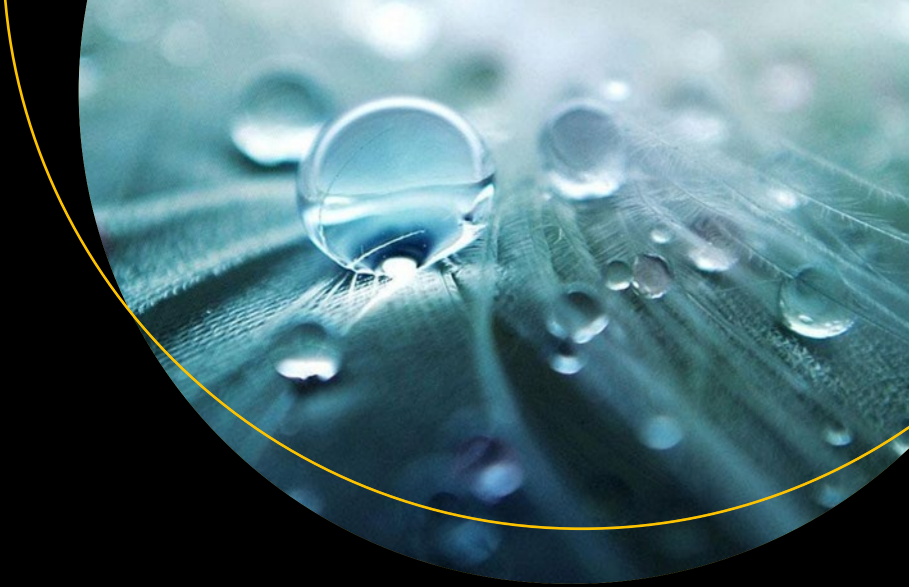

R Quick Syntax Reference

A Pocket Guide to the Language, APIs and Library

Second Edition

Margot Tollefson

Apress®

R Quick Syntax Reference

A Pocket Guide to the Language, APIs and Library

Second Edition

Margot Tollefson

Apress®

R Quick Syntax Reference: A Pocket Guide to the Language, APIs and Library

Margot Tollefson
Stratford, IA, USA

ISBN-13 (pbk): 978-1-4842-4404-3 ISBN-13 (electronic): 978-1-4842-4405-0
https://doi.org/10.1007/978-1-4842-4405-0

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Steve Anglin
Development Editor: Matthew Moodie
Coordinating Editor: Mark Powers

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail editorial@apress.com; for reprint, paperback, or audio rights, please email bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484244043. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

Table of Contents

About the Author

Margot Tollefson is a retired consulting statistician residing in the tiny town of Stratford in the corn and soybean fields of north-central Iowa. She started using the S-Plus language in the early 1990s and happily switched to R about ten years ago. Margot enjoys writing her own functions in R—to do plots and simulations as well as to implement custom modeling and use published statistical methods. She earned her graduate degrees in statistics from Iowa State University in Ames, Iowa.

Acknowledgments

I would like to thank the writers of the R Development Core Team at the Comprehensive R Archive Network. Without their help pages, this book could not have been written. I would also like to thank the editors at Apress, Steve Anglin and Matthew Moodie, for guiding my progress; and my husband, Clay Conard, for his support and patience over the last few months.

Introduction

R is a programming language that provides the user with powerful data and graphical analysis options. R is both flexible and broad. From tasks as simple as adding two numbers to tasks as complex as fitting an ARIMA model, R is capable of crunching the numbers.

The purpose of *R Quick Syntax Reference* is to provide the reader with the basic syntax of R. Often an R user gets stuck if, for example, a mode is incorrect or a logical test does not work. Because the full spectrum of R packages uses the same fairly simple syntax, *R Quick Syntax Reference* provides the reader with the necessary information to get unstuck and run and create all R functions and code.

The R language is based on the language S, a high-level programming language developed mainly by Richard A. Becker, John M. Chambers, and Allan R. Wilks in the AT&T laboratories in 1975. The R version of the language first became available in 1993 and was developed by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand.

R is open source and is a *GNU project*. As open-source code, the R language is free and constantly being improved. The R Foundation for Statistical Computing maintains the program, and the R Development Core Team currently does the development. Packages for specific analysis techniques are added often. At the present time, there are over 10,000 packages available in R. Most users will use only a few packages. We discuss using R at the command prompt in R Quick Syntax Reference. We also cover the integrated development environment (IDE), R Studio. RStudio was founded by J.J. Allaire and became available in a beta version in 2011 and as a regular version in 2017. The Chief Scientist at R Studio is Hadley Wickham.

This book is about the S3 and S4 versions of R—*S3* and S4 standing for the third and fourth versions of S, the commercial program on which R is based. The two versions run concurrently. Even though version S4 is quite different from S3, it is necessary to know the syntax of S3 in order to use S4. And S3 remains a powerful, flexible language in its own right—hence, this book.

Part I covers the basics of R. Chapter 1 describes how to download and install R and R Studio for the Windows, Mac, and Linux operating systems and also how to download packages. Because keeping separate folders for different projects is very useful, Chapter 1 gives instructions for running R and R Studio from different folders. It also gives the methods for updating the R and R Studio programs themselves.

Chapter 2 introduces the R prompt, gives a sample calculation, and describes the three parts of R—objects, operators, and assignments. We also describe the four windows of R Studio and what R scripts are. Chapter 3 covers the assignment of names to objects, demonstrates the `ls()` function that allows you to see the objects in a folder, and discusses the operators in R and R Studio.

Part II describes R objects. Objects have modes, classes, and types. Chapter 4 lists the modes and types and describes some of them. It also shows how modes and types differ. Chapter 5 discusses some of the classes and how classes differ between S3 and S4 R.

Part III covers functions. Chapter 6 starts with a list of the 30 default packages in R and follows with instructions on how to use functions. Because packaged functions all have help pages, the chapter provides instructions on how to access and use the help page of a function in both R and R studio. Chapter 7 describes how to create a function. Chapter 8 explains how to run a function—with a detailed approach to the argument list.

Part IV focuses on importing and exporting data in R and R Studio and methods for creating and manipulating some kinds of object. Chapter 9 describes several methods for importing data, gives a number of functions to create data objects, and discusses some random-number generators. It gives an example of data mining Twitter. Chapter 10 gives several methods for exporting from R and R Studio. A table of matched importing and exporting functions is included. Chapter 11 gives a number of functions that operate on objects—to bind objects together, to find descriptive qualities of an object, to assign qualities to an object, to aggregate an object in some way, or to apply functions to portions of an object.

Part V covers flow conditioning commands and functions. Chapter 12 presents the flow conditioning statements, and Chapter 13 supplies examples of them. Chapter 14 describes the two flow conditioning functions and gives examples.

Part VI discusses functions related to formatting and outputting output, looks at the results from packaged functions and at what some of the default packages contain, and provides some tips for using R and R Studio. Chapter 15 gives some rounding functions and some functions for outputting from a function. It also gives some functions that vary according to the class of the object on which the function operates and that summarize the results of the function, either textually or visually. Chapter 16 takes a look at the contents of the packages base, stats, and graphics and glances at the datasets, grDevices, methods, and utils packages. Chapter 17 describes how to deal with some common frustrations in R. More information is given on outputting from functions, plus an example of a recursive function and some advice on using R.

PART I

R Basics

CHAPTER 1

Downloading R and Setting Up a File System

The first step in using R and R Studio is to download the two programs from the Internet. R must be downloaded first. R and R Studio can be downloaded for the modern operating systems Windows, Mac OS X, and Linux. In this chapter, you will learn how to download and install R plus the 30 basic packages and R Studio, as well as how to install other packages and update R. Updating packages in R Studio is covered in Chapter 2. You will also learn how to use R in individual folders within the file system of the computer.

Downloading R and R Studio

You can download R from the web site of the Comprehensive R Archive Network (CRAN). CRAN updates the installation process from time to time; however, the instructions in this book are for the current steps at time of publication. CRAN provides instructions on the web site if the process has changed.

M. Tollefson, *R Quick Syntax Reference*, https://doi.org/10.1007/978-1-4842-4405-0_1

Begin the download process by going to the web site `https://www.r-project.org/`. At the web site, click on the link to choose the CRAN mirror. Choose a mirror near you. Links to the current versions for Windows, Mac OS X, and Linux systems are listed at the top of the window that opens when the mirror link is clicked. Select the appropriate version.

Windows

On the page that opens with the Windows link, select the link **base**, which is the top link. In the next window, click on the **download** link for the given Windows version. (Currently, the link is **Download R 3.5.1 for Windows**.) If R has not already been installed on the computer, the downloader will create a default folder in the **Documents** folder to hold R files. Unless there is a reason to change the folder name or location, accept the default. R will begin to download.

When the program finishes downloading, find the downloaded file in your file system. Downloads are put in `C://Users/User_folder/Downloads`, where ***User_folder*** is the folder of the user, unless another folder was specified earlier in the installation. Click on the downloaded file, which is an `.exe` installation file (currently `R-3.5.1-win.exe`.) A question about the safety of the program may pop up. The installation program is safe, so run the program.

The installation wizard will open. The installation process steps through several pages. On the first page, read the GNU GENERAL PUBLIC LICENSE; then, click on **Next**. For the rest of the pages, accepting the defaults on each page is fine, so click on **Next** on each page.

At the page of additional choices, click on **Next**, and the program will begin to install. When the installation is finished, click on **Finish** to complete the installation. The program and the 30 base packages are now installed. An icon for **R** will be on the computer desktop and, for Windows 10, in the start menu. To run **R**, click on the icon.

Mac OS X

On the page that opens from the Mac OS X link, first read the section under **R for Mac OS X**. The R project gives the advice to check the files for viruses and other problems.

Under **Latest release:** the package choice is the current version. Selecting the current version (the `.pkg` link, currently `R-3.5.1.pkg`) will download the package. When the packages have finished downloading, open the download folder under the username in **Finder**.

Select the R version .pkg file in the download folder. Opening the version will open the installer. With the installer open, click on **Continue** to go to the next page of the installer. Read the message from CRAN; then click **Continue**. Again, read the message from CRAN; then click **Continue**.

On the next page, you will find the license. After reading the license, click **Agree** to download **R**. On the next page, select either of the choices; then click on **Continue.** (The **Continue** button will not light up until a choice is made.)

On the next page, select **Install**. The installation program will ask for a password. After you have entered a password, the installation will begin. When the installation is finished, click on **Close**. You will next have the choice of keeping the installation or discarding it. If you keep **R**, **R** will be in the applications folder and on the launchpad and the 30 base packages will be loaded. Start **R** by opening the launchpad and selecting the **R** icon or by clicking on **R** in the applications folder.

Linux

At the CRAN site, CRAN provides source code for R for the Linux distributions Debian, openSuse, and Ubuntu. The Debian and Ubuntu distributions have been updated in 2018. The openSuse distribution dates from 2012.

The developers state that R is available through the package management system for most distributions of Linux. Look under GNU R. If the command line version of R is not available using the package management system, installing R directly from the terminal is an option. At `http://cran.r-project.org/bin/linux/distribution`, where *distribution* is debian, suse, or ubuntu, you can find instructions for installing R from the terminal command prompt.

The link to Red Hat at the CRAN site goes nowhere.

R Studio

At the R Studio site, R studio provides free source code for R Studio, as well as versions that cost. R Studio is available for Windows, Mac OS X, and the Linux distributions Debian, Red Hat, openSUSE, Ubuntu, and Fedora. To download the free version of R Studio, go to `https://www.rstudio.com/products/rstudio/download/` and go to the heading, **Installers for Supported Platforms**. Click on the link for your operating system and download and run the installer program. Follow the directions of the installer for your operating system. The instructions are similar to those for R.

On the Mac OS X system, the file RStudio-1.1.456.dmg in Filer must be opened each time the computer is booted up in order to have R Studio available in the system.

For Linux, R Studio may be available in the package manager. Search under R Studio.

Installing and Updating Packages

When initially installed, by default R comes with 30 packages. Often the user will want to use the power of the many other packages available in R. Installing and updating a package is straightforward.

Using the command line in R, for any of the operating systems, if the name of a package is known, typing

```
install.packages("package name")
```

at the R command prompt, where *package name* is the name of the package, will install the package. To update packages, typing

```
update.packages()
```

at the R command prompt will find those packages with updates and update the packages. To see which packages are already installed on the computer, enter

```
installed.packages()
```

at the R prompt.

If the name of the package is not known (also for known names), using the installer for the operating systems Windows and Mac OS X is easy. For Linux, instructions can be found at the CRAN web site, `http://cran.r-project.org`. Here you can find instructions for Windows and Mac OS X.

Installing and updating packages in R Studio is much easier. How will be given in Chapter 2, when the R Studio windows are described.

Windows

To install a package in Windows not using the command line, start by opening R. On the menu bar at the top of the screen, select **Packages**. A menu will drop down. **Select Install package(s)...**. Either the CRAN mirror window or the Packages window will come up. If the CRAN mirror window comes up, select a close mirror and click **OK**, which will bring up the Packages window.

The Packages window consists of a list of all of the available packages. Scroll down the list to find the package(s) you wish to install and select the package(s). Click on **OK** to begin the installation. As the installation proceeds, the steps of the installation will scroll on the R console. When the R prompt returns to the screen, the installation is complete.

To update packages not using the command line, select **Packages** on the menu bar and then select **Update packages…**. The Packages window to be updated will open, and it will have a list of all of the installed packages with updates. If there are none, the window will be empty. Choose the packages for updating and click on the **OK** button. If a question about using a personal library pops up, choose **Yes**. The packages will update. When the R prompt returns to the screen, the updates are complete.

Mac OS X

To install packages in Mac OS X, start by opening R. On the drop-down menu bar at the top of the screen, select **Packages & Data**. From the drop-down menu, select **Package Installer**, which brings up the R Package Installer. Click on **Get List** for a full list of packages or use the **Package Search** option to search for a package. Under either option, select the package(s) to be installed from the list.

Below the list of packages are choices for the location to put the packages. Hover over the list of location options for more information. Usually, one of the first two options will be correct. To the right of the location options are the **Install Selected** and **Update All** buttons. Before clicking on **Install Selected**, check the **Install Dependencies** box to make sure that any necessary packages are installed. Click on **Install Selected** to start the installation process. The selected packages will install.

To update packages, select **Packages & Data** from the menu bar at the top of the screen. From the drop-down menu, select **Package Installer**, which opens up the R Package Installer. At the bottom right of the Installer, select **Update All** and follow instructions.

Updating R

Since CRAN does not provide automatic updates for R, you must update it manually. The processes for Windows and Mac OS X are easy. For the Linux distributions Debian, Suse, and Ubuntu, instructions can be found in the `ReadMe` files at `http://cran.r-project/bin/linux/distribution`, where *distribution* is either Debian, Suse, or Ubuntu.

Windows

The first step in updating R in Windows is to open R and install the package **installr** if the package has not already been installed. Next, use the function **library** to provide access to **installr**. Type

```
library(installr)
```

at the command prompt and press **enter**. Then, to update R, type

```
updateR()
```

at the command prompt and press **enter**. R will either do an update or give a message that the program is up-to-date and return **False**.

Once **installr** has been installed, **installr** does not need to be installed again. The library must be accessed every time R is run.

Mac OS X

The first step in updating R in Mac OS X is to open R and select **R** from the drop-down menu bar at the top of the page. To run the updater, select **Check for R Updates** in the drop-down menu under **R** and follow instructions.

Using R in Separate Folders

Separate workspace images for R can be maintained in separate folders for Windows, Mac OS X, and Linux. This property of R is very handy for using R on separate projects. While the process of opening R in a given folder varies by the operating system, once in a folder, saving the workspace image is straightforward. When closing an R session, the program asks if the user would like to save the workspace image. If **Yes** is selected, then `.RData` and `.Rhistory` (`.Rapp.history` for Mac OS X) files are saved in the current directory. (For Mac OS X, the files are hidden, but the files are there.)

The `.RData` file contains the objects that were in R at the beginning of the session plus any objects that were added during the session minus any objects that were erased during the session. The `.Rhistory` (`.Rapp.history` for Mac OS X) file contains the history of the lines input at the R console. By default, all lines up to the last 512 lines are saved in Windows. For Mac OS X and Linux, the default is 250 lines. Access to the lines carries over from session to session if the history is saved.

Windows

To initially set up R in a folder, open R at the desktop. (Click on the **R** icon on the desktop or click on **R** in the list of programs or, in Windows 10, the Start menu.) Select **File** on the menu bar at the top of the screen. From the drop-down menu, select **Change dir…**. The **Browse to folder** window will open. Navigate to the folder of choice.

When exiting R, save the workspace image and R will create `.RData` and `.Rhistory` files in the folder. The `.RData` file will have a blue **R** icon associated with the file. In the future, going to the folder and clicking on the **R** icon will open R, and the history and objects saved within the folder will be present.

As a note for the initial setup, any objects in the desktop R will still be in R when the folder is changed. You can easily remove the objects. Type **`rm(list=ls())`** at the command prompt to remove all objects from the folder.

Mac OS X

For working within different folders in Mac OS X, there are two ways: dragging and dropping or using the terminal. For R in the **Applications** menu of **Finder**, if R is not open, dragging the folder in the **Documents** menu of **Finder** to the **R** application will open R in the folder using the `.RData` and `.Rapp.history` for that folder. (An image of the **R** application can be put in the **Documents** folder to make the dragging easier.)

To open R using the terminal, open the terminal (located under **Applications/Utilities** in **Finder**.) and type

```
open -a R folder
```

where *folder* is the location of the folder. Be sure to include the Documents folder in the name and to quote the name. R will open in the folder using the `.RData` and `.Rapp.history` files for that folder.

Linux

To open R in a given folder in Linux, change the directory to the folder and type **R** at the command prompt.

Projects in R Studio

Another way to work with separate projects is by opening new projects in R Studio. Each project has its own name and can be created using the menus in R Studio. The project can be accessed by clicking on the name in the directory where the project is saved. The extension for a project is .Rproj.

CHAPTER 2

The R Prompt and the R Studio Windows

This chapter covers the R prompt and the R Studio windows. It starts with descriptions of the three parts of R: objects, operators, and assignments. It continues with a discussion of working with the R prompt, followed by an example of doing a calculation at the R prompt. Afterward, it describes the four R Studio windows.

In Windows and macOS, R runs in GUIs: *RGUI* in Windows and *R.app GUI* in macOS. Both RGUI and R.app GUI open an R Console and run from the R prompt in the R Console. GUIs are available in Linux, but this book only covers running R from the terminal window R prompt. R in R Studio, for the three operating systems, is covered.

The Three Parts of R: Objects, Operators, and Assignments

There are basically three parts of R: objects, operators, and assignments.

> *Objects* contain information and can be, among other things, data, functions, or the results of functions. Objects always have a name. Users create

M. Tollefson, *R Quick Syntax Reference*, https://doi.org/10.1007/978-1-4842-4405-0_2

some objects, which are automatically saved on creation. Other objects are constants, functions, and datasets contained in the packages of R.

Operators manipulate objects, numbers, strings, and/or logical variables. For example, entering **a** = **2*b** at the R prompt would multiply **b** by two and assign the result to **a**. The objects **a** and **b** are numeric objects, and * is the multiplication operator. The equal sign makes an assignment of two times **b** to **a**. Operators are a type of function.

Assignments assign an expression to an object. *Expressions* can consist of objects, numbers, logical variables, strings, lists, other expressions, and/or functions, which are operated on by operators.

Expressions can be evaluated from the R prompt, instead of being evaluated and assigned to an object. (The other places where assignments and operations occur are within functions and within flow control.)

The R Prompt

All of R flows from the R prompt. R is essentially the running of functions and the doing of calculations. Functions and calculations can be run at the R prompt with or without an assignment to an object. Functions and calculations can also be run as part of another function, but everything starts at the R prompt.

Using R from the R prompt may seem daunting at first. R opens with some writing, and then a lonely little greater-than sign (>), which is the R prompt. The opening writing gives the R version number and some other information about the program, including the fact that the program runs with no warranty.

R remembers every line that is entered into the program, up to a set number of lines. A very handy side of R is that the up and down arrows on the keyboard will step through the lines. You only need to enter an expression once. Corrections to expressions are easy to do without typing the entire expression again.

To close R, enter **q()** at the R prompt or, for Windows and macOS, close the window. R will close with the option to save the workspace. In Linux, if the terminal window is closed without using **q()**, the current workspace will be lost.

The workspace consists of any objects present in R at the time the program is closed and the current history. Closing R without saving the workspace will result in reverting to the workspace present at the time the R session started.

An Example of a Calculation

The simplest use of R is as a calculator. The following calculation was done from the R prompt. There is no assignment in the calculation, so the result is returned on the screen.

```
> (1 + 3 + 7)/5
[1] 2.2
>
```

The first line gives the expression to be evaluated and the second line gives the result. The **[1]** in the second line is a label that tells the user that the result is the first value returned from the expression. Many expressions return more than one value. At the third line, the R prompt comes back and R is ready for another task.

The Four R Studio Windows

On opening the program for the first time, R Studio presents you three sub-windows. On the left side of the main R Studio window is a smaller single window. On the right are two vertically aligned smaller windows. Across the top, above the three windows, are two menus that provide several options for working with R Studio. Both menus extend the full width of the main window.

In Windows and Linux, the upper menu is the main R Studio window, while in macOS, the top menu is on the main macOS menu bar. In macOS, if R Studio is expanded to the full screen, you must hover over the top of the page to see the upper menu. One nice thing about R Studio is that, when you type, it provides autocompletion for object names. But, R Studio can be slow to load.

The upper menu has the buttons "File", "Edit", "Code", "View", "Plots", "Session", "Build", "Debug", "Profile", "Tools", and "Help". Each button opens a dropdown menu. The dropdown menus are self-explanatory.

The lower menu contains icons for (from left to right): opening new things (an R Script, an R Notebook, an R Markdown document, a Shiny Web App, a text file, an C++ file, an R Sweave document, an R HTML document, an R presentation, or an R documentation file), opening a new project, opening a file on the computer, saving the contents of the Source window (see the section on the fourth window), saving the contents under all of the Source window tabs, printing the contents of the active sub-window, searching for and opening files in the working directory (it searches for the letters from left to right and does not appear to work on macOS), adjusting the look and positions of the sub-windows, and adding add-ins.

The First Sub-window

The sub-window on the left opens to the standard R console, under a tab labeled "console." Commands are entered at the R prompt in the same way as in R. To the right of the console tab is a tab labeled "terminal," which gives access to the terminal of the computer.