

1 Getting Started

CHAPTER OUTLINE

Learning Objectives

Overview

R, RStudio, and R Markdown

Objects and Functions

Getting Started in RStudio

Navigating RStudio With R Markdown

Using R Markdown Files Versus R-Scripts

A Little Practice

Summary

Common Problems

Review Questions

Practice on Analysis and Visualization

Annotated R Functions

Answers

LEARNING OBJECTIVES

- Explain the difference between R, RStudio, and R Markdown.
- Distinguish between functions and objects.
- Download, install, and navigate both R and RStudio.
- Identify the main features of R Markdown.
- Create a Word or HTML file from R Markdown.

Overview

Learning how to use a high-powered microscope in biology class is a useful analogy to our task in this chapter. Learning how to manipulate and examine data in RStudio is analogous to positioning biological material on a slide, correctly placing it underneath the lens, and bringing it into focus. This analogy can be taken one step further. In biology class, you also learn how to take lab notes and present your findings. RStudio makes it easy to record your analysis and to present your results with professional-looking documents.

The goal of this chapter is to introduce you to R, RStudio, and R Markdown. We will be using RStudio to execute commands in R. RStudio makes many tasks in R easier since it provides an intuitive interface between you and R. Within RStudio, you will use R Markdown to take notes and generate professional-looking documents.

Mastering statistical software is an important part of analyzing data. Good analysis follows from scientists' skillful use of their lab equipment; scientists skilled in the lab are more likely to make interesting discoveries. The same holds true for you.

R, RStudio, and R Markdown

R is a statistical programming language based on S, a programming language developed at Bell Labs in the 1960s. R is an open-source language, administered by the R Core Development Team and the R Foundation. The program is free. Thousands of users contribute by writing their own R code and making it available in **packages**. Thousands of packages exist, all designed to help with specific aspects of data analysis. Packages are add-on features that you install to help make your analysis easier and more effective. Users submit packages to the R online community to test the code. Once testing is complete, the code is made available to all (<https://www.r-project.org>).

The interface for R, what the user actually sees when running it, is rather austere and based primarily on typing instructions to a prompt. In 2010, J. J. Allaire developed RStudio, an integrated development environment (IDE) that makes using R much easier. While the work you'll be doing in this book will technically be in R, you will be executing the commands, taking notes, and producing reports using RStudio.

Within RStudio, there is a note-taking and document-generating feature called **R Markdown**. R Markdown is a lightweight markup language for working with R based on Markdown, a more general lightweight markup language. Why bother with R Markdown? Why not simply cut and paste graphics and statistical output into a Word file? While you can simply cut and paste graphs and statistical output into Word, I recommend using R Markdown. R Markdown serves two important purposes. First, it makes combining data graphics with text extremely easy, generating documents in HTML, Word, or L^AT_EX.¹ Second, R Markdown encourages the careful practice of taking notes so that you can easily replicate your results. Communicating results, both to yourself and to a wider audience, is an important but often overlooked part of data analysis. Not here.

This book is designed to be read with your computer at your side. Once you download R and RStudio onto your computer (which we will describe in more detail next), you will be using RStudio to do all of the analysis. You do not need to open R, just RStudio. Almost every table and figure in this book was created with RStudio. There are exceptions, in which case I began with RStudio and edited the figure with Adobe Illustrator. To follow along with this book, you will only need RStudio.

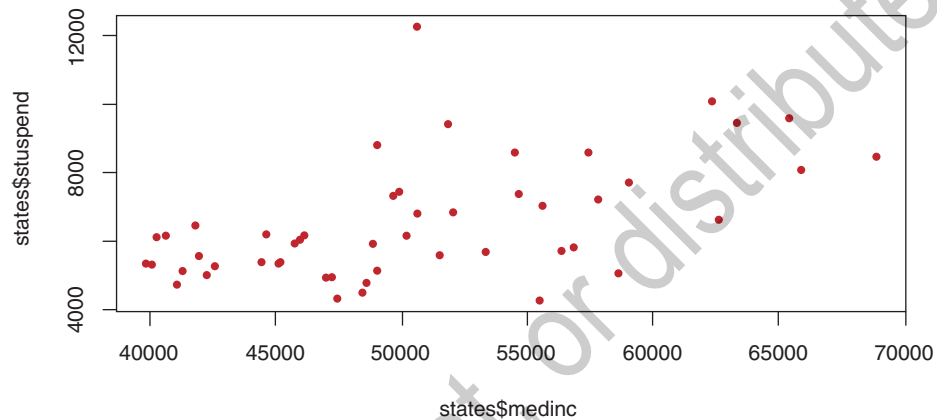
This book is designed so that the examples can be easily replicated. The instructions you type into RStudio (the R code) for each table and figure are presented before the actual table and figure appear in the text so that you can take the code, implement it, and reproduce the same tables and figures. The code will always appear in what will be referred to as a “code chunk” just above the table or figure produced. Just to give an example, the code in Code Chunk 1-1 produces a figure called a scatter plot. I'll go into more detail on scatter plots in this and subsequent chapters. For now, it's important to establish the direct connection between the code chunk and the figure that follows. In Figure 1-1, I plot the relationship between state spending on education per student and the state's median income.

¹ PDF files can only be produced if the appropriate L^AT_EX files are loaded onto your machine. L^AT_EX accomplishes the same thing as R Markdown but is a much more complete and therefore complicated markup language. If you do have the proper L^AT_EX files on your computer, when you generate a PDF file RStudio converts the R Markdown markup language to L^AT_EX. While files generated this way are of exceptional quality since they use the formatting language based on TeX, it can open up a whole new world of syntax that requires mastery. To produce PDF documents, a suite of files must be downloaded. The required software can be found at <https://www.latex-project.org/get/#tex-distributions> and is available for Mac, Windows, and Linux platforms.

Code Chunk 1-1

```
plot(states$medinc, states$stuspend,
      main = "Figure 1-1: Spending is Higher in Wealthier States",
      col = "#bf0000", pch = 20, font.main = 1)
```

■ **FIGURE 1-1** Spending Is Higher in Wealthier States



Art and Practice of Data Visualization

COPY RELEVANT BITS OF CODE

Experienced practitioners rely on several pieces of code that they know produce specific graphs and statistical analysis. Having a file where that code can be found, copied, and edited is highly recommended. RStudio makes that easy. Rarely are entire analyses created by typing code from scratch (there's simply too much to remember and therefore there are too many possibilities for mistakes).

KNOWLEDGE CHECK: Explain the difference between R, RStudio, and R Markdown.

1. Which of the following statements accurately describes R?
 - a. A statistical programming language based on S
 - b. An integrated development environment (IDE)
 - c. A lightweight markup language
 - d. A statistical programming language based on RStudio

2. Which of the following statements accurately describes RStudio?
 - a. A statistical programming language based on S
 - b. An integrated development environment for R
 - c. A lightweight markup language
 - d. A statistical programming language based on R
3. Which of the following statements accurately describes R Markdown?
 - a. A statistical programming language based on RStudio
 - b. A lightweight markup language
 - c. A feature in RStudio that helps replicate results
 - d. A feature in RStudio that helps presents results

Objects and Functions

While this is not a book dedicated solely to learning R, there are a couple of concepts worth mentioning before we embark on our journey. For comprehensive treatments of the R language, I find Kurt Gaubatz's *A Survivor's Guide to R* the most helpful (Gaubatz, 2015). Instead of a full-blown chapter (or two or three) on R, I want to briefly discuss two features of R that I think will help make sense of the instructions you'll be typing or copying into RStudio.

Let's start with functions. In R, the words function and command are interchangeable. **Functions** tell R to take specific actions. They are like the functions you learned in high school math or like the verbs you use every day: *open* the door, *eat* your broccoli, or worse, *finish* your homework. In these examples, the commands or functions are *open*, *eat*, or *finish*. Also note that each function is paired with an object: *door*, *broccoli*, *homework*. In R, functions are denoted by a word directly followed by a set of parentheses that contain an object. R code for the examples above would look like *open(door)*, *eat(broccoli)*, or *finish(homework)*. Since R really doesn't physically open doors, eat broccoli, or finish homework (although one could argue it does open doors and it at least helps finish homework), let's use a more realistic example from data we will be using in the book. Here is an actual example from R:

```
hist(states$hsdiploma)
```

Here *hist* is the function and *states\$hsdiploma* is the object. In this simple example, we're telling R to create a histogram (a simple picture of the data we'll use in our first session). I will provide all of the data we use in this book, which includes information on the 50 U.S. states, information about respondents from a survey, and information collected from countries throughout the world. In this example, the data we will use is the percentage of each U.S. state's adult population that finished high school. In the data, it is called *states\$hsdiploma*. Whenever you see a word before parentheses, you'll know that word is a function. There are lots of functions in R, and you can define your own.

The other basic concept is the **object**. In the everyday examples above, the objects are *door*, *broccoli*, and *homework*. While functions might be considered verbs, objects would be nouns. In the R example above, `states$hsdiploma` is the object on which the function `hist` operates. R is what we call an object-oriented language. That means R is designed to work with objects. In the example above, the object `states$hsdiploma` is a list of numbers. An object can be a list of numbers, several lists of numbers (what we'll call a data set), or a variety of other things. You can define your own objects. R code that defines objects can be recognized by the combination of the *lesser than* sign and the dash ("`<-`"), which essentially means "is equal to." Consider the following example:

```
myobject <- hist (states$hsdiploma)
```

In this example, I am defining an object named *myobject* as a histogram of the measure named `states$hsdiploma`, the percentage of each state's adult population with a high school diploma. The structure of `states$hsdiploma` is simply a list of numbers, one number for each state. To help solidify the concept of functions and objects, let's define a function and an object with some numbers we'll simply type into the computer. Let's start by defining an object:

```
newobject <- c(1,1,1,1,3,4,5,5,5,5,6,6)
```

The code above tells R to create an object called *newobject* and to define it as a list of 12 numbers. Here I decided to make a list of 12 numbers ranging from 1 to 6. The list of numbers has the same structure as `states$hsdiploma`, only we're using numbers we entered ourselves. The list called `states$hsdiploma` contains 50 numbers (one number for each state). The `c` in this instance tells R to combine the numbers into a list. Once we have *newobject* defined as a list of 12 numbers, we can use a function to make a histogram of that list, which is now an object called *myobject*. The command is written as follows:

```
hist(newobject)
```

This barely scratches the surface as an introduction to the object-oriented programming language R. The primary reason we're not dwelling too long on the basics of R is simple: the best way to learn data analysis using R is *to do* data analysis using R. Second, current practice in coding, regardless of the language (R, C++, JavaScript, Python, etc.), involves identifying useful chunks of code (often from a variety of different sites on the Internet), copying them, and using them in your work. Being able to recognize a function and an object will carry you a long way. Rest assured, this book contains all of the chunks of code you'll need to do sophisticated, effective, and insightful work.

There is one more note before we proceed. Most of the code you'll see in R is a function (or command) with a set of parentheses. As I noted, in those parentheses is an object of some sort. Just like our example of a door, broccoli, and homework, the object can be many different things. You'll notice as we proceed that in the parentheses there are often additional instructions we give, always separated by commas. Those extra commands allow us to be more specific. In our example of `open(door)`, we could have `open(door, slowly)` or `open(door, just a crack)`. These commands can get fairly long, but they give you that much more control over how the data are manipulated, providing even more insight.

KNOWLEDGE CHECK: Distinguish between functions and objects.

4. Given the analogy between verbs and functions in R, which of the following makes sense?
 - a. `homework()`
 - b. `finish()`
 - c. `combine()`
 - d. `numbers()`
5. Given the analogy between nouns and objects in R, which of the following makes sense?
 - a. `finish(homework)`
 - b. `numbers(combine)`
 - c. `homework(finish)`
 - d. `combine(numbers)`

Getting Started in RStudio

To follow the material in this book, you need to download four separate things: R, RStudio, and two files from SAGE. Once you've loaded the correct version of R onto your computer (<https://www.r-project.org>), you can download and use RStudio (<https://www.rstudio.com>). Once R and RStudio have been installed on your computer, two files are available from the SAGE server that are required: *Getting_Started_with_R.Rmd* and *Art_and_Practice.RData*. Both files can be found at edge.sagepub.com/brownstats1e. The first file contains material that will help you get started with RStudio and R Markdown. The second file contains the data you'll be using and a few functions I've created.



Art and Practice of Data Visualization

KEEP IT TOGETHER

It's best to create a file folder somewhere on your computer to keep your files together in one place. In terms of best practices, if you keep this folder somewhere in the Cloud (using Dropbox, iCloud, or any equivalent service), your work will always be available from any computer and won't be lost if something happens to your computer.

- Step 1. Open the RStudio application. It should look like Figure 1-2.

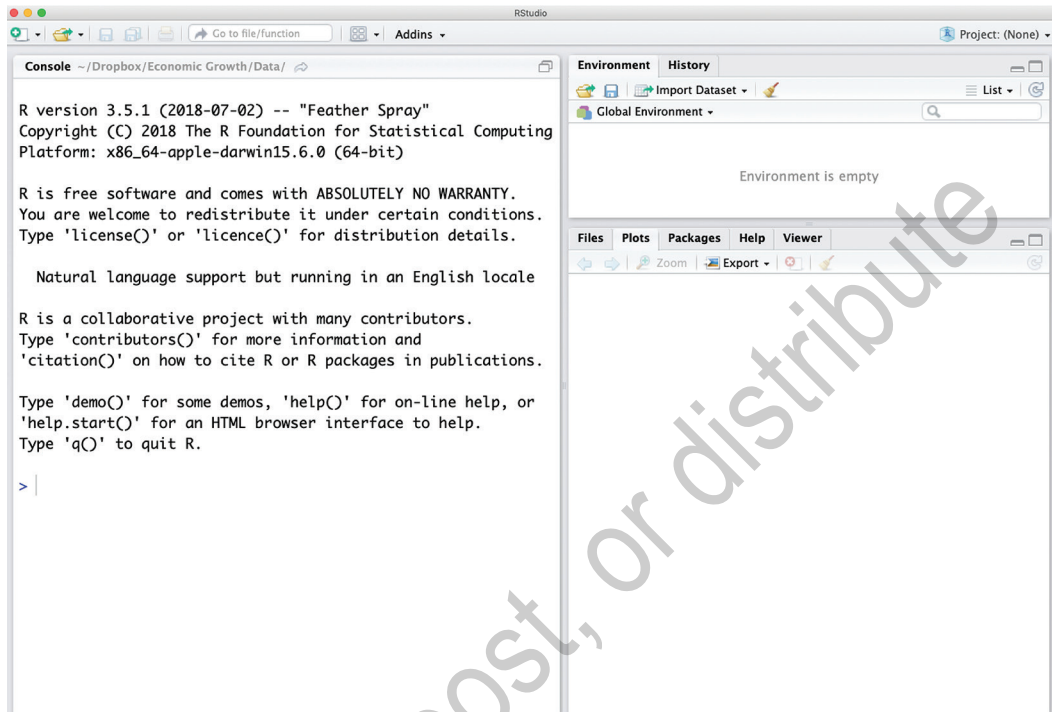
There are a number of ways to work with R. In this book, we will use R Markdown so we can create professional-looking documents in Word, HTML, or $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

- Step 2. Navigate to the upper-left pane and select the yellow folder button .

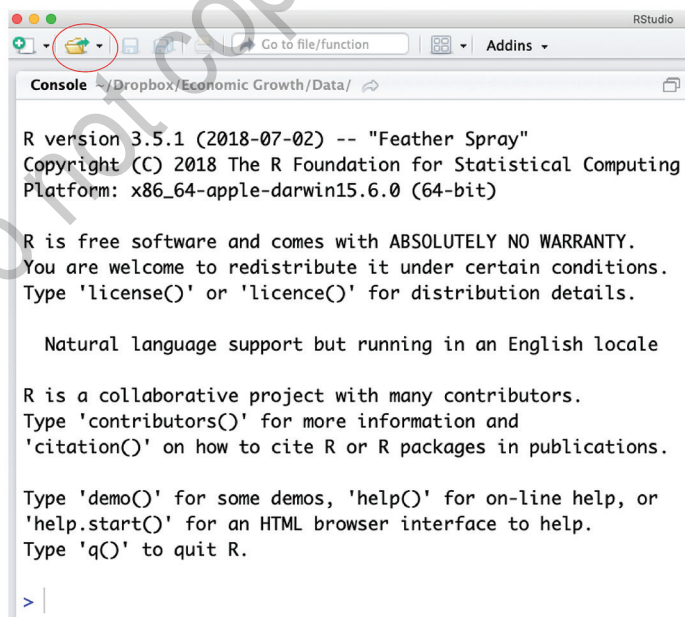
This step is illustrated in Figure 1-2a.

Copyright ©2022 by SAGE Publications, Inc.

■ **FIGURE 1-2** View When RStudio Is Opened




■ **FIGURE 1-2A** Open the R Markdown File



It will open a browsing window where you can navigate to the R Markdown file *Getting_Started_with_R.Rmd*. When you've opened the file, you should see a window pane open up in the upper-left corner of RStudio. Figure 1-3 illustrates what you should see.

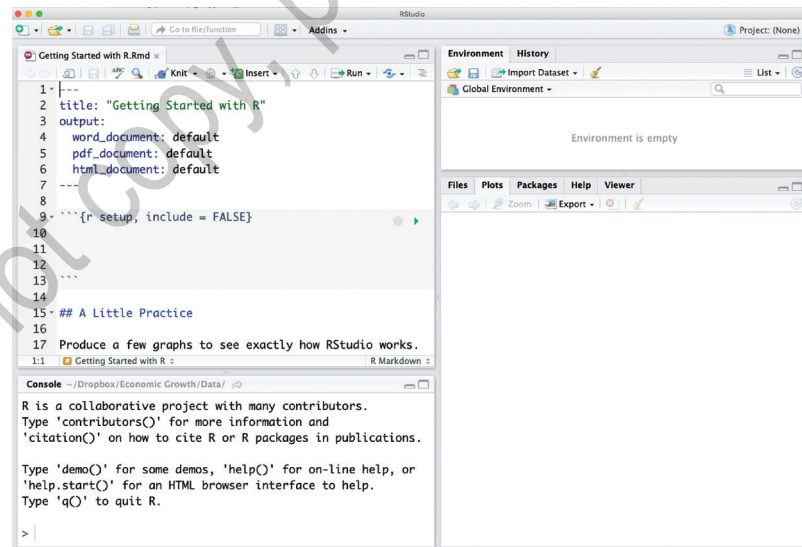
Notice there's a gray area within the R Markdown file. The gray area is called a **code chunk**, which is where we place our instructions (the actual R code). Since this is the first code chunk that appears in the window, I've entitled it *setup* and it should include instructions to load the data and the address of where they reside on your computer.

- Step 3. Navigate to the upper-right pane and click on the yellow folder icon , which will allow you to browse for the data file (Figure 1-4a).² Find *Art_and_Practice.RData* on your computer and then select it. If you complete this step correctly, you should see three data sets in the upper-right pane: *nes*, *states*, and *world* (Figure 1-4a).

Note that when you select the file, the load command with the actual address of that file appears in the lower-left pane (the console pane). It will look similar to what is depicted in Figure 1-4b but will vary depending on where you placed the file on your computer: `load("~/Art_and_Practice.RData")`. You can copy that command starting with the "l" in *load*.

Once you've copied that command, paste it on any line in the code chunk (Figure 1-4c). In Figure 1-4c, I drew a red ellipse that indicates where you can paste the code. That tells R Markdown where your data are so that you can **knit**³ this file into a Word, HTML, or L^AT_EX document.

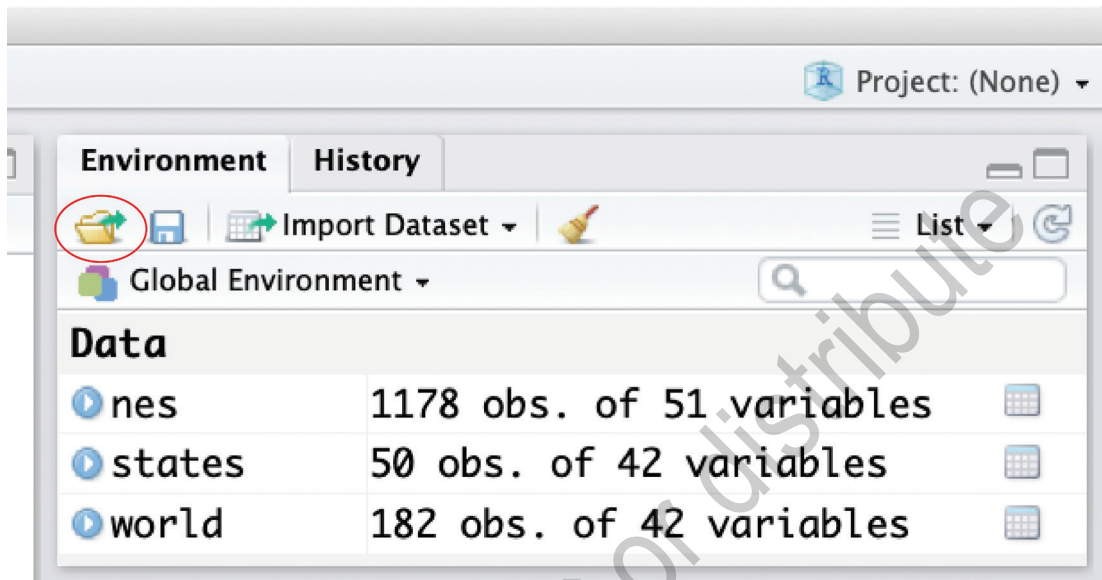
■ **FIGURE 1-3** R Markdown File Appears



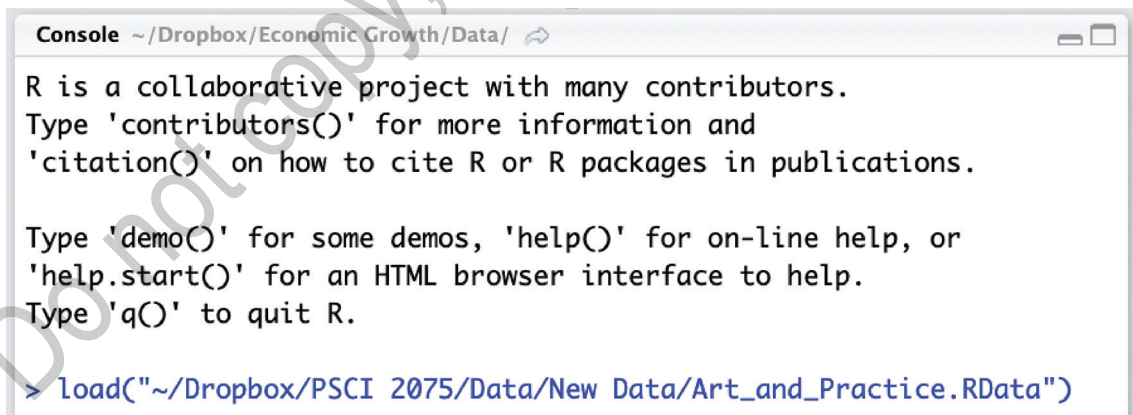
²The data file *Art_and_Practice.RData* includes data but also some functions. By data, I mean the actual numbers you'll need to make graphs, perform statistical tests, and start exploring important subjects (e.g., gun control laws in the United States, democracy across the world). Defining data and data sets more fully is saved for Chapter 2.

³"Knitting" is the term we use when converting the R Markdown file to a document.

■ **FIGURE 1-4A** Opening the Data File



■ **FIGURE 1-4B** Copying the *load* Command



- Step 4. Install the packages you'll need for this book. Execute the function I defined as *installD()* by typing it at the prompt in the console window (see Figure 1-5a). Included in the data file are a few functions written for this book. The function *installD()* is one of them and it installs all of the packages you need.

■ **FIGURE 1-4C** Paste the Command Into the Code Chunk

```

1 ---
2 title: "Getting Started with R"
3 output:
4   html_document: default
5   pdf_document: default
6   word_document: default
7 ---
8
9 ```{r setup, include = FALSE, echo = TRUE}
10
11
12
13
14
15 ...
16
17 ## A Little Practice

```

Once you've typed `installD()` and pressed enter to the prompt in the console pane (lower-left), the computer will start installing the packages. This might take 3–5 minutes or a bit longer depending on the speed of your computer. Once all of the packages are installed, the console pane will return you to the prompt.

- Step 5. Type `libraries()` at the prompt in the console pane (lower-left) and press enter to load all of the packages you just installed into RStudio so that you can use them (see Figure 1-5b). This is a specific command written for this book.

The `libraries()` command is similar to the `library()` command, which you can use to load one package at a time. You'll see some activity in the console as each package is loaded. This procedure usually lasts only a manner of seconds. Once you're back to the prompt in the console, you're ready to go!

To review, the first time you open RStudio, you need to type `installD()` in the console in order to install the packages for the book. The `installD()` command grabs all of the packages you need from the Internet. As you progress beyond this book, installing additional packages will be necessary. Fortunately, the RStudio interface makes installing packages easy. The process involves nothing more than clicking the *Packages* tab in the lower-right pane, clicking the

■ **FIGURE 1-5A** Install the Packages for the Book

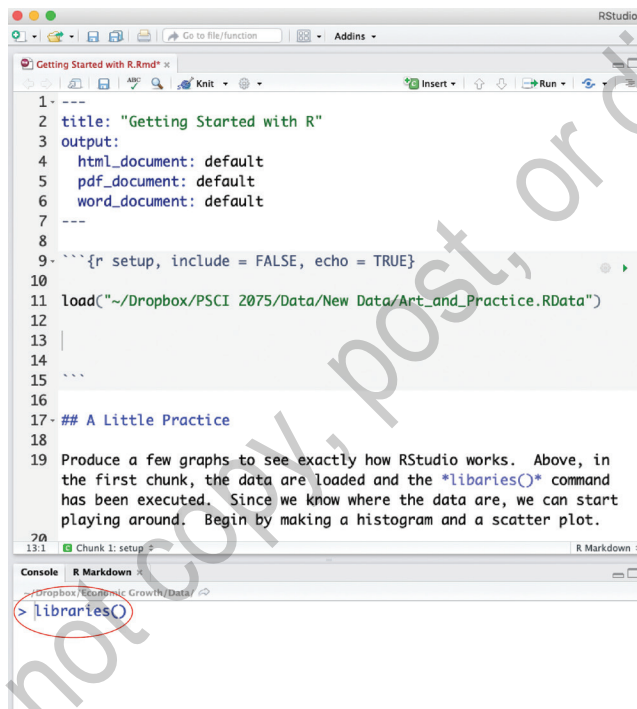


```

Console R Markdown x
~/Dropbox/Economic Growth/Data/
> load("~/Dropbox/PSCI 2075/Data/New Data/Art_and_Practice.RData")
> installD()

```

■ **FIGURE 1-5B** Loading Packages



```

RStudio
Getting Started with R.Rmd x
1 ---
2 title: "Getting Started with R"
3 output:
4   html_document: default
5   pdf_document: default
6   word_document: default
7 ---
8
9 {r setup, include = FALSE, echo = TRUE}
10
11 load("~/Dropbox/PSCI 2075/Data/New Data/Art_and_Practice.RData")
12
13 |
14
15 ...
16
17 ## A Little Practice
18
19 Produce a few graphs to see exactly how RStudio works. Above, in
the first chunk, the data are loaded and the *libraries()*
command has been executed. Since we know where the data are, we can start
playing around. Begin by making a histogram and a scatter plot.
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
25
```

Before we continue, a note about packages is in order. One of R's strengths is that there are thousands of packages available. At the same time, that is one of its weaknesses. With so many packages written by so many different contributors, some packages have functions with the same name. For example, the *select()* function exists in the 'dplyr' package and in the package called *MASS*. Sometimes one package will "mask" another. When you call a function, R will simply run the function with the package last loaded. In order to specify the function you want, sometimes it's necessary to type the package name followed by two colons ("::"). For example, in Chapter 3 you'll see that the *select()* command is preceded by *dplyr::* in order for the function to run properly. In this book, there are just a few code chunks where this happens. In each case, I will draw your attention to those instances.

KNOWLEDGE CHECK: Download and install both R and RStudio.


6. Which of the following accurately describes the *installD()* command?
 - a. It installs and loads R packages.
 - b. It is a base R command that loads all packages.
 - c. It is an author-defined command that loads all packages for this book.
 - d. It is an author-defined command that installs all packages for this book.
7. Which of the following is true about the *libraries()* command?
 - a. It installs and loads R packages.
 - b. It is a base R command that loads all packages.
 - c. It is an author-defined command that loads all packages required for this book.
 - d. It is an author-defined command that installs all packages required for this book.
8. Which of the following statements are true?
 - a. The *installD()* command must be typed in at the beginning of every session.
 - b. The *libraries()* command must be typed in at the beginning of every session.
 - c. The file *Art_and_Practice.RData* contains the data you'll need for this class.
 - d. You should only need to type the *installD()* command once in this class.

Navigating RStudio With R Markdown

There are many different ways to work with RStudio. While this book shows how to work with R Markdown, you can follow along by simply copying or typing in commands to the console pane. I use R Markdown for several reasons. First, it places your code next to the images produced, providing a direct connection between your code and analysis. Consequently, it encourages good note-taking and careful analysis. Second, it helps us and others replicate our work, an important aspect of the scientific process. With the notes and code placed together, others can follow how you arrived at your conclusions. Perhaps even more importantly, it makes it easy for *you* to remember how you arrived at your conclusions. Finally, R Markdown generates professional-looking documents in a variety of formats. How you present your

findings can be just as important as how you generated them in the first place. This section provides a brief introduction on how to use R Markdown in RStudio.

Before we continue, let's get oriented with the basics: the YAML, the code chunk, and how to execute commands in R Markdown. Let's pick up where we left off in the last section. Your screen should resemble Figure 1-6a. The R Markdown file *Getting_Started_with_R.Rmd* is open and we already added the *load()* command that brings in our data file and the *libraries()* command that loads the packages we'll need.

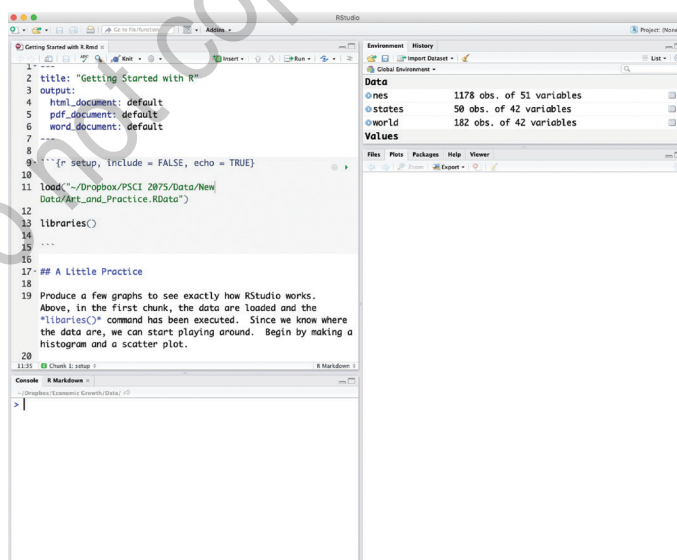
For future reference, to create a new R Markdown file, simply click on the arrow next to the file icon with the green plus sign  and select R Markdown. It is located at the very top left of the RStudio window.

At the beginning of the R Markdown file, demarcated by three horizontal dashes, there are instructions that specify the main title and a list of the documents we can create: PDF, Word, and HTML (see Figure 1-6b). This section is referred to as the **YAML**, which stands for *YAML Ain't Markup Language*. It contains commands that set features for the entire document. When you open a new R Markdown file, it will have basic YAML instructions already included. We're going to keep the YAML simple for now, but if you want to take advantage of additional commands, consult the *R Markdown Cheat Sheet* available at the RStudio website (<https://resources.rstudio.com/rstudio-developed/rmarkdown-2-0>).

In addition to downloading the cheat sheet from the RStudio website, you can access the entire set of cheat sheets by simply going to the RStudio help menu at the very top of the screen. In Figure 1-6c, you can see how to access the various cheat sheets without leaving RStudio.

The more important part of the R Markdown document is the gray area called the *code chunk* (see Figure 1-6d). Code chunks include the actual R code that executes commands. They are created by typing three apostrophes and then a bracketed {r}, which marks the

■ **FIGURE 1-6A** RStudio With R Markdown File



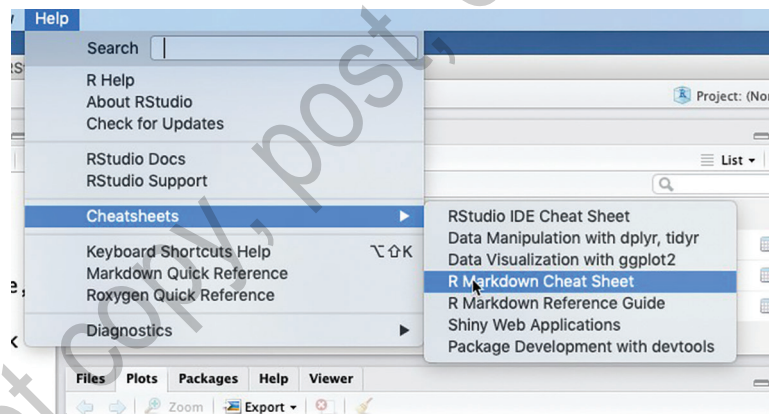
■ **FIGURE 1-6B** YAML

```

1 ---
2 title: "Getting Started with R"
3 output:
4   html_document: default
5   pdf_document: default
6   word_document: default
7 ---

```

■ **FIGURE 1-6C** Accessing the Cheat Sheet for R Markdown



■ **FIGURE 1-6D** The Code Chunk

```

8
9- ```{r setup, include = FALSE, echo = TRUE}
10
11 load("~/Dropbox/PSCI 2075/Data/New
12   Data/Art_and_Practice.RData")
13
14 libraries()
15 ```
16



```

beginning. Three more apostrophes indicate where the code chunk ends. You can also create a code chunk by typing *control+command+i* on a Mac or *control+alt+i* on a Windows machine.

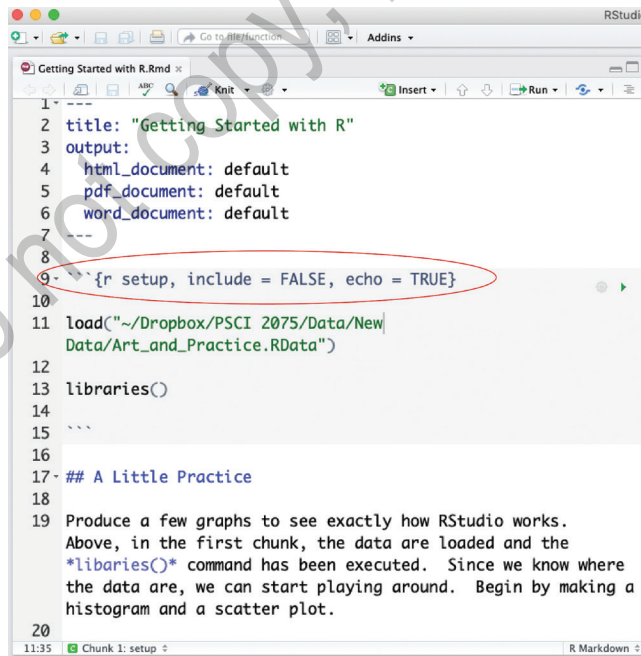
The code located at the top of the code chunk (in between the brackets) tells R Markdown to do several things. First, the *include = FALSE* statement tells R Markdown to not include any extraneous output from that specific code chunk in the document. For example, if the *include* statement is set on *TRUE*, the knitted document will show all of the output produced when the *libraries()* command was executed. We don't want all of that to appear in our document, so we type "FALSE".

Second, the *echo = TRUE* instruction tells R Markdown to include the code you entered in the code chunk, which is a good way to show your work (see Figure 1-6e). In each case, if you change the capitalized TRUE and FALSE, the opposite will happen.

The instructions (the actual R code) you either type or paste are placed inside the gray area (see Figure 1-6f). The command *load()* indicates, for example, what data file you are going to use. For this book, the name of the file is *Art_and_Practice.RData*. Once the command is executed, the data are loaded into RStudio.

Now that we've specified the code chunk options we want and we have the actual commands entered into the code chunk, we can execute them by clicking on the *Run* button  at the top of the upper-left pane or by clicking on the green arrow  in the code chunk (see Figure 1-6g). Again, I've placed red ellipses marking the two locations where you can execute the code. You can also execute each line of code separately by simply placing the cursor anywhere in the line and typing *command+enter* on a Mac or *control+enter* on a Windows machine.

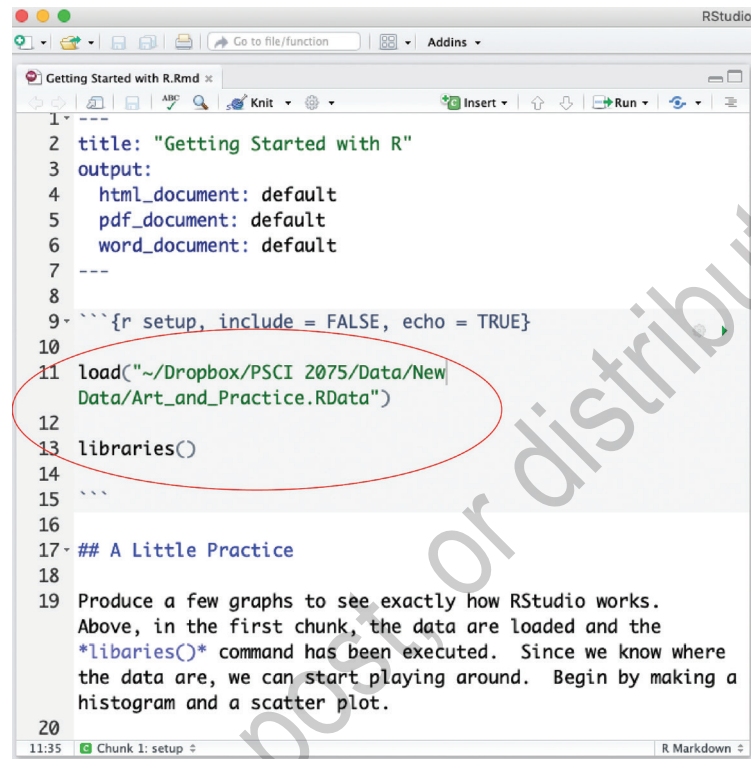
■ **FIGURE 1-6E** Chunk Options



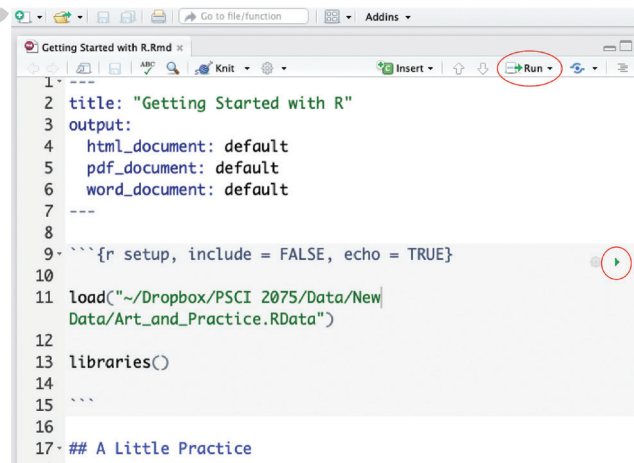
```

1 ---
2 title: "Getting Started with R"
3 output:
4   html_document: default
5   pdf_document: default
6   word_document: default
7 ---
8
9 {r setup, include = FALSE, echo = TRUE}
10
11 load("~/Dropbox/PSCI 2075/Data/New
12   Data/Art_and_Practice.RData")
13
14 libraries()
15
16
17 ## A Little Practice
18
19 Produce a few graphs to see exactly how RStudio works.
20 Above, in the first chunk, the data are loaded and the
21 *libraries()* command has been executed. Since we know where
22 the data are, we can start playing around. Begin by making a
23 histogram and a scatter plot.

```

FIGURE 1-6F The R Code That Is Executed

```
1 ---
2 title: "Getting Started with R"
3 output:
4   html_document: default
5   pdf_document: default
6   word_document: default
7 ---
8
9 ```{r setup, include = FALSE, echo = TRUE}
10
11 load("~/Dropbox/PSCI 2075/Data/New
12   Data/Art_and_Practice.RData")
13
14 libraries()
15
16
17 ## A Little Practice
18
19 Produce a few graphs to see exactly how RStudio works.
20 Above, in the first chunk, the data are loaded and the
21 *libraries()* command has been executed. Since we know where
22 the data are, we can start playing around. Begin by making a
23 histogram and a scatter plot.
24
25
```

FIGURE 1-6G Executing Commands

```
1 ---
2 title: "Getting Started with R"
3 output:
4   html_document: default
5   pdf_document: default
6   word_document: default
7 ---
8
9 ```{r setup, include = FALSE, echo = TRUE}
10
11 load("~/Dropbox/PSCI 2075/Data/New
12   Data/Art_and_Practice.RData")
13
14 libraries()
15
16
17 ## A Little Practice
18
19
```


Finally, anything appearing in the white areas is simply the text you'd like to include in your document (your notes). Type anything you'd like into this space (see Figure 1-6h). R Markdown is a stripped-down version of more elaborate markup languages. This is intentional so that you don't have to spend time learning more code. All of the commands you'll need to produce a professional-looking document are available from the drop-down menu that selects the R Markdown cheat sheet (refer back to Figure 1-6c).

There are a number of ways to work with RStudio. Described below are a set of best practices to get started. Once you are more comfortable with executing commands, analyzing output, and knitting documents, develop your own workflow when analyzing data. Before we proceed, let's review the different panes in the RStudio window.

■ **FIGURE 1-6H** Where to Type Text

```

1 ---
2 title: "Getting Started with R"
3 output:
4   html_document: default
5   pdf_document: default
6   word_document: default
7 ---
8
9 ```{r setup, include = FALSE, echo = TRUE}
10
11 load("~/Dropbox/PSCI 2075/Data/New
12 Data/Art_and_Practice.RData")
13 libraries()
14
15 ```
16
17 ## A Little Practice
18
19 Produce a few graphs to see exactly how RStudio works.
20 Above, in the first chunk, the data are loaded and the
21 *libraries()* command has been executed. Since we know where
22 the data are, we can start playing around. Begin by making a
23 histogram and a scatter plot.


```



Art and Practice of Data Visualization

A MULTITUDE OF OPTIONS

There is no single correct way to operate in RStudio. There are multiple options available to achieve the same estimates, pictures, and ways to present the results. More than anything, a healthy curiosity and sense of humor are required to develop your own effective workflow that originates with a data set and ends with important insights.

- *The upper-left pane.* Most of your work will occur in the upper-left pane of RStudio. This is where you can create R Markdown files that record your work, notes, and observations. It's a good idea to develop the habit of executing commands in this manner, since they are all saved and easy to access.
- *The lower-left pane (the console).* In the console pane, you can observe the commands being executed. You can type commands directly into the console if you don't necessarily need to record them. For example, before running more sophisticated analyses, you may want to see a simple picture of the data before you proceed.⁴
- *The lower-right pane.* The pane in the lower-right corner is used to manage packages and to view the figures and tables you create. If you're using an R Markdown file and you want tables and figures to appear in the lower-right pane, find the gear wheel  in the upper-left pane and select Chunk Output in Console. Once they appear in the lower-right pane, they can be enlarged, saved as files, and exported as individual items.
- *The upper-right pane (the global environment).* This pane shows what data sets are available and what functions you've defined. It helps you keep track of the data and objects you are working with.

Using R Markdown Files Versus R-Scripts

As noted earlier, this book was written in R Markdown. All of the code you'll need to execute is contained in the sections I've labeled as "code chunks" in the book. They proceed each table and figure. Before each code chunk, I provide descriptions of the code to explain what is going on. You'll simply need to become familiar with where that code is, type it into your own R Markdown file, make any necessary adjustments, and execute. To aid in locating the code you'll need, at the end of each chapter is a guide that lists all of the R functions I use in the chapter and in what labeled code chunk they first appear.

R-scripts are files that contain code you want to save. You can't generate nice-looking documents from them, but they serve as more informal ways to save your code for a specific task so that you can retrace your steps. If you don't plan on writing in-depth commentary or producing a document, operating with R-scripts is highly recommended. Note that they don't include code chunks like R Markdown. The entire R-script document can be thought of as one big code chunk in R Markdown. As long as there's no hashtag in front of the command, it can be executed as if it were in the console. As mentioned earlier, this book is structured to follow along using R Markdown rather than R-scripts. Nevertheless, operating RStudio with R-scripts and following this book is easily done if that's what you prefer.

KNOWLEDGE CHECK: Identify the main features of R Markdown.


9. Indicate which statements describe the YAML section of an R Markdown document.
 - a. It provides a place to execute commands in R.
 - b. It contains commands that set features for the entire document.

⁴Note that you can generate the previous command in the console pane by typing the up arrow on your keyboard.

- c. It is where you record the most often-used commands.
 - d. It always appears at the end of the R Markdown document.
10. Which of the following features of a code chunk is true?
- a. It provides a place to execute commands in R.
 - b. Code chunks appear as gray boxes in an R Markdown file.
 - c. The only way to execute a code chunk is by clicking the green arrow.
 - d. You should indicate which data file to use in the first code chunk.
11. Which of the following describes navigation in RStudio?
- a. You can only execute commands in the upper-left pane.
 - b. You can execute commands in the console pane.
 - c. Typing *command+enter* (Mac) or *control+enter* (Windows) executes commands.
 - d. Typing *control+command+i* (Mac) or *control+alt+i* (Windows) generates a code chunk.
12. What is the difference between an R Markdown file and an R-script?
- a. R Markdown files should be used for producing documents for presentation.
 - b. R-scripts should be used for producing documents for presentation.
 - c. R Markdown files should only be used for saving often-repeated or complex combinations of code.
 - d. R-scripts files should only be used for saving often-repeated or complex combinations of code.

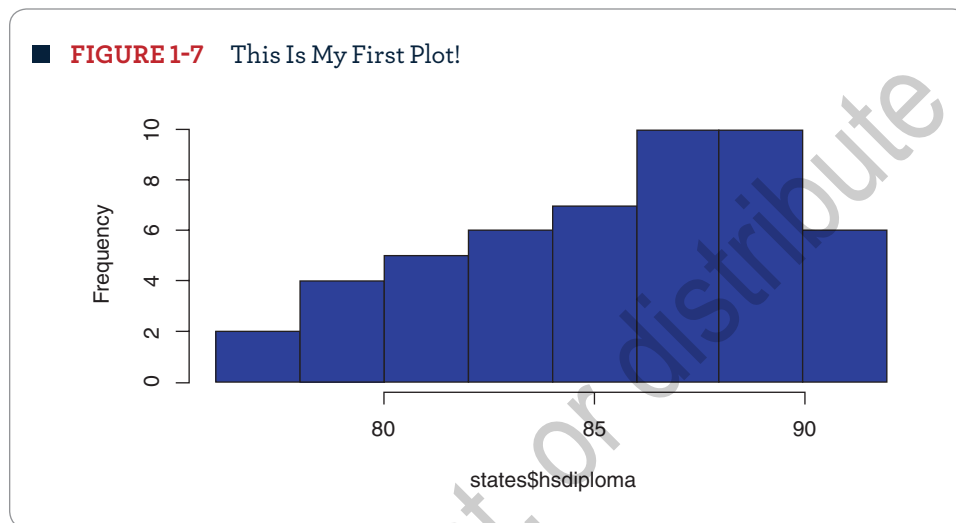
A Little Practice

Let's produce a picture of our data to see how everything works. To review, we loaded the data, executed the *libraries()* command, and opened up the R Markdown file *Getting_Started_with_R.Rmd*. Now we can start playing around. Let's start by making a histogram that reveals the data's distribution. Suppose we're interested in educational attainment. We have a **variable**, a list of numbers, that records the percentage of a state's adult population with a high school degree (*hsdiploma*). A collection of variables is called a **data set**. Stated properly, *hsdiploma* is a variable in the *states* data set. I give a more complete description of variables and data sets in Chapter 2. For now, let's see what the variable called *hsdiploma* actually looks like.

Code Chunk 1-2 draws a histogram, a useful picture that indicates the lowest, highest, and most common values of a variable (Figure 1-7). Code Chunk 1-2 tells R to draw a histogram. Remember, the code can either be typed into an R Markdown code chunk or executed in the console pane. Since we're using *Getting_Started_with_R.Rmd*, the code is included in the code chunk. Execute the code by clicking on the green arrow  found at the upper-right corner of the code chunk to produce Figure 1-7.

Code Chunk 1-2

```
hist(states$hsdiploma, main = "Figure 1-7: This Is My First Plot!",
      font.main = 1, col = "#0000bf")
```



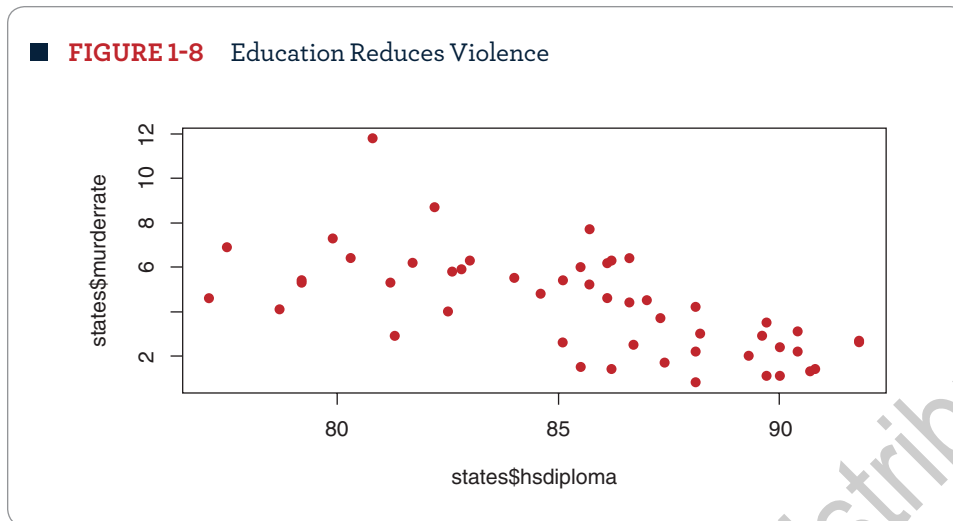
As I explained earlier when introducing functions and objects, all functions in R begin with the name of the function and a set of parentheses where you insert information that the function needs (the object). Note that we use the dollar sign (\$) to separate the data set name from the variable name. By specifying the data set and the variable name, R knows which data set you're using and specifically which variable you'd like to see. In Code Chunk 1-2, we use the function `hist()` to draw the histogram. Specify which variable to use (`states$hsdiploma`) within the parentheses. You can also add a title (`main =`), specify the font for the title (`font.main =`), and indicate a color (`col =`).


By clicking on the right-pointing green arrow  in the upper-right corner of the code chunk, R studio will execute the command and draw the histogram.


Let's draw another picture, a scatter plot of the two variables `murderrate` and `hsdiploma` from the `states` data set. Scatter plots show how two different variables relate to one another. In the following example, we'll look at educational attainment and murder rates (the number of homicides per 100,000 population) in the U.S. states. Note the appearance of `pch =`, which specifies the shape of the dots.

Code Chunk 1-3

```
plot(states$hsdiploma, states$murderrate,
      main = "Figure 1-8: Education Reduces Violence \n",
      font.main = 1, col = "#bf0000", pch = 19)
```




You can change the settings so that your output will appear in the bottom-right pane for closer inspection. To do that, go to the gear icon at the top of this window  and choose *Chunk Output in Console*. Now that we've loaded the data, loaded the packages, and produced a few plots, let's *knit* the document.

Once you've checked to make sure all of the code chunks in the R Markdown file run properly by clicking the green button in each one, choose the small arrow by the *Knit* icon  at the top of this window and choose either *Word* or *HTML*. Once you've done that, you'll see RStudio going through the process of generating a document (you'll see its progress in the console window). If done correctly, you should have a nice-looking HTML or Word document of this brief analysis.

As an additional example, let's make a picture of some data we've created by hand to demonstrate the power of R as an object-oriented programming language. Earlier, we discussed the difference between executing a function and defining an object. We've executed two functions already to create a histogram and a scatter plot.

For clarity, we'll use the example from our discussion on functions and objects. In *Getting Started with R.Rmd*, you'll see a code chunk that defines an object called *myobject* as a list of 12 numbers (the example we used earlier). Once that object is defined, we can put it inside the parentheses for the function to act on it. R knows whether it's the right kind of object or not. If you try to execute a function on an object that is inappropriate, R will object!

To explain why R might object, let's refer back to our example *eat(broccoli)*. R expects some kind of food when you're using the command *eat()*. If you typed in *eat(glass)*, R would spit it out (give you an error message). It knows *glass* is not the right kind of object given the function *eat*.

Since the commands are already included in the code chunk, simply click the green arrow  on the upper-right side of the chunk to execute. You should see a histogram that looks like the one in Figure 1-9. This is the beauty of object-oriented languages: once we've defined an object, there are a multitude of functions we can apply to it.

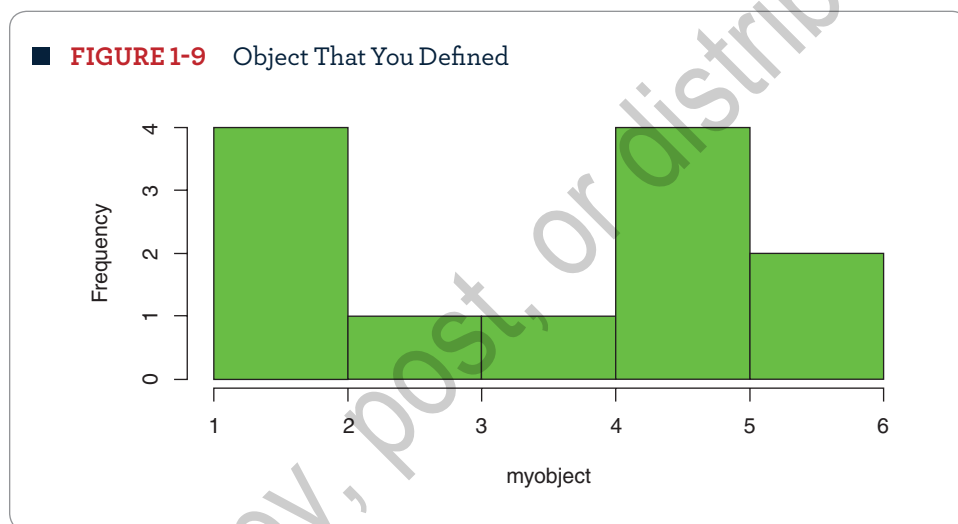
In Code Chunk 1-4, there are two commands. First, an object (*myobject*) is defined. The function *c()* tells R to combine the elements separated by commas into a list (a variable in this

case). Having defined an object called *myobject* that contains 12 numbers, we draw a histogram of that variable. We use the same *hist()* function as in Code Chunk 1-2, only this time we color it green.

Code Chunk 1-4

```
myobject <- c(1,1,1,1,3,4,5,5,5,5,6,6)
```

```
hist(myobject, main = "Figure 1-9: Object that You Defined",
      font.main = 1, col = "#00ff00")
```



As we discussed earlier, the purpose of this chapter is to get you up and running with RStudio. Just copying the code and running it yourself is an important step. But don't stop there. See how small changes to the code change the output. Gaining the facility necessary to be a good data analyst comes with practice. Let's take the previous example and make some modifications.

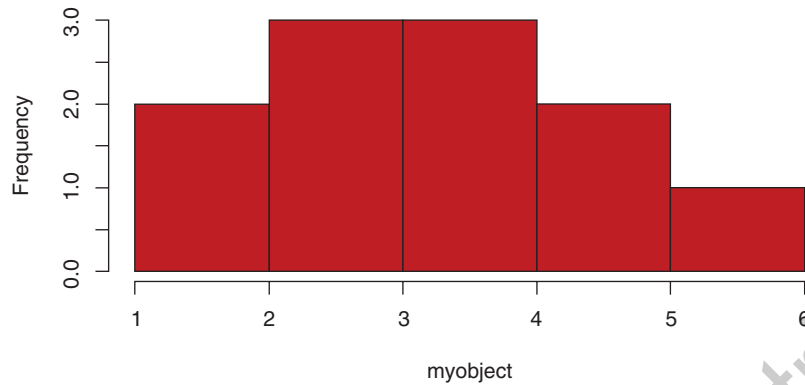
Take the code in the last code chunk and alter it. Just to get you started, I copied the previous code chunk, pasted it into Code Chunk 1-5, and played around with the numbers in the variable *myobject* to change the shape of the histogram. I converted numbers to threes and fours to emphasize the middle of the range. I also took the color option and changed it from green (#00ff00) to red (#bf0000). The result is displayed in Figure 1-10.

Code Chunk 1-5

```
myobject <- c(1,2,3,3,3,4,4,4,5,5,6)
```

```
hist(myobject, main = "Figure 1-10: Object that You Defined",
      font.main = 1, col = "#bf0000")
```

■ **FIGURE 1-10** Object That You Defined



As you'll learn, colors and shapes can easily be overdone, so much so that they distract from understanding what the data actually say. We'll pay more attention to that as we progress through the chapters. For now, to quote famous school teacher, Miss Frizzle:

Take chances. Make mistakes. Get messy!

KNOWLEDGE CHECK: Create a Word or HTML file from R Markdown.

13. Identify which commands will likely produce an error message.
 - a. `play(football)`
 - b. `play(homework)`
 - c. `eat(pizza)`
 - d. `eat(spoon)`
14. What does *knitting* do?
 - a. Executes the R commands
 - b. Generates an HTML document
 - c. Generates a Word document
 - d. Generates a PDF
15. Which of the following statements are true?
 - a. Used within the `hist()` command, `main =` allows you to specify the color of a histogram.
 - b. Used within the `hist()` command, `states$hsdiploma` specifies the variable to be plotted.
 - c. Used within the `hist()` command, `col =` specifies the column number to use.
 - d. Used within the `plot()` command, `pch =` specifies the shape of each point to be plotted.

SUMMARY

This chapter introduced R, RStudio, and R Markdown. You also learned how to load data, load the packages necessary for executing commands, and generate some simple pictures of the data. While this chapter is no substitute for helpful books devoted to learning the R programming language, you now have enough skill to start exploring.

If you play around with RStudio in these beginning chapters, you'll quickly develop the skills necessary to do some amazing work and to make some interesting discoveries. A few short commands and basic knowledge of RStudio change everything.



Art and Practice of Data Visualization

CURIOSITY AND HUMOR

Very much like writing, the way we go about coding and organizing our work reveals a lot about ourselves. Our level of patience will be on full display. One of the strengths and drawbacks of R is that there are a multitude of ways to do the same thing. While there is a lot of online help and advice, it will differ. Occasionally, since much of what we use in R is written by many different people, there are inconsistencies that can make it difficult to troubleshoot. Add to that, learning a new system of syntax can lead to despair. Learning the basics of R is not for the faint-hearted. The more you develop a sense of calm and a sense of humor throughout the enterprise, the faster you'll learn. Getting stuck and developing an effective way to get unstuck can be very rewarding and can be translated from this realm into many others. It's called critical thinking.

COMMON PROBLEMS

- Most of the problems I've seen at this stage include one of two things: (1) the versions of R, RStudio, and the computer's system software are not compatible; or (2) when the *installD()* command is executed, a clean and complete installation of the packages does not occur. So, pay attention to the version numbers of R and RStudio and make sure they are compatible with your system's software.
- Pay attention to what is happening once the *installD()* command is executed. Be on the lookout for packages that don't load or if the process ends abruptly. If the following *libraries()* command does not execute, go back and try the *installD()* command again. Sometimes, it's helpful to reinstall both R and RStudio and start again. Taking the extra 5–10 minutes to reinstall the applications and to install all of the packages properly will prevent problems down the road.
- Once you're up and running, you may find that something isn't working correctly. There are all sorts of sites to find help, but here are some of the main problems I've run across at this stage of the journey:
 - There was a typo in the command.
 - A copy of R from another class already existed on the computer.

- When installing packages, some files got corrupted.
- Not all of the packages from the *installD()* command got installed.
- Compatible versions of R and RStudio were not downloaded.
- The *libraries()* command was not executed when RStudio was opened.
- The *installD()* command was never executed.
- The versions of R and RStudio were not compatible with the computer's system software.
- The *Art_and_Practice.RData* file was not opened in the correct pane.

REVIEW QUESTIONS

1. What is RStudio?
2. What is R Markdown?
3. What does the console pane in RStudio do?
4. What is the difference between an object and a function in R?
5. How are commands executed in R Markdown?
6. When should you use an R-script?
7. What does the keystroke combination *command+enter* achieve? (Mac)
8. What are the advantages in using R Markdown when navigating RStudio?
9. What is *knitting*?
10. What color is #00ff00?

PRACTICE ON ANALYSIS AND VISUALIZATION

1. What is the difference between installing and loading packages?
 - a. Loading packages has to be performed each time RStudio is reopened.
 - b. Installing packages has to be performed each time RStudio is reopened.
 - c. Installing packages occurs after they've been loaded.
 - d. Loading packages occurs after they've been installed.
2. Which statements generally characterize navigating RStudio?
 - a. There is a correct and widely accepted way to navigate RStudio.
 - b. R-scripts are the best way to save your notes and commands as you are performing analysis.
 - c. R Markdown files are the best way to save your notes and commands as you perform your analysis.
 - d. Always execute commands in the console pane.
3. Which of the following would be likely functions?
 - a. eat
 - b. pray
 - c. calculate
 - d. Sally
4. Which of the following would be likely objects?
 - a. eat
 - b. pray
 - c. prayer
 - d. broccoli

5. Which commands make sense in the R context?
 - a. `eat(sandwich)`
 - b. `broccoli(eat)`
 - c. `eat(broccoli, steamed, everyday)`
 - d. `eat(steamed, broccoli, everyday)`
6. What is a code chunk in R Markdown?
 - a. Defines features for the entire R Markdown document
 - b. Contains executable code in R
 - c. R code that is knitted in an R Markdown file
 - d. A great place to write comments on your code
7. Which of the following statements are true?
 - a. RStudio will run without having R open and running.
 - b. R will run without RStudio open and running.
 - c. RStudio will run if R has been installed on your computer.
 - d. RStudio will run only if R is open first.
8. Which of the following statements are true?
 - a. `installD()` installs all of the packages you'll need for the book.
 - b. `libraries()` is an author-defined command that installs the packages you'll need for this book.
 - c. `libraries()` is an author-defined command that loads the packages you'll need for this book.
 - d. you need to execute `installD()` every time you open RStudio.
9. What best describes the activity in the console pane?
 - a. Execution of single commands
 - b. Execution of commands you don't need to save
 - c. Execution of long combinations of code
 - d. A good place to save commands for future use
10. What will you need to navigate RStudio successfully?
 - a. A background in differential calculus
 - b. At least one class in programming
 - c. A previous class in statistics
 - d. A sense of humor

ANNOTATED R FUNCTIONS

The following functions appear in this chapter. They are listed in order of their first appearance (with the code chunk number in parentheses) and annotated here to give a very brief description of their use. Some are not stand-alone functions and only work in combination with other commands. As a reminder, the code in every chapter will work properly if executed in the order it appears. Proper execution also depends on typing the author-defined `libraries()` command, which loads the required R packages.

plot(): basic R command to generate a plot. In this chapter, it is used to produce a scatter plot. (1-1)

hist(): basic R command to generate a histogram. (1-2)

c(): function used to combine elements separated by commas into a list. (1-4)

ANSWERS

KNOWLEDGE CHECK

- | | |
|------------|-------------|
| 1. a | 9. a, d |
| 2. b | 10. b, d |
| 3. b, c, d | 11. b, c, d |
| 4. b, c | 12. b, d |
| 5. a, d | 13. b, d |
| 6. b | 14. b, c, d |
| 7. a, b, d | 15. b, d |
| 8. b, c, d | |

PRACTICE ON ANALYSIS AND VISUALIZATION

- | | |
|------------|------------|
| 1. a, d | 6. b, c |
| 2. c | 7. a, b, c |
| 3. a, b, c | 8. a, c |
| 4. c, d | 9. a, b |
| 5. a, c | 10. d |



Access digital resources, including datasets, at <http://edge.sagepub.com/brownstats1e>.

Do not copy, post, or distribute