Sta141
Statistical Computing
Fall 2015

Programming, Computational Reasoning
Modern Data Technologies,
Modern Statistical Methods &
Real Data and Problems

In Class Coordination

Feel free to call me Duncan

I want you to raise questions and discuss problems, questions, concepts in class.
Feel free to raise your hand anytime.

The beginning of every class, I ask for questions. I expect there to be some. If not, then you are not working on the assignments.

If I speak too quickly, ask me to slow down.

If I say something you don’t understand, but you have tried to follow, ask me to explain it a different way.

Class Web site
http://eeyore.ucdavis.edu/stat141

Online Class Forum
http://piazza.com/uc_davis/fall2015/sta141

Join that ASAP!

Send non-personal messages to the entire class.

Personal email to my personal address or via piazza’s private post.
Today’s outline

- Basic Administrative material
- High-level themes & topics in this course.
- Style of this course
- And “Getting Started with R”.
- Tuesday: Why statistics and computing are so important these days

This is the era of “Data Science”

- Combination of
  - statistical methods,
  - reasoning about randomness,
  - computing,
  - being able to manipulate data
  - display results
  - problem solving

Schedule

- 2 lectures per week
- 1 discussion section.
- Office hours – really important.
  - After you have tried to do the assignments, come to office hours with your ideas and attempts and we can figure things out together. This is a huge part of the course.
- Class forum – send questions to the class, carefully describing what you have done and what the problem is.
- Other students and the TAs and I will answer.
- Counts as participation towards your grade.

This is the era of “Data Science”

- More than 90% of your stat. classes focus on “Statistical Methods and Modeling”, but
- More than 50% of data analysis activity is in the other stages that we don’t explicitly teach.
- Computing is essential to all of these stages and computing (now and in the future) is as fundamental as mathematics.
- Yet you have had > 12 years of math and ~ 1 course in computing.
We are going to focus on learning the language(s) of working with data so fundamental to data analysis and decision making.

This is the “introductory” class for statistical computing. We start from the beginning.

We focus on computational thinking, being able to reason about and know how to take a “data analysis” task and perform it on the computer.

This is more high-level and about new technologies rather than minute details of algorithms that are already implemented in every statistical environment.

In other words, we are interested in general aspects of programming, manipulating data in forms that we can use for exploring interesting questions, creating visual displays of data that inform us, presenting summaries using HTML, SVG & Web technologies, Google Earth, Google Maps.

We’ll also cover some statistical & numerical methods and computations, e.g. classification trees, k-nearest neighbors, numerical optimization, random number generation, bootstrap, cross-validation.

The focus is on computing, but we also take this opportunity to explore real data and to do common sense data analysis.

This does not necessarily mean using complex, sophisticated statistical methodology (unless it is appropriate). It is more about finding evidence within data and illustrating your conclusions, identifying conjectures/hypotheses, ....

High-level Themes

- Scientific Computing & Programming
- Working with data
- Data Technologies
- Visualization
- Exploratory Data Analysis & Common Sense
Syllabus Topics

- R programming language
- basics & formal fundamentals
- data manipulation & processing
- graphics/data visualization
- programming & developing reusable functions
- Data input & output
- Text processing & Regular expressions
- Shell language & commands
- XML, HTML & Web Scraping & Web Services - XPath
- Relational Databases & SQL
- Web pages & graphics

This class will be different from most of your statistics classes. We won’t have theorems and proofs; we won’t be (just) applying “this week’s” statistical method to data.

Instead, I want you to problem solve and learn to think about computational tasks.

There is a lot of latitude in how you do things, both the programming and the data analysis, so I want you to be creative and thoughtful.

There will be a lot of technical material to cover.

I will introduce much of it in class. But I expect you to read before class.

I do expect you to read help files, my handouts & notes, tutorials on the Web, chapters/sections of books, etc. to complete your understanding.

This is a very important part of the course - the ability to learn about new languages, technologies, ... by yourself.

And I expect you to come to class with lots of questions. So I will help you considerably, but you have

Through lectures, I’ll introduce you to new material
Then you go off and use it to solve problems. That’s when you’ll have questions.
Pose them on the class forum and/or in class and office hours and we’ll work through them for everyone.
Assignments as Labs

* Lecture courses offer the opportunity to show whether you can get the right answer. Labs on the other hand offer an essential opportunity for students to learn about the practice of science and this practice includes presenting one’s work in a clear and compelling fashion. – Moskovitz & Kellogg. Science, 29 July, 2011.

* This is a lab class with guidance/instructions in lectures.

I want to develop a collaborative relationship with you so that you learn, and not one where I am an adversary or task master and you compete to “pass” or get a grade.

Programming and computing require gaining experience. You must gain that through your own hard work. I can guide you and help you progress when you run into an obstacle.

“True understanding only comes through the student actively constructing their own understanding through a process of mentally building on their prior thinking and knowledge through ‘effortful study’”

Carl Weiman (Nobel Laureate)

http://www.scientificblogging.com/carl_wiemann/optimizing_the_university_why_we_need_a_new_educational_model_for_a_new_century

You are learning language and how to express yourself in these.

It takes practice and experience to become fluent.

You have to practice everyday, not just intensively once every 2 weeks!

So do a little bit every day.
Assignment 1

- Due in 2 parts (but one assignment)
- Get started immediately.

Computing is very different from mathematics from the perspective of learning.
- The computer is immensely fussy and you can waste a lot of time and frustration on tiny little mistakes.
- Unlike mathematics, the computer tells you if you are right or not.
- You get “immediate” feedback which you can learn from to adapt your next step.
- It is a different medium with lots of benefits that you should learn to exploit and learn the tools to help you exploit (e.g. debuggers, profilers, ...)

Most importantly, computing places you in a very active role for your learning.
- You are not trying to uncover truths about mathematics by manipulating symbols. Rather you are constructing an engine that will do what you tell it.

Process & Experience

- You only learn by programming and working with the computer and the data.
- You make mistakes, figure out what went wrong and try other things in a directed, intelligent manner.
- Go back and figure out why what you did first wasn’t correct and what was wrong with it and your approach.
- The iterative approach takes a lot of time, but gets faster the more experience you have.
- NOTE: Absolutely essential you don’t leave the assignments to the “day before” it is due. You won’t finish it.
Text Book(s)

- Unfortunately, there is no single book that covers all the topics we do.
- Paul Murrell’s (free) “Introduction to Data Technologies” is a good book that covers most of them.
- For those who want to learn R for the future, Robert Gentleman’s “R Programming for Bioinformatics”
- Norm Matloff’s “The Art of R Programming” also good.
- R Graphics Cookbook & R in a Nutshell.
- “Learning R” Richard Cotton
- Many topic-specific books for R, e.g., R Graphics.
- The R manuals are also a good resource
- “Data Science in R: Case Studies...” – Nolan & Temple

Learning to learn

- The absence of a single text book has a benefit.
- You have to use the Web and different resources to find information you need.
- This is a highly non-trivial skill
  - composing the question/goal
  - finding resources
  - honing queries (Web or human) to get the relevant information
  - knowing when to detour and when not to

Learning New Technologies

- As computing technologies continue to evolve, it is vital that you
  - learn the fundamentals which transfer across higher-level changes
  - can learn details of new technologies by yourself

- We’ll use R as the primary programming language
  - Open source and free!
  - De facto standard for statistics research, statistical computing, sophisticated/non-standard data analysis
  - Widely used in statistics, computational biology, finance, industry, ...
  - Flexible, full-featured programming language
    - many built-in statistical functions
    - graphics/visualization – grz, lattice, grid
    - > 7500 add-on “packages” available from CRAN (http://cran.r-project.org) & other repositories
Why R?

- It is not R itself, but general concepts of high-level programming and languages that are important to learn.
- R is a complete programming language and learning about it allows you to transfer the concepts to other languages.
- Syntax and available libraries may differ between languages, but how you approach a computational task and reason about the computations is the goal in this class.
- So not just about how to use one function or another, but about the programming process.

Grading

- 2-3 “shorter” (week long) assignments.
- more programming & graphics oriented
- 4-5 “longer” < 2 week assignments
- more involved data analysis & programming.
- like a focused homework, but uses material from several different topics in the course, i.e. doing an entire task from beginning to end.
- Participation: In Class and on-line forum (12.5%)
- Possibly 1 mid term – 7.5%
- Assignments count for 80% of the grade.

Work...

- Exercises – small tasks that I want you to do between classes, but that you don’t turn in.
- Assignments – more significant, complete work that you do hand in to me and the TAs
  - both printed version and email electronic version.

“Sharing” Work

- Make certain to identify and attribute any ideas, code, etc. that you use from other people and sources
- You can use code from the Web. (You need to understand it.)
- You can use information and code that is on the class discussion forum.
- You can talk with your classmates.
- You CANNOT submit other people’s code within your writeup – only small snippets.
- You must do the work yourself.
Class Forum

- Ask and answer each others questions, with myself and TAs participating too.
- Search the posts for previous questions and answers before sending mail.
- Participation is asking well-phrased questions, not just answering them.
- Participation on forum counts as part of your grade, along with participating in class and office hours.

When asking a question...

- There is an art to asking questions about computing and it takes time to compose questions in a way that contains enough, but no more information than is necessary.
- You have to think about how somebody who is not with you can understand where you are.
- It is very common to describe the problem in terms only you can understand based on what you have done up to that point.
- The rest of us are coming from a fresh start with no clue about what you have done.

Identify variables that you have created in earlier stages that you are using in the question.

Basic information:

- Tell us what operating system (OS) you are using, e.g. Windows 8, 10, OS X, Redhat linux, Ubuntu,…
- This is important information so people can help with OS-specific problems.