Data Mining
W4240 Section 001

Yixin Wang

Columbia University, Department of Statistics

September 14, 2015

Intro to R

Outline for today:

- R workspace
- R data types and objects
- basic math
- James 2.3
- scripts vs. console
- loops and vector operations
- functions and packages
- plotting

R Workspace

Open R

A *working directory* is where you are reading data from/writing data to. Let's make a working directory for this class.

A *workspace* is the collection of all the objects you have created. We now have a blank workspace.

R Objects

Types of data in R:

- *numeric*: real numbers (like doubles)
- *integer*: integers
- *character*: holds non-numeric values (like strings)
- *logical*: holds true/false values (like booleans)

R Objects

R allows you to create a different set of objects than most programming languages:

- *vector*: store a one-dimensional, ordered set of objects in same class
- *matrix*: store a two-dimensional, indexed set of objects in same class
- *data frame*: store a two-dimensional, indexed, named set of objects in multiple classes
- *array*: store an *n*-dimensional, indexed set of objects in same class
- *list*: store a one-dimensional ordering of any collection of objects

Concatenation: `c()`, `cbind()`, `rbind()`

R Objects

You can see which objects are in your workspace:

- `ls()` lists all objects
- `rm()` removes an object
- `attach()` attaches a data frame (so columns are now objects under their names)
- `detach()` detaches an object
- `search()` lists all attached objects (and packages)

R Basic Math

Computations in R:

- *Simple algebraic*: $+$, $-$, $*$, $/$
- *Matrix computations*: element-by-element is simple algebraic, matrix is $\% * \%$
- *Exponential/Log*: `exp()`, `log()`

R Basic Math

What's with $< -$? Why not just use $=$ to assign values?

- they do similar things, but have a different scope
- $=$ is for concrete instantiation
- $< -$ can be declared within a function... and exist outside of the function
- (some of you will have learned that such behavior is bad encapsulation)
- ex: `mean(x = 1:10)` vs. `mean(x <- 1:10)`. Does `x` exist in the workspace?

A few more R Warnings

R confuses some programmers:

- R uses $ in a manner analogous to the way other languages use dot.
- R has several one-letter reserved words: `c`, `q`, `s`, `t`, `C`, `D`, `F`, `I`, and `T`.
- (not really, but pretend)
- advice: do not use `T` or `F`. Ever.
- python friends: beware `x[-3]`
- Careful about vectors. Think `C`, not linear algebra
- (try `x*y`; try again with different lengths!)

- Enjoy the bugs.

James Section 2.3

An Introduction to Statistical Learning has a number of "Lab" sections. Let's run through the first one. You should do this with every "Lab" section.

Data can be found at `http://www-bcf.usc.edu/%7Egareth/ISL/data.html`.

R Scripts

The *console* executes a single command right away

Scripts allow you to save a set of commands

- save a set of executable commands
- write a function, which applies an action to a set of inputs
- to run a script, `source("demoscript.R")`
- to make a function available for use:
  1. save latest version of function
  2. run source file for that function

- Let's write a function to calculate a mean

- Let's modify it to exclude data above a particular threshold

Functions and Packages

One of the most useful parts of `R` is the package library

- `R` has lots of built in functions, like `mean()`, `min()`, `max()`, etc
- sometimes you want to do something fancy and `R` does not have a built in function (ex: support vector machines)
- often, there will be a package to do what you want
- a *package* is a library of functions that you can call
- download a package by Packages & Data > Package Installer (install all dependencies!)
- attach a package by `library(package name)`
- then use the functions in the package
- `search()` also displays all attached packages

Plotting

Plotting in `R` works by layers:

- `plot()` plots the inputs on a new plot
  - `type` controls type of plotting ("p", "l", "o", etc)
  - `pch` controls point symbols
  - `lty` controls line type
  - `col` controls color
  - `lwd` controls line width
  - `cex` controls point size
- `points()` adds a set of points to your plot
- `lines()` adds a set of lines
- `hist()` creates a new histogram

Homework 1

Homework 1 is designed to be an intro to `R` and the eigenfaces mini-project. Let's get started.

When in doubt, `www.google.com`