[1] Big Data: Inference at Scale

Matt Taddy, University of Chicago Booth School of Business

faculty.chicagobooth.edu/matt.taddy/teaching

[1] getting oriented

Introduction: goals, material, syllabus

Computing

- R: examples, resources, and how we'll learn.
- Big data, distribution, storage, connectivity.

Visualization, Statistics, and Dimension Reduction

Testing and Discovery: False Discovery Rates

Introduction

This is a class about Inference at Scale

We're here to make sure you have the tools for making good decisions based on large and complicated data.

A mix of practice and principles

- Solid understanding of essential statistical principles.
- Concrete analysis ability and best-practice guidelines.

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

A hands-on subject: the idea that MBAs can just pass data analysis off to number crunchers is an out-of-date cartoon.

What is in a name?

Big Data, Econometrics, Statistics, and Machine Learning

There are many labels for what we do...

Econometrics → Statistics
→ Data Mining / Big Data / Data Science
→ Machine Learning (ML) and Artificial Intelligence (AI)

Along this spectrum, you move from heavy focus on what things you are measuring (what real phenomena they correspond to) to a more pragmatic 'useful is true' pattern discovery approach.

The similarities are much bigger than any distinctions.

The ‘BD’ name comes from computer scientists working to do aggregation on data that is too big to fit on a single machine.

As aggregation became analysis, BD got closer to stats + ML. And after a healthy dose of hype, everything is fairly confused...

My take: *Data Science* is the umbrella term for inference in a world that is messier that in your old statistic textbook, and *Big Data* is DS focused on business and industrial applications.

- Infer patterns from complex high dimensional data.
- Simplicity and scaleability of algorithms is essential.
- We keep an eye on both *useful* and *true*.
- The end product is a *decision*.

A big aspect of Big Data is ‘pattern discovery’ or ‘data mining’

Economists think data mining is a dirty word ...

Economics and the Lucas Critique

... but they only know bad data mining.

Good DM is about inferring useful signal at massive scale.

Our goal is to summarize really high dimensional data in such a way that you can relate it to structural models of interest.

$\Rightarrow$ Variable Selection and Dimension Reduction

We also want to predict! If things don't change too much...

$\Rightarrow$ Probabilistic Prediction and Classification Rules

We need to constantly beware of false discovery.

What does it mean to be ‘big’?

Big in both the number of observations (size ‘n’)
and in the number of 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 (*F-test*).
Plot every variable to look for interactions or transformations.

Some BD tools are straight out of previous statistics classes (linear regression) and some are totally new (trees, PCA).
All require a different approach when *n* and *p* get really big.

Course Schedule

subject to change...

[1] Data: Computing, plotting, and principles. [False?] discovery.

[2] Regression: A grand overview, linear and logistic.

[3] Model Selection: penalties, information criteria, cross-validation

[4] Treatment Effects: HD controls, propensity scores, bootstrap

[5] Classification: Multinomials, KNN, sensitivity/specificity, DMR

Midterm!

[6] Networks: co-occurance, directed graphs, Page Rank

[7] Clustering: Mixture models, k-means, and association rules.

[8] Factors: Latent variables, PCA, PCR, and PLS.

[9] Trees: CART and random forests, ensembles

[10] Text Mining: topic models, sentiment prediction, deep learning

We’ll be working with real data analysis examples

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

Many are applicable to marketing, but we’re far more general.

All of our analysis will be conducted in R

This is the real deal: industrial strength software for data analysis. It’s free, cross platform, and hugely capable.

Academics (stats, marketing/finance, genetics, engineering), companies (EBay, Google, Microsoft, Boeing, Citadel, IBM), and governments (Rand, DOE National Labs, Navy) use R.

Since R is free, you’ll always be able to use it.

A huge strength of R is that it is open-source.
This is also why it is sometimes a bit unpolished.

R has a core, to which you can add contributed packages.
These add-ons are as varied as the people who write them.
Some are specific, others general, some are great, some suck.

R is not without flaws, but neither are the other options.
e.g., I like python, but the community of stats developers is smaller and you need to be a more careful programmer.

Some students prefer to wrap R in an IDE; e.g. R-studio.

The barrier of entry for R is its command line interface:

You type commands to get what you want.

The learning curve can be steep, but is very worthwhile.

- You have code of saved commands, so you know what you've done and can easily repeat similar operations.
- Interacting with computers in this way is a Big Data skill.

All code for lectures and homework will be available online.
The best way to learn software is through imitation.

There are a ton of resources: see the website and **syllabus**.

Computing: R

To start, click on R or R or just type ‘R’ in a terminal.

At its most basic, R’s just a fancy calculator. (e.g., `*`, `/`, `+`, `-`).

Everything is based on assigning names
(`<-` works pretty much the same as `=`).

`A <- 2` → 2.

`B <- c(1,2,3)` → 1 2 3 (same as `B=1:3`).

`C = A + B[3]` → 5.

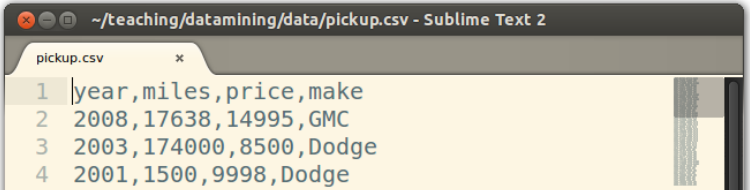
The `c( )` function and `:` operator build vectors.
`length(B)` will tell you how many elements it holds (3).

To inspect a variable, just type the name (do this often!).

R can read almost any data format

First, set the working directory to where you store data.
It's good practice to create folders just for this.
Use `cmd-D` (Mac), `File → ChangeDir` (Windows)
or just type `setwd('/path/to/my/working/dir')`.
Set a default with *preferences* or shortcut *properties*.

In this directory, you'll store flat files: text tables of data.
We'll use mostly `.csv` files: comma separated values.
Any file in Excel can be saved as '.csv', and vice versa.
Beware of of Excel automatically changing data values!



```
year,miles,price,make
2008,17638,14995,GMC
2003,174000,8500,Dodge
2001,1500,9998,Dodge
```

To load data, type `trucks <- read.csv("pickup.csv")`.
The data are then in working memory (RAM).

`trucks` is a `dataframe`: a matrix with names.
You can use index names and numbers to access the data.

```
trucks[1,] # the first observation
trucks[1:10,] # the first 10 observations
trucks[,1] # the first variable (year)
trucks$year # same thing
trucks[,’year’] # same thing again
trucks[trucks$miles>200000,] # some real clunkers
```

And call R’s functions on the data.

```
nrow(trucks) # sample size
summary(trucks) # summary of each variable
```

A note on data storage

Much of the world revolves around flat files (.txt, .csv, etc).
There are two other dominant broad storage/analysis modes.

Structured Query Language (SQL) databases

A model for fast relational queries on structured data:

```
select apple_id,apple_price from grocerylist where apple_col = green
```

Examples are MySQL, Oracle, SQLite, Teradata, ...

The analyst typically pulls data from the DB with an SQL query, writes this to a flat file, then reads the flat file into R.

Packages like RSQLite can also read directly from SQL DBs.

After extraction, you analyze the data *in memory*.

Distributed File Systems (DFS)

When the data is unstructured (log files, raw text documents) or too big to fit on a single server, we turn to DFS models.

Examples are Hadoop HDFS, Amazon S3.

The data is scattered across many machines,
with a key that tells you where all the pieces live.

Analysis uses algorithm frameworks like MapReduce:
partition data into small chunks, analyze each independently.

This is Big Data

Each chunk analysis (‘reduce’) uses the sort of tools we’ll cover in class. We’ll talk more about distributed algorithms later, and throughout class we always focus on scalable techniques.

Back to R

Basic Elements in R: numeric, factor, logical, character

The values in our dataframes all have a class.

- numeric values are just numbers (1, 2, 0.56, 10.2).
- factor variables are categories with levels ('lowfat','reg')
- logical values are either TRUE or FALSE.
- character strings are just words or messages ('hi mom').

We have plenty of tools to investigate and manipulate these:

`as.numeric(X), factor(X), class(X), levels(X)`

R has functions that look like `f(arg1, arg2, ...)`**.**

e.g., create new variables: `lprice = log(truck$price)`.

And add them to your data: `truck$lprice = lprice`.

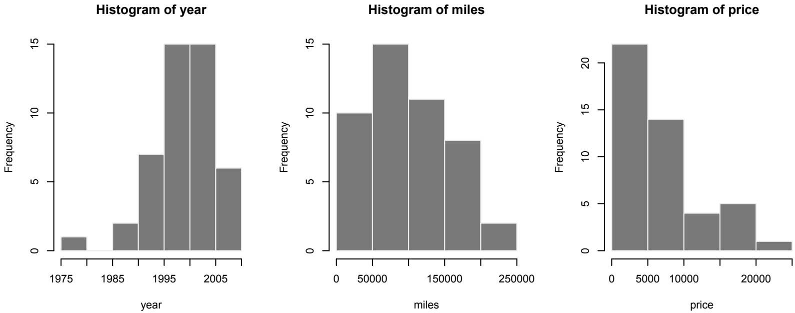
To find out how a function works, type `?f` or `help(f)`.

Plotting is super intuitive

Use `plot(mydata$X, lY)` or `plot(lY ~ mydata$X)`

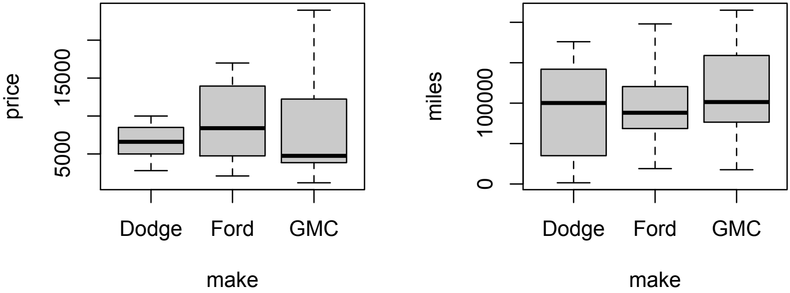
Let’s look at some basic plots...

The simple histogram for continuous variables



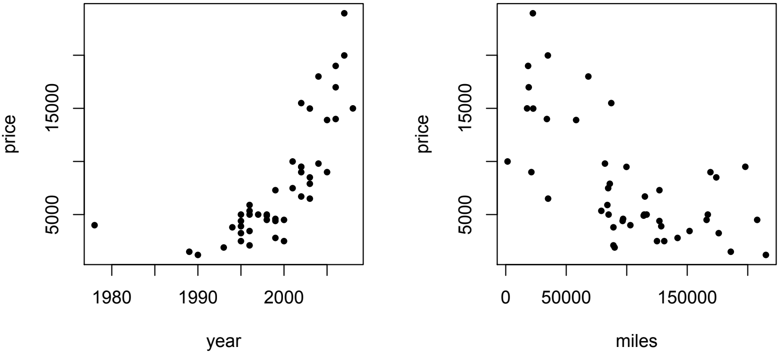
Data is binned and plotted bar height is the count in each bin.

Boxplots: summarizing conditional distributions

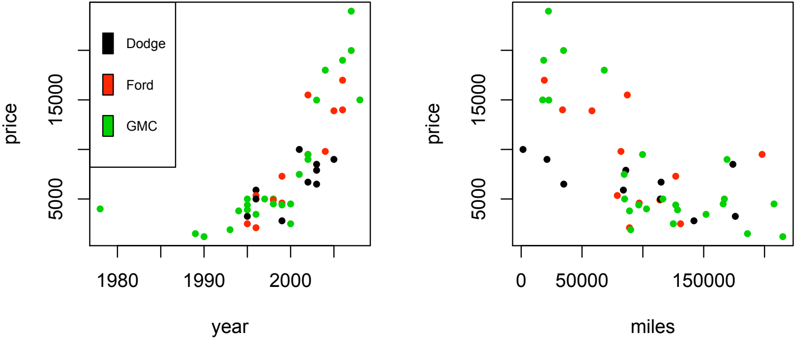


The box is the Interquartile Range (IQR; i.e., 25th to 75th %), with the median in bold. The whiskers extend to the most extreme point which is no more than 1.5 times the IQR width from the box.

Use scatterplots to compare variables.



And color them to see another dimension.



The scatterplot is mightier than you think...

Scatterplots are a fundamental unit of statistics.

If you're able to find and compare meaningful low-dimensional summary statistics, then you are winning the DM game.

- Humans are good at comparing a few variables.
- If we can put it in a picture, we can build intuition.
- Prediction is easy in low dimensions.

The key to good graphics is to reduce high-dimensional data to a few very informative summary statistics, then plot them. We'll focus on info visualization throughout this course.

A note on data visualization

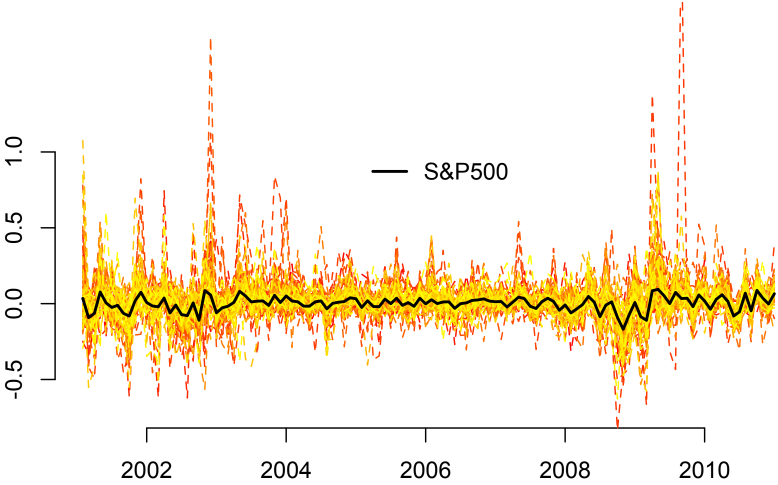
A plot without a concept of a variable is useless graphic.

Three broad pillars:

- Statistics: reducing dimension of your data to a few rich variables for comparison. Can be just picking two features to scatterplot, or can involve more complicated projections.
- Design: effective communication – with shapes, space, and color – for a given set of variable observations.
- Language: making it easy to move from Stats to Design.

They're all interconnected, but we'll focus on statistics.

Fancy plot: monthly stock returns



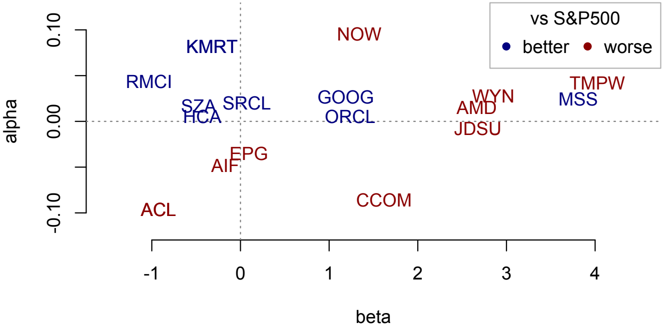
What do we learn?

Useful plot: market model coefficients

Fit a line between stock returns R_t and market returns M_t (SP).

$$R_t \approx \alpha + \beta M_t$$

α is money you make regardless of what the market does.
β is the asset's sensitivity to broad market movements.



Regression is king

```
fit <- glm(log(price)~ year+make, data=trucks)
summary(fit)  # glm = generalized linear model
...
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -195.22731   25.18120  -7.753 1.24e-09 ***
year           0.10196    0.01259   8.099 4.07e-10 ***
makeFord       0.13987    0.19786   0.707    0.484
makeGMC        0.16202    0.17586   0.921    0.362
```

This is the model (familiarize yourself with the notation!)

$$\mathbb{E}[\log(\texttt{price})] = \beta_0 + \texttt{year}\beta_{\texttt{year}} + \mathbb{1}_{[\texttt{ford}]}\beta_{\texttt{ford}} + \mathbb{1}_{[\texttt{gmc}]}\beta_{\texttt{gmc}}.$$

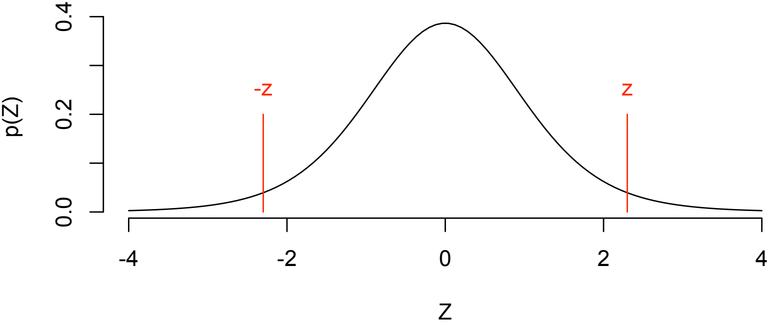
See other models and syntax in `pickups.R`.
We'll see more next week: review the basics before then.

Review: Hypothesis Testing

What is `Pr(>|t|)`? Why the $\star\star\star$?

For a test of $\beta = 0$ vs $\beta \neq 0$, the test stat is $z_\beta = \hat{\beta}/s_{\hat{\beta}}$: how many standard deviations is our estimate away from zero?

The p-value is then $\mathrm{P}(|Z| > |z_{\hat{\beta}}|)$, with $Z \sim \mathrm{N}(0, 1)$.



A single test

A p-value is probability of a test stat farther into the distribution tail than what you've observed, *if the null hypothesis is true*.

Testing procedure: Choose a cut-off 'α' for your p-value '*p*', and conclude significance (e.g., variable association) for $p < \alpha$.

This is justified as giving only α probability of a false-positive. For example, in regression, $\hat{\beta} \neq 0$ only if its *p*-value is less than the accepted risk of a false discovery for each coefficient.

The problem of multiplicity

α is for a single test. If you repeat many tests, about $\alpha \times 100\%$ of the null tests should erroneously pop up as significant.

Suppose that 5 of 100 regression coefficients are actually influential, and that you find all of them significant.

Test the rest of them at $\alpha = 0.05$:

Since you reject H_0 for 5% of the useless 95 variables,
$4.75/9.75 \approx 50\%$ of significant tests are false discoveries!

This is called the False Discovery Proportion (FDP).
It can be really big with a small true non-Null rate.

False Discovery Rate

Big data is about making *many* tough decisions.

Instead of focusing on single tests, we'll consider

$$\text{FD Proportion} = \frac{\#\text{ false positives}}{\#\text{ tests called significant}}$$

FDP is a property of our fitted model. We can't know it.

But we can control its expectation:

False Discovery Rate, FDR $= \mathbb{E}[\text{FDP}]$.

It is the multivariate (aggregate) analogue of α.

False Discovery Rate control

Suppose we want to be sure that FDR $\leq q$ (say, 0.1).

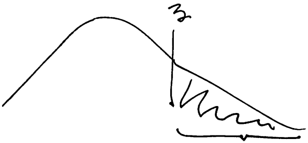
The Benjamini + Hochberg (BH) algorithm:

- Rank your N p-values, smallest to largest, $p_{(1)} \dots p_{(N)}$.
- Set the p-value cut-off as $p^{\star} = \max \left\{ p_{(k)} : p_{(k)} \leq q \frac{k}{N} \right\}$.

If your rejection region is p-values $\leq p^{\star}$, then FDR $\leq q$.
Caution: assumes (rough) independence between tests.

♦ Motivating FDR control

P-values are uniformly distributed under the Null.



$\varphi(z) = P(Z > z)$

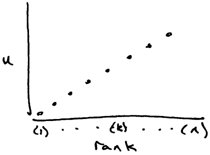
- Uniform CDF is $P(U < u) = u$
- $P(\varphi(Z) < \varphi(z))$

  $= P(Z > z) = \varphi(z)$

  $\Rightarrow \varphi(z) \sim$ Uniform.

Rank statistics from a uniform have Expectation $\mathbb{E}[u_{(k)}] = \frac{k}{n}$ for sample size 'n'.

$\Longrightarrow$ We can plot the mean w/ slope $\frac{1}{n}$



q is the slope of a shallower line defining our rejection region.

♦♦ Demonstrating FDR control

Given our independence assumption the proof is simple.

N = total # tests, N_0 = # that are Null.

Say $R(u)$ is total w̄ p-val $\le u$, & $r(u)$ is #Null w̄ $p \le u$.

A cut-off equivalent to BH is

$$u^* = \max\left\{u : u \le q\frac{R(u)}{N}\right\}$$

and then $\frac{1}{R(u^*)} \le q/Nu^*$.

$$\Longrightarrow \text{FDP is } \frac{r(u^*)}{R(u^*)} \le q\frac{r(u^*)}{N u^*} \text{ w̄ } \boxed{\text{Expectation } q\frac{N_0}{N}}$$

(since $E\left[r(u)/u\right] = N_0$ for unif. & indep. Null p-vals.)

FDR roundup

We introduced the problem of *multiplicity* by saying that given α (p-value cutoffs) can lead to big FDR: $\alpha \to q(\alpha)$

B+H reverse the relationship – it's a recipe for the $\alpha(q)$ that will give you whatever FDR q you want: $q \to \alpha(q)$

FDR is *the* way to summarize risk when you have many tests. You'll never think about testing the same again!

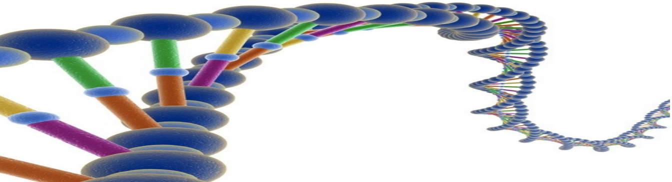
Example: multiple testing in GWAS

GWAS: genome-wide association studies
Scan large DNA sequences for association with disease.

Single-nucleotide polymorphisms (SNPs) are paired DNA locations that vary across chromosomes. The allele that occurs most often is major (A), and the other is minor (a).

Question: Which variants are associated with increased risk?

Then investigate why + how.



Allele Association

A common SNP summary is Allele Frequency (AF):

freq(a): AA $\rightarrow$ 0 Aa/aA $\rightarrow$ 1 aa $\rightarrow$ 2

Question: which SNP AF distributions that vary with disease?

Answer: a huge number of AF×disease contingency tables.

An example for type-2 Diabetes mellitus:

	minor AF: `rs6577581`		
DM2 STATUS	0	1	2
case	357	72	27
control	428	54	1

χ^2 *stat is 32 on 2 df for p* $= 9\times 10^{-8}$

Cholesterol

Willer et al, Nat Gen 2013 describe meta-analysis of GWAS for Cholesterol levels. We'll focus on the 'bad' LDL Cholesterol.

At each of 2.5 million SNPs, they fit the linear regression

$$\mathbb{E}[\mathit{LDL}] = \alpha + \beta \mathit{AF}$$

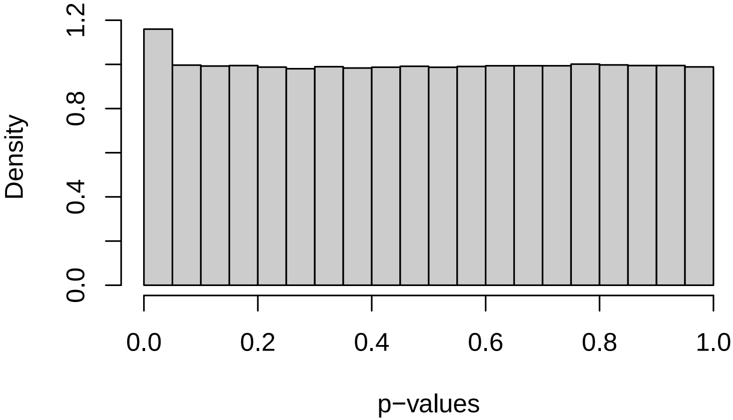
Where AF is allele frequency for the 'trait increasing allele'.

2.5 mil SNP locations

$\Rightarrow$ 2.5 mil tests of $\beta \neq 0$

$\Rightarrow$ 2.5 mil p-values!

Cholesterol GWAS P-values: Which are significant?

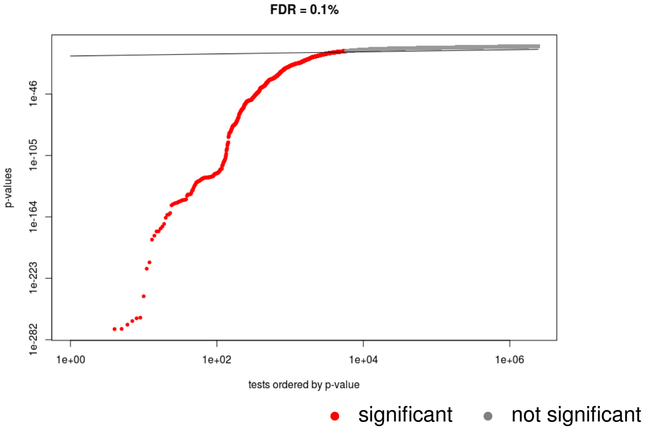


The tiny spike down by zero is our only hope for discovery.

Recall: p-values from the null distribution are uniform.

Controlling the False Discovery Rate

The slope of the FDR-cut line is q/[# of variables].



Lots of action!

4000 significant tests at FDR of `1e-3`

(so only 4-5 are false discoveries).

All there is to know about FDR

p-values from the null distribution are uniform, and *N* of them ranked and plotted should lie along a line with slope $1/N$.

FDP is the number of false discoveries divided by the number of test results called significant. You don't know the FDP.

You can control $FDR = \mathbb{E}[FDP]$ to be $\leq q$ amongst *N* tests

- rank and plot p-values against rank/*N*
- draw a line with slope q/N
- find the max point where p-values cross this line, and use that point to set your rejection region.

R roundup

Quitting R: usually save the script, not the workspace.
The workspace is an .rda image, while the .R script is just text.

Stay up-to-date: Make sure you have latest versions.
Stay organized: keep track of your work.
Keep things simple, and don't panic.

Consider using shared drives to collaborate. Or, even better, create a group github repo and use version control.

R is a fantastic tool and there is tons you can do, but don't worry about learning everything right away.

Plenty of resources out there. Also don't hesitate to use me and your friends (and google!) to avoid frustration.

Week 1 Homework

Homescan data from Nielson

70K Households
and all of their purchases.

Homescan data dictionary
is on the course site.

Ben and Jerry's subset is in
`BenAndJerry.csv`

Week 1 Homework

See `benjerry_start.R` for code to get you started.

[1] Explore the data and visualize: what variables are interesting? Choose a few, plot them together, and tell a story.

[2] Describe the regression model in the code. Improve it?

[3] Take the p-values from your regression and look for evidence of association. Relate what you learn to your story from [1]. How many true discoveries do you think we have?

[+] Why should we be worried about our FDR control here?

You are encouraged to work in groups of up to 4.
Tell me what you find (and how). Do not hand in R-code!
Homeworks are out of 5: Amazing, Great, Good, Poor, Ouch.