

4.12: Getting help

The very last topic I want to mention in this chapter is where to go to find help. Obviously, I've tried to make this book as helpful as possible, but it's not even close to being a comprehensive guide, and there's thousands of things it doesn't cover. So where should you go for help?

4.12.1 Other resources

- The Rseek website (www.rseek.org). One thing that I really find annoying about the R help documentation is that it's hard to search properly. When coupled with the fact that the documentation is dense and highly technical, it's often a better idea to search or ask online for answers to your questions. With that in mind, the Rseek website is great: it's an R specific search engine. I find it really useful, and it's almost always my first port of call when I'm looking around.
- The R-help mailing list (see <http://www.r-project.org/mail.html> for details). This is the official R help mailing list. It can be very helpful, but it's *very* important that you do your homework before posting a question. The list gets a lot of traffic. While the people on the list try as hard as they can to answer questions, they do so for free, and you *really* don't want to know how much money they could charge on an hourly rate if they wanted to apply market rates. In short, they are doing you a favour, so be polite. Don't waste their time asking questions that can be easily answered by a quick search on Rseek (it's rude), make sure your question is clear, and all of the relevant information is included. In short, read the posting guidelines carefully (<http://www.r-project.org/posting-guide.html>), and make use of the `help.request()` function that R provides to check that you're actually doing what you're expected.

Taken together, Chapters 3 and 4 provide enough of a background that you can finally get started doing some statistics! Yes, there's a lot more R concepts that you ought to know (and we'll talk about some of them in Chapters 7 and 8), but I think that we've talked quite enough about programming for the moment. It's time to see how your experience with programming can be used to do some data analysis...

References

Fox, J., and S. Weisberg. 2011. *An R Companion to Applied Regression*. 2nd ed. Los Angeles: Sage.

-
41. Notice that I used `print(keeper)` rather than just typing `keeper`. Later on in the text I'll sometimes use the `print()` function to display things because I think it helps make clear what I'm doing, but in practice people rarely do this.
 42. More precisely, there are 5000 or so packages on CRAN, the Comprehensive R Archive Network.
 43. Basically, the reason is that there are 5000 packages, and probably about 4000 authors of packages, and no-one really knows what all of them do. Keeping the installation separate from the loading minimizes the chances that two packages will interact with each other in a nasty way.
 44. If you're using the command line, you can get the same information by typing `library()` at the command line.
 45. The `logit` function a simple mathematical function that happens not to have been included in the basic R distribution.
 46. Tip for advanced users. You can get R to use the one from the `car` package by using `car::logit()` as your command rather than `logit()`, since the `car::` part tells R explicitly which package to use. See also `:::` if you're especially keen to force R to use functions it otherwise wouldn't, but take care, since `:::` can be dangerous.
 47. It is not very difficult.
 48. This would be especially annoying if you're reading an electronic copy of the book because the text displayed by the `who()` function is searchable, whereas text shown in a screen shot isn't!
 49. Mind you, all that means is that it's been removed from the workspace. If you've got the data saved to file somewhere, then that *file* is perfectly safe.
 50. Well, the partition, technically.
 51. One additional thing worth calling your attention to is the `file.choose()` function. Suppose you want to load a file and you don't quite remember where it is, but would like to browse for it. Typing `file.choose()` at the command line will open a window in which you can browse to find the file; when you click on the file you want, R will print out the full path to that file. This is kind of handy.
 52. Notably those with `.rda`, `.Rd`, `.Rhistory`, `.rdb` and `.rdx` extensions
 53. In a lot of books you'll see the `read.table()` function used for this purpose instead of `read.csv()`. They're more or less identical functions, with the same arguments and everything. They differ only in the default values.
 54. Note that I didn't do this in my earlier example when loading the `.Rdata`

55. A word of warning: what you *don't* want to do is use the “File” menu. If you look in the “File” menu you will see “Save” and “Save As...” options, but they don't save the workspace. Those options are used for dealing with *scripts*, and so they'll produce .R files. We won't get to those until Chapter 8.
56. Or functions. But let's ignore functions for the moment.
57. Actually, I don't think I *ever* use this in practice. I don't know why I bother to talk about it in the book anymore.
58. Taking all the usual caveats that attach to IQ measurement as a given, of course.
59. Or, more precisely, we don't know how to measure it. Arguably, a rock has zero intelligence. But it doesn't make sense to say that the IQ of a rock is 0 in the same way that we can say that the average human has an IQ of 100. And without knowing what the IQ value is that corresponds to a literal absence of any capacity to think, reason or learn, then we really can't multiply or divide IQ scores and expect a meaningful answer.
60. Once again, this is an example of *coercing* a variable from one class to another. I'll talk about coercion in more detail in Section 7.10.
61. Some users might wonder why R even allows the `==` operator for factors. The reason is that sometimes you really do have different factors that have the same levels. For instance, if I was analysing data associated with football games, I might have a factor called `home.team`, and another factor called `winning.team`. In that situation I really should be able to ask if `home.team == winning.team`.
62. Note that, when I write out the formula, R doesn't check to see if the `out` and `pred` variables actually exist: it's only later on when you try to use the formula for something that this happens.
63. For readers with a programming background: what I'm describing is the very basics of how S3 methods work. However, you should be aware that R has two entirely distinct systems for doing object oriented programming, known as S3 and S4. Of the two, S3 is simpler and more informal, whereas S4 supports all the stuff that you might expect of a fully object oriented language. Most of the generics we'll run into in this book use the S3 system, which is convenient for me because I'm still trying to figure out S4.

This page titled [4.12: Getting help](#) is shared under a [CC BY-SA 4.0](#) license and was authored, remixed, and/or curated by [Danielle Navarro](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.