

LEARNING MADE EASY



# R Projects

for  
**dummies**<sup>®</sup>  
A Wiley Brand

A large circular graphic on the left side of the cover. It features a dark blue background with a grid of small white dots. Overlaid on this are several vertical lines of varying heights, some colored red and some white, with small circles at their ends, resembling a data visualization or a stylized world map.

Learn a wide range of  
R applications

Work through R projects  
and sharpen your skillset

Understand how to execute  
your own R projects

**Joseph Schmuller, PhD**





# R Projects

by Joseph Schmuller, PhD

for  
**dummies**<sup>®</sup>  
A Wiley Brand

## R Projects For Dummies®

Published by: **John Wiley & Sons, Inc.**, 111 River Street, Hoboken, NJ 07030-5774, [www.wiley.com](http://www.wiley.com)

Copyright © 2018 by John Wiley & Sons, Inc., Hoboken, New Jersey

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the Publisher. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Trademarks:** Wiley, For Dummies, the Dummies Man logo, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002. For technical support, please visit <https://hub.wiley.com/community/support/dummies>.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit [www.wiley.com](http://www.wiley.com).

Library of Congress Control Number: 2017964027

ISBN: 978-1-119-44618-7; 978-1-119-44617-0 (ebk); 978-1-119-44616-3 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

# Contents at a Glance

<b>Introduction</b>	1
<b>Part 1: The Tools of the Trade</b>	5
CHAPTER 1: R: What It Does and How It Does It	7
CHAPTER 2: Working with Packages	31
CHAPTER 3: Getting Graphic	43
<b>Part 2: Interacting with a User</b>	77
CHAPTER 4: Working with a Browser	79
CHAPTER 5: Dashboards — How Dashing!	107
<b>Part 3: Machine Learning</b>	143
CHAPTER 6: Tools and Data for Machine Learning Projects	145
CHAPTER 7: Decisions, Decisions, Decisions	167
CHAPTER 8: Into the Forest, Randomly	185
CHAPTER 9: Support Your Local Vector	201
CHAPTER 10: K-Means Clustering	221
CHAPTER 11: Neural Networks	237
<b>Part 4: Large(ish) Data Sets</b>	253
CHAPTER 12: Exploring Marketing	255
CHAPTER 13: From the City That Never Sleeps	275
<b>Part 5: Maps and Images</b>	291
CHAPTER 14: All Over the Map	293
CHAPTER 15: Fun with Pictures	305
<b>Part 6: The Part of Tens</b>	319
CHAPTER 16: More Than Ten Packages for Your R Projects	321
CHAPTER 17: More than Ten Useful Resources	327
<b>Index</b>	331



# Table of Contents

<b>INTRODUCTION</b>	1
About This Book	2
Part 1: The Tools of the Trade	2
Part 2: Interacting with a User	2
Part 3: Machine Learning	2
Part 4: Large(ish) Data Sets	2
Part 5: Maps and Images	2
Part 6: The Part of Tens	3
What You Can Safely Skip	3
Foolish Assumptions	3
Icons Used in This Book	3
Beyond the Book	4
Where to Go from Here	4
 <b>PART 1: THE TOOLS OF THE TRADE</b>	 5
<b>CHAPTER 1: R: What It Does and How It Does It</b>	7
Getting R	7
Getting RStudio	8
A Session with R	11
The working directory	11
Getting started	12
R Functions	15
User-Defined Functions	16
Comments	18
R Structures	18
Vectors	18
Numerical vectors	19
Matrices	21
Lists	24
Data frames	25
Of for Loops and if Statements	28
 <b>CHAPTER 2: Working with Packages</b>	 31
Installing Packages	31
Examining Data	33
Heads and tails	33
Missing data	33
Subsets	34
R Formulas	35
More Packages	36
Exploring the tidyverse	37

<b>CHAPTER 3: Getting Graphic</b>	43
Touching Base	43
Histograms	44
Density plots	45
Bar plots	47
Grouping the bars	49
Quick Suggested Project	51
Pie graphs	53
Scatterplots	53
Scatterplot matrix	55
Box plots	56
Graduating to ggplot2	57
How it works	58
Histograms	59
Bar plots	61
Grouped bar plots	62
Grouping yet again	64
Scatterplots	67
The plot thickens	68
Scatterplot matrix	72
Box plots	73
 <b>PART 2: INTERACTING WITH A USER</b>	 77
<b>CHAPTER 4: Working with a Browser</b>	79
Getting Your Shine On	79
Creating Your First shiny Project	80
The user interface	83
The server	84
Final steps	85
Getting reactive	86
Working with ggplot	89
Changing the server	90
A few more changes	92
Getting reactive with ggplot	94
Another shiny Project	96
The base R version	97
The ggplot version	104
Suggested Project	106
 <b>CHAPTER 5: Dashboards — How Dashing!</b>	 107
The shinydashboard Package	107
Exploring Dashboard Layouts	108
Getting started with the user interface	109
Building the user interface: Boxes, boxes, boxes	110



Lining up in columns . . . . .	117
A nice trick: Keeping tabs . . . . .	121
Suggested project: Add statistics . . . . .	125
Suggested project: Place valueBoxes in tabPanels. . . . .	126
Working with the Sidebar . . . . .	126
The user interface . . . . .	128
The server . . . . .	131
Suggested project: Relocate the slider . . . . .	133
Interacting with Graphics . . . . .	135
Clicks, double-clicks, and brushes — oh, my! . . . . .	135
Why bother with all this? . . . . .	138
Suggested project: Experiment with airquality . . . . .	141
<b>PART 3: MACHINE LEARNING . . . . .</b>	<b>143</b>
<b>CHAPTER 6: Tools and Data for Machine Learning Projects . . . . .</b>	<b>145</b>
The UCI (University of California-Irvine) ML Repository . . . . .	146
Downloading a UCI dataset . . . . .	146
Cleaning up the data . . . . .	148
Exploring the data . . . . .	150
Exploring relationships in the data . . . . .	152
Introducing the Rattle package . . . . .	157
Using Rattle with iris . . . . .	159
Getting and (further) exploring the data . . . . .	159
Finding clusters in the data . . . . .	162
<b>CHAPTER 7: Decisions, Decisions, Decisions . . . . .</b>	<b>167</b>
Decision Tree Components . . . . .	167
Roots and leaves . . . . .	168
Tree construction . . . . .	168
Decision Trees in R . . . . .	169
Growing the tree in R . . . . .	169
Drawing the tree in R . . . . .	171
Decision Trees in Rattle . . . . .	173
Creating the tree . . . . .	174
Drawing the tree . . . . .	175
Evaluating the tree . . . . .	176
Project: A More Complex Decision Tree . . . . .	177
The data: Car evaluation . . . . .	177
Data exploration . . . . .	179
Building and drawing the tree . . . . .	180
Evaluating the tree . . . . .	181
Quick suggested project: Understanding the complexity parameter . . . . .	181
Suggested Project: Titanic . . . . .	182

<b>CHAPTER 8: Into the Forest, Randomly</b>	185
Growing a Random Forest	185
Random Forests in R	187
Building the forest	187
Evaluating the forest	189
A closer look	190
Plotting error	191
Plotting importance	193
Project: Identifying Glass	194
The data	194
Getting the data into Rattle	195
Exploring the data	196
Growing the random forest	198
Visualizing the results	198
Suggested Project: Identifying Mushrooms	200
<b>CHAPTER 9: Support Your Local Vector</b>	201
Some Data to Work With	201
Using a subset	202
Defining a boundary	202
Understanding support vectors	203
Separability: It's Usually Nonlinear	205
Support Vector Machines in R	207
Working with e1071	207
Working with kernlab	212
Project: House Parties	214
Reading in the data	216
Exploring the data	217
Creating the SVM	218
Evaluating the SVM	220
Suggested Project: Titanic Again	220
<b>CHAPTER 10: K-Means Clustering</b>	221
How It Works	221
K-Means Clustering in R	223
Setting up and analyzing the data	223
Understanding the output	224
Visualizing the clusters	225
Finding the optimum number of clusters	226
Quick suggested project: Adding the sepals	229
Project: Glass Clusters	231
The data	231
Starting Rattle and exploring the data	232
Preparing to cluster	233

Doing the clustering .....	234
Going beyond Rattle .....	234
Suggested Project: A Few Quick Ones .....	235
Visualizing data points and clusters .....	235
The optimum number of clusters .....	236
Adding variables .....	236
<b>CHAPTER 11: Neural Networks .....</b>	<b>237</b>
Networks in the Nervous System .....	237
Artificial Neural Networks .....	238
Overview .....	238
Input layer and hidden layer .....	239
Output layer .....	240
How it all works .....	240
Neural Networks in R .....	241
Building a neural network for the iris data frame .....	241
Plotting the network .....	243
Evaluating the network .....	244
Quick suggested project: Those sepals .....	245
Project: Banknotes .....	245
The data .....	245
Taking a quick look ahead .....	246
Setting up Rattle .....	247
Evaluating the network .....	249
Going beyond Rattle: Visualizing the network .....	249
Suggested Projects: Rattling Around .....	251
<b>PART 4: LARGE(ISH) DATA SETS .....</b>	<b>253</b>
<b>CHAPTER 12: Exploring Marketing .....</b>	<b>255</b>
Project: Analyzing Retail Data .....	255
The data .....	256
RFM in R .....	257
Enter Machine Learning .....	265
K-means clustering .....	265
Working with Rattle .....	267
Digging into the clusters .....	268
The clusters and the classes .....	270
Quick suggested project .....	271
Suggested Project: Another Data Set .....	272

<b>CHAPTER 13: From the City That Never Sleeps</b>	275
Examining the Data Set	275
Warming Up	276
Glimpsing and viewing	276
Piping, filtering, and grouping	277
Visualizing	279
Joining	280
Quick Suggested Project: Airline names	283
Project: Departure Delays	283
Adding a variable: weekday	283
Quick Suggested Project: Analyze weekday differences	284
Delay, weekday, and airport	285
Delay and flight duration	287
Suggested Project: Delay and Weather	289
 <b>PART 5: MAPS AND IMAGES</b>	291
 <b>CHAPTER 14: All Over the Map</b>	293
Project: The Airports of Wisconsin	293
Dispensing with the preliminaries	293
Getting the state geographic data	294
Getting the airport geographic data	295
Plotting the airports on the state map	298
Quick Suggested Project: Another source of airport geographic info	299
Suggested Project 1: Map Your State	299
Suggested Project 2: Map the Country	299
Plotting the state capitals	301
Plotting the airports	302
 <b>CHAPTER 15: Fun with Pictures</b>	305
Polishing a Picture: It's magick!	305
Reading the image	306
Rotating, flipping, and flopping	307
Annotating	308
Combining transformations	309
Quick suggested project: Three F's	309
Combining images	310
Animating	311
Making your own morphs	312
Project: Two Legends in Search of a Legend	313
Getting Stan and Ollie	313
Combining the boys with the background	314
Explaining image_apply()	314
Getting back to the animation	316
Suggested Project: Combine an Animation with a Plot	316

<b>PART 6: THE PART OF TENS</b>	319
<b>CHAPTER 16: More Than Ten Packages for Your R Projects</b>	321
Machine Learning	321
Databases	322
Maps	322
Image Processing	324
Text Analysis	324
<b>CHAPTER 17: More than Ten Useful Resources</b>	327
Interacting with Users	327
Machine Learning	328
Databases	328
Maps and Images	329
<b>INDEX</b>	331



# Introduction

---

If you're like me, you think the best way to learn is by doing. Don't just read about something — practice it! If you want to be a builder, then build. If you want to be a writer, then write. If you want to be a carpenter, then carpenter. (Yes, that noun and verb are the same. *Carpent* is not a word.)

I based this book on that learning-by-doing philosophy. My objective is for you to expand your R skill set by using R to complete projects in a variety of areas, and to learn something about those areas, too.

Even with those noble intentions, a book like this one can fall into a trap. It can quickly become a cookbook: Use this package, use these functions, create a graphic — and presto, you've finished a project and it's time to move on.

I didn't want to write that book. Instead, beginning in Part 2 (which is where the projects start), each chapter does more than just walk you through a project. First, I show you some background material about the subject area, and then (in most chapters) you work through a scaled-down project in that area to get your feet wet, and then you complete a larger project.

But a chapter doesn't end there. At the end of each chapter, you'll find a Suggested Project that challenges you to apply your newly minted skills. For each of those, I supply just enough information to get you started. (Wherever necessary, I include tips about potential pitfalls.)

Along the way, you'll also encounter Quick Suggested Projects. These are based on tweaks to projects you've already completed, and they present additional challenges to your growing skill set.

One more thing: Every subject area could be the basis for an entire book, so I can only scratch the surface of each one. Chapter 17 directs you toward resources that provide more information.

# About This Book

---

I've organized this book into six parts.

## Part 1: The Tools of the Trade

Part 1 is all about R and RStudio. I discuss R functions, structures, and packages, and I show you how to create a variety of graphics.

## Part 2: Interacting with a User

The projects begin in Part 2, where you learn to create applications that respond to users. I discuss the `shiny` package for working with web browsers, and the `shinydashboard` package for creating dashboards.

## Part 3: Machine Learning

This is the longest part of the book. I begin by telling you about the University of California–Irvine Machine Learning Repository, which provides the data sets for the projects. I also discuss the `rattle` package for creating machine learning applications. The projects cover decision trees, random forests, support vector machines, k-means clustering, and neural networks.

## Part 4: Large(ish) Data Sets

The two projects in Part 4 deal with larger data sets than you encounter earlier in the book. The first project is a customer segmentation analysis of over 300,000 customers of an online retailer. A follow-up analysis applies machine learning. The second project analyzes a data set of more than 500,000 airline flights.

## Part 5: Maps and Images

Two projects are in Part 5. The first is to plot the location (along with other information) of airports in one of the US states. The second shows you how to combine an animated image with a stationary one.



## Part 6: The Part of Tens

The first chapter in Part 6 provides information about useful packages that can help you with future projects. The second tells you where to learn more about the subject areas of this book.

## What You Can Safely Skip

Any reference book throws a lot of information at you, and this one is no exception. I intended it all to be useful, but I didn't aim it all at the same level. So if you're not deeply into the subject matter, you can avoid paragraphs marked with the Technical Stuff icon, and you can also skip the sidebars.

## Foolish Assumptions

I'm assuming that you

- » Know how to work with Windows or the Mac. I don't spell out the details of pointing, clicking, selecting, and other actions.
- » Can install R and RStudio (I show you how in Chapter 1), and follow along with the examples. I use the Windows version of RStudio, but you should have no problem if you're working on a Mac.

## Icons Used in This Book

You'll find icons in all *For Dummies* books, and this one is no exception. Each one is a little picture in the margin that lets you know something special about the paragraph it sits next to.



TIP

This icon points out a hint or a shortcut that helps you in your work and makes you an all-around better person.



REMEMBER

This one points out timeless wisdom to take with you as you continue on the path to enlightenment.



WARNING

Pay attention to this icon. It's a reminder to avoid something that might gum up the works for you.



TECHNICAL  
STUFF

As I mention in the earlier section “What You Can Safely Skip,” this icon indicates material you can blow past if it's just too technical. (I've kept this information to a minimum.)

## Beyond the Book

In addition to what you're reading right now, this product comes with a free access-anywhere Cheat Sheet that presents a selected list of R functions and describes what they do. To get this Cheat Sheet, visit [www.dummies.com](http://www.dummies.com) and type **R Projects For Dummies Cheat Sheet** in the Search box.

## Where to Go from Here

You can start the book anywhere, but here are a couple of hints. Want to introduce yourself to R and packages? You'll find the info in Chapters 1 and 2. Want to start with graphics? Hit Chapter 3. For anything else, find it in the table of contents or in the index and go for it.

If you're a cover-to-cover reader, turn the page. . . .

# 1

## **The Tools of the Trade**

## IN THIS PART . . .

Learn about R and RStudio

Understand R Functions and Structures

Create your own R Functions

Examine data

Use base R graphics

Graduate to ggplot2 graphics

- » Getting R and RStudio on your computer
- » Plunging into a session with R
- » Working with R functions
- » Working with R structures

## Chapter 1

# R: What It Does and How It Does It

**S**o you're ready to journey into the wonderful world of R! Designed by and for statisticians and data scientists, R has a short but illustrious history.

In the 1990s, Ross Ihaka and Robert Gentleman developed R at the University of Auckland, New Zealand. The Foundation for Statistical Computing supports R, which is growing more popular by the day.

## Getting R

If you don't already have R on your computer, the first thing to do is to download R and install it.

You'll find the appropriate software on the website of the Comprehensive R Archive Network (CRAN). In your browser, type this web address if you work in Windows:

```
cran.r-project.org/bin/windows/base
```

Type this one if you work on the Mac:

```
cran.r-project.org/bin/macosx
```

Click the link to download R. This puts a `win.exe` file in your Windows computer or a `pkg` file in your Mac. In either case, follow the usual installation procedures. When installation is complete, Windows users see two R icons on their desktop, one for 32-bit processors and one for 64-bit processors (pick the one that's right for you). Mac users see an R icon in their Application folder.



TIP

Both addresses provide helpful links to FAQs. The windows-related one also has the link Installation and Other Instructions.

## Getting RStudio

Working with R is a lot easier if you do it through an application called RStudio. Computer honchos refer to RStudio as an IDE (*Integrated Development Environment*). Think of it as a tool that helps you write, edit, run, and keep track of your R code, and as an environment that connects you to a world of helpful hints about R.

Here's the web address for this terrific tool:

```
www.rstudio.com/products/rstudio/download
```

Click the link for the installer for your computer's operating system — Windows, Mac, or a flavor of Linux — and again follow the usual installation procedures.



TIP

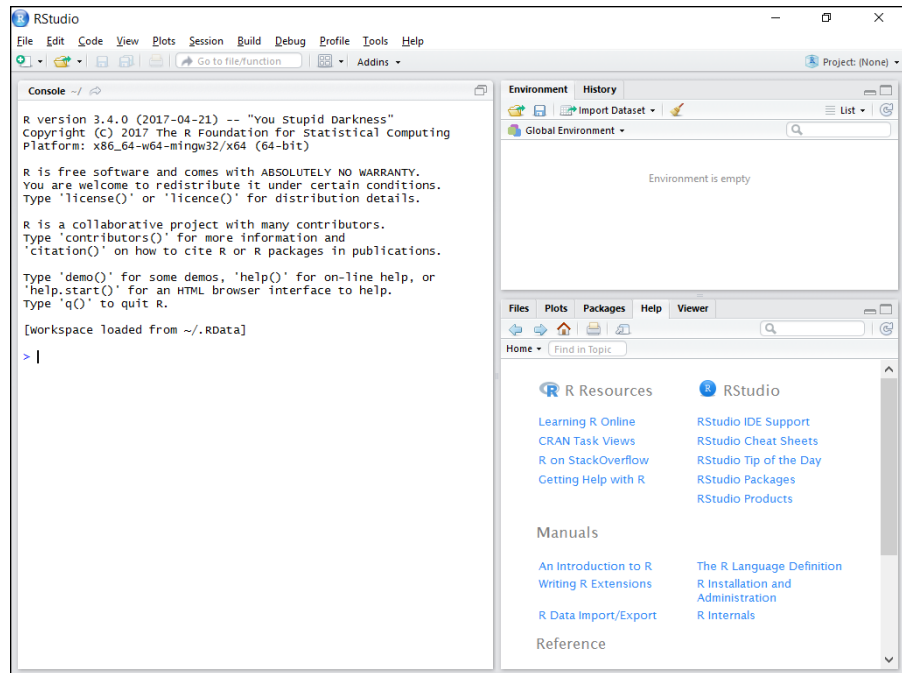
In this book, I work with R version 3.4.0 and RStudio version 1.0.143. By the time you read this, later versions of both might be available.

After you finish installing R and RStudio, click on your brand-new RStudio icon to open the window shown in Figure 1-1.

The large Console pane on the left runs R code. One way to run R code is to type it directly into the Console pane. I show you another in a moment.

The other two panes provide helpful information as you work with R. The Environment/History pane is in the upper right. The Environment tab keeps track of the things you create (which R calls objects) as you work with R. The History tab tracks R code that you enter.

**FIGURE 1-1:**  
RStudio,  
immediately after  
you install it and  
click on its icon.



**TIP**

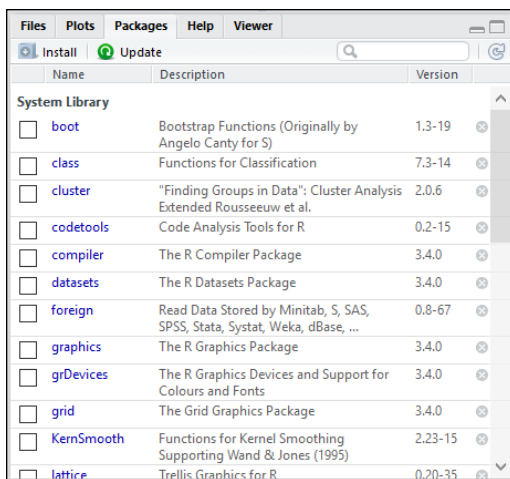
Get used to the word *object*. Everything in R is an object. The Files/Plots/Packages/Help pane is in the lower right. The Files tab shows files you create. The Plots tab holds graphs you create from your data. The Packages tab shows add-ons (called *packages*) that have downloaded with R. Bear in mind that *downloaded* doesn't mean "ready to use." To use a package's capabilities, one more step is necessary, and trust me — you'll want to use packages.

Figure 1-2 shows the Packages tab. I discuss packages later in this chapter.

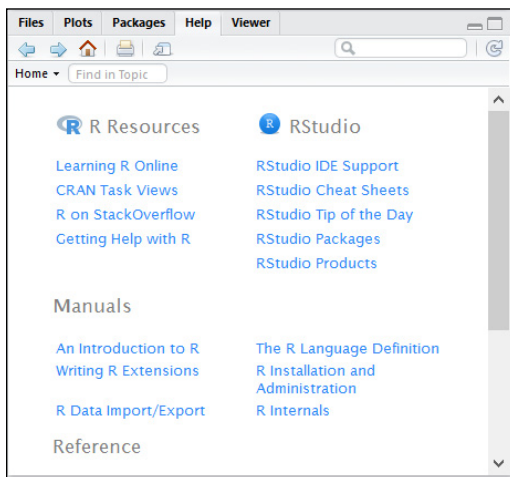
The Help tab, shown in Figure 1-3, links you to a wealth of information about R and RStudio.

To tap into the full power of RStudio as an IDE, click the icon in the upper right corner of the Console pane. That changes the appearance of RStudio so that it looks like Figure 1-4.

**FIGURE 1-2:**  
The RStudio  
Packages tab.



**FIGURE 1-3:**  
The RStudio  
Help tab.



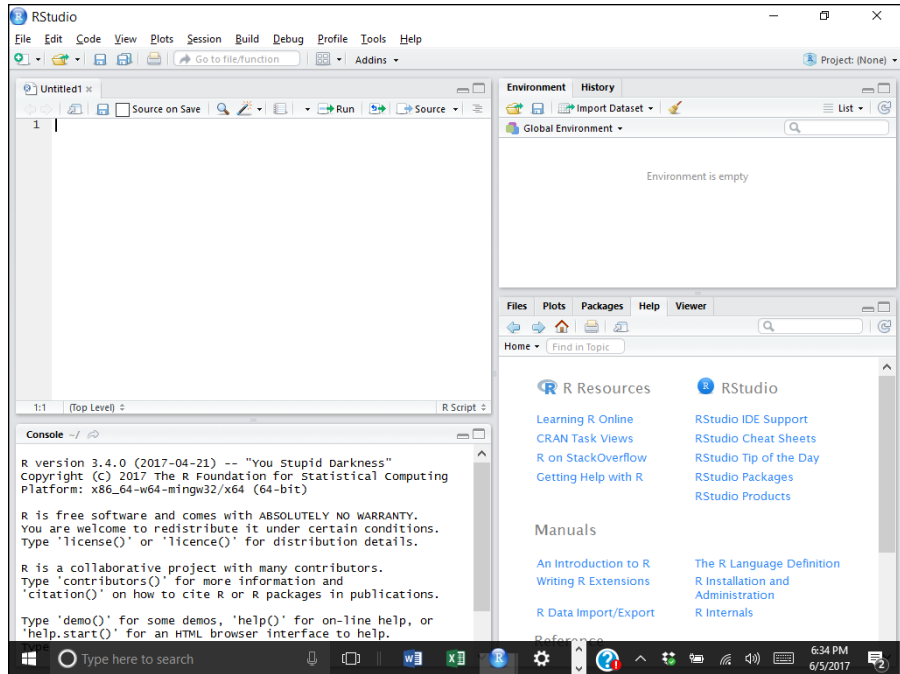
The Console pane relocates to the lower left. The new pane in the upper left is the Scripts pane. You type and edit code in the Scripts pane by pressing Ctrl+R (Command+Enter on the Mac), and then the code executes in the Console pane.



**TIP**

Ctrl+Enter works just like Ctrl+R. You can also highlight lines of code in the Scripts pane and select Code ⇄ Run Selected Line(s) from RStudio's main menu.





**FIGURE 1-4:** RStudio after you click the icon in the upper right corner of the Console pane.

## A Session with R

Before you start working, select **File** → **Save As** from the main menu and then save your work file as **My First R Session**. This relabels the tab in the Scripts pane with the name of the file and adds the `.R` extension. This also causes the filename (along with the `.R` extension) to appear on the Files tab.

## The working directory

What exactly does R save, and where does R save it? What R saves is called the *workspace*, which is the environment you're working in. R saves the workspace in the *working directory*. In Windows, the default working directory is

```
C:\Users\<User Name>\Documents
```

On a Mac, it's

```
/Users/<User Name>
```

If you ever forget the path to your working directory, type

```
> getwd()
```

in the Console pane, and R returns the path onscreen.



TIP

In the Console pane, you don't type the right-pointing arrowhead at the beginning of the line. That's a prompt.

My working directory looks like this:

```
> getwd()
[1] "C:/Users/Joseph Schuller/Documents"
```

Note the direction the slashes are slanted. They're opposite to what you typically see in Windows file paths. This is because R uses `\` as an *escape character* — whatever follows the `\` means something different from what it usually means. For example, `\t` in R means *Tab key*.



TIP

You can also write a Windows file path in R as

```
C:\\Users\\<User Name>\\Documents
```

If you like, you can change the working directory:

```
> setwd(<file path>)
```

Another way to change the working directory is to select Session ⇨ Set Working Directory ⇨ Choose Directory from the main menu.

## Getting started

Let's get down to business and start writing R code. In the Scripts pane, type

```
x <- c(5,10,15,20,25,30,35,40)
```

and then press Ctrl+R.

That puts this line into the Console pane:

```
> x <- c(5,10,15,20,25,30,35,40)
```

As I say in an earlier Tip paragraph, the right-pointing arrowhead (the greater-than sign) is a prompt that R puts in the Console pane. You don't see it in the Scripts pane.

Here's what R just did: The arrow-sign says that `x` gets assigned whatever is to the right of the arrow-sign. Think of the arrow-sign as R's *assignment operator*. So the set of numbers 5, 10, 15, 20 . . . 40 is now assigned to `x`.



REMEMBER

In R-speak, a set of numbers like this is a *vector*. I tell you more about this concept in the later section “R Structures.”

You can read that line of code as “`x` gets the vector 5, 10, 15, 20.”

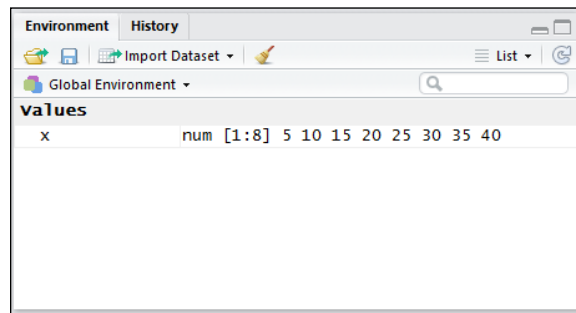
Type `x` into the Scripts pane and press Ctrl+R, and here's what you see in the Console pane:

```
> x
[1]  5 10 15 20 25 30 35 40
```

The 1 in square brackets is the label for the first line of output. So this signifies that 5 is the first value.

Here you have only one line, of course. What happens when R outputs many values over many lines? Each line gets a bracketed numeric label, and the number corresponds to the first value in the line. For example, if the output consists of 23 values and the eighteenth value is the first one on the second line, the second line begins with [18].

Creating the vector `x` causes the Environment tab to look like Figure 1-5.



**FIGURE 1-5:**  
The RStudio  
Environment tab  
after creating the  
vector `x`.



TIP

Another way to see the objects in the environment is to type `ls()` into the Scripts pane and then press Ctrl+R. Or you can type `> ls()` directly into the Console pane and press Enter. Either way, the result in the Console pane is

```
[1] "x"
```

Now you can work with `x`. First, add all numbers in the vector. Typing **sum(x)** in the Scripts pane (be sure to follow with Ctrl+R) executes the following line in the Console pane:

```
> sum(x)
[1] 180
```

How about the average of the numbers in vector `x`?

That would involve typing **mean(x)** in the Scripts pane, which (when followed by Ctrl+R) executes

```
> mean(x)
[1] 22.5
```

in the Console pane.



TIP

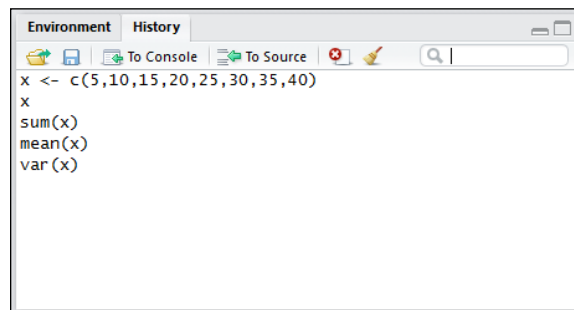
As you type in the Scripts pane or in the Console pane, you see that helpful information pops up. As you become experienced with RStudio, you learn how to use that information.

*Variance* is a measure of how much a set of numbers differ from their mean. Here's how to use R to calculate variance:

```
> var(x)
[1] 150
```

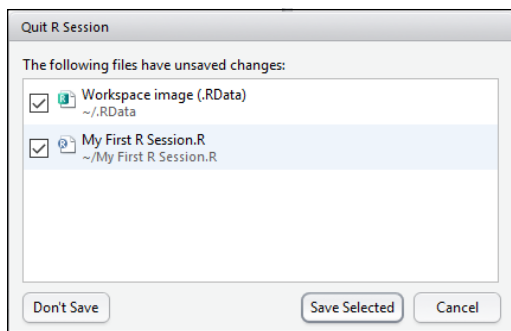
What, exactly, is variance and what does it mean? (Shameless plug alert.) For the answers to these and numerous other questions about statistics and analysis, read one of the most classic works in the English language: *Statistical Analysis with R For Dummies* (written by yours truly and published by Wiley).

After R executes all these commands, the History tab looks like Figure 1-6.



**FIGURE 1-6:**  
The History tab,  
after creating and  
working with a  
vector.

To end a session, select **File** ⇨ **Quit Session** from the main menu or press **Ctrl+Q**. As Figure 1-7 shows, a dialog box opens and asks what you want to save from the session. Saving the selections enables you to reopen the session where you left off the next time you open RStudio (although the Console pane doesn't save your work).



**FIGURE 1-7:**  
The Quit R  
Session  
dialog box.



REMEMBER

Moving forward, most of the time I don't say "Type this code into the Scripts pane and press **Ctrl+Enter**" whenever I take you through an example. I just show you the code and its output, as in the `var()` example.



REMEMBER

Also, sometimes I show code with the `>` prompt, and sometimes without. Generally, I show the prompt when I want you to see R code and its results. I don't show the prompt when I just want you to see R code that I create in the Scripts pane.

## R Functions

The examples in the preceding section use `c()`, `sum()`, and `var()`. These are three *functions* built into R. Each one consists of a function name immediately followed by parentheses. Inside the parentheses are *arguments*. In the context of a function, *argument* doesn't mean "debate" or "disagreement" or anything like that. It's the math name for whatever a function operates on.



REMEMBER

Sometimes a function takes no arguments (as is the case with `ls()`). You still include the parentheses.

The functions in the examples I showed you are pretty simple: Supply an argument, and each one gives you a result. Some R functions, however, take more than one argument.

R has a couple of ways for you to deal with multi-argument functions. One way is to list the arguments in the order that they appear in the function's definition. R calls this *positional mapping*.

Here's an example. Remember when I created the vector `x`?

```
x <- c(5,10,15,20,25,30,35,40)
```

Another way to create a vector of those numbers is with the function `seq()`:

```
> y <- seq(5,40,5)
> y
[1] 5 10 15 20 25 30 35 40
```

Think of `seq()` as creating a “sequence.” The first argument to `seq()` is the number to start the sequence *from* (5). The second argument is the number that ends the sequence — the number the sequence goes *to* (40). The third argument is the increment of the sequence — the amount the sequence increases *by* (5).

If you *name* the arguments, it doesn't matter how you order them:

```
> z <- seq(to=40,by=5,from=5)
> z
[1] 5 10 15 20 25 30 35 40
```

So when you use a function, you can place its arguments out of order, if you name them. R calls this *keyword matching*. This comes in handy when you use an R function that has many arguments. If you can't remember their order, use their names, and the function works.



TIP

For help on a particular function — `seq()`, for example — type `?seq`. When you run that code, helpful information appears on the Help tab and useful information pops up in a little window right next to where you're typing.

## User-Defined Functions

R enables you to create your own functions, and here are the fundamentals on how to do it.