HS616 Syllabus

Robert Horton

October 15, 2014

Statistical Computing for Biomedical Data Analytics

Description

This is an intensive introduction to statistical computing with R. Programming assignments will draw from a wide range of computational and applied mathematical concepts required for biomedical data analytics, including probability, statistics, linear algebra, optimization, data manipulation, visualization, linear modeling and model diagnostics.

Overview

This course is designed to develop core computing skills, to complement traditional coursework in mathematical statistics, and to lay the groundwork for further study of data analytics. Students will learn to use the R statistical programming language and environment by solving structured problems in a wide range of application areas relevant to biomedical data analysis.

The programming exercises will emphasize interaction between R and other computational resources, particularly SQL databases, networked resources, the Linux environment, and Python. Special emphasis will be placed on approaches that scale to large data sets.

Objectives

On completing this course, students will be able to:

- take advantage of the large number of software modules available from the R language
- clean, merge, filter, and arrange data into formats suitable for regression or machine learning analysis
- create sophisticated graphs of functions or data
- use linear algebra approaches to find optimal solutions for systems of equations
- be able to run advanced SQL queries (including GIS extensions, stored procedures, and in-database analytics) from the R environment
- develop and deploy interactive visualizations and simple data products
- use scripting and literate analysis tools to create fully documented and reproducible data analysis workflows

Recommended texts

- R in Action (2nd edition): *This is the primary text* for the course. It presents a broad and readable introduction to using the R environment for data analysis, but it does not focus on programming. Note that the second edition is officially coming out this spring, but you can purchase an "early access" subscription that lets you read the latest revisions as they are ready. You can often find discount codes for books from this publisher (look on the Quick R site, for example)
- Introductory Statistics with R: I'm looking at this as an alternative; the first chapter is recommended by Hadley Wickham before you dive into ggplot.

- An Introduction to R: The "sample session" in Appendix A is a useful tutorial; otherwise, this book is pretty technical, and mostly good for reference.
- Advanced R: also available online. This is a new text written by Hadley Wickham, the author of many of the packages we will focus on (ggplot2, tidyr, dplyr, etc). It is not an introductory book, but we will use selected sections.

Websites

- Interactive Tutorials
  - swirl: This is an R package that walks you through various tutorials that you run within R.
- Questions
  - RSeek: interface to the Google search engine that focuses on R; this can be useful since the letter "R" is used for other things.
  - StackOverflow
- Reference
  - Reference Card
  - Quick R
  - Cookbook for R
  - About R on StackOverflow
- Code
  - CRAN The Comprehensive R Archive Network is the main repository for R packages.
  - Bioconductor is a separate repository for bioinformatics packages
  - GitHub, especially the course repository and Hadley Wickham's site

Syllabus

R is both a programming language and an environment for statistical analysis. It has well-developed facilities for software development, data manipulation, data visualization, and (of course) statistics; the power of the system is the ability to bring all of these capabilites together. Unfortunately, students can face a daunting challenge if they need to learn all of these aspects before they can reap the benefits of R. The answer, of course, is to break the learning task into small iterations; learn something about programming, something about data management, some graphical technique, and some statistcal approach, then use them to address a problem. Repeat until you have learned all the aspects (this may take a while). The textbook is structured around several such iterations. This course, on the other hand, is laid out in one big iteration, with four sections emphasizing programming, data manipulation, graphics, and analysis. The division is not crisp, however, since most interesting results require a combination of these things. We will cover a variety of interesting and important ideas, applications and examples in class, but you will need to take the initiative to develop basic R skills on your own.

Section 1: R Programming

- Week 1: Teach yourself R: ***We cannot cover everything you need to know in class***
  - R == programming + data + graphics + statistics
    - synergy: the different fields complement each other
    - iterate: you can't learn it all at once
  - Resources

    - Textbook: *R in Action*
      - the book presents multiple cycles: Getting Started, Beginning, Intermediate, Advanced
      - this course is (mostly) organized in one big cycle
    - Electronic:
      - tutorials (swirl)
      - help system and reference sites
      - code repositories and installing packages
      - Coursera
  - Literate calculation
    - Introduction to RStudio and “Literate Analysis”
    - R as a calculator
    - typesetting mathematical equations in LaTeX
    - Exercise: Documenting calculations in RStudio
  - Idioms matter
    - How to baffle R: the “loop that allocates memory” ploy
    - Hello functional programming
    - Simple performance metrics: plotting timing results > vectorization
  - Set up github accounts
- Week 2: Probability
  - Pseudorandom numbers
    - deterministic “randomness”
    - the *Randumb* package: installing from a zip file
  - Rolling dice
    - simulating the Binomial distribution
  - Curve fitting
    - using lm and formulas to fit a polynomial to a set of points
    - parameter scanning to fit bell curve to Binomial simulation results
  - Sensitivity and specificity exercise
    - structuring code with functions
    - organizing data into dataframes
  - Bootstrapping
    - estimating probability distributions by sampling
    - parallelization
    - profiling with lineprof: installing from github
- Week 3: Calculus and linear algebra review
  - Checking derivatives
    - functions as parameters
  - Numerical integration
    - area under (part of) a bell curve
  - Matrix manipulation
    - dot product (“weighted combination”) of two vectors
    - multiplying a matrix by a vector
    - Gauss-Jordan elimination
    - solving systems of equations
- Week 4: Hypothesis testing

    - Distributions under null and alternative hypotheses; alpha, beta
        - Interactive graphs using “manipulate”
    - Multiple testing adjustments
        - Bonferroni, Benjamini-Hochberg

Section 2: Loading and Munging Data

- Week 5: Loading and cleaning data
    - Spreadsheets and data frames
    - data cleansing
        - Regular expressions
        - Un*x pipes
    - merging and aggregation
    - key-value lookups
        - linear scanning, binary search, hash tables
- Week 6: Using SQL from R
    - sqldf: bidirectional promoters in a mammalian genome
        - self-join on a table of transcription start sites.
    - Direct database access
    - Sophisticated SQL with Postgres
- Week 7: Tidy Data: rows are examples (with outcome), columns are attributes
    - Classification example
    - dplyr & tidyr
        - swirl exercise
- Week 8: Dimension reduction
    - **Midterm Exam**
    - Singular Value Decomposition

Section 3: Visualization and User Interaction

- Week 9: ggplot2
    - graphics menagerie
        - continuous / categorical
        - univariate / multivariate
- Week 10: Shiny
    - Interactive ODE model
    - Interactive analysis
- Week 11: Clustering and Graphs
    - Heat maps
    - Graphing graphs
- Week 12: Visualizing Regression
    - regression as an optimization problem
    - interactive visualization of line, data, and cost function in parameter space.

Section 4: Analytics

- Week 13: Linear models
  - Feature selection
  - Performance profiling revisited
  - Generalized linear models
  - Regularization
- Week 14: Interactions
  - Simpson's paradox
  - Categorical inputs
  - Dimension reduction
- Week 15: Diagnostics
  - Model diagnostics
  - Prediction diagnostics
    - ROC curves: sensitivity/specificity tradeoff
    - Learning curves: bias/variance tradeoff
  - Hypersonic survey of Machine Learning algorithms
- Week 16: Finals Week
  - **Final Exam**
  - Project presentations