.





**Save the results** In principle, we can always copy the numerical output from the Output Window and the graphical output from the graphics window and paste them into our favorite word processor. This has however several disadvantages: pasted graphics are generally poor quality, and even if we keep the calculated results, it may be hard or impossible to reconstruct how the results were generated in detail without extra documentation. The recommendation is therefore to save both the Output Window (which contains both commands and results) and any interesting plot to file. For the output window, this can be done through the menu **File/Save output** (or **Save output as**). The resulting file is a pure text file with extension `.txt` and can be read in any text editor (e.g. Notepad).

Note that there are other saving options in the File menu, which we discuss below in the next section. For graphs, we have the menu **Graphs/Save graph to file** with the options **as bitmap** and **as PDF/Postscript/EPS**. For professional quality graphs that look reasonable even when zoomed or changed in size, we recommend using the second option, which brings us to the dialogue shown on the right.



Apart from the file type, the user can specify the image size in (inches, where 1 inch = 2.54 cm) and font size in points. Generally, it is sufficient to use the defaults and to change the size of the resulting graph when including it in a word processing document.

**Quitting the R Commander** This can be done in the usual way, by clicking the Close button in the upper right corner, or via the menu **File/Exit/From Commander**; if the content of the Script or Output Window has been changed, the Commander offers to save them.

To quit R, either choose the menu **File/Exit/From Commander and R**, or click the Close button on the R window. R will *always* offer to save the workspace, i.e. all the currently defined objects (data sets, analysis results etc.) in the working memory; these can be stored as `.RData` files and later be re-loaded via the R menu **File/Load workspace**. However, if you only work with the R Commander, and if you have saved the script- and output windows previously, this will be rarely necessary.

### 1.3 Statistical data

Statistical data sets are generally arranged like spreadsheets, as a rectangular grid of rows and columns, where

- rows correspond to different subjects (patients, mice, cell lines...),
- columns correspond to different measurements on this set of subjects (age, sex, height, weight of patients; weight and tumor size of mice; proliferation rate and gene expression in cell line).

This makes it possible to e.g. collect data in a spreadsheet program like MS Excel (or OOcalc or Gnumeric) and to import it into your favorite statistical software later on. However, the format for statistical data is much stricter, it is *only* rows and columns in a rectangle: no comments, formulas, graphs, decorations, empty lines/columns...

Many statistical programs only allow working with one data set at a time; examples are Stata, SPSS, and R Commander (active data set). Other programs allow the use of multiple data sets in parallel (e.g. SAS or R).

## 2 Common tasks

### 2.1 Enter data manually

1. Start a new dataset through menu `Data/New data set`
2. Choose a name for the new data set
3. Define variables (columns) through clicking on the column label; choose name and type (numeric = quantitative, character = qualitative)
4. Enter the data in the cells
5. Click the Close button on the Edit window when done

### 2.2 Import a text/Excel file

Menu `Data/Import data/From Excel`, choose a name for the data set to be imported (as above), and select the file.

### 2.3 Load plug-ins

Extending the R Commander itself: menu `Tools/Rcmdr plugin(s)` (e.g. `TeachingDemos`)

### 2.4 Save plots for MS Word

1. Right-click the plot
2. Choose `Save as metafile`
3. In MS Word: insert graphics from file.

### 2.5 Copy and paste with MS Office

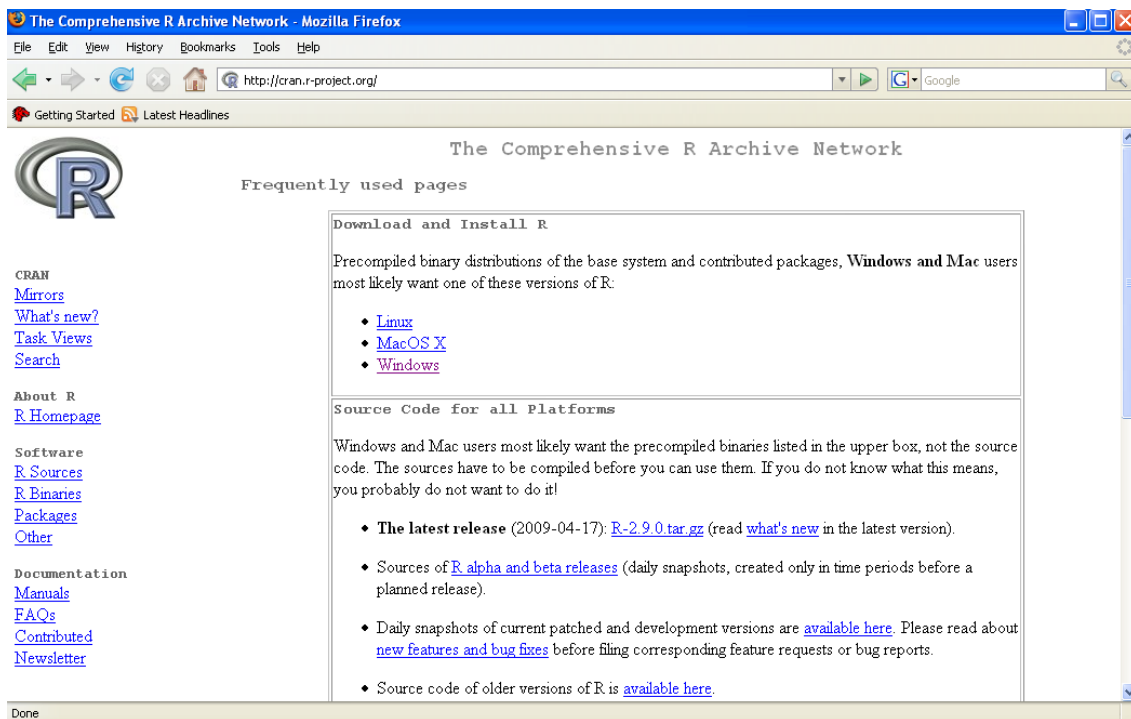
Especially useful for tables:

1. Load the Rcmdr plugin `Export`
2. Generate the desired output (stored internally until next calculation)
3. Menu `Export/Export objects using xtable`
4. Choose Export format HTML and set captions, labels etc, as desired
5. Select and copy the generated HTML code
6. Paste into MS Excel
7. Copy and paste the table into other applications as required

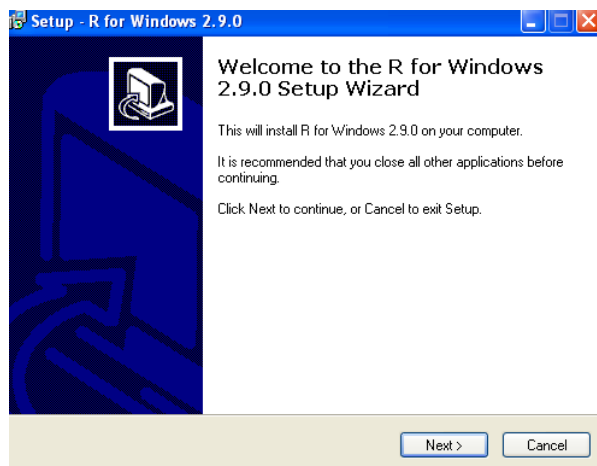
## A Installation

### A.1 Install R

Download the installer from the website:

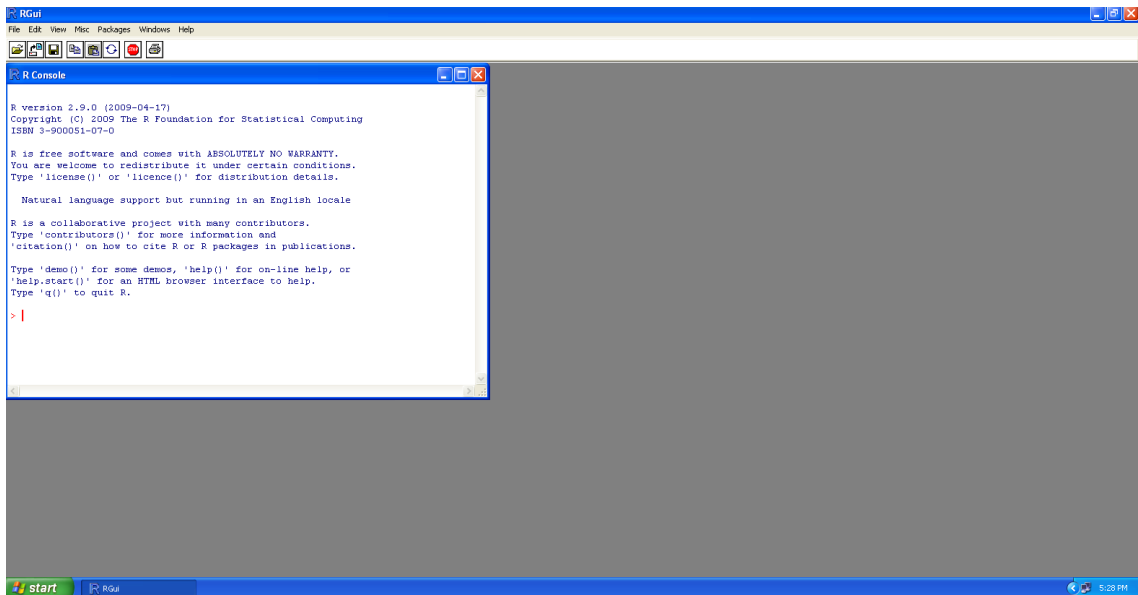


Run the installer and accept all defaults:



## A.2 Install Rcmdr

Start R:



Easiest from the menu **Packages/Install packages**. This will require the following steps:

1. Select a CRAN mirror – choose the geographically closest one to minimize network load (Sweden).
2. Select **Rcmdr** from the list of add-on packages and confirm with **Ok**. This will install the add-on package locally as part of the R installation.
3. Load **Rcmdr**: in order to make an add-on package and its data and commands available to the user, it has to be loaded first. This can be done via the menu **Packages/Load package**, or from the command line as `library(Rcmdr)`.
4. The first time you load **Rcmdr**, it will detect that many required add-on packages are not installed yet on your system. When it offers to install them collectively, confirm. *This step may take a while.*
5. There is no Step 5.

## Session Information

```
> toLatex(sessionInfo())
```

- R version 2.10.1 (2009-12-14), i386-apple-darwin8.11.1
- Locale: sv\_SE.UTF-8/sv\_SE.UTF-8/C/C/sv\_SE.UTF-8/sv\_SE.UTF-8
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils