Tidbits

Parting Wisdom

Paul E. Johnson[1] Pascal Deboeck[2]

[1]Department of Political Science

[2]Dept. of Psychology, University of Kansas

2013

Don't Forget: R Documents Itself

```
help.start()
```

- launches a browser (if it can find one) that overviews the
  - delivered documents
    - Introduction to R (a book)
    - R FAQ
  - package documents

Package help

- Remember also package "vignettes", pdf "article" documents that are delivered with many packages.
- Every time you run

```
library(some-package-name)
```

- You should also run

```
library(help = "some-package-name")
##or, equivalently
help(package = "some-package-name")
```

Search for functions and packages

- help.search("regression")
  - is OK if you are just looking for installed functions
- RSiteSearch("regression")
  - opens a browser with options on what gets searched
- R package "sos" offers search tools for finding material in CRAN
  - Install that, then run

```
library(sos); library(help = sos); vignette
    ("sos")
```

Email and Forum Support

- Google searches for R material may help, but may also lead to bad advice from people who don't know more than you do.
- Two solutions
  1. Join r-help, or at least read/search its archives
  2. Read Stack Overflow
     - http://stackoverflow.com/questions/tagged/r

Feel the Power of the Source, Luke

- For example, go to `http://rweb.quant.ku.edu/cran`. You can get the R source code (e.g, R-3.0.1.tar.gz)
  - Explore in src/library
- When you are perplexed with a package, download source "tarball"
- They are easily downloaded from CRAN package listing. Look for "Contributed extension packages"
- All packages follow a standard format, with the R code in the R folder, and the documentation and examples are under inst.

Ways to step through the Source code

- Don't be afraid to debug the R source code, or the code for any package
- Easiest way: Run

```
> debug(lm)
```

- Then watch what happens. Try and see!

```
> x <- rnorm(100); y <- rnorm(100)
> m1 <- lm(y ~ x)
```

- Type "n" to step into the code. After than, "n" (or just hit Enter) to step to next command
- Type "c" to continue and finish function
- While in debug mode, run commands to inspect data and variables.

```
ls()
```

- You will see the state of all variables "inside" the function.

[containsverbatim]debug package

- Slightly fancier mtrace function in "debug" package (Mark Bravington)

```
library(debug)
x <- rnorm(100); y <- 0.5* x + 4*rnorm(100)
mtrace(lm)
mod <- lm(y~x)
```

- pops open a "code browser" window
- Hit return in R window to go step-by-step.

How to not Betray yourself as a Stranger in R-land

- Refer to "packages", not libraries. library() is a function that opens packages.
- Use "<-", not "=" for assignments
- Avoid unnecessary "for loops," use automatic "vectorization" instead or (l-s)apply instead

How to ask a question

- We usually become tired and frustrated, and then send out emails like "I can't make XYZ work".
- Generally, those messages are not helpful because we need to know EXACTLY what you tried and we need to know WHAT SYSTEM and WHICH PACKAGES you have.
- r-help has a (longish) "posting guide" (that strains my patience).
- The best advice is this: every time you ask a question, provide:
  - Output from R command "sessionInfo()".
  - a COMPLETE and run-able example of the "problem"
- Informative subject heading (NOT "Need R help, please")

R Bible has Four Books

- VR: William Venables and Brian Ripley, *Modern Applied Statistics with S* (package: MASS)
  - Ripley has been a tireless code contributor and R maintainer. The R authorities expect you have read MASS.
- PB: Pinheiro and Bates, *Mixed Effects Models in S and S-Plus* (packages: nlme, lme4)
  - If you want to talk about "hierarchical models", learn to think of it in the way R folks think of it
- F: John Fox, *Applied Regression Analysis, Linear Models, and Related Methods* (and the *Companion to Applied Regression*) (packages: car, Rcmdr)
- H: Frank Harrell, *Regression Modeling Strategies* (packages, rms Hmisc)
  - Original author of Proc Logistic in SAS

There are Lots of Ways to Estimate the "Same" Model

OLS:	stats::lm, rms::ols
ordinal:	MASS::polr, rms::lrm, ordinal::clm, VGAM::vglm
multinomial:	MASS::multinom, VGAM::vglm

- These are generally compatible, but different as well. If you have trouble with the ones provided with R, then r-help is an appropriate venue to ask.
- If you are using other packages, you should seek out the author or the discussion forums that they want you to follow.

http://pj.freefaculty.org/guides

- Now you know me, you might understand these stat write-ups. Better-than-average writeups on OLS, logistic/probit models, GLM, and Distributions.
- If you use Linux or are trying to do High Performance ("Cluster") Computing, `http://crmda.ku.edu/computing`
- If you are trying to administer Windows with Stat programs, I've got a growing collection of tips as well in the WinStat Admin pages. (Notepad++, Emacs -> R)

Have a Nice Summer

Don't forget to wear sun block.