CVEN 5454 Intro. R-Session Robert Cruickshank & William Raseman 14 Feb 2017

Motivation. Assist students in:

1. Efficient processing of homework problems,
2. Creating a powerful and well documented R-project with elegant solutions, and
3. "Turning ideas into software quickly and faithfully"[1]

Learning objectives. Students will be able to:

1. Understand some of the needs addressed by R
2. Open and exit both R and R-studio
3. Create simple programs to:
    a. Read and write files (local and web)
    b. Manipulate data: vectors and matrices (scalars don't exist!)
    c. Document results with text, figures and tables
4. Share origin of R and predecessor language S S v1. 1976 Fortran®, S v2. 1980 Unix®

Teaching Points:

1. Input/Output (I/O): Human→Machine, Machine→Machine, Functions, Lego-like qualities of nested commands
2. Invoking arithmetic operations: +, -, mean…, on vectors and matrices. Beware recycling effects!
3. Notation for selecting: lists, elements, columns, rows…
4. Visualizations and typesetting
5. Documenting results: Knit and R-Markdown

Review from Last Class R-Session 1: http://civil.colorado.edu/~balajir/CVEN5454/R-sessions/sess1/

1. Data Reading
2. Compute basic statistics - including robust measures
3. Line and boxplots - including multiple plots
4. Source Functions

Learning Activities:

1. R Reference Card by Tom Short on CRAN
2. Getting help
3. Input / Output
4. Data creation: c("foo"), cbind(), rbind()
5. Slicing & extraction: [,] [[ ]], [-1] and x$name
6. Variable conversion: as.data.frame()…
7. Variable information: length(), dim(), ncols(), nrows(), testing with is.na() and is.null() …
8. Data selection and manipulation: which.min(), which.max(), sort(), rev(sort()), sample(), unique(), and order()[2] which list elements in ascending order.
9. Math: =, <-, +, - , rank()…
10. Matrices: t(), rowsum(), colmean()…
11. Advanced data processing: lapply(X,FUN=)…
12. Strings: grep(pattern,X)…
13. Dates & times
14. Plotting & typesetting examples
15. Graphical parameters: par()
16. Lattice (Trellis) graphics
17. Optimization & model fitting
18. Statistics
19. Distributions
20. Programming: Functions!
21. Nesting commands and using unlist()[2]
22. Install.packages("foo")[2], library(foo)[2], Beware of package commands masking other commands.
23. qgplot()[2] ← beyond our scope, but check it out!

A way to think about what R can do. What would you need to provide telephone service?

1. *Phonebook with list of names* (which implies a file system and I/O), ability to sort, typesetting, drawing map of Area Codes and ZIP Codes.
2. *Directory Assistance* with powerful search and match capabilities.
3. *Route telephone calls* including processing of dialed digits, 7-digit local vs. 10-digit long distance, check if first digit is a 1 and if so process the next three digits as an area code
4. *Provide enough capacity* for Mother's Day☺

Historical Context Question: Why might R and Unix commands often be so cryptically short, for example, the letter "t" for transpose? Clue: ASR-33

Where to go for additional help?

1. Comprehensive R Archive Network CRAN
2. Quick-R http://www.statmethods.net/
3. Stack Exchange http://stackexchange.com/
   If unable to find answer after carefully searching, *join the Stack Exchange Community* and follow simple rules to post a question. You'll have the 1st answer in tens of minutes or less, and peer reviews of the answer and additional answers shortly thereafter☺

[1] *John Chambers: Original motivation of R-Developers☺*

[2] *Command not included in Short's R Reference Card*