Penultimate

We finished our treatment of relational databases and expanded the kinds of queries we make to the Reality Mining data

We reviewed the SQL commands we introduced in the previous lecture and extended them to deal with multiple tables; we got closer to reconstructing the results from the Reality Mining group

Present

We have some choices; I wrestled with the following topics for the final lecture

1. Data models, and in particular XML (the Extensible Markup Language); you will see more and more data encoded this way and you should know about it

2. Processing, a simple scripting language for generating dynamic visualizations

3. Emacs and R, or more to the point ESS, Emacs Speaks Statistics

4. Version control software, and specifically SVN, the Subversion software

5. Package construction in R; the tools that exist to help you share your code
Pointer for future practice

XML encodes information with meaningful structure and semantics that both computers and humans can understand.

It finds greatest traction in contexts where you have to exchange data.

In short, it allows for easy definition, transmission, validation and interpretation of data.

Prerogative, mine

But, ultimately, I decided to dodge the glories of XML and instead introduce Processing and use it as an anchor for reviewing some of the important themes in the class.

Hold on...

Pushing forward, pressing on...

Throughout this course we have seen that different languages or tools are better at performing different tasks.

R, for example, allows you to be more expressive with statistical computations like model fitting and residual analysis; while Perl feels more natural for working with text data.

Part of your job is to recognize first that tools exist to accomplish your computing task (that it’s doable) and then to find the “best,” or at least not the worst, tool to perform the task.

Prelude

Last time, we mentioned that code has become a large component of many artists’ practices.

What kinds of “computations” should be natural in an environment that supports this community? What kinds of “data” should be easy to manipulate and display?

Also, many of the examples we saw were meant to be experienced online, that is, through a browser; should the environment support “exhibiting” work and what would make that easy?
Processing

As an environment, Processing includes a text editor for composing programs, a display window for viewing running programs, and a compiler that takes the program from code to "runnable" form.

Given Processing's lineage, it should not surprise you that its basic unit of production is called a sketch; a sketch is simply a directory that contains code and data (compare it to an R package, for example).

Pause

Processing is essentially a Java library; we haven't learned Java, but you've got enough coding under your belt to understand what's going on here.

To become a Processing super-user, Java would be important to know in the long run; for now, we can treat it as its own language and see how far we get.
To exhibit the code you've written means distributing a program in some way; here is where the connection between Processing and Java comes in.

Most browsers will execute Java "programs" in the form of an Applet; Processing will take all the pain out of assembling the code and generating the web pages.
Possibilities

So, Processing is strong on animation, on images, on dynamics; it also includes simple functions to collect user input via the mouse or other devices.

How might we put this to work? What does the publishing aspect give us?
Point?

Hopefully this class has made you think a little about the act of computing; that there are expressive languages out there that might change what you think about data, about analysis, about your “practice”

The more tools you have at your disposal, the richer your practice will be; you can get involved earlier in projects and to some extent, you can even initiate projects

There is no reason to believe that you have to wait for others to hand you statistical questions to answer...

Precocious computing

A background in computing might lead you places your mathematical tools aren’t quite ready to describe; this isn’t bad, not bad at all...

Probability, mathematical statistics, they’re just one way of describing the patterns in data; hopefully in this class you’ve seen that computation comes with its own descriptions and its ways of “knowing”

Purview

Computing and the flow of data are everywhere and mediate even our physical interactions with places

How much of this becomes the purview of statistics depends entirely on our ability to “join in,” to swim in the new data substrates that have formed around us

Computing and in general information technologies will be your point of entry!

Thank you.