The data scientist's toolbox

STA 380
UT-Austin

What is this course about?

This class is about gaining knowledge from raw data. You'll learn to use large and complicated data sets to make better decisions.

A mix of practice and principles:

- Solid understanding of essential statistical ideas
- Concrete data-crunching skills
- Best-practice guidelines.

We’ll learn what to trust, how to use it, and how to learn more.

First half: supervised learning.

- Given past data on outcomes y paired with features x, can we find patterns that allow us to predict y using x?
- Key characteristic: there is a single privileged outcome y.
- Example: a house has 3 bedrooms (x_1), 2 bathrooms (x_2), 2100 square feet (x_2), and is located in Hyde Park (x_4). What price (y) should it sell for?

In real life, there might be hundreds or thousands of features. If you know regression: this is like regression on steroids!

Second half: unsupervised learning.

- We still have multivariate data and want to find patterns.
- But there is no single privileged outcome. ("Everything is y.")
- Example: "Here's data on the shopping basket of every Whole Foods customer at 6th and Lamar last month. Find some patterns that we can use to improve product placement."

An alphabet soup of labels...

Statistical learning, data mining, data science, ML, AI… there are many labels for what we're doing!

- Econometrics, statistics: focused on understanding the underlying phenomena and formally quantifying uncertainty.
- Business analytics, data science, data mining: traditionally focused on pragmatic data-analysis tools for applied prediction problems.
- Machine learning, pattern recognition, artificial intelligence: focused on algorithms with engineering-style performance guarantees.

An alphabet soup of labels...

How our goals fit into all this: we keep an eye on what is both *useful* and *true*:

- Learn actionable patterns from noisy, complex data (data mining).
- If at all possible, do so using simple, scalable algorithms (machine learning, AI).
- If necessary, provide error bars (statistics).
- Always be aware of the problem context or decision at hand (econometrics).

About "data mining"...

Among economists, “data mining” is a dirty word. Example: the “Lucas critique”:

- Fort Knox has never been robbed.
- Thus historically, there's a zero correlation between security spending at Fort Knox (x) and the likelihood of being robbed (y).
- Naive “data mining” conclusion: leave Fort Knox unguarded!

This is a total caricature. We'll strive to give data mining a better reputation :-)

What does it mean for data to be "big"?

Big in either or both:

- the number of observations (size n)
- and in the number of features or predcitor variables (dimension p).

In these settings, you cannot:

- Look at each individual variable and make a decision (t-tests).
- Choose amongst a small set of candidate models (specification tests from stats or econometrics).
- Plot every variable to look for interactions or transformations.

Good data mining = inference at scale

Some data-mining tools are familiar, or familiar with a twist:

- linear regression
- p-values
- automatically select a set of relevant feature variables, then fit a linear model

Some are totally new:

- PCA
- K-means

All require a different approach when n and p get really big.

People use these tools everywhere

- Mining client information: Who buys your stuff, what do they pay, what do they think of your new product?
- Online behavior: Who is on what websites, what do they buy, how do/can we predict or nudge behavior?
- Collaborative filtering: predict preferences from people who do what you do; recommender engines.
- Text mining: Connect blogs/emails/news to sentiment, beliefs, or intent. Parsing unstructured data, e.g. EMR.
- Big regression: mining data to predict asset prices; using unstructured data as controls in observational studies.

The four pillars of data science

1. Data collection
2. Data cleaning (pre-processing/hacking/“munging”)
3. Analysis
4. Summary (figures + prose)

This course focuses a little on 2, heavily on 3-4, and not at all on 1.

Data collection and cleaning: principles

On collection, management, and storage: a full subject unto itself. (I’m happy to provide references, but this isn’t the part of data science we cover in this course.)

On cleaning: I defer to Jeff Leek’s description of “How to Share Data with a Statistician.” (See course readings.) Always provide:

1. The raw data.
2. Tidy data.
3. A variable “code book” written in easily understood language.
4. A complete, fully reproducible recipe of how the clean data was produced from the raw.

Data analysis and summary: principles

You will analyze a lot of data in this course. Our watchwords are *transparency* and *reproducibility*.

- The end product: you will write a report with beautiful figures, and someone else will marvel at it.
- Data science is hard enough already: there is zero room for ambiguity or confusion about data or methods.
- Any competent person should be able to read your description and reproduce exactly what you did.

Data analysis and summary: principles

The ideal: “hit-enter” reproducibility.

- Someone hits enter; your analyses and figures are reproduced from scratch and merged with prose, before their eyes.
- We will rely on a handful of easily mastered software tools to put this ideal into practice: R, Markdown, and Git

Data analysis and summary: principles

All reports involve three main things:

1. A question: what are we doing here?
2. Evidence: a set of figures, tables, and numerical summaries based on the analyses performed.
3. Conclusions: what did we learn?

Data analysis and summary: principles

The basic recipe for writing a statistical report:

1. Make the key figures and tables first.
2. Write detailed, self-contained captions for each one.
3. Put these figures and tables in order (question, then answer).
4. Write the story around these main pieces of evidence.

This helps avoid “fear of the blank page”!

Our software toolkit

- R: for data analysis
- Markdown and RMarkdown: for writing reports
- GitHub: for collaboration and dissemination or results

R

R is the real deal: an immensely capable, industrial-strength platform for data analysis.

It's used everywhere:

- Academic research (stats, marketing/finance, genetics, engineering)
- Industry (Google, Microsoft, EBay, Boeing, Citadel, IBM, NY Times)
- Governments/NGOs (Rand, DOE, National Labs, Navy)

R is free and looks the same on all platforms, so you'll always be able to use it.

R

A huge strength of R is that it is open-source. R has a *core*, to which anyone can add contributed *packages*.

- $\approx$ 10,000 packages c. Jan 2017, as varied as the people who write them.
- Some are specific, others general.
- Some are great, some decent and unpolished, some terrible.

R has flaws, but so do all options (e.g. Python is great, but the community of stats developers is smaller, interactive data analysis is less slick, and you need to be a more careful and sophisticated programmer.)

Most students prefer to use R via an IDE. We'll use *RStudio.* It's awesome.

Markdown

- A simple markup language for generating a wide variety of output formats (HTML, PDF, etc) from plain text documents.
- Two pillars: (1) a formatting language; (2) a conversion tool.
- Much simpler than, for example, HTML.

This presentation was written in Markdown.

Markdown

```
## Markdown

- A simple markup language for generating a
wide variety of output formats (HTML, PDF,
etc) from plain text documents.
- Two pillars: (1) a formatting language; (2)
a conversion tool.
- Much simpler than, for example, HTML.

This presentation was written in Markdown.
```

This is what the raw text looked like for the last slide; it got rendered as a bulleted list under a title.

Git and GitHub

git:

- software for version control.
- ideal for collaborative work.
- the basic unit in the git universe is a *repository*, aka “repo”: a collection of files/directories all related to a single task, project, or piece of software.
- Example: the class website is a git repo.

Git and GitHub

GitHub:

- a git repository hosting service.
- a location to store your code in the cloud and easily sync it across multiple machines and multiple collaborators
- the coolest place on the Internet :-)

Git and GitHub

The git repo for our class website is stored both on GitHub (the `remote` copy) and my own computer (the `local` copy).

Basic workflow:

- Make changes to files in the `local` copy of the repo.
- `commit` those changes, thereby creating a snapshot of the repo at a single moment in time that can always be restored.
- `push` those changes to `remote`

Git and GitHub

You can use git either through:

- the command line in a Unix/Linux shell (the hard-core coder's approach)
- a graphical front-end (e.g. GitHub Desktop, SourceTree). I strongly recommend this for git first-timers!

Your assignment for the first day

Look through the tutorials on the course website, and:

- Create a GitHub account.
- Create your first GitHub repository.
- Inside that repository (on your local machine), create a toy RMarkdown file that does something—e.g. simulates some normal random variables and plots a histogram.
- Knit that RMarkdown file to a Markdown (.md) output.
- Push the changes to GitHub and view the final (knitted) .md file as a web page.

These instructions will make sense after you read the tutorials!