Package Exploration

Pritam Dalal

What Are Packages?

- "Packages are collections of R functions, data, and compiled code in a well-defined format. The directory where packages are stored is called the library. R comes with a standard set of packages. Others are available for download and installation. Once installed, they have to be loaded into the session to be used." *Quick-R Powered by DataCamp.*

- "A package is a suitable way to organize your own work and, if you want to, share it with others. Typically, a package will include code (not only R code!), documentation for the package and for the functions inside, some tests to check everything works as it should, and data sets." *R Packages: A Beginner's Guide*

Packages on Your Computer or Session

`install.packages("package")` - install package from CRAN.

`library(package)` - load package in current session.

`.libPaths()` - where the packages are stored on your machine.

`library()` - all packages installeed on your machine.

`search()` - list of packages loaded in current session.

Keeping Packages Up to Date

`install.packages()` - single package update.

`update.packages()` - updates all packages.

NOTE: when you're an active R coder, you will install new packages a few times a week (and occasionally in a single day).

Packages in the Wild (Repositories)

A *repository* is a place where packages are located where you can install them. The most popular ones are online for everyone to use.

CRAN - this is the official one that you download from when you use `install.packages()`. It is coordinated by the R foundation, and they have requirements for packages that are listed here.

Github - this is popular because of unlimited space for open source, and for the integration with Git version control. But there is no review process, so this is more of a wild-west. To install from here you use `devtools::install_github()`.

Discovering New Packages

I ususally discover pacakges in the following ways (in order of frequency):

1. I need to do something that I don't know how to do (e.g. rolling standard deviation calculations, multiple plots in a single figure), and I Google around to try to find the answer.

2. In the course of reading about a different package. (But try to avoid the rabbit hole.)

3. Exploring packages related to a particular field or subject area. (I rarely do this.)

CRAN Task Views

- A convenient page on the CRAN website that separates out important packages by their subject matter.
- Google: “cran task views”
- Check out: finance (first and foremost), and if you have time - time series, econometrics, machine learning, econometrics.
- Some of my favorite packages: `bizdays`, `fOptions`, `tidyquant`, `RMariaDB`, `rugarch`.

Documentation

Official PDF: every package on CRAN has a LateX document. It's usually not very good, but sometimes you can get what you want from it.

Vignettes: if you are lucky (you often are not) the package creators have made a vignette which is supposed to be more of a narrative-based introduction/exploration of the package (e.g. `Rcpp`, `tidyquant`).

Tidyverse: many packages for the tidyverse have really good online documentation (yet another reason to stay in the tidyverse if possible).

`help()` - the built-in help function can often answer simple questions about a particular function in a package.

Tutorials: don't be afraid to google around for tutorials, R nerds love to write about R.