

Introduction to R

Baburao Kamble

August, 2014

1 Introduction to R

R is a language and environment for statistical computing and graphics. It is a GNU (GNU GENERAL PUBLIC LICENSE) project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

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. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- an effective data handling and storage facility,

- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

R, like S, is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R dialect of S, which makes it easy for users to follow the algorithmic choices made. For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly.

Many users think of R as a statistics system. We prefer to think of it of an environment within which statistical techniques are implemented. R can be extended (easily) via *packages*. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

R has its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both on-line in a number of formats and in hardcopy.

This document contains explanations, examples and exercises for Data Analysis Using R Course at the University of Nebraska-Lincoln, which can also be understood (hopefully) by people without any programming experience.

2 Getting started with R

2.1 Install R

To install R on your computer (windows/mac/linux), go to the website of the Comprehensive R Archive Network (CRAN) <http://cran.r-project.org/>.

You can use precompiled binary distributions of the base system and contributed packages for **Windows/Linux/Mac** operating system.

Windows and Mac users most likely want to download the precompiled binaries

- For windows OS choose base R installation <http://cran.r-project.org/bin/windows/base/R-3.1.1-win.exe>
- For mac OS **R 3.1.1** binary for Mac OS X 10.6 (Snow Leopard) or Mac OS X 10.9 (Mavericks). <http://cran.r-project.org/bin/macosx/R-3.1.1-snowleopard.pkg>
<http://cran.r-project.org/bin/macosx/R-3.1.1-mavericks.pkg>
- For Linux OS install binaries depends on your Linux system (debian/redhat/Suse/ubuntu).

After finishing R installation, you should see "R" icon (32 and or 64 bit) on desktop. Click on icon to start the standard R interface.



In R Console, you can type simple commands after the ">" prompt and R will then execute your command. You can use console and calculator.

```
> 9+10
[1] 19
> 9*10
[1] 90
> 9/10
[1] 0.9
> 4^2
[1] 16
> pi
[1] 3.141593
> sin(30)
[1] -0.9880316
> cos(pi/2)
[1] 6.123032e-17
> sin(pi)
[1] 1.224606e-16
> log(30)
[1] 3.401197
> log10(2)
[1] 0.30103
```

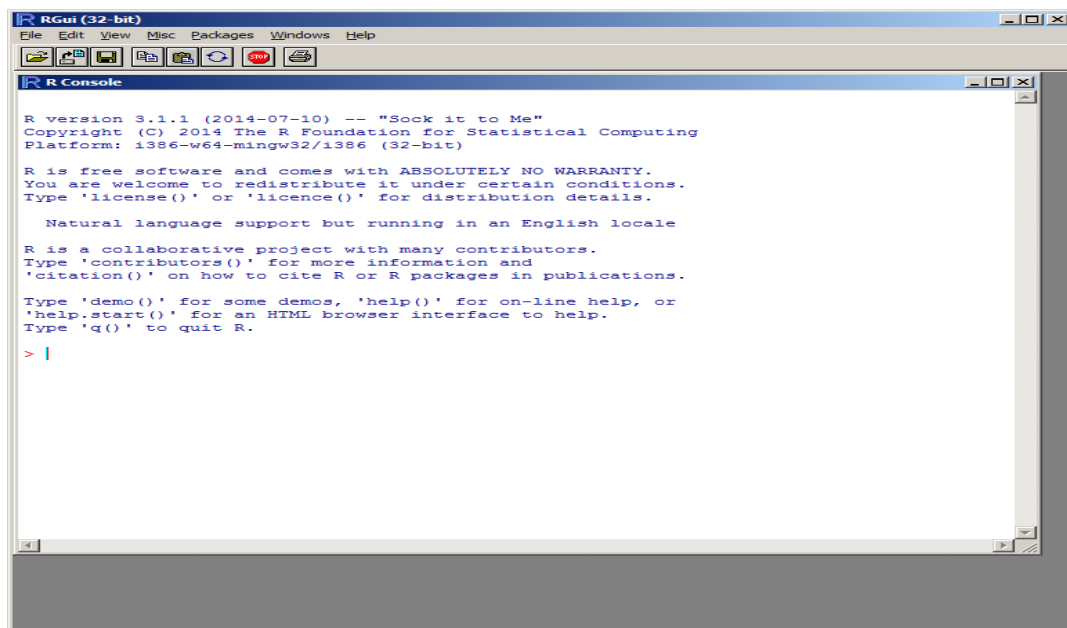


Figure 1 The Console Window of R.

2.2 Install RStudio

RStudio is integrated development environment (IDE), a powerful and productive user interface for R. It's free and open source, and works great on Windows, Mac, and Linux.

RStudio includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

To install RStudio, go to: <http://www.rstudio.org/>

- click Download RStudio
- click Download RStudio Desktop
- Go to Installers for ALL Platforms
- Download and install the specific installer for your Windows/mac/Linux

RStudio is the premier integrated development environment for R. It is available in open source and commercial editions and runs on the desktop

(Windows, Mac, and Linux) or over the web with RStudio Server and RStudio Server Pro.

2.3 RStudio layout

The RStudio interface consists of several windows (see Figure 1).

- Bottom left: **console window** (also called **command window**). Here you can type simple commands after the ">" prompt and R will then execute your command. This is the most important window, because this is where R actually does stuff.
- Top left: **editor window** (also called **script window**). Collections of commands (scripts) can be edited and saved. When you don't get

At the moment of writing 3.0.3 was the latest version.

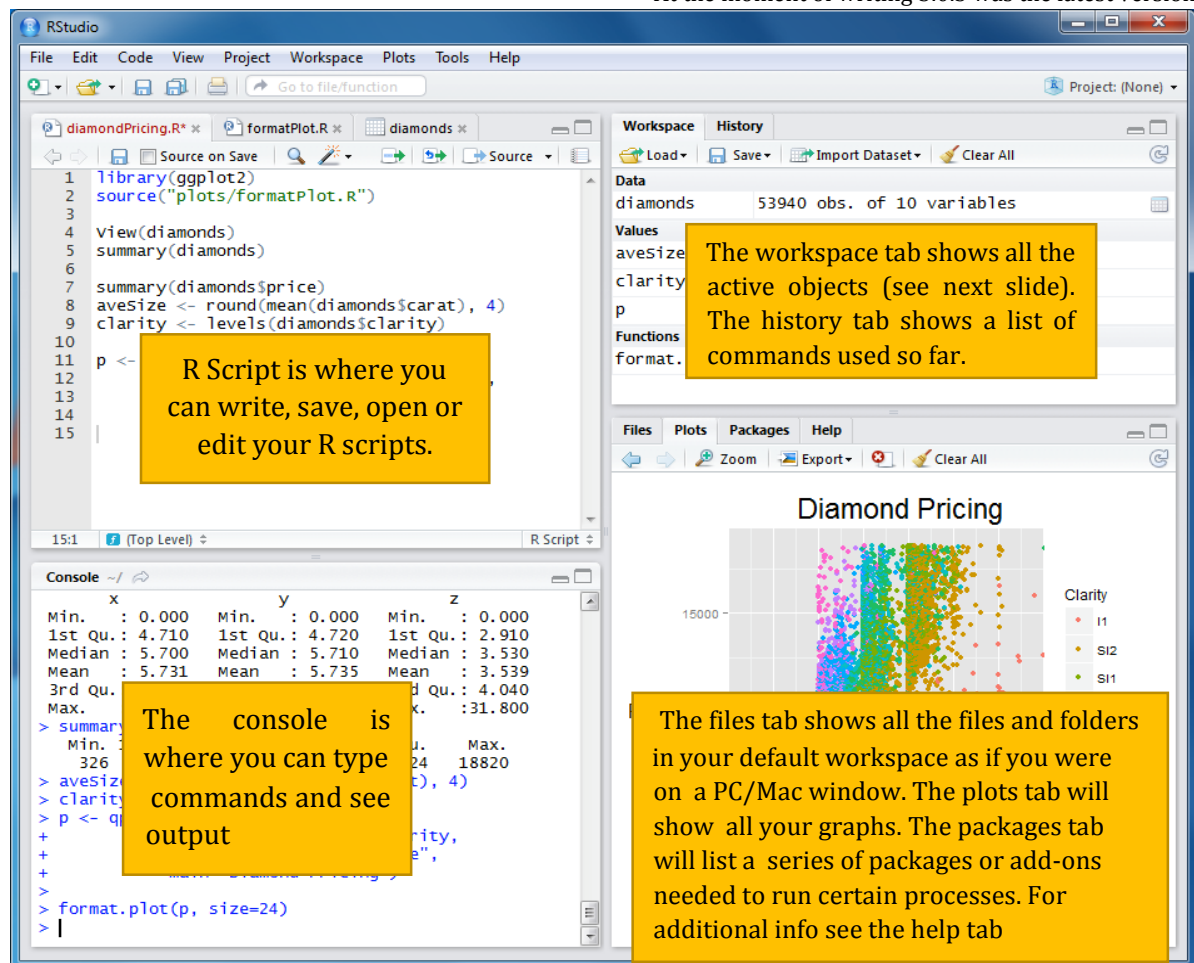


Figure 1 The editor, workspace, console and plots windows in RStudio

Choose the most recent one.

‡ There are many other (freeware) interfaces, such as TinnR.
this window, you can open it with File →

New→R script

Just typing a command in the editor window is not enough, it has to get into the command window before R executes the command. If you want to run a line from the script window (or the whole script), you can click Run or press CTRL+ENTER to send it to the command window.

- Top right: **workspace / history window**. In the workspace window you can see which data and values R has in its memory. You can view and edit the values by clicking on them. The history window shows what has been typed before.
- Bottom right: **files / plots / packages / help window**. Here you can open files, view plots (also previous plots), install and load packages or use the help function.

You can change the size of the windows by dragging the grey bars between the windows.

2.4 Working directory

Your *working directory* is the folder on your computer in which you are currently working. When you ask R to open a certain file, it will look in the working directory for this file, and when you tell R to save a data file or figure, it will save it in the working directory.

Before you start working, please set your working directory to where all your data and script files are or should be stored.

Type in the command window:
setwd("directoryname"). For example:

```
> setwd("M:/DataAnalysis/R/")
```

Make sure that the slashes are forward slashes and that you don't forget the apostrophes (for the reason of the apostrophes, see section 10.1). R is case sensitive, so make sure you write capitals where necessary.

Within RStudio you can also go to Tools / Set working directory.

2.5 Libraries

R can do many statistical and data analyses. They are organized in so-called *packages* or *libraries*. With the standard installation, most common packages are installed.

To get a list of all installed packages, go to the packages window or type library() in the console window. If the box in front of the package name is ticked, the package is loaded (activated) and can be used.

There are many more packages available on the R website. If you want to install and use a package (for example, the package called "geometry") you should:

- Install the package: click install packages in the packages window and type geometry or type install.packages("geometry") in the command window.
- Load the package: check box in front of geometry or type library("geometry") in the command window.

3 Some first examples of R commands

3.1 Calculator

R can be used as a calculator. You can just type your equation in the command window after the ">":

```
> 10^2 + 36
```

and R will give the answer

```
[1] 136
```

ToDo

Compute the difference between 2014 and the year you started at this university and divide this by the difference between 2014 and the year you were born. Multiply this with 100 to get the percentage of your life you have spent at this university. Use brackets if you need them.

If you use brackets and forget to add the closing bracket, the ">" on the command line changes into a "+". The "+" can also mean that R is still busy with some heavy computation. If you want R to quit

what it was doing and give back the ">", press ESC (see the reference list on the last page).

3.2 Workspace

You can also give numbers a name. By doing so, they become so-called variables which can be used later. For example, you can type in the command window:

```
> a = 4
```

You can see that a appears in the workspace window, which means that R now remembers what a is.[§] You can also ask R what a is (just type a ENTER in the command window):

```
> a [1] 4
```

or do calculations with a:

```
> a * 5 [1] 20
```

If you specify a again, it will forget what value it had before. You can also assign a new value to a using the old one.

```
> a = a + 10
```

```
> a [1] 14
```

To remove all variables from R's memory, type

```
> rm(list=ls())
```

or click "clear all" in the workspace window. You can see that RStudio then empties the workspace window. If you only want to remove the variable a, you can type rm(a).

ToDo

Repeat the previous ToDo, but with several steps in between. You can give the variables any name you want, but the name has to start with a letter.