This course is not:

A class in **Computational Statistics**

We will not explicitly cover matrix decompositions or algorithms for solving least squares problems

This course is not:

A class in **Applied Statistics**

We will not explicitly cover algorithms for fitting a linear model, interrogating the fit, using diagnostics to suggest various elaborations

---

Statistics 202a

*Data Technologies and Programming Principles*

Instructor: Mark Hansen  
8951 Mathematical Sciences  
x68375  
cocteau@stat.ucla.edu

Office Hours: Tues/Thurs TBD or by appointment

Grading: 80% Projects, 20% in-class participation
This course is not:

- A class in **Mathematical Statistics**

We will not explicitly cover anything related to the large-sample behavior of estimators, rates of convergence, or minimaxity

This course is:

An introduction to the idea of **exploratory computing**, that is, the computational skills necessary to conduct all phases of a statistical analysis

Why?

- Our ability to compute shapes and limits the kinds of methods we can perform
- Our ability to compute colors our very notion of what data are or can be
- Without basic skills, we are banished to the hinterlands and consulted only when someone needs a *p*-value

Some history

- R. A. Fisher is often credited as the single most important figure in 20th century statistics
- Before Fisher, statistics was an “ingenious collection of ad hoc devices” (Efron, 1996)
- *Statistical Methods for Research Workers* (1958) covers many of the basic techniques still found in introductory textbooks

![Image of Ronald A. Fisher](www.csse.monash.edu.au/~lloyd/tildeImages/People/Fisher.RA)
Some history

He is responsible for creating a mathematical framework for statistics; but equally important was his impact on data analysis.

Fisher once commented that over his calculator, “he had learned all he knew”.

* From “The Technical Tools of Statistics” by J. W. Tukey (1964)
  stat.bell-labs.com/who/tukey/memo/techtools.html

Ronald Aylmer Fisher (1890-1962)

What happened between Fisher’s time (the 1920’s and 30’s) and Tukey’s (the 1960’s and 70’s)?

The computer

In 1947 the transistor was developed

In 1964 the IBM 360 was introduced and “quickly becomes the standard institutional mainframe computer”.

By the end of the 1960’s computer resources were generally available at major institutions.

* Taken from a PBS history of the computer, www.pbs.org/wnet/timeline/micro.html

John Wilder Tukey (1915-2000)

Some history

John Tukey is another pioneer of statistical theory and practice.

In his texts *Exploratory Data Analysis I and II* (1970), Tukey creates graphical tools for exploring features in data.

Many of the descriptive techniques presented in introductory textbooks are due to Tukey.

His style is iterative, advocating many different analyses.

From the Bell Laboratories archives, circa 1965
  stat.bell-labs.com/who/tukey

What happened between Fisher’s time (the 1920’s and 30’s) and Tukey’s (the 1960’s and 70’s)?

The computer

In 1947 the transistor was developed

In 1964 the IBM 360 was introduced and “quickly becomes the standard institutional mainframe computer”.

By the end of the 1960’s computer resources were generally available at major institutions.

* Taken from a PBS history of the computer, www.pbs.org/wnet/timeline/micro.html

John Wilder Tukey (1915-2000)
The computer

The introduction of the computer changed the way statistics is practiced

Automated graphics had an enormous impact on statistical methodology and is really at the core of Tukey’s EDA

So too is the idea that we can try out many analyses, we can fit models and perform computations easily and quickly and uncover what the data might have to tell us

As the computer revolution finally penetrates into the technical tools of statistics, it will not change the essential characteristics of these tools, no matter how much it changes their appearance, scope, appositeness and economy. We can only look for:

1. more of the essential erector-set character of data analysis techniques, in which a kit of pieces are available for assembly into any of a multitude of analytical schemes,
2. an increasing swing toward a greater emphasis on graphicality and informality of inference,
3. a greater and greater role for graphical techniques as aids to exploration and incisiveness,
4. steadily increasing emphasis on flexibility and on fluidity,
5. wider and deeper use of empirical inquiry, of actual trials on potentially interesting data, as a way to discover new analytic techniques, and
6. greater emphasis on parsimony of representation and inquiry, on the focussing, in each individual analysis, of most of our attention on relatively specific questions, usually in combination with a broader spreading of the remainder of our attention to the exploration of more diverse possibilities.

From “The Technical Tools of Statistics” by J. W. Tukey (1964)

stat.bell-labs.com/who/tukey/memo/techtools.html

Home computers and the Internet

First personal computer in 1975 (MITS Altair 8800)

In 1977 Apple introduces the Apple II at a price of $1,195; 16K of RAM, no monitor

The first spreadsheet, VisiCalc, ships in 1979 and is designed for the Apple II

The Apple Macintosh appears in 1984

Microsoft Windows 1.0 ships in 1985

* Taken from a PBS history of the computer, www.pbs.org/nerds/timeline/micro.html

Home computers and the Internet

In the 1990’s there is a migration to “ubiquitous computing”: There are small but powerful computers in phones, PDAs, cars, you name it

The internet (or rather a nationwide fiber optic network) connects us, with wireless access becoming standard

At the same time, technologies for data collection, and in particular those associated with environmental monitoring are undergoing a small revolution in the form of sensor networks

Shouldn’t all this have some impact on the practice of statistics?
Statistics today

Undeniably, Tukey’s prediction that computing will provide us with a kind of Swiss Army knife of tools has come to pass; as we will see, programs like R have grown out of that vision.

What he was unable to anticipate was the extent to which information technologies would revolutionize the way we observe the world, fundamentally changing what we think of as data.

The complexity and size of data, our ability to measure and record phenomena related to our physical (and virtual) surroundings are at an all time high.

Bin Yu at UCB speaks of many of the interesting problems in statistics as now existing in the IT-regime.

Outline

Emacs, Unix tools, Perl
Regular expressions, manipulating text
R, data types, basic computations, writing functions, objects and methods, language definition
Code distribution, R packages, software licenses
Scripting visualization via Processing
Databases, SQL, R interface to MySQL, XML, XSLT
Statistical computation in realtime systems

Back to this course

The tools we will cover are an admittedly biased sample of computer languages and tools.

They have served me and my colleagues well in a variety of different applications and comprise a kind of core that you should know about.

Many of these tools are what computer scientists would classify as scripting languages.

Computer languages are created for varying purposes and tasks — different kinds and styles of programming. One common programming task is known as scripting, or connecting diverse pre-existing components to accomplish a new related task. Those languages which are suited to scripting are typically called scripting languages. Many languages for this purpose have common properties: they favor rapid development over efficiency of execution; they are often implemented with interpreters rather than compilers; and they are strong at communication with program components written in other languages.

Many scripting languages emerged as tools for executing one-off tasks, particularly in system administration. One way of looking at scripts is as “glue” that puts several components together; thus they are widely used for creating graphical user interfaces or executing a series of commands that might otherwise have to be entered interactively through keyboard at the command prompt. The operating system usually offers some type of scripting language by default, widely knowns as a shell script language.

Scripts are typically stored only in their plain text form (as ASCII) and interpreted, or compiled each time prior to being invoked.

Some scripting languages are designed for a specific domain, but often it is possible to write more general programs in that language. In many large-scale projects, a scripting language and a lower level programming language are used together, each lending its particular strengths to solve specific problems. Scripting languages are often designed for interactive use, having many commands that can execute individually, and often have very high level operations (for example, in the classic UNIX shell (sh), most operations are programs themselves).

A (silly) example

Many organizations make data available via the web; here is a plot of temperature readings at James Reserve for the last 48 hours.

Data are pulled from a central database in response to your request; you specify the time frame, the reporting rate, and so on.

From the Wikipedia on “scripting language”
A (silly) example

Suppose we want to do this regularly? What is the best way to get access? Regularly navigate through the site, "select" a large chunk of data and load it into Excel?

That seems remote; can we get the data in a more automated fashion? The site buries data in HTML; we will see that languages like Perl let us scrape out the contents...

Or, is it better to do all the processing in R?

Or, if we have more direct access to the database, can we fetch records directly using the MySQL library in R?

Over the course of this quarter, we will be able to evaluate these tradeoffs, and in the process learn a bit more about data technologies like XML.
The hard sell...

In my experience, statistical computing inevitably leads to questions of data representation and introduces a myriad of connections with how data are collected, displayed and analyzed; resolving these questions can be some of the most rewarding aspects of statistical practice.

... or at very least

Through this class we hope to create a culture within our department in which students routinely discuss computation, sharing experiences or reporting on new computing platforms and emerging technologies.

We hope to instill basic problem solving skills so that you can learn languages on your own, cull online documentation, or find manuals online.

We hope that no matter what kinds of projects you start either at UCLA or beyond, you will not stumble over the computing.

Other byproducts

Problem-solving techniques; carving the problem into smaller pieces; sequential analyses.

An aesthetic sense for which tools to apply to which task; we will emphasize discussion in this class.

Debugging skills.

Experience auditioning new methods, code and searching for documentation.

... but you won’t go it alone

The class is organized around a series of projects.

You will often work in groups, organized to so that each represents a diverse set of computing skills.

At other times, you will work on your own; but we will make use of the group structure to help discuss what you did and compare your approach to that of the others.
Dartmouth data project

When a wireless device connects to an AP, we first see messages stating that the device has Authenticated and then that it has Associated with the AP.

When a device switches to a different AP it Reassociates, and when the user logs off they Deauthenticate.

Enron email corpus

This data set consists of the emails from 151 employees of Enron, mostly upper-management.

The data are stored in mail folders, one for each user; our project will be to apply social network metrics to see if the pattern of email exchange is changing as the scandal unfolds.

Computing

You are welcome to perform your computations anywhere; my only constraint is that you use the same tools we are using in class (if something strikes you as better, please speak up!)

We have a machine available (dual processor G4 with 2Gb of RAM; should that prove to be underpowered, it will be swapped with a dual processor G5 with 8Gb of RAM).

A sudden collapse

July 2001

New CEO Jeff Skilling gets hit in the face with pig from a protester who blamed Enron for California's energy crisis. California consumer groups and politicians accused Enron of price gouging during California's power shortage. Enron stock drops to $42.93.

July 2001

The California power crisis provokes Enron, Pacific Gas and Electric Company files for bankruptcy, and the enronwhile.com website is shutdown. Pacific Gas and Electric Co. is one of the top three utility companies, along with Southern California Edison and San Diego Gas & Electric.

Aug. 14, 2001

CEO Jeff Skilling resigns for “personal reasons.”

Oct. 12, 2001

Enron discloses a $658 million loss in its third quarter (for this fiscal year).

Oct. 24, 2001

Andrew Fastow named as chief financial officer due to questionable business transactions and partnerships. Jeff McElmoyn takes over as CEO.

Oct. 31, 2001

Securities and Exchange Commission upgrades its inquiry into Enron’s business transactions and accounted events.

Nov. 8, 2001

Enron reviews financial statements to induce earnings by an additional $550 million over the past four years, in huge part due to losses from impeached partnerships. It is also disclosed that Fastow made $30 million in fees and profits from his involvement with the discredited partnerships.

Enron announces it must pay a $990 million in debt, as another $6 billion by next year.

Enron's CEO Lay calls the U.S. Secretary of the Treasury, Paul O'Neill, to discuss Enron's financial problems, according to a Treasury Dept. spokesperson.

Computing

The machine is called

lab-compute.stat.ucla.edu

and you can access it via ssh even outside our firewall; I will hand out login accounts today together with passwords.
Computing

You will need access to a Unix system (or something close); you can either use one of our machines or if you only have access to a Windows computer you can get what you need with Cygwin

www.cygwin.com

I have instructions about how to install it on the course web site; otherwise you can bring in your (laptop) computer and I can help you after class

Next time

We will begin with the basics of the Unix operating system; the purpose of an operating system is to organize and control hardware and software so your computer behaves in a flexible but predictable way

Unix is just one, various flavors of Windows are other examples

Unix also comes in flavors, some of which share a common source, some of which run on special hardware; AT&T Unix, BSD (Berkeley Software Distribution), FreeBSD, OSX or Darwin (based on BSD), Linux...

Why spend time on Unix tools?

Practical. These tools help us interact with data, shape it, share it, organize it, store it; the more experienced you are with these tools, the more conveniently you can manipulate data

Historical. R (the programming language we will cover in a couple weeks) is a “free version” of something called S which grew out of Bell Labs; there are a number of parallels between the spirit and final form of the Unix shell and the R command line

But first, your quarter-long project...
Mapping

Mapped or spatial data have always been an important thread of statistical research

In recent years, the broad idea of mapping and understanding geographically coded data has entered the public psyche

With Google Maps and Amazon’s A9 Maps (Beta) we have seen the price of creating maps dip to zero
Your project

Groups will report weekly (starting next week) on their progress

We will use a course Wiki to keep track of the different projects; the weekly updates will let other class members comment on what you’re up to

Projects will be presented during the scheduled class final period