

## R help on the internet:

<http://www.ats.ucla.edu/stat/r/>

<http://data.princeton.edu/R/>

<http://hadley.nz/> (Hadley Wickham is as close as you'll find to a celebrity in R and the statistical programming world. He's written a plurality, if not a majority, of the packages available in R. He has great tutorials and a very helpful Twitter account. He's also written books.)

<http://www.janelawrencesumner.com/teaching.html> (The dropdown menu for teaching, and some links on the page itself, have sample code that I wrote for when I teach more mathematically-intensive courses, but the code itself is annotated and can be useful for learning some basic and some more advanced commands.

## R help on campus:

<http://latis.umn.edu/services-and-programs/research-support/2016-research-workshop-series> (LATIS runs a series of workshops, some of which deal with R.)

## R help books:

- The Monogan book is really valuable and you should have access to supplementary data and files through the library's website. If you can't find it, let me know and I can email it to you.

- I personally taught myself R using the book "Introductory Statistics with R" by Peter Dalgaard. It's slightly out of date now, but should be available in the library.

- Braun and Murdoch's "A First Course in Statistical Programming with R" is very popular and is often assigned to first-year graduate students. (It also took me forever to find the title because I usually just refer to it as "the owl book".)

- Here are two lists of other books that may be helpful: <https://www.r-bloggers.com/r-programming-books-updated/> and <https://www.r-project.org/doc/bib/R-books.html>

Additionally, below is a guide to debugging that I wrote for my graduate students.

-----

## A General Guide to Debugging R and Debugging Yourself

(1) Error messages. While some package authors do better than others, R is not known for its user-friendly and intuitive error messages and warnings. If you get an error you don't understand, the first thing to do is look at the help page for the command you're using. That may provide insight into what's wrong. If that doesn't work, copy and paste "R" and then the message into Google. Odds are good someone has posted a similar question to Stack Overflow and if you read a couple of them, it'll probably tell you what's wrong. (I tend to find Stack Overflow more useful than the other fora, but your mileage may differ.) Failing that, you can always sign up for an account on Stack Overflow and post your own question. If you do so, you should be sure to follow general posting custom, including not introducing yourself and providing a minimum working example (MWE): <http://stackoverflow.com/help/mcve>. (This is actually good advice for asking for help from anyone.)

(2) Debugging yourself. Sometimes the problem is R, but sometimes the problem is improper translation between your thoughts and the code. Something I was taught to do when I was learning R that I thought was really annoying but is actually very helpful (and which I still do regularly) is writing things out in words. This works in (at least) two scenarios:

(A) You can't seem to adapt existing code for your purposes. Here it's a good idea to do two things. First, write out a paragraph or two that details very specifically what you want the code to do. Second, write out a paragraph or two explaining exactly what you think the existing code does. Often in the process of doing this, you'll discover that you don't know exactly what the existing code is doing or what you want it to do, or you'll independently discover the disconnect between your intentions and the code. If you can't, this is a good thing to email me, because I can then identify what the issue is and ask you leading questions until you independently discover it.

(B) Things just don't work like you think they should (or at all). Annotate your code. Every line of it. Either comment every line with a specific explanation of what it does, or write out several paragraphs explaining, step by step, what the code does. This too will often help you discover that you don't exactly understand a line of code or that a line of code you wrote isn't doing what you want it to do. This can be time-consuming but is usually very, very helpful. When I was first learning R, I was required to do this every time I asked a question. I resented it at the time because no one really wants to confront the implication that they don't know what they're doing, but it always helped, and I see the value in it now.