Introduction to R

George Lesica

May, 2013

What is R?

- Pedantic
  - Open source implementation of S
  - Largely compatible with commercial S-PLUS
- Useful
  - Language and framework for data-oriented computing
  - By statisticians, for statisticians (this is important)
  - Includes a number of convenience libraries and tools

Up and running in three slides

- R Project: `http://www.r-project.org`
- RStudio: `http://www.rstudio.com/`
  - Kick-ass IDE - I use it and I'm a die-hard Vim guy
  - Especially good for exploration / interactive use
  - Can also run in a browser

Installing R

- Linux
  - Debian / Ubuntu: `sudo apt-get install r-base`
  - Archlinux: `sudo pacman -S r`
  - Fedora: `sudo yum install R`
  - Generic: `http://cran.rstudio.com/bin/linux/`
- Mac OS: `http://cran.rstudio.com/bin/macosx/`
- Windows: `http://cran.rstudio.com/bin/windows/`

Installing RStudio

`http://www.rstudio.com/ide/download/desktop`

- Packages available for Debian and Fedora
- Installers available for Windows and Mac OS X
- Tarball install also available

Sample code and data

`https://github.com/glesica/r-talk`

- Build slides: `make slides` (or just use Pandoc)

Sample data credits

- Google Flu Trends: `http://www.google.org/flutrends/us/data.txt`
- Movie ratings `https://github.com/glesica/r-talk/raw/master/data/movies.csv`
- AFINN Corpus `https://github.com/glesica/r-talk/raw/master/data/AFINN-111.txt`

Useful resources

- `http://rseek.org/` - R-centric search engine (because googling for "R" is maddening)
- `http://www.johndcook.com/R_language_for_programmers.html` - introduction to R for programmers
- `http://www.computerworld.com/s/article/9239625/Beginner_s_guide_to_R_Introduction` - just-published introduction to R with tons of links

Let's talk tools - RStudio

- Great IDE for both coding and interactive work
- In-window plotting and plot history
- Built-in documentation browser
- Code completion
- Variety of color themes
- Version control integration
- Some refactoring support (reflow, extract function, jump to definition)

The R language

- C-inspired syntax
- Anonymous functions
- Data-centric data structures
  - Vector
  - List
  - Matrix
  - Data Frame
- Vectorized operations
- Call-by-value (memory!)

R in thirty seconds - variables

- Character (string)
- Integer
- Numeric
- Logical (boolean)
- NA (missing value)
- NULL (nothing)

R in thirty seconds - operators

- Assignment: `<-`
- Comparison: `==`, `!=`, `<`, `>`, `<=`, `>=`
- Logical: `&&`, `&`, `||`, `|`, `!` (short forms are element-wise)
- Math: `+`, `-`, `*`, `/`, `^`
- Special:
  - NA: `is.na()`
  - NULL: `is.null()`
  - Be careful comparing to these weird values

R in thirty seconds - functions

- Declaration: `f <- function() { ... }`
- Calling: `f()`
- Call by value (watch your memory!)
- Functions are first-class so higher-order functions are fine:

```
> f <- function() { function() { 2 }}
> g <- f()
> g()
[1] 2

> h <- function(x) { x() + 2 }
> h(g)
[1] 4
```

R in thirty seconds - flow control

- Conditional: `if (...) { ... } else { ... }` - else clause optional
- Loops:
  - `for (i in ...) { ... }`
  - `while (...) { ... }`
  - `break` and `next` may be used inside loops

Use the vectors, Luke!

Vectors are everywhere!

```
> length('hello world')
[1] 1
> length(c('hello world'))
[1] 1
> length(c('hello', 'world'))
[1] 2
> length(5)
[1] 1
> length(c(5))
[1] 1
```

WTF?

Show me the codez!

A simple demonstration of several aspects of R. Note that there are two types of return, explicit and implicit.

```
power.up <- function(x, p=NULL) {
  if (is.null(p)) {
    return(x ^ 2)
  }
  x ^ p
}

a <- 5
b <- 1:5
c <- c(2,4,5)
```

More advanced example

- Grab the repo at `https://github.com/glesica/r-talk`
- Read the code!
- Source `traffic.r`:

```
source('traffic.r', echo=TRUE)
```