Research Tools: R

Data Structures, Stats, Plotting

James MacGlashan

What is R?

- Command line statistical package
- Based off the 'S' programming language
- Free!
- Built-in functions for typical statistics and plotting
- You can write your own functions
- Works on Windows/Unix/Linux/Mac OS
- http://www.r-project.org/

Basics

- Launches with a simple window
- Can be used like a basic calculator
- Command History
- Command auto complete
- Weakly typed variables set with '='
- Create vectors: c(x1, x2, x3)
  - index base 1: x[1], x[2], ...
- Create lists: list(e1, e2, e3)
  - or: list(name1=e1, name2=e2, name3=e3)
  - x[[1]], x$name
- Help pages for more info: help(function)

Vector Manipulation

- Create quick series: start:end ; end:start
- Length: length(vec)
- Element-wise arithmetic: x + y ; x * y ; ...
- Scalar to vector arithmetic: 5 * x ; 5 + x; ...
- Selective Indexing: x[c(3, 5, 7)] ; x[c(-1, -2)]
- Inner Product: x %*% y

Matrices

- Can also construct matrix:
  - matrix(elements, nrow=c, ncol=k)
  - matrix(elements, nrow=c, ncol=k, byrow=TRUE)
  - cbind(col1, col2, ...)
- Index: x[r,c]
- Extract row/column: x[r,] ; x[,c]
- Matrix Multiplication: x %*% y
- Crossproduct: crossprod(x, y)
- Transpose: t(x)
- Determinant det(x)

Data Frames

- Matrix of different types (specialized list)
- Usual data type when reading a file
- Set your root workspace directory with "change working directory"
  - relative paths will be from here
- read.table(path, header=TRUE, sep=",")
- attach(frame) import frames variables to current scope

Basic Statistics

- Random: rnorm(n) ; rexp(n) ; runif(n) ; rpois(n)
  - n is number of observations, can also set parameters like “mean”
- mean(x)
- std(x)
- var(x)
- summary(x)
- colMeans(matrix)
- rowMeans(matrix)

More Statistics

- Noisy line: runif(20, 0.8, 1.2) * 1:20
- Correlation: cor(x, y)
- T-Tests: t.test(x, mu=mean) ; t.test(x, y, ...)
  - see help page for tons of info
- Anova: aov(formula, dataFrame)
  - predict~groups
  - a = aov(Weight~Type)
  - summary(a)
- Linear Regression: lm(formula, dataFrame)
  - fit1 = (Mileage~Weight+Disp.)
  - summary(fit1)

Simple plotting

- plot(x, ...) or plot(x, y, ...)
  - type - what kind of plot?
  - col - stroke color
  - lwd - line width
  - lty - line type
  - pch - 1-25, and characters
  - bg - background color of open symbols

Annotating a Graph

- From plot() or title():
  - main - title of the plot
  - sub - sub title
  - xlab/ylab - title for axes
  - cex - text size
    - cex.axis
    - cex.main
    - cex.sub

Multiple Figures

- Start with plot of first figure
- Add figures with:
  - lines()
  - points()
- Make sure plot has right range first
  - set xlim and/or ylim using range
    - range(x1, x2) - returns range of values
- Alternative: matplot
  - matplot(matrixX, matrixY, ...)

Legend

- legend(x, y, legend, fill, ...)
  - x/y - position
  - may set x to things like “top”, “left”, “topleft”, ...
  - legend - vector of strings for figure names
  - fill - color of legend box (default transparent)
  - typical par parameters like lty (but vector to define each figure)

Error bars

- Some packages will have better functions for this
- Manual way requires:
  - a vector of the mean values and indices,
  - a vector of lower confidence bounds
  - a vector of upper confidence bounds
- Plot graph of mean values, then use “arrows”
  - plot(x, y, type=”o”)
  - arrows(x, upper, x, lower, code=3, angle=90, length=0.1)

Rendering to Vector PDF

- Vector PDFs are small in file size and scale exceptionally well (great for papers)
- Same plotting commands as usual, with an additional command before and after
  - pdf("pathToFile.pdf")
  - ----plotting commands----
  - dev.off()
- Note that path is relative to working directory (or absolute if you use an absolute path)

Special Plotting

- The plot command can take special objects
  - Try a fitted model from lm
- ablines adds a complete line across a plot
  - use to show how well fit a regression is
    - fit = lm(Mileage~Weight)
    - plot(Mileage)
    - abline(fit)
- Automatic Histograms
  - hist(data, ...)
- Automatic Boxplots
  - boxplot(data, ...)

Control structures

- General looping control exists such:
  - while(x < 5){ }
  - for(e in vector){}
    - for(i in 1:n){}
- If commands
  - if(x == y){
    statements
    }
    else if(x == z){
    statements
    }
    else{
    statements
    }
  - typical C-like logical operators (e.g., &&, ||)

Functions

- All functions created with “function” keyword
- Functions are named by assigning a value to it
- addThree = function(x){ return(x+3) }

```
> addThree(5)
[1] 8
```