Recomendations for learning R at all levels from the TS community

06 May, 2015

Contents

0.1 Beginners

Starting with R can be challenging, but it is no longer a challenge to find starting guids, tutorials or books. Yet, now, the problem becomes the vast number of choices in "starting guides". This is why we have made a list of options those beginning with R should enjoy. Here are some great introductions to basics of R (besides the intro manuals (http://cran.r-project.org/manuals.html)), including free websites that we would highly recommend and two books (both read and own by TS members):

0.1.1 Web based content

- Website: http://www.statmethods.net/

0.1.2 Books

- Intro Book (same author as the website above): http://www.manning.com/kabacoff/
- Andy Field Book on R (tremendously approachable author that combines the programming with stats): http://www.amazon.com/Discovering-Statistics-Using-Andy-Field/dp/1446200469

0.1.3 Videos

Intro R video. Here's Roger Peng. He's awesome; great at explaining. This was developed for Coursea:

- http://blog.revolutionanalytics.com/2012/12/coursera-videos.html

0.2 Intermediate

0.2.1 Data manipulation

R has built in data reshaping via the reshape function that comes "out of the box" with a base R installation. Trinker did a series of blog lectures using this function here: - http://trinkerrstuff.wordpress.com/2012/05/03/reshape-from-base-explained-part-i/ - http://trinkerrstuff.wordpress.com/2012/05/06/reshape-from-base-explained-part-ii/

If you are going to have to manipulate data often, we would recommend learning the reshape2 package though. Here's some nice tutorials:

- http://tgmstat.wordpress.com/2013/10/31/reshape-and-aggregate-data-with-the-r-package-reshape2/
- http://www.seananderson.ca/2013/10/19/reshape.html
- http://www.slideshare.net/jeffreybreen/reshaping-data-in-r

0.2.2 Graphics

R has three basic plotting platforms: base, lattice and ggplot. All have their various pros and cons. Graphics in base are limitless but can be tedious to code, ggplot2 is the go to guy for quick-and-easy graphics, lattice is something in between. We recommend to really know one graphics system well and have a second as a backup.

Best book that bakes everything in - R Graphics Cookbook (free here: http://it-ebooks.info/book/1316/)

Great intro video: - https://www.youtube.com/watch?v=HeqHMM4ziXA - https://www.youtube.com/watch?v=n8kYa9vu1l8

ggvis is in beta state but here's the package on GitHub and a recent video where Hadley discussed ggvis: - GitHub ggvis: https://github.com/rstudio/ggvis - intro: http://ggvis.rstudio.com/ - video: https://www.youtube.com/watch?feature=player_embedded&v=LOXe6Eu59As

Also worth mentioning is Ramnath's (author of slidify) rcharts package but that requires a bit more knowledge and skill. It also plays well with slidify and shiny:

- Gallery: http://rcharts.io/gallery/
- Home site:: http://rcharts.io/
- http://rcharts.io/howitworks/
- Intro video https://www.youtube.com/watch?v=nYr5KzaR1Vc

0.2.3 R and Reproducable research

R has become a powerful platform for reproducable research with a variety of packages that help you document all stages of manuscript writing.

Then the document management and publishing platforms: github, pandoc/Rwhatever, Rmarkdown, github-blogging.

For document management above all learn knitr. It is the key to reproducible research in R and it works very well with RStudio.

- A good way to learn is by running the demos: http://yihui.name/knitr/demo/minimal/
- This video is an excellent knitr intro: https://www.youtube.com/watch?v=ovdP3AOE5AE

From there learn Rmarkdown to produce html docs (These are Rmd files) here's a link on Rmarkdown: - http://www.rstudio.com/ide/docs/authoring/using_markdown

To make a LaTeX doc you use a Rnw file. Here's an absolute beginner script: - https://github.com/yihui/knitr-examples/blob/master/002-minimal.Rnw

Github is a great place to store and manage anything. It's DropBox on steroids. It's wee suited for code sharing, including projects. It's free if the code is publicly available. It costs for private repos unless you're a student (you get 5 free). GitHub is based on the git language (or program; depends on your view). Dropbox uses git as well. RStudio is also set up to work with GitHub so it makes learning the actual git language unnecessary. With RStudio you only need to know a few commands and that's it. - https://github.com/

There's another git repo management we'd recommend second that has free private repos called bitBucket. Here is a link and also a video made by Trinker on using bitbucket: - Overview: https://www.atlassian.

com/software/bitbucket/overview - The site: https://bitbucket.org/ - video (slightly outdated now): https://www.youtube.com/watch?v=jGeCCxdZsDQ

The blogging aspect or webpage facet of GitHub is discussed here: - https://pages.github.com/

It's more than blogging. It's a full web hosting capability. If you have your own domain name you can even use that and GitHub hosts for free. Here's an example of a simple page where using hitgub to host things:

- http://trinker.github.io/card/contact
- http://marcodvisser.github.io/aprof/Quickstart.html
- http://trinker.github.io/hlm/#1
- https://github.com/trinker/embodied

Pandoc converts between documents awesomely. It's pretty easy to use. There is an R wrapper in the reports package, that makes it easier to use (as does the pander package). But straight from the command line is easy enough. Pandoc's website has a ton of example conversions that work very well. Pandoc isn't a necessary document tool to learn but it's handy indeed. Here's the examples: - http://johnmacfarlane.net/pandoc/demos.html

0.2.4 Presenting, Publishing and Interactive web-applications

We already mentioned knitr in publishing. RStudio really is a terrific way to publish as it links GitHub and knitr + R. You can also throw things up quickly vie Rpubs (http://blog.rstudio.org/2012/06/04/announcing-rpubs/) which are documents made in RStudio that you want to share. You can send all kinds of documents to the net just by clicking a a button. Examples: - http://rpubs.com/trinker

slidify is a terrific way to make html presentations (and even more) but RStudio also has a quick way to make presentations as seen here: - http://www.rstudio.com/ide/docs/presentations/overview

Trinker's take on the two is summed up here: - https://github.com/ramnathv/slidify/issues/278

Now slidify requires some learning and is not well documented yet. It also requires learning some html to fine tune things. Here's Ramnath's opening video for slidify:

- www.youtube.com/embed/I95GOmLc7TA

A good way to learn slidify is by looking at the excellent examples and source code found here:

- Example slides (at bottom): https://github.com/ramnathv/slidifyExamples
- Source code: https://github.com/ramnathv/slidifyExamples/tree/gh-pages/examples

Shiny is a great way to make interactive graphs, for the web, with R. Shiny takes the most experience of anything mentioned thus far: it's a more advanced task. It's powerfull and really nifty but it really intergrates a lot of the other things I've mentioned together: - http://shiny.rstudio.com/

0.2.5 Coding style

Proper coding is vital to good science practice. There are many of types of errors and inefficiency stemming from programming style (Kernighan & Plauger 1978). These bad practices are prolific and have even lead to retraction of publications (see e.g. Merali 2010). Below we supply some links, which may help you learn to code better and more efficiently. This is a small investment that should pay off big in the future, as bad coding practises once learnt are difficult to "unlearn".

Web-based:

Daniel Falster's piece on writing nice R code

The Google R Style Guide

Best practices for programming in Science

References on programming style:

Merali, Z. (2010) Computational science: Error, why scientific programming does not compute. Nature, 467, 775-777.

Kernighan, B.W. & Plauger, P.J. (1978) The Elements of Programming Style. McGraw Hill, New York, 2nd edition.

0.3 Advanced

0.3.1 Coding efficiency in R

A guide on how to program effectively in R, from easy to advanced The R inferno

A quick guide on using profiling to find bottlenecks in your code is given here.

0.3.2 High performance computing (HPC)

The CRAN taskview on HPC

Computing is central to modern science, the following paper provides a gentle introduction to high performance computing - aimed at biologists - but it is suited for any-level R users. A tutorial on HPC for biologist

The paper includes a detailed tutorial, suitable for classroom use, provinding step-by-step explanations of profiling (using aprof), parallel computing and calling C from R is provided in the supplemental material.

1 Contribute

As R, this repro should always be evolving and in a stage of development. Do you have any ideas on how we can improve it? We'll be happy to hear them so please contribute! You can make suggestion here, or just fork the repro and make a pull request!