

Introduction to R

LEARNING OUTCOME

At the end of this chapter, you will be able to:

- ▶ Install R
 - ▶ Install any R package
 - ▶ Work with any R package using functions such as `find.package()`, `install.packages()`, `library()`, `vignette()` and `packageDescription()`
-

1.1 INTRODUCTION

Statistical computing and high-scale data analysis tasks needed a new category of computer language besides the existing procedural and object-oriented programming languages, which would support these tasks instead of developing new software. There is plenty of data available today which can be analysed in different ways to provide a wide range of useful insights for multiple operations in various industries. Problems such as the lack of support, tools and techniques for varied data analysis have been solved with the introduction of one such language called R.

1.1.1 What is R?

R is a scripting or programming language which provides an environment for statistical computing, data science and graphics. It was inspired by, and is mostly compatible with, the statistical language S developed at Bell laboratory (formerly AT & T, now Lucent technologies). Although there are some very important differences between R and S, much

of the code written for S runs unaltered on R. R has become so popular that it is used as the single most important tool for computational statistics, visualisation and data science.

1.1.2 Why R?

R has opened tremendous scope for statistical computing and data analysis. It provides techniques for various statistical analyses like classical tests and classification, time-series analysis, clustering, linear and non-linear modelling and graphical operations. The techniques supported by R are highly extensible.

S is the pioneer of statistical computing; however, it is a proprietary solution and is not readily available to developers. In contrast, R is available freely under the GNU license. Hence, it helps the developer community in research and development.

Another reason behind the popularity and widespread use of R is its superior support for graphics. It can provide well-developed and high-quality plots from data analysis. The plots can contain mathematical formulae and symbols, if necessary, and users have full control over the selection and use of symbols in the graphics. Hence, other than robustness, user-experience and user-friendliness are two key aspects of R.

Why Learn R?

The following points describe why R language should be used (Figure 1.1):

- If you need to run statistical calculations in your application, learn and deploy R. It easily integrates with programming languages such as Java, C++, Python and Ruby.
- If you wish to perform a quick analysis for making sense of data.
- If you are working on an optimisation problem.
- If you need to use re-usable libraries to solve a complex problem, leverage the 2000+ free libraries provided by R.
- If you wish to create compelling charts.
- If you aspire to be a Data Scientist.
- If you want to have fun with statistics.

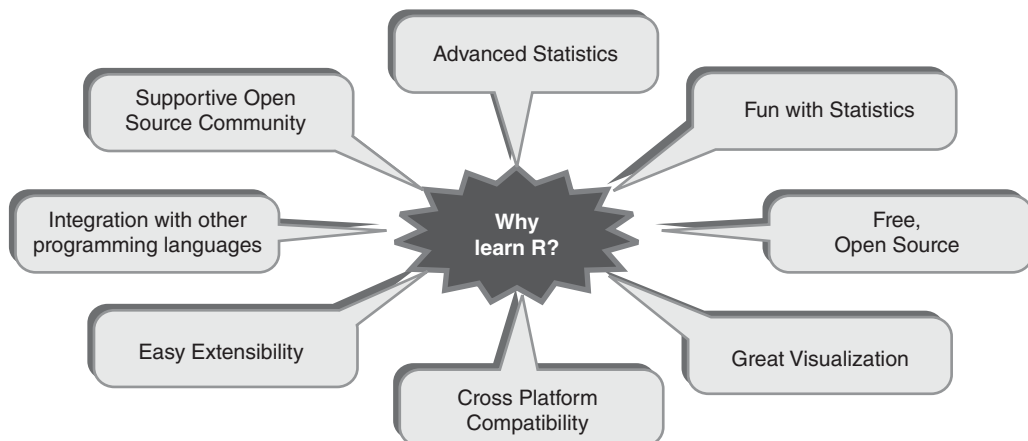


FIGURE 1.1 Advantages of learning R language

- R is free. It is available under the terms of the Free Software Foundation's GNU General Public License in source code form.
- It is available for Windows, Mac and a wide variety of Unix platforms (including FreeBSD, Linux, etc.).
- In addition to enabling statistical operations, it is a general programming language so that you can automate your analyses and create new functions.
- R has excellent tools for creating graphics such as bar charts, scatter plots, multi-panel lattice charts, etc.
- It has an object oriented and functional programming structure along with support from a robust and vibrant community.
- R has a flexible analysis tool kit, which makes it easy to access data in various formats, manipulate it (transform, merge, aggregate, etc.), and subject it to traditional and modern statistical models (such as regression, ANOVA, tree models, etc.)
- R can be extended easily via packages. It relates easily to other programming languages. Existing software as well as emerging software can be integrated with R packages to make them more productive.
- R can easily import data from MS Excel, MS Access, MySQL, SQLite, Oracle etc. It can easily connect to databases using ODBC (Open Database Connectivity Protocol) and ROracle package.

1.1.3 Advantages of R Over Other Programming Languages

Advanced programming languages like Python also support statistical computing and data visualisation along with traditional computer programming. However, R wins the race over Python and similar languages because of the following two advantages:

1. Python needs third party extensions and support for data visualisation and statistical computing. However, R does not require any such support extensively. For example, the `lm` function is present for linear regression analysis and data analysis in both Python and R. In R, data can be easily passed through the function and the function will return an object with detailed information about the regression. The function can also return information about the standard errors, coefficients, residual values and so on. When `lm` function is called in the Python environment, it will duplicate the functionalities using third party libraries such as SciPy, NumPy and so on. Hence, R can do the same thing with a single line of code instead of taking support from third party libraries.



SciPy is used for performing data analysis tasks and NumPy is used for representing the data or objects.

2. R has the fundamental data type, i.e., a vector that can be organised and aggregated in different ways even though the core is the same. Vector data type imposes some limitations on the language as this is a rigid type. However, it gives a strong logical base to R. Based on the vector data type, R uses the concept of data frames that are

like a matrix with attributes and internal data structure similar to spreadsheets or relational database. Hence, R follows a column-wise data structure based on the aggregation of vectors.



Just Remember

There are also some disadvantages of R. For example, R cannot scale efficiently for larger data sets. Hence, the use of R is limited to prototyping and sandboxing. It is rarely used for enterprise-level solutions. By default, R uses a single-thread execution approach while working on data stored in the RAM which leads to scalability issues as well. Developers from open source communities are working hard on these issues to make R capable of multi-threading execution and parallelisation. This will help R to utilise more than one core processor. There are big data extensions from companies like Revolution R and the issues are expected to be resolved soon. Other languages like SPlus can help to store objects permanently on disks, hence, supporting better memory management and analysis of high volume of massive datasets.

Check Your Understanding

1. What is R?

Ans: R is an open source programming language for data science and statistical computing.

2. What is the predecessor of R?

Ans: The statistical computing language, S is the predecessor of R.

3. What is the fundamental data type of R?

Ans: The fundamental data type of R is a vector.

4. What is the disadvantage of using R in enterprise-level large-scale solutions?

Ans: R language cannot scale up for large data sets. Hence, it is difficult to use R for large-scale data analysis tasks for enterprise-level solutions.

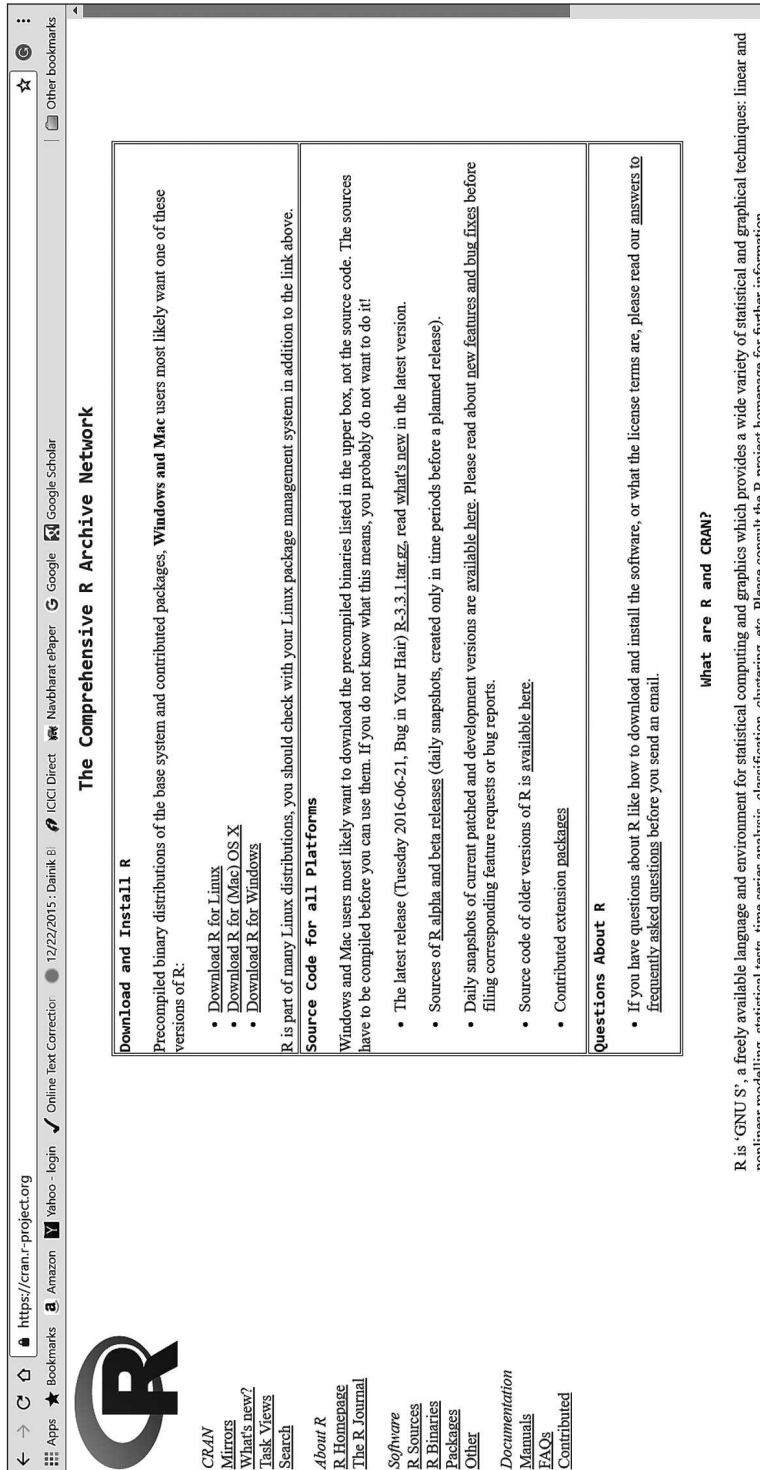
1.2 DOWNLOADING AND INSTALLING R

The integrated development suite for R language can be downloaded from the Comprehensive R Archive Network (CRAN)¹. The network includes mirror websites for downloading the suite from different countries.

1.2.1 Downloading R

To download R, users need to visit the CRAN mirror page and click on the URL of the chosen mirror that will redirect them to the respective site (Figure 1.2).

¹ URL of CRAN—<https://cran.r-project.org/mirrors.html>



The screenshot shows a web browser window displaying the CRAN website. The address bar shows the URL <https://cran.r-project.org>. The page title is "The Comprehensive R Archive Network". The main content area is titled "Download and Install R" and contains the following text and links:

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (Tuesday 2016-06-21, Bug in Your Hair) [R-3.3.1.tar.gz](#), read [what's new](#) in the latest version.
- Sources of R [alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features](#) and [bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- [Contributed extension packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

What are R and CRAN?

R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the [R project homepage](#) for further information.

FIGURE 1.2 CRAN website for downloading R

R is offered as a precompiled binary distribution of a base system and contributing packages. Different distributions of R are available for different operating systems (OS) like Windows, Mac and Linux.



In some Linux OS, R distributions are included by default. Hence, it is a good idea to check the package management system of a Linux OS platform before installing R on it.

Downloading R for Windows

Windows users need to first download and install binaries for the base distribution. The current version of the base binary distribution is R 3.3.1. Users can check and download previous contributions and versions of R, Rtools from the mirror website. Rtools is used for building R and its packages (Figure 1.3).

Downloading R for Mac

R works on Mac OS version 10.6 or more. The downloadable directory contains the base distribution and packages for downloading and installing R on Mac (Figure 1.4).

Downloading R for Linux

Different distributions of R are available for different distributions of Linux like Ubuntu, Debian, RedHat and SUSE (Figure 1.5). On the Command Line Interface (CLI), the following command will download the binary on a Linux machine—`$ wget http://cran.rstudio.com/src/base/R-3/R-3.1.1.tar.gz`

1.2.2 Installing R

After downloading R distribution binaries for the correct OS platform, R is installed.

Installing R on Windows

Installing R on Windows is simple. Users need to double click on the downloaded binary, named R-3.3.1-win.exe, on a graphical interface. Command line installation options are available for Windows (Figure 1.6).



Two versions are available for 32-bit and 64-bit Windows OS. By default, both the versions are installed. Hence, users need to select the desired version manually during installation.

[CRAN](#)
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

[About R](#)
[R Homepage](#)
[The R Journal](#)
[Software](#)
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Other](#)

[Documentation](#)
[Manuals](#)
[FAQs](#)
[Contributed](#)

[https://cran.r-project.org](#)

[Bookmarks](#)
[Amazon](#)
[Yahoo - login](#)
[Online Text Corrector](#)
[12/22/2015: Dainik B](#)
[ICICI Direct](#)
[Newbharat ePaper](#)
[Google](#)
[Google Scholar](#)

[Other bookmarks](#)

R for Windows

Subdirectories:

base
contrib
old contrib
Roots

Binaries for base distribution (managed by Duncan Murdoch). This is what you want to **install R for the first time**.
 Binaries of contributed CRAN packages (for R >= 2.11.x; managed by Uwe Ligges). There is also information on **third party software** available for CRAN Windows services and corresponding environment and make variables.
 Binaries of contributed CRAN packages for outdated versions of R (for R < 2.11.x; managed by Uwe Ligges).
 Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

<https://cran.r-project.org/bin/windows/>

FIGURE 1.3 Downloading R for Windows

The screenshot shows a web browser window displaying the CRAN website page for 'R for Mac OS X'. The browser's address bar shows the URL 'https://cran.r-project.org'. The page content includes a navigation menu on the left with links like 'CRAN', 'Mirrors', 'What's new?', 'Task Views', 'Search', 'About R', 'R Homepage', 'The R Journal', 'Software', 'R Sources', 'R Binaries', 'Packages', 'Other', 'Documentation', 'Manuals', 'FAQs', and 'Contributed'. The main content area features a large 'R' logo and a heading 'R for Mac OS X'. Below the heading, there is a paragraph explaining that the directory contains binaries for a base distribution and packages to run on Mac OS X (release 10.6 and above). It mentions Mac OS 8.6 to 9.2 (and Mac OS X 10.1) are no longer supported but can be found in the 'old' directory. A note states that CRAN does not have Mac OS X systems and cannot check these binaries for viruses. A section titled 'R 3.3.1 "Bug in Your Hair" released on 2016/06/21' provides instructions to check the MD5 checksum of the downloaded image and to use the 'md5' command in the Terminal to verify the signature. A 'Files' section lists two download links: 'R-3.3.1.pkg' (ca. 71MB) and 'R-3.2.1-snow/leopard.pkg' (ca. 68MB). The page also includes a note about XQuartz requirements for OS X 10.6 and higher.

https://cran.r-project.org

Other bookmarks

Google Scholar

Google

Navharat ePaper

ICICI Direct

12/22/2015, Dainik B

Online Text Connector

Yahoo - login

Amazon

R for Mac OS X

This directory contains binaries for a base distribution and packages to run on Mac OS X (release 10.6 and above). Mac OS 8.6 to 9.2 (and Mac OS X 10.1) are no longer supported but you can find the last supported release of R for these systems (which is R 1.7.1) [here](#). Releases for old Mac OS X systems (through Mac OS X 10.5) and PowerPC Macs can be found in the [old](#) directory.

Note: CRAN does not have Mac OS X systems and cannot check these binaries for viruses. Although we take precautions when assembling binaries, please use the normal precautions with downloaded executables.

As of 2016/03/01 package binaries for R versions older than 2.12.0 are only available from the [CRAN archive](#) so users of such versions should adjust the CRAN mirror setting accordingly.

R 3.3.1 "Bug in Your Hair" released on 2016/06/21

Please check the MD5 checksum of the downloaded image to ensure that it has not been tampered with or corrupted during the mirroring process. For example

```
type
md5 R-3.3.1.pkg
pkgutil --check-signature R-3.3.1.pkg
```

In the *Terminal* application to print the MD5 checksum for the R-3.3.1.pkg image. On Mac OS X 10.7 and later you can also validate the signature using

Files:

R 3.3.1 binary for Mac OS X 10.9 (Mavericks) and higher, signed package. Contains R 3.3.1 framework, R.app GUI 1.68 in 64-bit for Intel Macs, Tcl/Tk 8.6.0 X11 libraries and Texinfo 5.2. The latter two components are optional and can be omitted when choosing "custom install", it is only needed if you want to use the `tcltk` R package or build package documentation from sources.

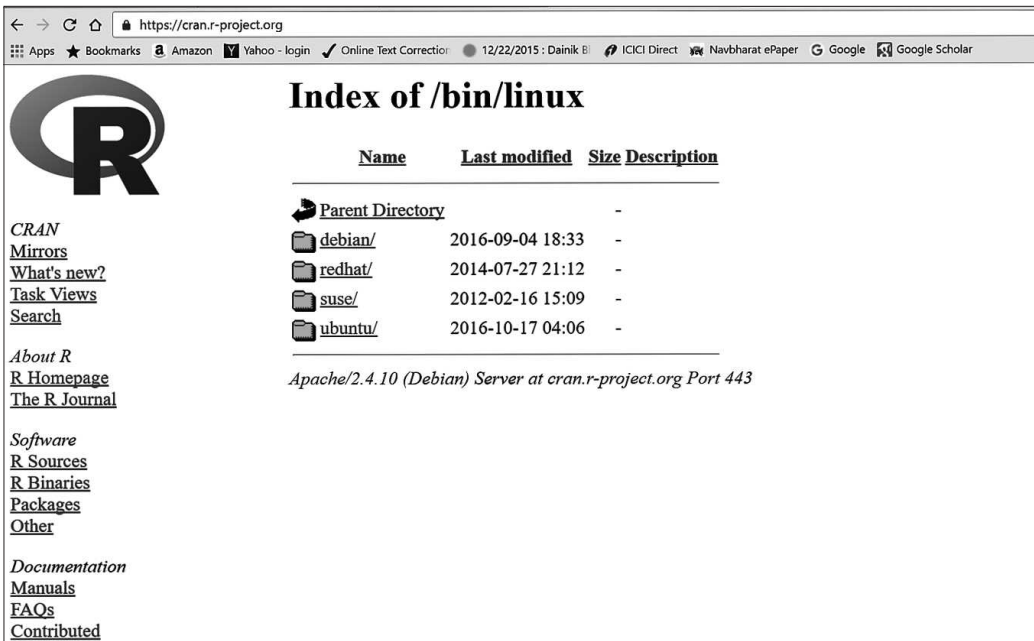
Note: the use of X11 (including `tcltk`) requires `XQuartz` to be installed since it is no longer part of OS X. Always re-install `XQuartz` when upgrading your OS X to a new major version.

R 3.2.1 legacy binary for Mac OS X 10.6 (Snow Leopard) - 10.8 (Mountain Lion), signed package. Contains R 3.2.1 framework, R.app GUI 1.66 in 64-bit for Intel Macs. This package contains the R framework, 64-bit GUI (R.app), Tcl/Tk 8.6.0 X11 libraries and Texinfo 5.2. GNU Fortran is **NOT** included (needed if you want to compile packages from sources that contain FORTRAN code) please see the [tools directory](#).

R-3.3.1.pkg
MD5-hash: 586a00131d0b57578bcb0b14d665
SHA1-hash: 1281d22616926162ab494c0f6edf1b96d19
(ca. 71MB)

R-3.2.1-snow/leopard.pkg
MD5-hash: 586a00131d0b57578bcb0b14d665
SHA1-hash: b0e6910b72ee232410d051174e0a06e0e040
(ca. 68MB)

FIGURE 1.4 Downloading R on Mac



CRAN

[Mirrors](#)

[What's new?](#)

[Task Views](#)

[Search](#)

[About R](#)

[R Homepage](#)

[The R Journal](#)

[Software](#)

[R Sources](#)

[R Binaries](#)

[Packages](#)

[Other](#)

[Documentation](#)

[Manuals](#)

[FAQs](#)

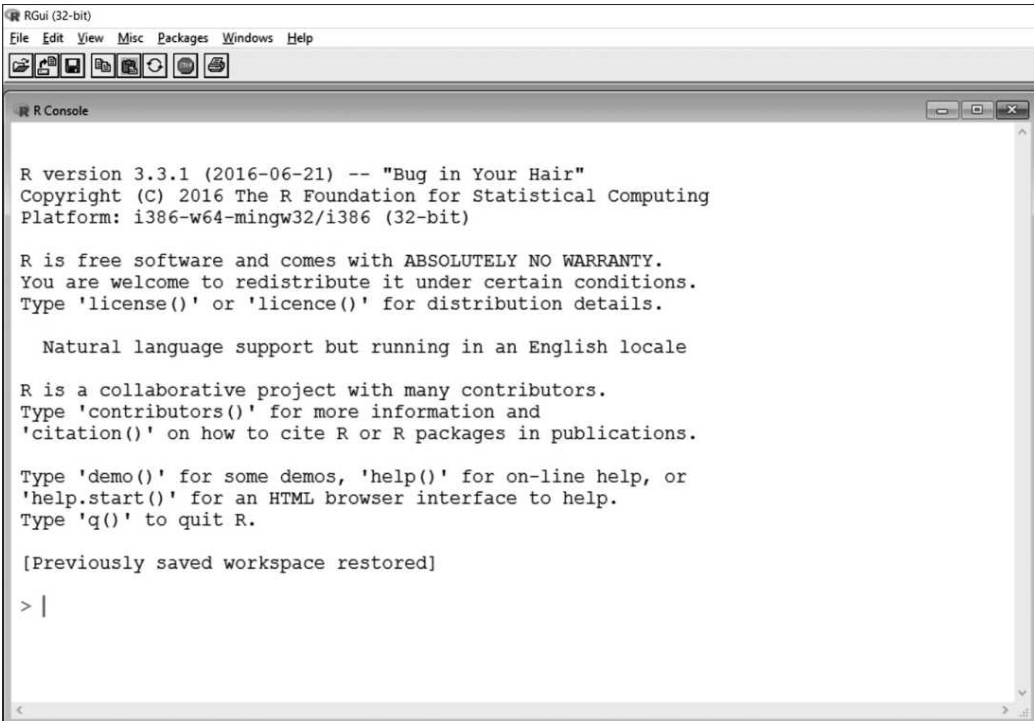
[Contributed](#)

Index of /bin/linux

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
Parent Directory		-	
debian/	2016-09-04 18:33	-	
redhat/	2014-07-27 21:12	-	
suse/	2012-02-16 15:09	-	
ubuntu/	2016-10-17 04:06	-	

Apache/2.4.10 (Debian) Server at cran.r-project.org Port 443

FIGURE 1.5 Downloading R for Linux distributions



```

RGui (32-bit)
File Edit View Misc Packages Windows Help
[Icons]

R Console

R version 3.3.1 (2016-06-21) -- "Bug in Your Hair"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-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.

Natural language support but running in an English locale

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.

[Previously saved workspace restored]

> |

```

FIGURE 1.6 R console on a 32-bit Windows PC

Installing Rtools

Rtools is an additional requirement for developing R packages under Windows OS environment. In addition to installing the R software on Windows, users need to install Rtools for the installed version of R.

Installing R on Mac

The process for installing R on Mac is similar to that for Windows. Users need to double click on the binaries downloaded from the CRAN website and follow the prompts.

Installing R on Linux

Users need to install R from the source on Linux distributions. This can be done by following commands in the supervisor mode. The following steps will install and configure R into a user-specific subdirectory within the home directory:

```
$ tar xvf R-3.1.1.tar.gz
$ cd R-3.1.1
$ ./configure --prefix=$HOME/R
$ make && make install
```



Setting the path on a Linux machine is very critical. Without the path, R and RScript do not work.

1.2.3 Primary File Types of R

Working with R involves working on two types of files—RScripts and R markdown documents.

RScript

RScript is a text file that contains commands for an R program. The same commands can be executed individually on the CLI of Integrated Development Environment (IDE) for R programming. An RScript can be also be developed and executed. However, there is a difference between executing a command directly on CLI and executing the same command through an R script. An RScript has a .R extension.

Command line interface is needed for quick and small data processing and checking operations. In large-scale solutions, it integrates multiple programs during prototyping and subsequent phases. In that case, RScripts are used for managing the integration process.

Markdown Documents

R markdown documents are produced for creating and authoring dynamic documents, reports and presentations from R. R markdown documents have a set of markdown

syntaxes derived from the core markdown syntaxes. These syntaxes are embedded into RScripts and codes. When these embedded codes and scripts are executed then the output is formatted based on the markdown syntaxes and hence becomes easily understandable. R markdown documents can be regenerated automatically if the underlying RScripts and codes or data are changed. The output format of an R markdown covers a wide range of formats including PDF, HTML, HTML5 slides, websites, dashboards, tufte handouts, notebooks, books, MS word, etc. The extension for R markdown document files is `.rmd`.

Check Your Understanding

1. How to locate an RScript file in a typical file system?

Ans: An RScript file can be located in a typical file system by verifying if the extension of the file is `.R`.

2. What is R markdown and how is it different from word documentation?

Ans: R markdown documents are dynamic and reproducible. Markdown files are used for making reports and documents with R. These markdown codes are embedded into files such as PDF, HTML, word files, etc. On the contrary, word files are text files only and do not support markdown.

1.3 IDEs AND TEXT EDITORS

Various text editors can be used for writing RScripts and codes. Table 1.1 describes some popular IDEs and text editors for writing and executing R codes.

TABLE 1.1 Some IDEs and text editors for writing and executing R codes

<i>Name</i>	<i>Platform(s)</i>	<i>License</i>	<i>Details and Usage</i>
Notepad and Notepad++ to R	Windows, Linux and Mac	GNU GPL	Notepad++ to R is an editor for R that is simple and robust. It supports extensions like close passing to Notepad++ editor, R GUI editor and optionally to a PuTTY window on a remote machine. It supports batch processing using shortcuts, monitoring of execution of RScripts and so on.
Tinn-R	Windows	GNU GPL	Tinn-R is a word processor and text editor that can process generic ASCII and UNICODE on Windows OS. This is well integrated into R and supports GUI and IDE for R.
Revolution Productivity Enhancer (RPE)		Commercial	Revolution productivity enhancer is an R productivity or enhanced environment. However, it can work as an IDE for new users. The usability features of RPE are very supportive. It includes features like IntelliSense for detecting completion of word, code snippets, and so on. Hence, RPE is an integrated IDE and editor with built-in visual debugging tools.

There are various IDEs used in R language. You will learn about these IDEs in the following section.

1.3.1 R Studio

R studio is the most widely used IDE for writing, testing and executing R codes (Figure 1.7). This is a user-friendly and open source solution. There are various parts in a typical screen of an R studio IDE. These are:

- Console, where users write a command and see the output
- Workspace tab, where users can see active objects from the code written in the console
- History tab, which shows a history of commands used in the code
- File tab, where folders and files can be seen in the default workspace
- Plot tab, which shows graphs
- Packages tab, which shows add-ons and packages required for running specific process(s)
- Help tab, which contains the information on IDE, commands, etc.

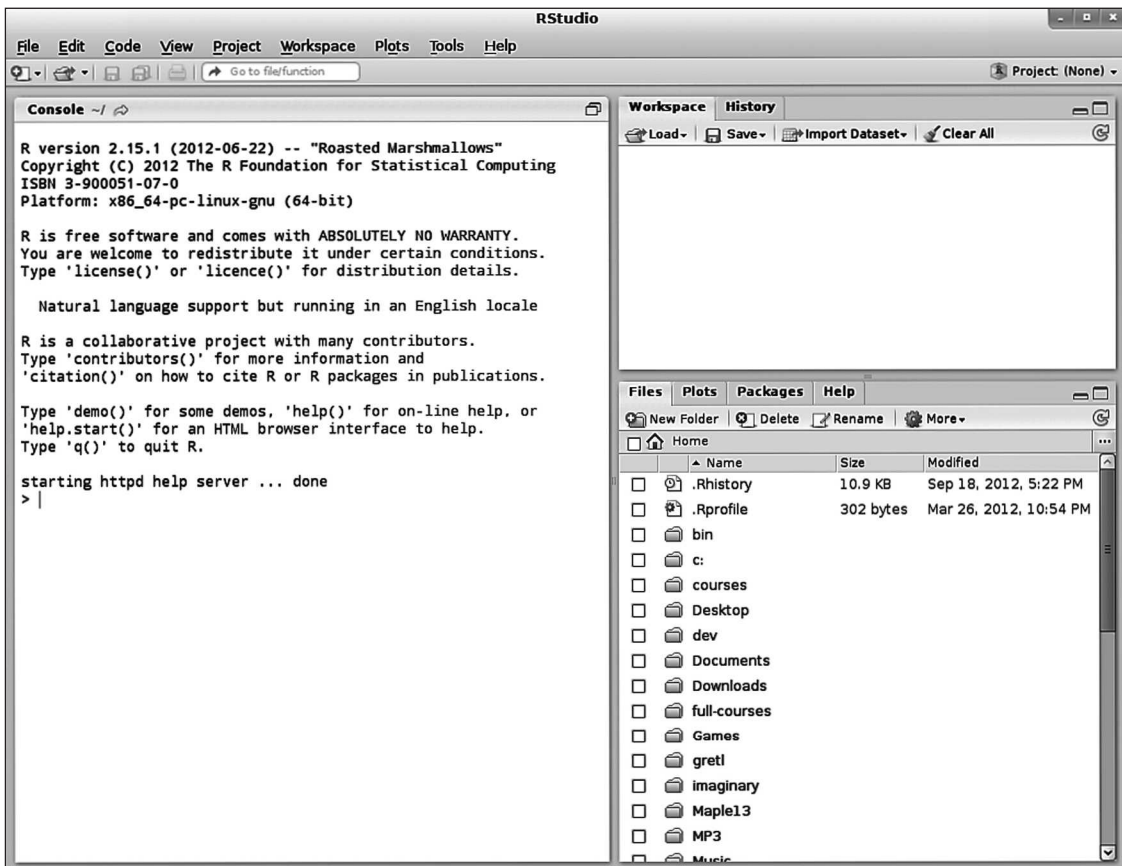


FIGURE 1.7 R Studio Interface

1.3.2 Eclipse with StatET

Eclipse is a well-known IDE for Java, C++, etc.; however, Eclipse can be used for statistical programming based on R also. The corresponding IDE is called Eclipse with StatET. Eclipse with StatET offers a set of tools that can be used for coding in R and building R packages. It supports one or more local and remote installations of R. Its functionalities can be expanded by using more add-ons like Sweave and Wiktetext. Different parts of the IDE are given below:

- Console for R
- Object browser
- Package manager
- Debugger
- Data viewer
- R help system.

1.4 HANDLING PACKAGES IN R

A package in R is the fundamental unit of shareable code. It is a collection of the following elements:

- Functions
- Data sets
- Compiled code
- Documentation for the package and for the functions inside
- Tests – few tests to check if everything works as it should.

The directory where packages are stored is called a library. R comes with a standard set of packages. Others are available for download and installation as per requirement. As on date, there are over 10,000 plus packages available in CRAN. This is also one of the reasons behind the huge popularity and success of R.

Packages are used to share codes with others. One can develop their own R package. Any R user can then download, install and learn to use the package. Packages, therefore allow for an easy, transparent and cross-platform extension of the R base system.

R is an open source language; thus, new packages are being developed and updated by developers daily. Some of these packages may not work properly or may have bugs. Hence, it is not a good idea to use every new and updated package on R development environment. This can affect the stability of the development environment. A stable environment requires the *sandboxing technique* (a security mechanism often used to execute untested or untrusted programs or code from unverified or untrusted third parties, users, etc., without damaging/maligning the host machine or operating system or production environment) to test new packages or update a package before installing it in the development environment.

In general, there is a single package library with each installation of R on a computer. Users can change the path to that library to install a package on a different location other than the default package library. The command `.libPaths()` can be used to get or set the path of the package library.

Example

```
> .libPaths()
```

Output

```
C:/R/R-3.1.3/library
```

This is the default package library location. The following command will change it into another path:

Example

```
> .libPaths("~/R/win-library/3.1-mran-2016-07-02")
```

Output

```
C:/Users/User1/Documents/R/win-library/3.1-mran-2016-07-02
```

R can be extended easily with the help of a rich set of packages. There are more than 10,000 packages available for R. These packages are used for different purposes. Tables 1.2 and 1.3 list some commonly used R packages for different purposes.

TABLE 1.2 Commonly used R packages for different purposes

<i>Data Management</i>	<i>Data Visualisation</i>	<i>Data Products</i>	<i>Data Modelling and Simulation</i>
dplyr, tidyr, foreign, haven etc.	ggplot, ggvis, lattice, igraph etc.	shiny, slidify, knitr, markdown etc.	MASS, forecast, bootstrap, broom, nlme, ROCR, party etc.

TABLE 1.3 Commonly used packages in R

<i>Author(s)</i>	<i>Package Name</i>	<i>Description</i>	<i>Available At</i>
Andrew Gelman, et al.	arm	It is used for hierarchical or multi-level regression models.	http://cran.r-project.org/web/packages/arm/
Douglas Bates, Martin Maechler, and Ben Bolker	lme4	It contains functions for generating generalised and linear mixed-effects models.	http://cran.r-project.org/web/packages/lme4/
Duncan Temple Lang	Rcurl	It provides an interface of R to the package library, libcurl. The interface helps in interacting with the HTTP protocols for importing raw data from the web.	http://www.omegahat.org/Rcurl/
Duncan Temple Lang	RJSONIO	It provides a set of functions to read and write JSON for analysing data from different web-based APIs.	http://www.omegahat.org/RJSONIO/
Duncan Temple Lang	XML	It provides functions and facilities for analysing HTML and XML documents to extract structured data from web-based sources.	http://www.omegahat.org/RXML/

(Continued)

<i>Author(s)</i>	<i>Package Name</i>	<i>Description</i>	<i>Available At</i>
Gabor Csardi	igraph	It contains routines for network analysis and making simple graphs to represent social networks.	http://igraph.sourceforge.net/
Hadley Wickham	ggplot	It contains a set of grammar rules for implementing graphics in R. The package is used for creating high-quality graphics.	http://cran.r-project.org/web/packages/glmnet/index.html
Hadley Wickham	lubridate	The package provides functions to use dates in R in an easier way.	https://github.com/hadley/lubridate
Hadley Wickham	reshape	It contains a set of tools for manipulation, aggregation and management of data in R.	http://had.co.nz/plyr/
Ingo Feinerer	tm	It contains functions to perform text mining in R. Text mining helps to work with unstructured data.	http://www.spatstat.org/spatstat/
Jerome Friedman, Trevor Hastie, and Rob Tibshirani	glmnet	It helps to work with the elastic-net and also regularised and generalised linear models.	http://had.co.nz/ggplot2/

1.4.1 Installing an R Package

R comes with some standard packages that are installed when a user first installs R and additional packages can be installed separately. Users need to navigate through the package library and install a package in the desired location. Following commands are used for navigating through R package library and installing R package.

1. To start R, follow either Step 2 or 3. The assumption is that R is already installed on your machine.
2. If there is an “R” icon on the desktop of the computer that you are using, double click on the “R” icon to start R. If there is no “R” icon on the desktop then click on the “Start” button at the bottom left of your computer screen, and then choose “All programs”, and start R by selecting “R” (or R X.X.X, where X.X.X gives the version of R, e.g. R 2.10.0) from the menu of programs.
3. The R console should show up.
4. Once you have started R, you can install an R package (e.g. the “ggplot2” package) by choosing “Install package(s)” from the “Packages” menu at the top of the R console. This will ask you for the website that you wish to download the package from. You can choose “Iceland” (or another country, if you prefer). It will also bring up a list of available packages that you can install, and you can choose the package that you want to install from that list (e.g. “ggplot2”).
5. This will install the “ggplot2” package.
6. The “ggplot2” package is now installed. Whenever you want to use the “ggplot2” package after this, after having successfully started R, you first have to load the package by typing into the R console: `library(“ggplot2”)`.
7. You can get help on a package by typing the following at the R prompt: `help(package = “ggplot2”)`

1.4.2 Few Commands to Get Started

installed.packages()

A user can check for all installed packages on the machine by using the `installed.packages()` function.

```
> installed.packages()
  Package      LibPath      Version      Priority
arules        "arules"      "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.3-1"    NA
arulesViz     "arulesViz"   "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.1-0"    NA
assertthat    "assertthat"  "C:/Users/seema_acharya/Documents/R/win-library/3.2" "0.1"      NA
BH            "BH"          "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.62.0-1" NA
bit           "bit"         "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.1-12"   NA
bitops        "bitops"      "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.0-6"    NA
boot          "boot"        "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.3-18"   "recommended"
caTools       "caTools"     "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.17.1"   NA
class         "class"       "C:/Users/seema_acharya/Documents/R/win-library/3.2" "7.3-14"   "recommended"
coin          "coin"        "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.1-2"    NA
colorspace    "colorspace"  "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.2-6"    NA
corpcor       "corpcor"     "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.6.8"    NA
curl          "curl"        "C:/Users/seema_acharya/Documents/R/win-library/3.2" "2.3"      NA
DBI           "DBI"         "C:/Users/seema_acharya/Documents/R/win-library/3.2" "0.5-1"    NA
dendextend    "dendextend"  "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.1.8"    NA
DEoptimR      "DEoptimR"    "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.0-8"    NA
devtools      "devtools"    "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.12.0"   NA
dichromat     "dichromat"   "C:/Users/seema_acharya/Documents/R/win-library/3.2" "2.0-0"    NA
digest        "digest"      "C:/Users/seema_acharya/Documents/R/win-library/3.2" "0.6.9"    NA
doParallel    "doParallel"  "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.0.10"   NA
dplyr         "dplyr"       "C:/Users/seema_acharya/Documents/R/win-library/3.2" "0.5.0"    NA
ff            "ff"          "C:/Users/seema_acharya/Documents/R/win-library/3.2" "2.2-13"   NA
foreach       "foreach"     "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.4.3"    NA
foreign       "foreign"     "C:/Users/seema_acharya/Documents/R/win-library/3.2" "0.8-66"   "recommended"
FRB           "FRB"         "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.8"      NA
gclus         "gclus"       "C:/Users/seema_acharya/Documents/R/win-library/3.2" "1.3.1"    NA
gdata         "gdata"       "C:/Users/seema_acharya/Documents/R/win-library/3.2" "2.17.0"   NA
ggdendro      "ggdendro"    "C:/Users/seema_acharya/Documents/R/win-library/3.2" "0.1-20"   NA
ggfortify     "ggfortify"   "C:/Users/seema_acharya/Documents/R/win-library/3.2" "0.4.1"    NA
ggplot2       "ggplot2"     "C:/Users/seema_acharya/Documents/R/win-library/3.2" "2.0.0"    NA
git2r         "git2r"       "C:/Users/seema_acharya/Documents/R/win-library/3.2" "0.18.0"   NA
gplots        "gplots"      "C:/Users/seema_acharya/Documents/R/win-library/3.2" "2.17.0"   NA
gridBase      "gridBase"    "C:/Users/seema_acharya/Documents/R/win-library/3.2" "0.4-7"    NA
gridExtra     "gridExtra"   "C:/Users/seema_acharya/Documents/R/win-library/3.2" "2.2.1"    NA
gtable        "gtable"      "C:/Users/seema_acharya/Documents/R/win-library/3.2" "0.2.0"    NA
gtools        "gtools"      "C:/Users/seema_acharya/Documents/R/win-library/3.2" "3.5.0"    NA
```

....

`remove.packages()` can be used to uninstall a package.

packageDescription()

“DESCRIPTION” file has the basic information about a package. It has details such as what the package does, who is the author, what is the version for the documentation, the date, the type of license its use, and the package dependencies, etc. To access the description file inside R, use the function, `packageDescription("package")`. The same can also be accessed via the documentation of the package by using `help(package = "package")`.

Let us look at the description for the “stats” package.

```
> packageDescription("stats")
Package: stats
Version: 3.2.3
Priority: base
Title: The R Stats Package
Author: R Core Team and contributors worldwide
Maintainer: R Core Team <R-core@r-project.org>
Description: R statistical functions.
License: Part of R 3.2.3
Suggests: MASS, Matrix, Suppdists, methods, stats4
Build: R 3.2.3; x86_64-w64-mingw32; 2015-12-10 13:03:29 UTC; windows

-- File: C:/Program Files/R/R-3.2.3/library/stats/Meta/package.rds
```

Or

```
> help(package="stats")
```

The output shown is partial.



help(package = "package")

To get an overview of all the functions and datasets in an R package, use the `help()` function.

```
> help(package = "datasets")
```

The above will provide an overview of all functions and datasets inside the package, "datasets". One of the dataset available in "datasets" package is "AirPassengers". To

access the dataset, “AirPassengers” inside the “datasets” package, use the code given below:

```
> datasets::AirPassengers
      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949 112 118 132 129 121 135 148 148 136 119 104 118
1950 115 126 141 135 125 149 170 170 158 133 114 140
1951 145 150 178 163 172 178 199 199 184 162 146 166
1952 171 180 193 181 183 218 230 242 209 191 172 194
1953 196 196 236 235 229 243 264 272 237 211 180 201
1954 204 188 235 227 234 264 302 293 259 229 203 229
1955 242 233 267 269 270 315 364 347 312 274 237 278
1956 284 277 317 313 318 374 413 405 355 306 271 306
1957 315 301 356 348 355 422 465 467 404 347 305 336
1958 340 318 362 348 363 435 491 505 404 359 310 337
1959 360 342 406 396 420 472 548 559 463 407 362 405
1960 417 391 419 461 472 535 622 606 508 461 390 432
```

If there will be frequent use of this package, it is worthwhile to load it into the memory. This can be achieved using the library function:

```
> library(datasets)
```

Note: the package name has to be specified without enclosing it in quotes. The `library()` function will load the package, “datasets” into the memory. Then any dataset within this package can be accessed by simply typing the name of the dataset at the R prompt.

```
> library(datasets)
> AirPassengers
      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949 112 118 132 129 121 135 148 148 136 119 104 118
1950 115 126 141 135 125 149 170 170 158 133 114 140
1951 145 150 178 163 172 178 199 199 184 162 146 166
1952 171 180 193 181 183 218 230 242 209 191 172 194
1953 196 196 236 235 229 243 264 272 237 211 180 201
1954 204 188 235 227 234 264 302 293 259 229 203 229
1955 242 233 267 269 270 315 364 347 312 274 237 278
1956 284 277 317 313 318 374 413 405 355 306 271 306
1957 315 301 356 348 355 422 465 467 404 347 305 336
1958 340 318 362 348 363 435 491 505 404 359 310 337
1959 360 342 406 396 420 472 548 559 463 407 362 405
1960 417 391 419 461 472 535 622 606 508 461 390 432
```

***find.package()* and *install.packages()* Command**

`find.package()` and `install.packages()` commands will find and install specific R package(s). There are two versions of this command. The first helps in installing one package at a time and the other is used to install multiple packages at once using a single command—`install.packages()`. More details on commands like `find.package()` and `install.packages()` can be retrieved using the `help()` command. For example, `help(installed.packages)` can show details like the version number of a function.

Example

To install a single package, the command is:

```
>find.package("ggplot2")
>install.packages("ggplot2")
```

Output

The first command will help to find if there is any package named “ggplot2” installed in the system or not. Then the `install.packages()` function will install the package named “ggplot2” CLI (Figure 1.8). It will download and install the package and all the dependencies of the package.

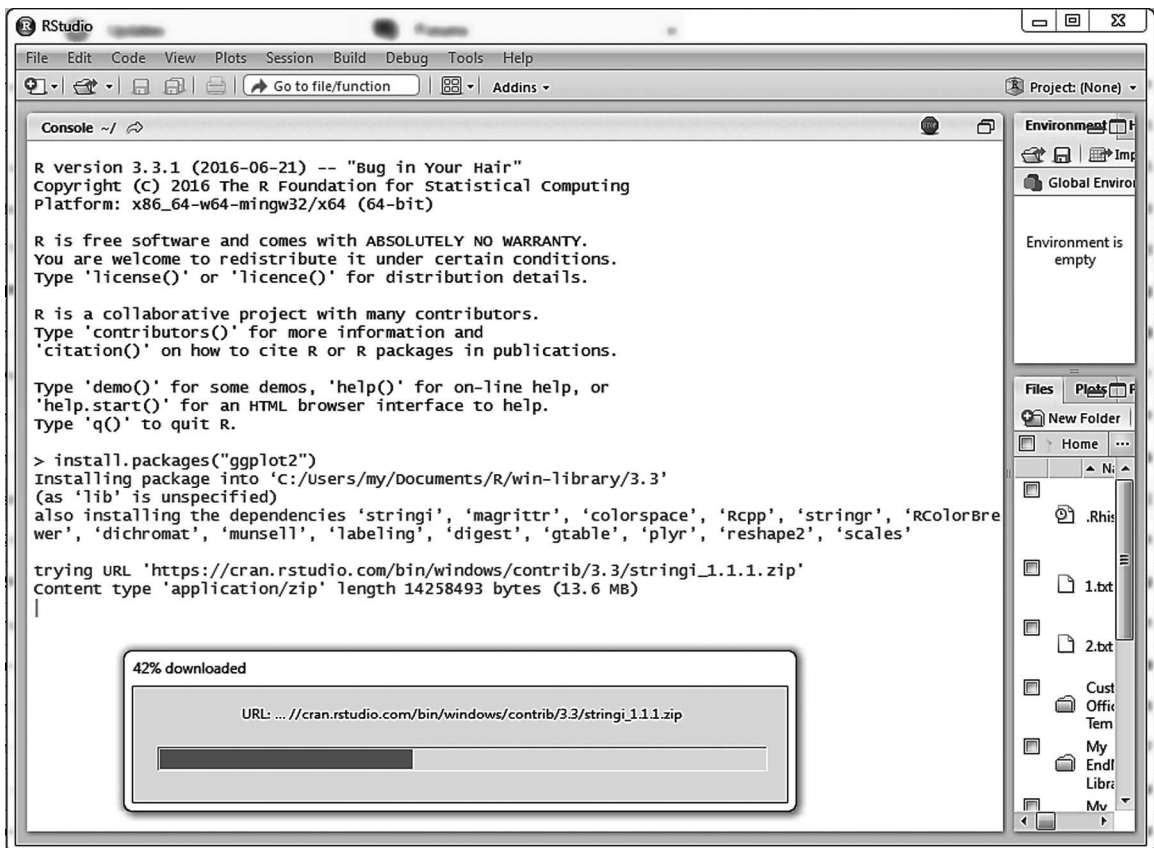


FIGURE 1.8 Example of installing a package

Example

To install more than one package(s) at a time, the `install.packages()` command will have the following format:

```
>install.packages(c("ggplot", "tidyr", "dplyr"))
```

Output

It will install packages `ggplot`, `tidyr` and `dplyr`.



The command to check whether a package is installed or not is the ‘if’ condition checking. The command for checking whether the package “`ggplot2`” is installed or not can be done by using:

```
>if (!require("ggplot2")){install.packages("ggplot2")}
```

`library()`

`library()` command loads a package.

Example

```
>library(ggplot2)
```

Output

It will load the package “`ggplot2`”.

`vignette()`

Vignettes are a very useful source of help with packages. They are provided by the package authors to demonstrate and highlight few functionalities of their package in detail. Use `browseVignettes()` function to get a list of all vignettes available with your installed packages.

```
> browseVignettes()
```

Vignettes found by “`browseVignettes()`”

Vignettes in package `arules`

- Introduction to `arules` - [PDF](#) [source](#) [R code](#)

Vignettes in package `arulesViz`

- Visualizing Association Rules: Introduction to `arulesViz` - [PDF](#) [source](#) [R code](#)

Vignettes in package `coin`

- A Lego System for Conditional Inference - [PDF](#) [source](#) [R code](#)
- `coin`: A Computational Framework for Conditional Inference - [PDF](#) [source](#) [R code](#)
- Implementing a Class of Permutation Tests: The `coin` Package - [PDF](#) [source](#) [R code](#)
- Order-restricted Scores Test - [PDF](#) [source](#) [R code](#)

Vignettes in package `colorspace`

- HCL-Based Color Palettes in R - [PDF](#) [source](#) [R code](#)

Vignettes in package `curl`

- The `curl` package: a modern R interface to `libcurl` - [HTML](#) [source](#) [R code](#)

Vignettes in package `DBI`

- A Common Database Interface (DBI) - [HTML](#) [source](#)
- A Common Interface to Relational Databases from R and S -- A Proposal - [HTML](#) [source](#)
- Implementing a new backend - [HTML](#) [source](#) [R code](#)

To view all vignettes for a specific package, e.g., “ggplot2”, use the `vignette()` function.

Vignettes in package ‘ggplot2’:

```
ggplot2-specs           Aesthetic specifications (source, html)
extending-ggplot2      Extending ggplot2 (source, html)
```

Check Your Understanding

1. Name a few packages used for data management in R.

Ans: `dplyr`, `tidyr`, `foreign`, `haven`, etc.

2. Name a few packages used for data visualisation in R.

Ans: `ggplot`, `ggvis`, `lattice`, `igraph`, etc.

3. Name a few packages used for developing data produces in R.

Ans: `shiny`, `slidify`, `knitr`, `markdown`, etc.

4. Name a few packages used for data modelling and simulation in R.

Ans: `MASS`, `forecast`, `bootstrap`, `broom`, `nlme`, `ROCR`, `party`, etc.

5. How can the default path to package library be changed in R?

Ans: To change the default package library in R, users need to follow the following steps on the console of R IDE:

Step 1: Check the current path to the package library

```
> .libPaths()
```

Step 2: Change the path using the following command.

```
> .libPaths("write the desired path here")
```

6. What is the command to check and install the “dplyr” package?

Ans: `if (!require("dplyr ")) {install.packages("dplyr")}`

7. How can we install multiple packages in R?

Ans: To install multiple packages in R the command is, `>install.packages(c("ggplot2", "tidyr", "dplyr"))`



Just Remember

To access help in RStudio, it can be accessed from the console and from the CLI (Figure 1.9). The command is `help()`.

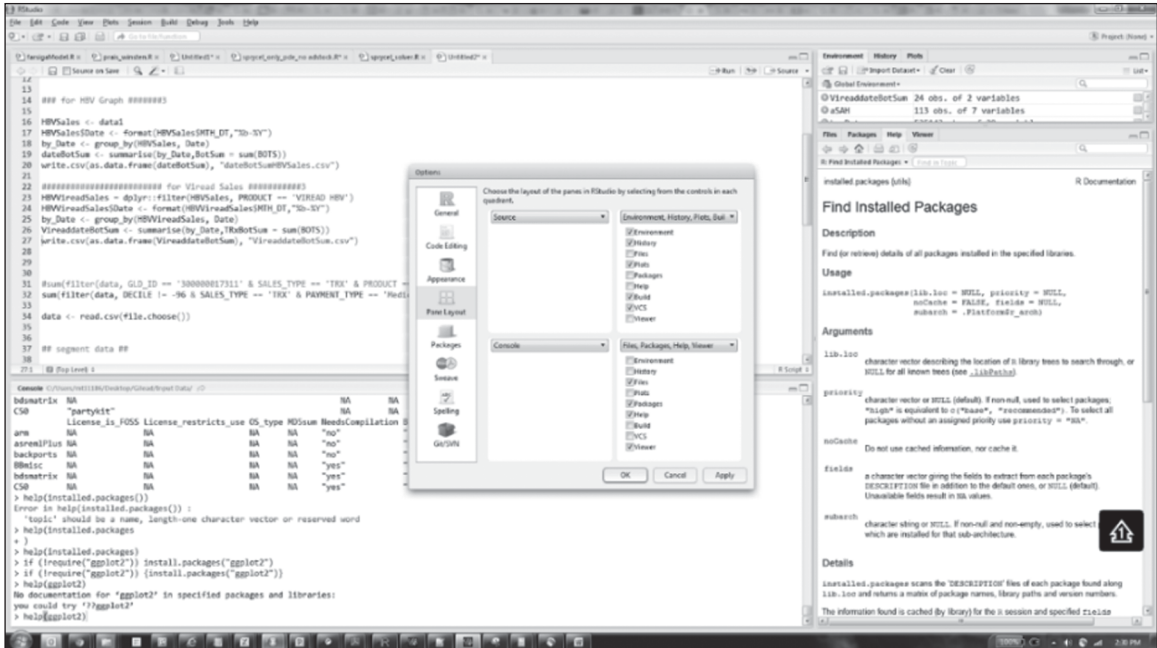


FIGURE 1.9 Accessing `help()` command from the console and CLI

Summary

- R is an open source and object-oriented programming language for statistical computing and data visualisation.
- R is a successor of the proprietary statistical computing programming language S.
- R can be downloaded and installed on different OS platforms like Windows, Linux and Mac.
- R has the fundamental data type of vector.
- Text editors like Notepad++ to R, Tinn-R and Rev R are more than just editors for R. These can support extended functionalities and IDE features.
- R has several IDEs like RStudio, Eclipse with StatET and so on.
- R has a rich library of more than 10,000 packages.
- R has two fundamental file types called RScripts and R markdown documents.
- R commands can be written in RScripts or through the command line interface.
- R has a rich collection of inbuilt data sets like `mtcars`, Biochemical Oxygen Demand (BOD), etc.



KEY TERMS

- **BOD:** An inbuilt data set in R, which contains data on the Biochemical Oxygen Demand.
- **CLI:** A console through which a user can interact with a computer. The interaction happens through successive lines of commands on the console.
- **IDE:** A special type of software that offers a set of comprehensive facilities to develop computer software. Usually, an IDE consists of a number of automation tools, a debugger and an editor for coding.
- **R:** An open source and object oriented programming language for statistical computing and data visualisation.



MULTIPLE CHOICE QUESTIONS

1. What is R?
 - (a) An object-oriented programming language
 - (b) An open source project from CRAN
 - (c) A programming language for statistical computing
 - (d) All of these
2. Which one of the following programming languages is a dialect of R language?
 - (a) Python
 - (b) C
 - (c) S
 - (d) Q
3. Which one of the following is a text editor of R?
 - (a) RStudio
 - (b) Microsoft word
 - (c) Notepad++ to R
 - (d) Tableau
4. Which of the following are IDEs for R?
 - (a) RStudio
 - (b) Both a and c
 - (c) Eclipse with StatET
 - (d) None of these
5. What is the primary file type of R?
 - (a) Vector
 - (b) Text file
 - (c) RScripts
 - (d) Statistical file
6. R can be downloaded from:
 - (a) CRAN website
 - (b) Google PlayStore
 - (c) None of these
 - (d) All of these
7. Which one of the following R packages is used for data management?
 - (a) haven
 - (b) igraph
 - (c) slidify
 - (d) forecast

8. Which one of the following R packages is used for data visualisation?
 - (a) haven
 - (b) igraph
 - (c) slidify
 - (d) forecast
9. Which one of the following R packages is used for data products?
 - (a) haven
 - (b) igraph
 - (c) slidify
 - (d) forecast
10. Which one of the following R packages is used for data modelling and simulation?
 - (a) haven
 - (b) igraph
 - (c) slidify
 - (d) forecast
11. The functionalities of R are divided among:
 - (a) Packages
 - (b) Domains
 - (c) Libraries
 - (d) None of these



SHORT QUESTIONS

1. What is R? What are the advantages of R programming language over other general purpose programming languages?
2. How can we install a package on R?
3. Give examples of two IDEs for R.
4. Give detailed examples of three packages used in R.
5. Give a detailed description of head() command used in R.
6. How can we install multiple R packages with a single command?
7. State the difference(s) between head() and tail() commands used in R.
8. State the difference(s) between ncol() and nrow() commands used in R.

Answers to MCQs:

1. (d) 2. (c) 3. (c) 4. (b) 5. (c) 6. (a) 7. (a)

8. (b) 9. (c) 10. (d) 11. (a)