



R Programming Overview

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

R programming content

- Data types
- Subsetting
- Reading and writing data
- Control structures
- Functions
- Scoping
- Vectorized operations
- Dates and times
- Debugging
- Simulation
- Optimization

Reading Lines of a Text File

`readLines` can be useful for reading in lines of webpages

```
## This might take time
con <- url("http://www.jhsph.edu", "r")
x <- readLines(con)
> head(x)
[1] "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">"
[2] ""
[3] "<html>"
[4] "<head>"
[5] "\t<meta http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8"
```

Something's Wrong!

How do you know that something is wrong with your function?

- What was your input? How did you call the function?
- What were you expecting? Output, messages, other results?
- What did you get?
- How does what you get differ from what you were expecting?
- Were your expectations correct in the first place?
- Can you reproduce the problem (exactly)?

lapply

`lapply` takes three arguments: a list `x`, a function (or the name of a function) `FUN`, and other arguments via its `...` argument. If `x` is not a list, it will be coerced to a list using `as.list`.

```
> lapply
function (X, FUN, ...)
{
  FUN <- match.fun(FUN)
  if (!is.vector(X) || is.object(X))
    X <- as.list(X)
  .Internal(lapply(X, FUN))
}
```

The actual looping is done internally in C code.