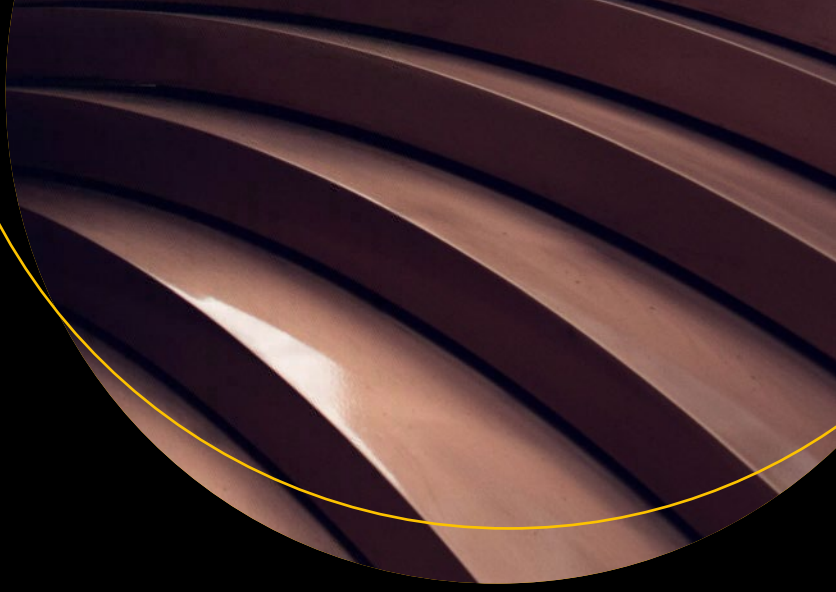

R 4 Quick Syntax Reference

A Pocket Guide to the Language, API's and Library

—

Third Edition

—

Margot Tollefson

Apress®

R 4 Quick Syntax Reference

A Pocket Guide to the Language, API's and Library

Third Edition

Margot Tollefson

Apress®

R 4 Quick Syntax Reference: A Pocket Guide to the Language, API's and Library

Margot Tollefson
Vanward Statistics, PO Box 286, 612 Teneyck Avenue, Stratford, IA, 50249-0286, USA

ISBN-13 (pbk): 978-1-4842-7923-6 ISBN-13 (electronic): 978-1-4842-7924-3
https://doi.org/10.1007/978-1-4842-7924-3

Copyright © 2022 by Margot Tollefson

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

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: James Markham
Coordinating Editor: Mark Powers

Cover designed by eStudioCalamar

Cover image by Nkululeko Jonas on Unsplash (www.unsplash.com)

Distributed to the book trade worldwide by Apress Media, LLC, 1 New York Plaza, New York, NY 10004, U.S.A. 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 booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail 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 (https://github.com/Apress). For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

This edition is dedicated to my parents,
Roy and Alice Marie, who stood by me.

Table of Contents

About the Author xv

About the Technical Reviewer xvii

Acknowledgments xix

Introduction xxi

Part I: R Basics 1

Chapter 1: Downloading R and RStudio and Setting Up a File System 3

Downloading R and RStudio 4

The Windows Systems 5

The macOS Systems 6

The Linux Systems 6

Installing and Updating Packages 8

The Windows Systems 9

The macOS Systems 9

The Linux Systems 10

Updating R 10

All Systems 11

The macOS Systems 11

Using R in Separate Folders 12
Projects in RStudio 13
The Windows Systems 13
The macOS Systems 14
The Linux Systems 14

Chapter 2: The R Prompt and the RStudio Main Menus and Subwindows 15

The Three Parts of R: Objects, Operators, and Assignments 16
The R Prompt 17
An Example of a Calculation 18
The Main Menus and the Four Subwindows in RStudio 18
The Main Menus 18
The First Subwindow 21
The Second Subwindow 22
The Third Subwindow 23
The Fourth Subwindow 24

Chapter 3: Assignments and Operators 29

Types of Assignment 29
Example of Three Types of Assignment 31
Listing and Removing Objects in RStudio and R 32
Operators 34
Logical Operators and Functions 34
Arithmetic Operators 37
Matrix Operators and Functions 38
Relational Operators 41
Subscripting Operators 42
Odds and Ends 49

Part II: Kinds of Objects 51

Chapter 4: Types and Modes of Objects 53

Overview of the Types and Modes 53

Commonly Used Types 54

Atomic, Recursive, Language, and S4 Kinds 55

Some Functions for the Atomic Kind of Types 56

The NULL Type 57

The Raw Type 57

The Logical Type 60

The Integer or Double Types and the Numeric Mode 62

The Complex Type 66

The Character Type 69

The Common Recursive Kind of Types 73

The List Type 73

The Closure, Special, and Builtin Types 77

The Environment Type 78

The S4 Type 85

The Less Common Types 86

The Language Kind of Objects 87

Notes 91

Chapter 5: Classes of Objects 93

Some Basics on Classes 93

Vectors 95

Some Common S3 Classes 97

The Matrix Class: matrix 97

The Array Class: array 103

Names for Vectors, Matrices, Arrays, and Lists 105
The Data Frame Class: data.frame 108
The Time Series Classes: ts and mts 116
The Factor Classes: factor and ordered 120
The Date and Time Classes: Date, POSIXct, POSIXlt, and difftime 123
The Formula Class: formula 136
The S4 Class 140
Reference Classes 143

Part III: Functions 153

Chapter 6: Packaged Functions 155

The Libraries 155
Loading and Removing Packages 156
Default Packages and Primitive Functions 157
Using the Help Pages 158
Identifier 159
Title 159
Description 159
Usage 159
Arguments 160
Details 161
Value 161
Some Other Optional Sections 162
References 162
See Also 162
Examples 163

Chapter 7: Scripts, User-Created Functions, and S4 and Reference Class Methods 165

Scripts 166

The Structure of a Function 168

How to Enter a Function into R 177

In RStudio 178

Using an Editor 178

Inline Entry 181

An Outside Editor: dget( ) and Copying and Pasting 182

Chapter 8: How to Use a Script or Function 201

Example of a Script: Mining Twitter 201

Calling a Function 205

Arguments 208

The Output from a Function 210

Part IV: I/O and Manipulating Objects 213

Chapter 9: Importing and Creating Data 215

Reading Data into RStudio and R, Including R Datasets 216

Connections 217

Reading Data Using RStudio 238

The scan( ) Function 241

The read.table( ) and Related Functions 258

The readLines( ), read.socket( ), readClipboard( ), and readline( ) Functions 264

The dget( ) and source( ) Functions 267

The loadhistory( ) Function 271

Some Other Functions That Read Textual Data 272

The load( ) and attach( ) Functions 272

The readRDS( ) Function 275
The readBin( ), readChar( ), and unserialize( ) Functions 277
R Datasets and the library( ), attach( ), and data( ) Functions 282
Probability Distributions and the Function sample( ) 288
Probability Distributions 288
The sample( ) Function 293
Manually Entering Data and Generating Data with Patterns 295
The c( ) Function 295
The seq( ) and rep( ) Functions 299
Combinatorics and Grid Expansion 305
The paste( ) and paste0( ) Functions 309

Chapter 10: Exporting from R 311

Connection Functions That Write 312
The sink( ) Function 313
The cat( ) and write( ) Functions 314
The write.table( ) and Related Functions 321
The writeLines( ), write.socket( ), and writeClipboard( ) Functions 326
The dput( ) and dump( ) Functions 330
The savehistory( ) Function 334
The save( ) and save.image( ) Functions 335
The saveRDS( ) Function 338
The writeBin( ), writeChar( ), and serialize( ) Functions 339
Matching Importing and Exporting Functions 348
Other Exporting Functions 350

Chapter 11: Descriptive Functions and Manipulating Objects.....353

Descriptive Functions 354

The dim() Function 354

The nrow(), ncol(), NROW(), and NCOL() Functions 355

The length() Function 357

The nchar() and nzchar() Functions.....362

Manipulating Objects 366

The cbind(), rbind(), cbind2(), and rbind2() Functions.....367

The Apply Functions 371

The eapply() Function.....391

The sweep() and scale() Functions.....393

The aggregate(), table(), tabulate(), and ftable() Functions.....398

Some Character String Functions.....409

Part V: Flow Control.....429

Chapter 12: Flow Control.....431

Brackets "{}" and the Semicolon ";" 431

The "if" and "if/else" Control Statements.....432

The "while" Control Statement 433

The "for" Control Statement.....434

The "repeat" Control Statement.....435

The Statements "break" and "next" 436

Nesting.....436

Chapter 13: Examples of Flow Control.....437

Nested "for" Loops with an "if/else" Statement 437

Using Indices 438

A "while" Loop 439

Using Indices 440

Nested “for” Loops....441
Using Indices....443
A “for” Loop, “if” Statement, and “next” Statement....445
Using Indices....446
A “for” Loop, a “repeat” Loop, an “if” Statement, and a “break” Statement....448
Using Indices....451

Chapter 14: The ifelse( ) and switch( ) Functions....455

The ifelse( ) Function....455
The switch( ) Function....462

Part VI: Some Common Functions, Packages, and Techniques....467

Chapter 15: Some Common Functions....469

The options( ) Function....470
The round( ), signif( ), and noquote( ) Functions....473
The round( ) Function....474
The signif( ) Function....476
The noquote( ) Function....478
The format( ), print( ), and plot( ) Functions....479
The format( ) Function....480
The print( ) Function....482
The plot( ) Function....483
The eval( ), attributes( ), attr( ), and summary( ) Functions....484
The eval( ) Function....484
The attributes( ) and attr( ) Functions....486
The summary( ) Function....489

The deriv(), numericDeriv(), and integrate() Functions 492
The deriv() and Related Functions 492
The numericDeriv() Function 500
The integrate() Function 502
The parse() and deparse() Functions 505
The parse() Function 506
The deparse() Function 514
Some Functions for Models: anova(), coef(), effects(), residuals(), fitted(), vcov(), confint(), and predict() 517

Chapter 16: The base, stats, and graphics Packages 523

The base Package 524
Reserved Words 524
Built-in Constants 524
Trigonometric and Hyperbolic Functions 525
Exponential and Log Functions 529
Beta- and Gamma-Related Functions 531
Miscellaneous Mathematical Functions 534
Complex Numbers 539
Matrices, Arrays, and Data Frames 540
A Few Other Functions and Some Comments 548
The stats Package 551
Basic Descriptive Statistics 552
Some Functions That Do Tests 556
Some Modeling Functions in stats 561
Clustering Algorithms and Other Multivariate Techniques 565
The graphics Package 566

Chapter 17: Tricks of the Trade.....571

Value Substitution: NA, NaN, Inf, and –Inf 571

If Statements and Logical Vectors 575

Lists and the Functions list() and c().....577

Paths and URLs.....579

Editing Objects Like Matrices and Data Frames 582

A Bit More 583

Recursive Functions.....584

Some Final Comments.....586

Index.....587

About the Author

Margot Tollefson is a self-employed consulting statistician residing in the tiny town of Stratford in the corn and bean fields of central Iowa. Her business is Vanward Statistics. She started using the S-Plus language in the early 1990s and was happy to switch to R over 18 years ago. Margot enjoys writing her own functions in R–to do plots and simulations as well as to implement custom modeling and to use published statistical methods. She earned her graduate degrees in statistics from Iowa State University in Ames, Iowa.

About the Technical Reviewer

Matt Wiley leads institutional effectiveness, research, and assessment at Victoria College, facilitating strategic and unit planning, data-informed decision making, and state/regional/federal accountability. As a tenured, associate professor of mathematics, he won awards in both mathematics education (California) and student engagement (Texas). Matt earned degrees in computer science, business, and pure mathematics from the University of California and Texas A&M systems.

Outside academia, he coauthors books about the popular R programming language and was managing partner of a statistical consultancy for almost a decade. He has programming experience with R, SQL, C++, Ruby, Fortran, and JavaScript.

A programmer, a published author, a mathematician, and a transformational leader, Matt has always melded his passion for writing with his joy of logical problem solving and data science. From the boardroom to the classroom, he enjoys finding dynamic ways to partner with interdisciplinary and diverse teams to make complex ideas and projects understandable and solvable.

Acknowledgments

First, I would like to acknowledge the R Core Team at the R Foundation for Statistical Computing, Vienna, Austria. The team provides for the help pages in R, without which this book would not have been possible. The technical editor, Matt Wiley, gave many good suggestions and corrections, for which I would like to thank him. I would also like to acknowledge the editors, Mark Powers, Nirmal Selvaraj, and Steve Anglin, at Apress for their help. And, per usual, I would like to thank my husband, Clay Conard, for his patience as I spent days writing and for his support.

Introduction

In this third edition of *R Quick Syntax Reference*, there is more of an emphasis on RStudio and the conveniences that RStudio provides. The book continues to cover how to download and update R and RStudio, how to work with both programs, the types (and modes) and classes of R objects and expressions, using the S4 object system (at the introductory level), working with and creating scripts and functions, and importing from and exporting to external files. Also, working with objects that have dimensions, working with character objects, controlling the flow of a function or group of expressions, running some useful functions (in more and less detail), and working with some trickier parts of R.

This edition focuses on object and expression types rather than modes. The coverage of the S4 object system remains the same as in the second edition, but the Reference Class object system now has a couple of sections. In Chapter 5, the coverage of date and time classes has been expanded. The chapters on importing and exporting external files have been expanded, with detailed coverage of connection functions (for connecting to external locations, including URLs and URIs). Table 10-1, in Chapter 10, has been expanded to cover most of the importing and exporting functions that are in the default packages loaded by R.

In Chapter 11, there is one more apply function (for applying a function recursively). Also, some functions for working with character strings have been added. In Chapter 13, the last example has been substantially changed (but still codes the same process). The other examples in the chapter remain the same, although some argument names have been changed.

A few more functions are presented in Chapters 15 and 16. In Chapter 15, functions for finding derivatives and integrals are now included. Sections on parsing and deparsing text have been added. Also, the functions that return attributes are described, as well as the function to evaluate expressions. In Chapter 16, the functions are organized by subject matter more than in the first two editions.

Chapter 17 now has a section on functions that work with paths that go to external locations. Also, a section on entering and editing data objects at the console has been added, as has a section on some new changes in R. The information about getting results out of a function has been moved to Chapter 7.

My purpose in writing and revising this book has been to provide a useful reference for using the versatile R language. I feel the book achieves the purpose of the book. I hope the reader does too.

Source Code

All source code associated with this book can be accessed at `http://github.com/apress/r4-quick-syntax-reference`.

PART I

R Basics

CHAPTER 1

Downloading R and RStudio and Setting Up a File System

R is an open source statistical programming language based on the commercial programming language S. The S language was developed at Bell Laboratories in the 1970s and 1980s, mainly by John Chambers. The last version of S, called S4, came out in 1998. The R language was initially developed by Ross Ihaka and Robert Gentleman at the University of Auckland (in New Zealand) through "clean-room" methods. R was written to mimic S, but with its own code. The first official version of the R language was released in 1995 and version v1.0 in 2000. The current version of R is v4.1.1 (as of September 2021). R is a GNU package and is released under the GNU General Public License. The Comprehensive R Archive Network (CRAN) makes the language and packages available and is a network of mirrors at institutions. CRAN was first deployed in 1997. (The preceding information is from the Wikipedia pages **S (programming language)** and **R (programming language)**.) The citation for R is

> *R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL* `https://www.R-project.org/`.

© Margot Tollefson 2022
M. Tollefson, *R 4 Quick Syntax Reference*, https://doi.org/10.1007/978-1-4842-7924-3_1

RStudio is an Integrated Development Environment for the R language. The citation for RStudio is

> *RStudio Team (2021). RStudio: Integrated Development Environment for R. RStudio, PBC, Boston, MA. URL* http://www.rstudio.com/.

The first step in using R and RStudio is to download the two programs from the Internet. R must be downloaded first. R and RStudio can be downloaded for the modern operating systems: the Windows systems, the macOS systems, and the Linux distributions (and other UNIX-like systems). In this chapter, you will learn how to download and install R with the 30 basic packages of R, plus how to install RStudio. You will learn how to use R to install other R packages and how to update R. Updating packages using RStudio is covered in Chapter 2. Also, you will learn how to place R history and data files in individual folders within the file system of the computer.

Downloading R and RStudio

You can download R from the website of CRAN. CRAN updates the installation process from time to time; however, the instructions in this book are for the current steps at the time of publication. CRAN provides instructions on the website if the process has changed. (Solutions also exist for downloading R to a server or root system and linking the program to remote computers, e.g., using the open source program RStudio Server supplied by RStudio.)

Begin the download process by going to the website www.r-project.org/. At the website, in the left menu, click on the link labelled **CRAN** (below **Download**), which opens a list of CRAN mirrors. Choose a mirror near you. Links to the current versions of R for Windows systems, macOS systems, and Linux systems are listed in the first section of the window. Select the appropriate version.

The Windows Systems

On the page that opens with the Windows system link, select the link **base** (which is a link under **Subdirectories**). In the next window, click on the **download** link for the given Windows system version. (Currently, the link is **Download R 4.1.0 for Windows**.) 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. Click on the downloaded file, which is an `.exe` installation file (currently `R-4.1.0-win.exe`.) A question about whether to run the program may pop up. The installation program should be safe, so run the program (as an administrator). Choose which language to use.

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

At the page of additional choices, to place an icon on the desktop and/or the Quick Start menu, check one, or both, of the first two checkboxes. Then, click **Next**, and the program will begin to install. When the installation is finished, click **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 if placing an icon was chosen and in the Start menu. To run **R**, click on the icon or on the R entry in the Start menu.

The macOS Systems

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

Under **Latest release**, the current package choices are `R-4.1.0.pkg` and `R-4.1.0-arm64.pkg`. `R-4.1.0.pkg` is for computers with a 64-bit Intel processor and macOS 10.13 or later. `R-4.1.0-arm64.pkg` is for computers with an Apple M1 or later processor and macOS 11 or later. (For earlier systems, there are links to earlier versions of R below on the page.) Selecting a version will download the package. When the packages have finished downloading, open the download folder in **Finder**.

Select the R version `.pkg` file in the download folder. Opening the version will open the installer. With the installer open, click **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 **Install**. The installation program will ask for a password. After you have entered the password (the password you use to get onto your computer), the installation will begin. When the installation is finished, click **Close**. You will next have the choice of keeping the `.pkg` file or discarding it. You can discard the file. **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 in **Finder**.

The Linux Systems

At the CRAN site, CRAN provides source code for R Linux distributions Debian, Fedora, Red Hat, openSUSE, and Ubuntu. The Debian and

Ubuntu distributions have been updated in 2021. The Fedora and Red Hat distributions date from 2020. The openSUSE distribution dates from 2016.

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. For the Debian, Fedora, Red Hat, and Ubuntu distributions, instructions for installing R can be found by clicking one of the links in the parentheses after the **Download R for Linux** link. Instructions for openSUSE are found by clicking on the link **Download R for Linux** and on the **suse** link in the page that follows.

RStudio

At the RStudio site, RStudio provides free source code for RStudio, as well as versions that cost. RStudio is available for Windows systems, macOS systems, and most Linux distributions. To download the free version of RStudio, go to `www.rstudio.com/products/rstudio/download/`. Scroll down to the blue **Download** buttons and click on the left button (under **RStudio Desktop**, **Open Source License**, **Free**). The computer focus will go to a link for your operating system. Use the link to download and run the installer program.

For Windows systems, follow the directions of the installer. The instructions are like those for R.

On the macOS systems, the file `RStudio-version_number.dmg` is the installation file, where *version_number* is the RStudio version number. The current number is `1.4.1717`. RStudio must be installed by hand each time the computer is booted if RStudio is to be used during the session.

For Linux distributions, RStudio should be available in the package manager. Search under RStudio.

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. The easiest way to install and update packages is to use RStudio. However, we cover installing and updating packages at the command line here and installing and updating packages using RStudio in Chapter 2, where we introduce the RStudio subwindows. Whether installed at the command line or by using RStudio (which just generates R commands and runs the R commands at command line), the packages are in the same folder and accessible to both programs.

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 Windows operating systems and the macOS operating systems is easy. In Linux distributions, finding the name of a package is a bit harder. Instructions for Windows systems, macOS systems, and Linux systems are given in the following.

The Windows Systems

To install a package in a Windows system 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 the available packages. Scroll down the list to find the package(s) you wish to install and select the package(s). Click **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 to be updated** window will open, and it will have a list of all the installed packages with updates. If there are none, the window will be empty. Choose the packages for updating and click 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.

The macOS Systems

To install packages in the macOS operating systems not using the command line, 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. Choose the repository in which to look for the package. Click **Get List** for a full list of packages in the repository 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. 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 **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 the R Package Installer. At the bottom right of the Installer, select **Update All** and follow instructions.

The Linux Systems

For Linux distributions, use the command line method given at the start of this section to install and update packages. To access a list of package names, enter `available.packages()` at the R prompt. Since there are over 30,000 package names, the list should be assigned to an object and the object examined (e.g., entering `ap=available.packages(); ap[ which( substr( ap, 1, 3 )=="MAL" ) ]` at the R prompt will return all packages that begin with MAL).

Updating R

Since CRAN does not provide automatic updates for R, you must update it manually. For all the operating systems, the following instructions can be used. An alternative process for the macOS systems is easier and given separately below.

All Systems

The first step in updating R is to open R and install the package **installr** if the package has not already been previously 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 loaded if the library is to be used during an R session.

(Within an R session, a package must be loaded to be used. A package is loaded at the command line by entering **library(*package_name*)** where **package_name** is the name of the package. However, a package can only be loaded if the package has been installed at some time in the past. Once loaded, R has access to the objects in the package.)

The macOS Systems

The first step in the alternative version of updating R in the macOS systems 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 systems, macOS systems, and Linux systems. 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.

For all operating systems, the function `setwd()` can be used to set the working directory, for example, `setwd("Documents/Margot's MacBook/R stuff")` on my macOS system. (Entering `getwd()` at the R prompt returns the address of the current folder.) There are also operating system specific ways to open R in a folder (see below).

When an R session is closed, the program asks if the user would like to save the workspace image. If there are objects in the workspace and **Yes** is selected, then the `.RData` and `.Rhistory` (and `.Rapp.history` for macOS systems) files are saved in the current directory. (For macOS systems and Linux systems, the files are hidden, but the files are there.) If there are no objects in the workspace, only `.Rhistory` is saved.

The workspace of an R or RStudio session contains all the user-defined objects that have been created but not removed and that have been part of a saved workspace, starting with the first time the workspace was opened and saved.

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 macOS systems, if the file exists) 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 systems. For macOS systems and Linux systems, the