Course Description

Stat 243 is being substantially revised relative to previous years’ offerings. The course will now be an introduction to statistical computing taught using R. The course will cover both programming concepts and statistical computing concepts. Programming concepts will include data and text manipulation, data structures, flow control, functions and variable scope, regular expressions, matrix manipulations, and debugging. Statistical computing topics will include numerical linear algebra, simulation studies and Monte Carlo methods, numerical optimization, and visualization. A goal is that coverage of these topics complement the models/methods discussed in the rest of the statistics graduate curriculum. We will also cover the basics of UNIX/Linux, in particular some basic shell scripting, operating on remote servers, and basic parallel processing.

Note that I aim to have the course be useful to those who already know a fair amount of R by (1) covering more advanced aspects of R and (2) through the extensive coverage of the statistical computing topics.

Informal prerequisites: If you are not a statistics or biostatistics graduate student, please chat with me if you’re not sure if this course makes sense for you. A background in calculus, linear algebra, probability and statistics is expected, as well as a basic ability to operate on a computer (but not necessary a UNIX variant). I do not expect you to know any R, but most students probably will, and those who don’t know any R will probably need to spend extra time in the initial weeks getting up to speed.

Objectives of the course

The goals of the course are that, by the end of the course, students be able to:

- operate effectively in a UNIX environment and on remote servers;
- program effectively in R with an advanced knowledge of R functionality and an understanding of general programming concepts; and
- understand in depth and be able to make use of principles of numerical linear algebra, optimization, and simulation for statistics-related research.

Personnel

- Instructor:
  - Chris Paciorek
e-mail: paciorek@stat.berkeley.edu; Room 339, Evans Hall; Phone: (510) 642-6190;
Office hours (in Evans 339): TBD, or just drop by if my door is open, or schedule an appointment.

**When to see me about an assignment:** I’m here to help, including providing guidance on assignments. You don’t want to be futilely spinning your wheels for a long time getting nowhere. That said, before coming to see me about a difficulty, you should try something a few different ways and try to define/summarize what is going wrong or where you are getting stuck.

- There is no GSI, and there are no sections.

**Course website**

All course materials will be posted on bspace. In some cases I may have a set of slides that will serve as the skeleton for class. I will do my best to post the slides and demo R code on bspace by 4 pm the day before class. I will not print out copies, so please bring your own copies if you want them in front of you. I may have an ongoing document for each unit that I add to over several class periods, so you can just print out the new pages for each class or week.

I will also make announcements and provide material through the website, with announcements also going out as email, so please check your email regularly.

For asking questions that are not of a personal nature (i.e., that relate to course material), please email the class email list, stat243fall2011@bspace.berkeley.edu (we might use the Forums tool on bspace instead...). This will serve as a discussion board so everyone can see my responses and so you can answer each other’s questions as well.

If you need to get my attention quickly or have an issue specific to an individual student, you can send an email to paciorek@stat.berkeley.edu.

**Course material**

Primary textbooks:

- For R (Chambers is more abstract and Adler more hands-on):
  - Chambers, John; Software for Data Analysis: Programming with R (available electronically through OskiCat: http://dx.doi.org/10.1007/978-0-387-75936-4)
  - Adler, Joseph; R in a Nutshell (available electronically through OskiCat: http://uclibs.org/PID/151634)

- For statistical computing topics: Gentle, James. Computational Statistics (available electronically through OskiCat: http://dx.doi.org/10.1007/978-0-387-98144-4)

- Assorted documents provided by me.

Other resources with more details on particular aspects of R:

- The R-intro and R-lang documentation. www.cran.r-project.org/manuals.html


Other resources with more detail on particular aspects of statistical computing concepts:

- Lange, Kenneth; Numerical Analysis for Statisticians, 2nd ed. (first edition is available electronically through OskiCat)
· Monahan, John; Numerical Methods of Statistics

And for bash:

· Newham, Cameron and Rosenblatt, Bill. Learning the bash Shell (available electronically through OskiCat: http://uclibs.org/PID/77225)

Computing Resources

All students will be provided with a computer account on the Statistical Computer Facility (SCF) network of Linux and Mac computers. The computer room in 342 Evans provides iMacs, and computers are also available in Evans 432; you can also remotely log in to the SCF system from other campus computers or from home. If you wish to do your assignments on some other computer, keep in mind that required programs/packages may be stored on the SCF system, and it is your responsibility to get the required resources working on the other computer. Also, assignments and topics are oriented towards the UNIX operating system, so if you wish to use a non-UNIX computer, you should make sure that the necessary resources are available (and in some cases, you will need to use a UNIX-based system). I can try to help with Windows-related issues but this help will necessarily be limited by my limited experience with Windows.

Course requirements and grading

The grade for this course is primarily based on computer assignments due every week or two that will be assigned throughout the semester. There is no final exam. 20% of the grade is based on class participation, which can be either through talking in class or participating in online discussions. This could involve asking questions or providing your input/expertise on a topic.

Problem sets will be open-ended, so those coming in at different levels may explore things with more or less sophistication. I’m also open to you defining your own assignment for a given topic, if you are working on a specific problem. E.g., instead of working on a particular text manipulation problem I assign, you might work with your own text data. Check with me before forging ahead.

I will be less willing to help you if you come to my door or my email inbox at the last minute. Working with computers can be unpredictable, so give yourself plenty of time for the assignments.

Please submit your assignments on paper, including all your code.

Class participation

My goal is to have classes be an interactive environment. This is both more interesting for all of us (hopefully) and more effective in learning the material. I encourage you to ask questions and will pose questions to the class to think about and discuss. To increase time for discussion and assimilation of the material in class, before some classes I may ask that you read material in advance of class.

Please do not use phones during class and limit laptop use to the material being covered.

Student backgrounds with computing will vary. For those of you with limited background on a topic, I encourage you to ask questions during class so I know what you find confusing. For those of you with extensive background on a topic (there will invariably be some topics where one of you will know more about it than I do), I encourage you to pitch in with your perspective. In general, there are many ways to do things on a computer, particularly in a UNIX environment, so it will help everyone (including me) if we hear multiple perspectives/ideas.
Feedback

I welcome comments and suggestions (and gripes) and will solicit feedback via a survey partway through the class. Comments at any other time are welcome, and if you prefer anonymity, you can leave a note in my mailbox or under my door.

Topics (in rough order with rough timing)

1. Introduction to UNIX and operating on a compute server (3 days)
2. Intro to R (1 day)
3. Data and text manipulation in R, including regular expressions and database operations (3 days)
4. R programming: data structures and types, object orientation, flow control, functions, efficient programming, parsing/expressions/formulas (7 days)
5. R debugging and profiling (2 days)
6. Computer storage, architecture, and arithmetic (2 days)
7. Numerical linear algebra (5 days)
8. Simulation studies and Monte Carlo (3 days)
9. Numerical integration and numerical differentiation, symbolic integration and differentiation (3 days)
10. Optimization (5 days)
11. Advanced graphics (2 days) (this may be moved to be after item #4)
12. Basics of parallel processing (2 days)