STA 141 Syllabus

This class is about scientific and statistical computing. It is intended to provide you with a strong foundation in computing skills that are increasingly necessary for a practicing statistician and scientists generally.

The main topics that we will learn about during the class are

- The R environment and programming language

    - Basics of R, data types & structures,
    - graphics - standard and lattice, formulae
    - vectorized operations
    - control flow, writing functions, debugging, testing.
    - Efficient algorithms & code, optimization & profiling.

    We will explore these in the context of exploratory data analysis, simulation, sampling.
    This topic will take 4-5 weeks of the course and be the primary focus. We will cover the basics to "get things done" and then how to think and reason about the language to gain a more comprehensive understanding of how to program.
- Data manipulation

    - Basic input techniques for rectangular data,
    - for non-rectangular data,

- Text manipulation & Regular Expressions
    Working with text data.
- Shell tools and programming and working with other languages.
- Web-related computing, Web services, **XML**
    Accessing data via the Web: Scraping data from HTML pages, HTML forms, REST services, SOAP, parsing XML.
    Creating graphics for the web, e.g., Google Earth, SVG animations, ...
- [Optional] Relational database management systems (RDBMS)
    concepts of databases, relational model, **structured query language (SQL)** and accessing databases from R.

See detailed topics for more information.

We will encounter these topics in the context of exploring real data. Much of the work will involve manipulating and exploring data and making sense of it through summaries and creative graphical displays. We will also use the computer and programming to perform simulations of stochastic processes. We will also use some statistical modeling, covering statistical methods that you may not have seen in other classes (e.g. k-nearest neighbors, cross validation, bootstrapping). We will cover these heuristically rather than with formal theory. So you will learn the computing topics by using them in actual settings.

The primary goals of the class are

- for you to become competent high-level programmers so that you can approach data analysis and simulation problems confidently and sensibly;
- for you to become aware of important available technologies and gain an understanding of how they work;
- for you you learn how to find out about new computing topics e.g. R functions, packages, new technologies.
- be able to explore, make sense of and summarize data and understand the importance of this aspect of statistics (rather than simply applying statistical methods to numbers)

Grading

- 85% assignments (6 or 7 made up of 2-3 short homeworks and 4 longer assignments)
- 15% Class participation
  This includes asking and answering questions in class, on the class forum, in office hours and generally being engaged.

Policies

1. You can discuss approaches to problems with other students.
2. You cannot copy code from other students.
3. You can look for hints, code and solutions on the Web, but you must acknowledge them in your writeups.
4. Reports:
   - You are to hand in printed reports to Gabe Becker and send an archive of the writeup and all the relevant code to dtemplelang@ucdavis.edu.
   - The reports should describe the context of the problem and your approach to it and provide details about the more difficult and non-basic elements of the programming involved in your work. You should discuss and contrast alternative approaches, even if you have not implemented them.
   - For data analysis problems, you are to write your answer as if for a scientist or journalist who is familiar with the basic ideas. The focus is on the discoveries or confirmation of conjectures and hypotheses, not the programming. However, do point out interesting programming tasks or things that you learned about important functions (e.g. lattice and legends). So your writeup

  - should include both computational issues and commentary and analysis of the primary problem/context.
  - The data analysis problems are intended to focus on exploratory data analysis and understanding the data. At times, we will fit statistical models. For the rest of the time, use common sense and find interesting aspects of the data based on the context of the data. Do not apply "arbitrary" statistical methods to data just for the sake of it!

Duncan Temple Lang
<duncan@wald.ucdavis.edu>
Last modified: Thu Sep 27 18:39:46 PDT 2012

Topics

The following is a list of the topics we will most likely cover, given time constraints and different interests of students. The order is approximate and we will tend to introduce topics to make you aware of them and then revisit them in greater detail.

- Getting started with R
    - Running R,
    - finding functions and objects,
    - issuing commands, getting help pages,
    - generating random data,
    - creating graphics, different graphics
- The R Language
    - basic data structures - vectors (integer, numeric, logical, character) and lists.
    - matrices
    - data frames
    - subsetting
    - details of function calls
        - argument matching
        - copying of arguments
    - vectorized operations
    - Basic graphics and devices
- Programming
    - control flow: if, for, while.
    - writing functions
    - vectorization
    - debugging tools & strategies
    - basics of methods and object-oriented programming in R
    - efficiency
    - testing code
- Graphics
    - traditional graphics
        - basic plot types
        - graphics parameters
        - annotating plots
        - layouts
    - lattice
        - basic plot types
        - formula language

        - conditioning
        - groups
        - legends/keys

- Input

    - reading non-standard data formats
    - regular expression language
    - connections

- Shell

    - Remote login
    - Shell command line language
    - Useful commands/tools.
    - Redirection (to and from files, pipes connecting output from one command as input to another)

- Access data from the Web

    - Scraping HTML documents.
    - HTML forms.
    - APIs via REST and SOAP.
    - Parsing XML and XPath.

- Graphics

    - Guidelines for good graphics
    - Graphics for the Web
        - interactivity
        - animation

- Statistical Techniques

    - cross validation
    - bootstrapping
    - k nearest neighbors
    - random number generation
    - simulation

Duncan Temple Lang *<duncan@wald.ucdavis.edu>*

Last modified: Tue Dec 6 10:32:45 PST 2011