The Data Science Track

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Why do data science?

"It is not the critic who counts: not the man who points out how the strong man stumbles or where the doer of deeds could have done better. The credit belongs to the man who is actually in the arena, whose face is marred by dust and sweat and blood, who strives valiantly, who errs and comes up short again and again, because there is no effort without error or shortcoming, but who knows the great enthusiasms, the great devotions, who spends himself for a worthy cause; who, at the best, knows, in the end, the triumph of high achievement, and who, at the worst, if he fails, at least he fails while daring greatly, so that his place shall never be with those cold and timid souls who knew neither victory nor defeat."

Theodore Roosevelt, 26th President of the United States

Statistics and the science game

The key challenge in data science

"Ask yourselves, what problem have you solved, ever, that was worth solving, where you knew knew all of the given information in advance? Where you didn't have a surplus of information and have to filter it out, or you didn't have insufficient information and have to go find some?"


Dan Myer, Mathematics Educator

The key word in data science is not data; it is science

About us

Data intensive statistics in biology and medicine

- Brian Caffo
  - Website http://www.bcaffo.com/
  - Twitter @bcaffo
  - Github https://github.com/bcaffo
- Jeff Leek
  - Website http://biostat.jhsph.edu/~jleek/, http://simplystatistics.org/
  - Twitter @jtleek
  - Github https://github.com/jtleek
- Roger Peng
  - Website http://www.biostat.jhsph.edu/~rpeng/,http://simplystatistics.org/
  - Twitter @rdpeng
  - Github https://github.com/rdpeng

Why data science?

http://www.economist.com/node/15579717

Why data science?

McKinsey Global Institute

June 2011

Big data: The next frontier for innovation, competition, and productivity

http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation

Why statistical data science?

The New York Times

For Today's Graduate, Just One Word: Statistics

By STEVE LOHR
Published: August 5, 2009

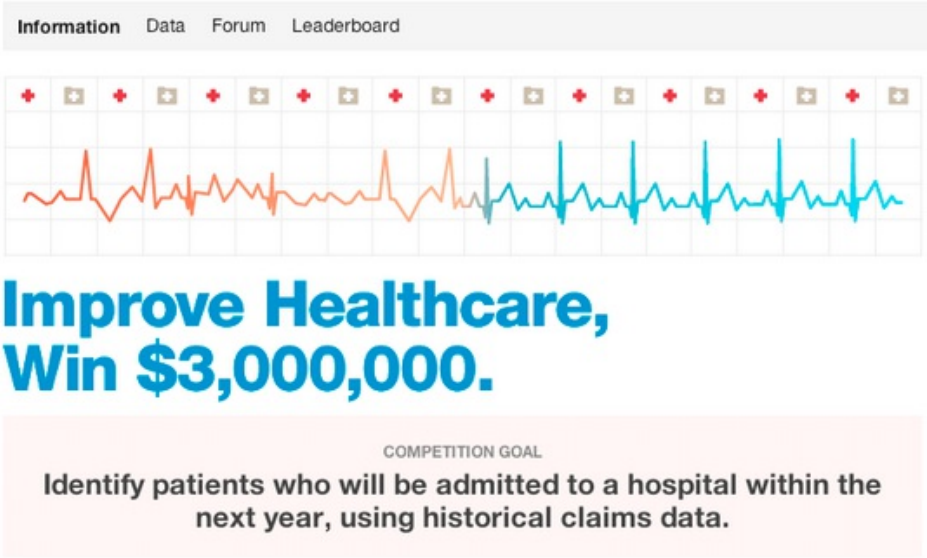
MOUNTAIN VIEW, Calif. — At Harvard, Carrie Grimes majored in anthropology and archaeology and ventured to places like Honduras, where she studied Mayan settlement patterns by mapping where artifacts were found. But she was drawn to what she calls "all the computer and math stuff" that was part of the job.

TWITTER
LINKEDIN
COMMENTS (58)
SIGN IN TO E-MAIL

http://www.nytimes.com/2009/08/06/technology/06stats.html?_r=0

Why are you lucky?

Why are you lucky?



Heritage Health Prize

Why R?

http://www.nytimes.com/2009/01/07/technology/business-computing/07program.html?pagewanted=all

Why R?

- It is free
- It has a comprehensive set of packages
  - Data access
  - Data cleaning
  - Analysis
  - Data reporting
- It has one of the best development enviroments - Rstudio http://www.rstudio.com/
- It has an amazing ecosystem of developers
- Packages are easy to install and "play nicely together"

Who is a data scientist?

Daryl Morey

Who is a data scientist?

Hilary Mason

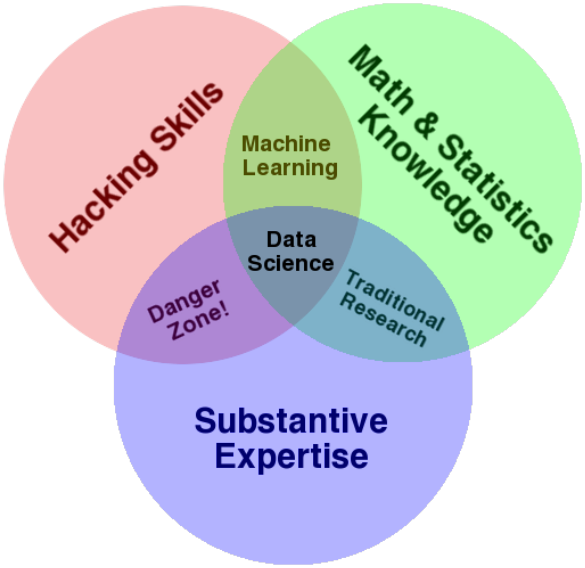
Who is a data scientist?

Daphne Koller

Who is a data scientist?

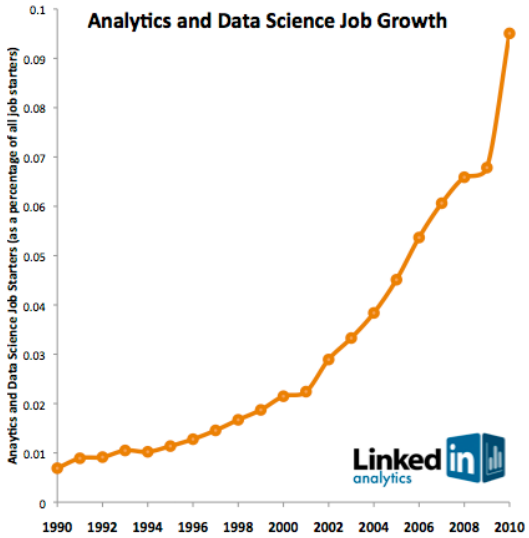
Nate Silver

Our goal



Drew Conway

Plus jobs



http://radar.oreilly.com/2011/09/building-data-science-teams.html

This course

- Introducing you to the track
- Getting tools set up
- Giving you basic background

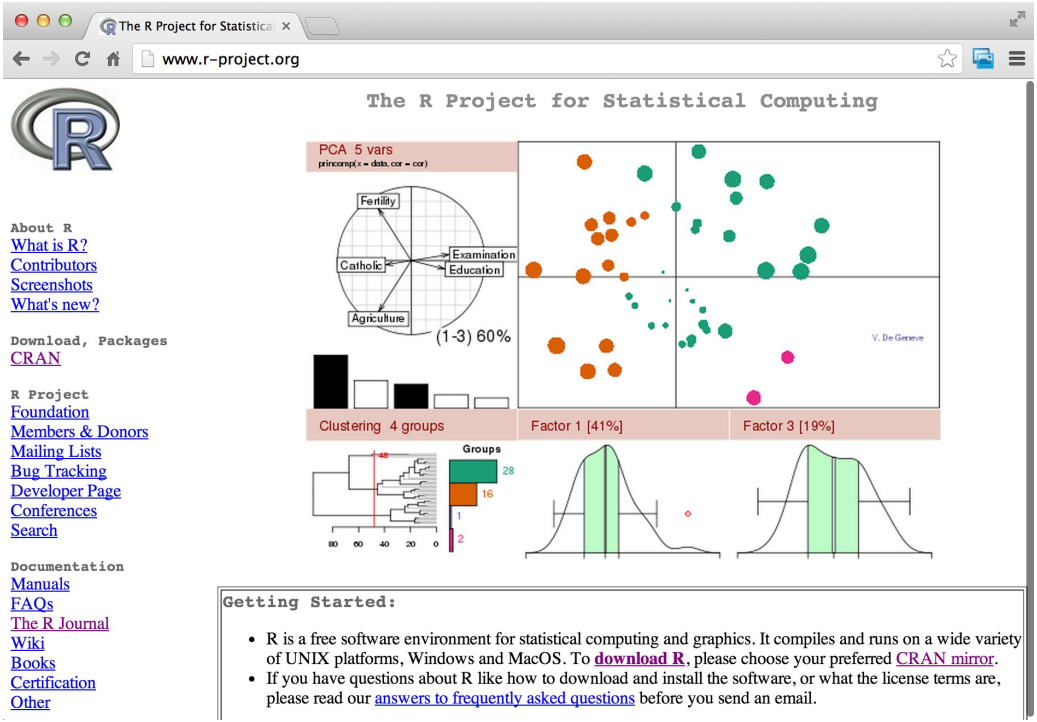
The Data Scientist's Toolbox

Johns Hopkins Bloomberg School of Public Health

What do data scientists do?

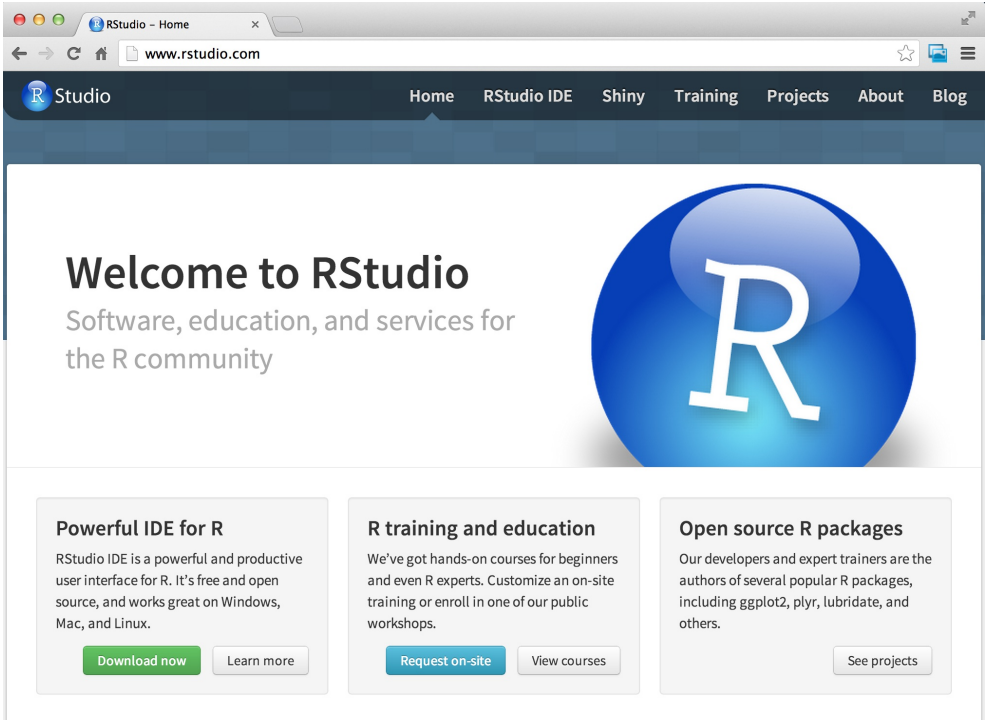
- Define the question
- Define the ideal data set
- Determine what data you can access
- Obtain the data
- Clean the data
- Exploratory data analysis
- Statistical prediction/modeling
- Interpret results
- Challenge results
- Synthesize/write up results
- Create reproducible code
- Distribute results to other people

The main workhorse of data science



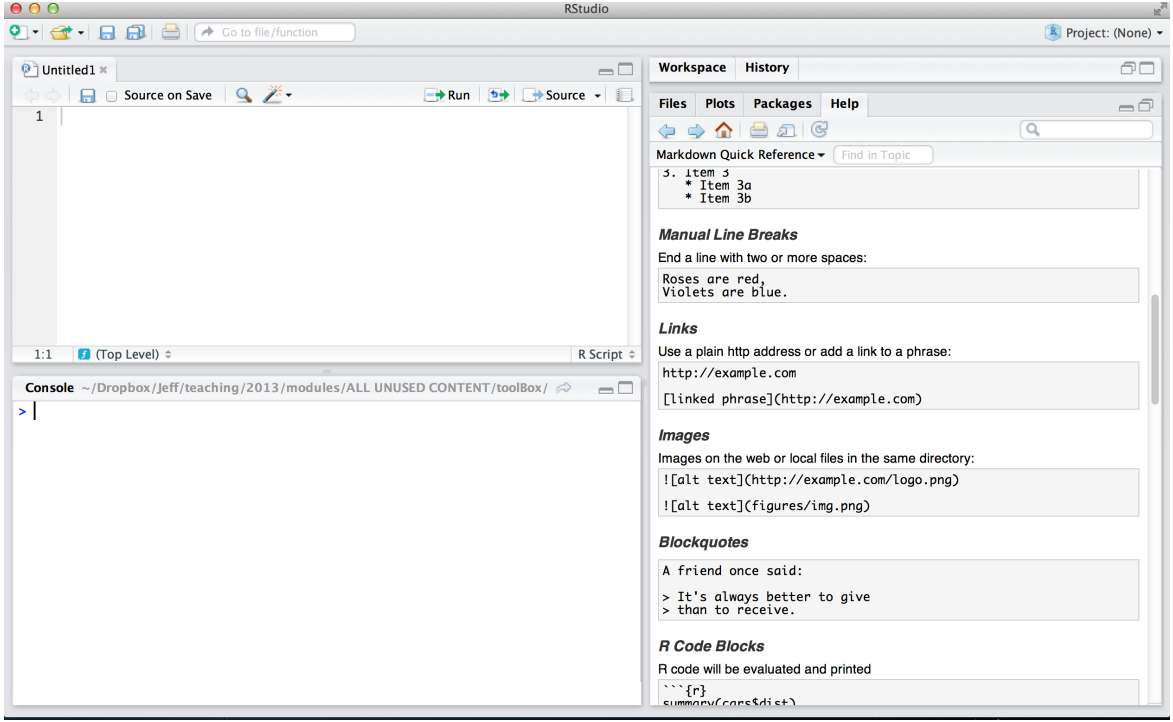
http://www.r-project.org/

Where we will work on coding



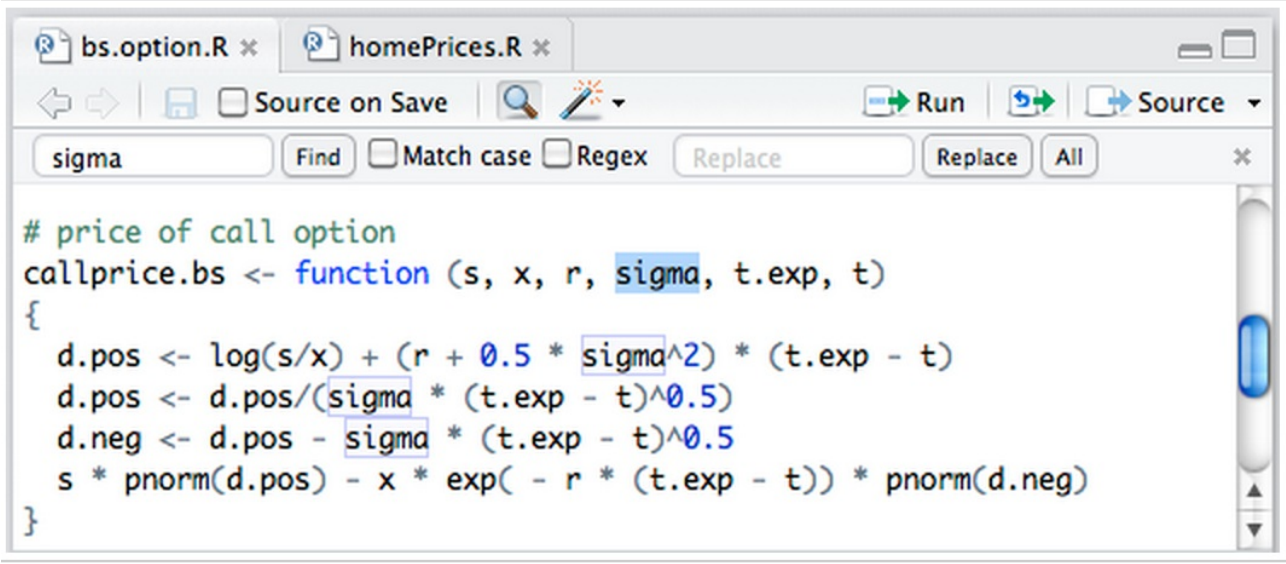
http://www.rstudio.com/

Rstudio's interface



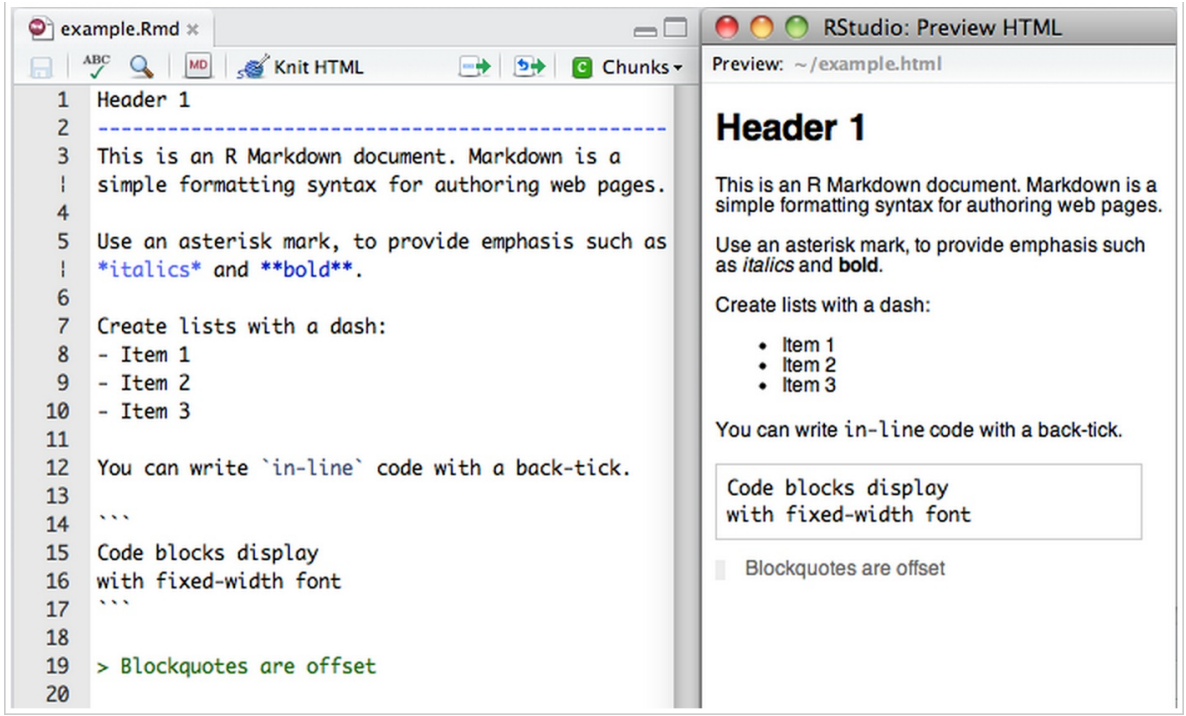
http://www.rstudio.com/

Primary file types - R script



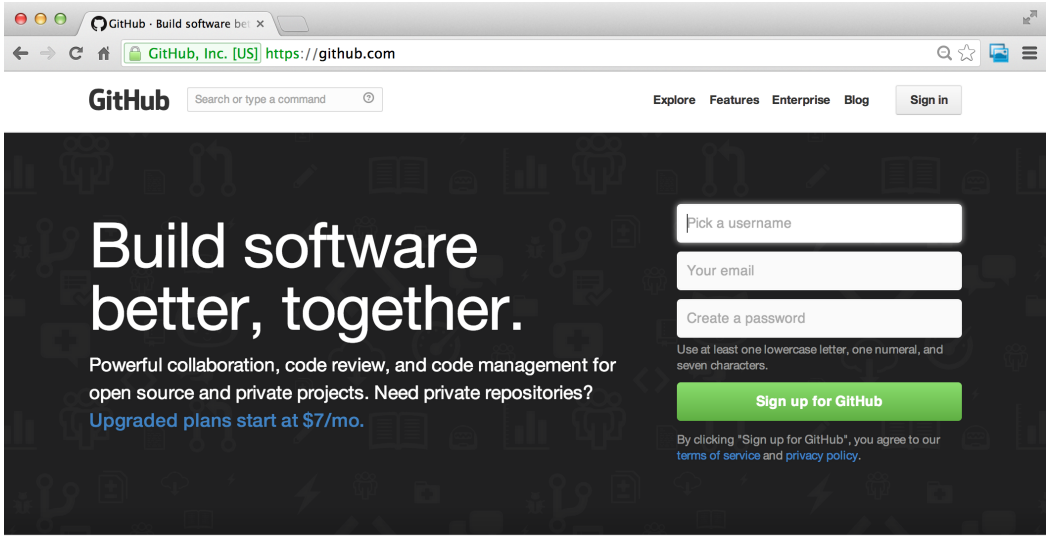
http://www.rstudio.com/ide/docs/using/source

Primary file types - R markdown document



http://www.rstudio.com/ide/docs/authoring/using_markdown

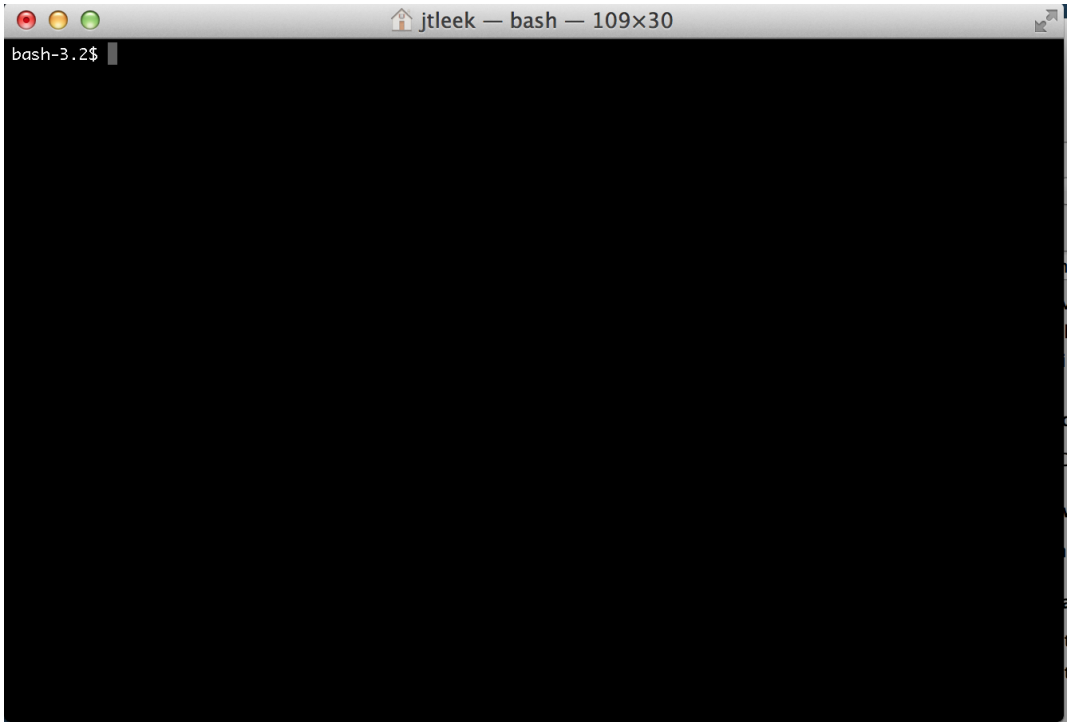
Sharing your results - Github & Git



Why you'll love GitHub.

Powerful features to make software development more collaborative.

Where to run Github commands - the shell



Getting help

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Asking questions

- In a standard class
  - There are 30-100 people
  - You raise your hand and ask a question
  - The instructor responds
- In a MOOC
  - There are almost 100,000 people
  - You post a question to the message board
  - Others vote on your questions
  - Your instructor responds (as often as possible)
  - Your peers respond (as often as possible)

Often the fastest answer is the one you find yourself

- It's important to try to answer your own questions first
- If the answer to your question is in the help file or the top hit on Google, the answer to your question will be, "Read the documentation" or "Google it" (http://lmgtfy.com/)
- If you figure out the answer and see the same questions on the forum, post the solution you found

Some important R functions

Access help file

```
?rnorm
```

Search help files

```
help.search("rnorm")
```

Get arguments

```
args("rnorm")
```

```
function (n, mean = 0, sd = 1)
NULL
```

Some important R functions

See code

```
rnorm
```

```
function (n, mean = 0, sd = 1)
.External(C_rnorm, n, mean, sd)
<bytecode: 0x7f9173a9f630>
<environment: namespace:stats>
```

R reference card

http://cran.r-project.org/doc/contrib/Short-refcard.pdf

How to ask an R question

- What steps will reproduce the problem?
- What is the expected output?
- What do you see instead?
- What version of the product (e.g. R, packages, etc.) are you using?
- What operating system?

How to ask a data analysis question

- What is the question you are trying to answer?
- What steps/tools did you use to answer it?
- What did you expect to see?
- What do you see instead?
- What other solutions have you thought about?

Be specific in the title of your questions

- Bad:
  - HELP! Can't fit linear model!
  - HELP! Don't understand PCA!
- Better
  - R 2.15.0 lm() function produces seg fault with large data frame, Mac OS X 10.6.3
  - Applied principal component analysis to a matrix - what are U, D, and V^T?
- Even better
  - R 2.15.0 lm() function on Mac OS X 10.6.3 -- seg fault on large data frame
  - Using principal components to discover common variation in rows of a matrix, should I use U, D or V^T?

Etiquette for forums/help sites: DOs

- Describe the goal
- Be explicit
- Provide the minimum information
- Be courteous (never hurts)
- Follow up and post solutions
- Use the forums rather than email

Etiquette for forums/help sites: DON'Ts

- Immediately assume you found a bug
- Grovel as a substitute for doing your work
- Post homework questions on mailing lists (people don't like doing your homework)
- Email multiple mailing lists at once/the wrong mailing list
- Ask others to fix your code without explaining the problem
- Ask about general data analysis questions on R forums.

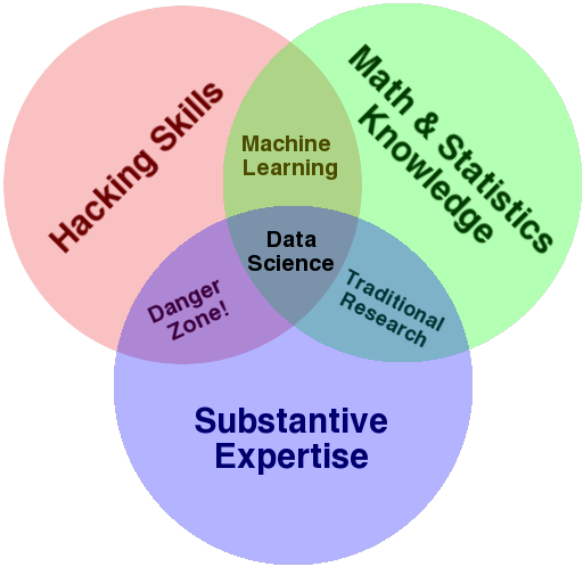
Credits

- Roger's Getting Help Video
- Inspired by Eric Raymond's "How to ask questions the smart way"

Finding answers

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

One of the key data science traits



Drew Conway

Key characteristics of hackers

- Willing to find answers on their own
- Knowledgeable about where to find answers on their own
- Unintimidated by new data types or packages
- Unafriad to say they don't know the answer
- *Polite but relentless*

Google knows it too

Where to look for different types of questions

- R programming (see also: http://bit.ly/Ufaadn)
  - Search the archive of the class forums
  - Read the manual/help files
  - Search on the web
  - Ask a skilled friend
  - Post to the class forums
  - Post to the R mailing list or Stackoverflow
- Data Analysis/Statistics
  - Search the archive of the class forums
  - Search on the web
  - Ask a skilled friend
  - Post to the class forums

A note on Googling data science questions

- The best place to start for general questions is the forums
- Stackoverflow (use the tag "[r]"),R mailing list for software questions, CrossValidated for more general questions
- Otherwise Google "[data type] data analysis" or "[data type] R package"
- Try to identify what data analysis is called for your data type
  - Biostatistics for medical data
  - Data Science for data from web analytics
  - Machine learning for data in computer science/computer vision
  - Natural language processing for data from texts
  - Signal processing for data from electrical signals
  - Business analytics for data on customers
  - Econometrics for economic data
  - Statistical process control for data about industrial processes

Credits

- Roger's Getting Help Video
- Inspired by Eric Raymond's "How to ask questions the smart way"

R Programming Overview

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

R programming content

- Data types
- Subsetting
- Reading and writing data
- Control structures
- Functions
- Scoping
- Vectorized operations
- Dates and times
- Debugging
- Simulation
- Optimization

Reading Lines of a Text File

`readLines` can be useful for reading in lines of webpages

```
## This might take time
con <- url("http://www.jhsph.edu", "r")
x <- readLines(con)
> head(x)
[1] "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 Transitional//EN\">"
[2] ""
[3] "<html>"
[4] "<head>"
[5] "\t<meta http-equiv=\"Content-Type\" content=\"text/html;charset=utf-8
```

Something's Wrong!

How do you know that something is wrong with your function?

- What was your input? How did you call the function?
- What were you expecting? Output, messages, other results?
- What did you get?
- How does what you get differ from what you were expecting?
- Were your expectations correct in the first place?
- Can you reproduce the problem (exactly)?

lapply

`lapply` takes three arguments: a list `X`, a function (or the name of a function) `FUN`, and other arguments via its ... argument. If `X` is not a list, it will be coerced to a list using `as.list`.

```
> lapply
function (X, FUN, ...)
{
    FUN <- match.fun(FUN)
    if (!is.vector(X) || is.object(X))
        X <- as.list(X)
    .Internal(lapply(X, FUN))
}
```

The actual looping is done internally in C code.

Getting and Cleaning Data Overview

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Getting and Cleaning Data Content

- Raw vs. tidy data
- Downloading files
- Reading data
  - Excel, XML, JSON, MySQL, HDF5, Web, ...
- Merging data
- Reshaping data
- Summarizing data
- Finding and replacing
- Data resources

Connecting and listing databases

```
ucscDb <- dbConnect(MySQL(), user = "genome", host = "genome-mysql.cse.ucsc.edu")
result <- dbGetQuery(ucscDb, "show databases;")
dbDisconnect(ucscDb)
result
```

Merging data - merge()

```
mergedData2 <- merge(reviews, solutions, by.x = "solution_id", by.y = "id",
    all = TRUE)
head(mergedData2[, 1:6], 3)
reviews[1, 1:6]
```

Raw versus processed data

Raw data

- The original source of the data
- Often hard to use for data analyses
- Data analysis *includes* processing
- Raw data may only need to be processed once

http://en.wikipedia.org/wiki/Raw_data

Processed data

- Data that is ready for analysis
- Processing can include merging, subsetting, transforming, etc.
- There may be standards for processing
- All steps should be recorded

http://en.wikipedia.org/wiki/Computer_data_processing

Exploratory Analysis Overview

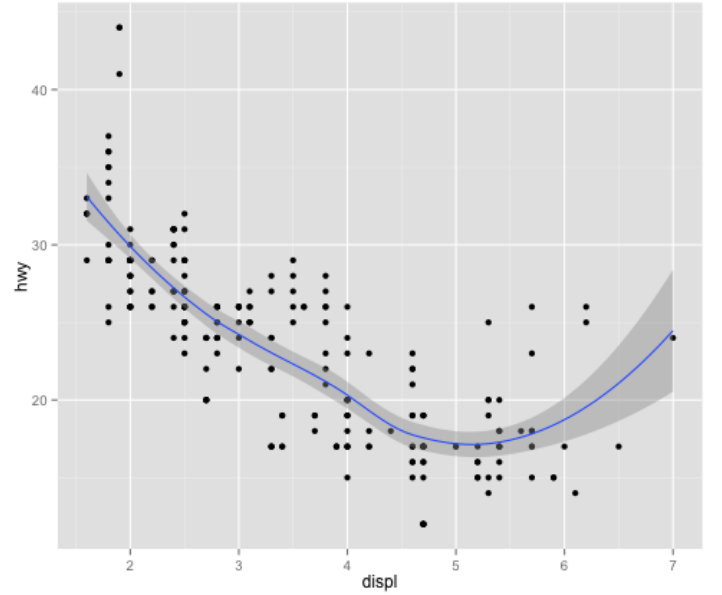
Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Exploratory Analysis Content

- Principles of Analytic Graphics
- Exploratory graphs
- Plotting Systems in R
  - base
  - lattice
  - ggplot2
- Hierarchical clustering
- K-Means clustering
- Dimension reduction

Adding a geom

```
qplot(displ, hwy, data = mpg, geom = c("point", "smooth"))
```

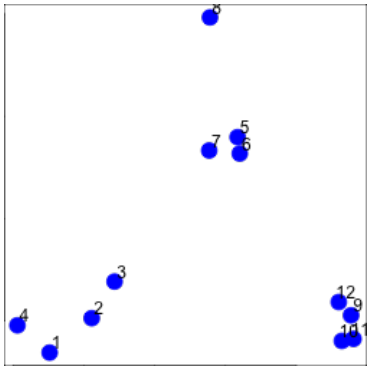


Principles of Analytic Graphics

- Principle 1: Show comparisons
- Principle 2: Show causality, mechanism, explanation
- Principle 3: Show multivariate data
- Principle 4: Integrate multiple modes of evidence
- Principle 5: Describe and document the evidence
- Principle 6: Content is king

K-means clustering - example

```
set.seed(1234)
par(mar = c(0, 0, 0, 0))
x <- rnorm(12, mean = rep(1:3, each = 4), sd = 0.2)
y <- rnorm(12, mean = rep(c(1, 2, 1), each = 4), sd = 0.2)
plot(x, y, col = "blue", pch = 19, cex = 2)
text(x + 0.05, y + 0.05, labels = as.character(1:12))
```



Reproducible Research Overview

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Reproducible Research Content

- Structure of a Data Analysis
- Organizing a Data Analysis
- Markdown
- LaTeX
- R Markdown
- Evidence-based data analysis
- RPubs

Steps in a data analysis

- Define the question
- Define the ideal data set
- Determine what data you can access
- Obtain the data
- Clean the data
- Exploratory data analysis
- Statistical prediction/modeling
- Interpret results
- Challenge results
- Synthesize/write up results
- Create reproducible code

Data analysis files

- Data
  - Raw data
  - Processed data
- Figures
  - Exploratory figures
  - Final figures
- R code
  - Raw scripts
  - Final scripts
  - R Markdown files (optional)
- Text
  - Readme files
  - Text of analysis

Statistical Inference Overview

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Statistical Inference Content

- Basic probability
- Likelihood
- Common distributions
- Asymptotics
- Confidence intervals
- Hypothesis tests
- Power
- Bootstrapping
- Non-parametric tests
- Basic bayesian statistics

Example

Suppose that the proportion of help calls that get addressed in a random day by a help line is given by

$$f(x) = \begin{cases} 2x & \text{for } 1 > x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Is this a mathematically valid density?

The normal distribution

- A random variable is said to follow a **normal** or **Gaussian** distribution with mean μ and variance σ^2 if the associated density is

$$(2\pi\sigma^2)^{-1/2}e^{-(x-\mu)^2/2\sigma^2}$$

  If X a RV with this density then $E[X] = \mu$ and $Var(X) = \sigma^2$

- We write $X \sim N(\mu, \sigma^2)$
- When $\mu = 0$ and $\sigma = 1$ the resulting distribution is called **the standard normal distribution**
- The standard normal density function is labeled ϕ
- Standard normal RVs are often labeled Z

Example bootstrap code

```
B <- 1000
n <- length(gmVol)
resamples <- matrix(sample(gmVol,
                           n * B,
                           replace = TRUE),
                    B, n)
medians <- apply(resamples, 1, median)
sd(medians)
[1] 3.148706
quantile(medians, c(.025, .975))
    2.5%    97.5%
582.6384 595.3553
```

Regression Models Overview

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Regression Models Content

- Linear regression
- Multiple Regression
- Confounding
- Residuals and diagnostics
- Prediction using linear models
- Model misspecification
- Scatterplot smoothing/splines
- Machine learning via regression
- Resampling inference in regression, bootstrapping, permutation tests
- Weighted regression
- Mixed models (random intercepts)

A historically famous idea, Regression to the Mean

- Why is it that the children of tall parents tend to be tall, but not as tall as their parents?
- Why do children of short parents tend to be short, but not as short as their parents?
- Why do parents of very short children, tend to be short, but not a short as their child? And the same with parents of very tall children?
- Why do the best performing athletes this year tend to do a little worse the following?

Basic regression model with additive Gaussian errors

- Least squares is an estimation tool, how do we do inference?
- Consider developing a probabilistic model for linear regression

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

- Here the ϵ_i are assumed iid $N(0, \sigma^2)$.
- Note, $E[Y_i \mid X_i = x_i] = \mu_i = \beta_0 + \beta_1 x_i$
- Note, $Var(Y_i \mid X_i = x_i) = \sigma^2$.
- Likelihood equivalent model specification is that the Y_i are independent $N(\mu_i, \sigma^2)$.

Multivariable regression analyses

- An insurance company is interested in how last year's claims can predict a person's time in the hospital this year.
  - They want to use an enormous amount of data contained in claims to predict a single number. Simple linear regression is not equipped to handle more than one predictor.
- How can one generalize SLR to incoporate lots of regressors for the purpose of prediction?
- What are the consequences of adding lots of regressors?
  - Surely there must be consequences to throwing variables in that aren't related to Y?
  - Surely there must be consequences to omitting variables that are?

Practical Machine Learning Overview

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Practical Machine Learning Content

- Prediction study design
- Types of Errors
- Cross validation
- The caret package
- Plotting for prediction
- Preprocessing
- Predicting with regression
- Predicting with trees
- Boosting
- Bagging
- Model blending
- Forecasting

Basic terms

In general, **Positive** = identified and **negative** = rejected. Therefore:

- **True positive** = correctly identified
- **False positive** = incorrectly identified
- **True negative** = correctly rejected
- **False negative** = incorrectly rejected

Medical testing example:

- **True positive** = Sick people correctly diagnosed as sick
- **False positive**= Healthy people incorrectly identified as sick
- **True negative** = Healthy people correctly identified as healthy
- **False negative** = Sick people incorrectly identified as healthy.

http://en.wikipedia.org/wiki/Sensitivity_and_specificity

Correlated predictors

```
library(caret)
library(kernlab)
data(spam)
inTrain <- createDataPartition(y = spam$type, p = 0.75, list = FALSE)
training <- spam[inTrain, ]
testing <- spam[-inTrain, ]

M <- abs(cor(training[, -58]))
diag(M) <- 0
which(M > 0.8, arr.ind = T)
```

```
##        row col
## num415  34  32
## direct  40  32
## num857  32  34
## num857  32  40
```

Basic idea behind boosting

1. Start with a set of classifiers $h_1, \ldots, h_k$
   - Examples: All possible trees, all possible regression models, all possible cutoffs.
2. Create a classifier that combines classification functions: $f(x) = \text{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$.
   - Goal is to minimize error (on training set)
   - Iterative, select one h at each step
   - Calculate weights based on errors
   - Upweight missed classifications and select next h

Adaboost on Wikipedia

http://webee.technion.ac.il/people/rmeir/BoostingTutorial.pdf

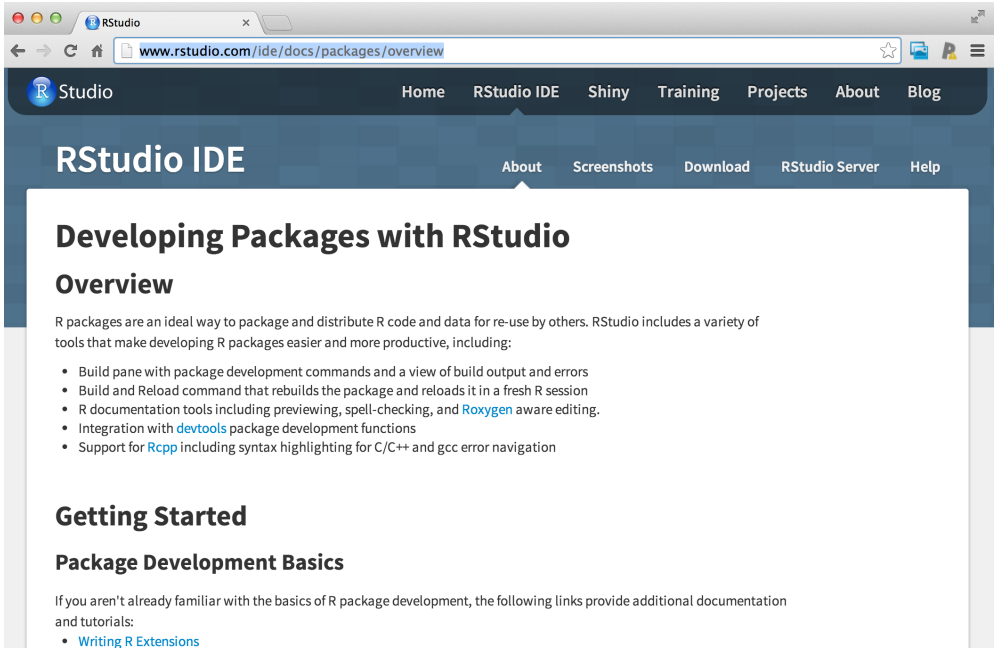
Building Data Products Overview

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Building Data Products Content

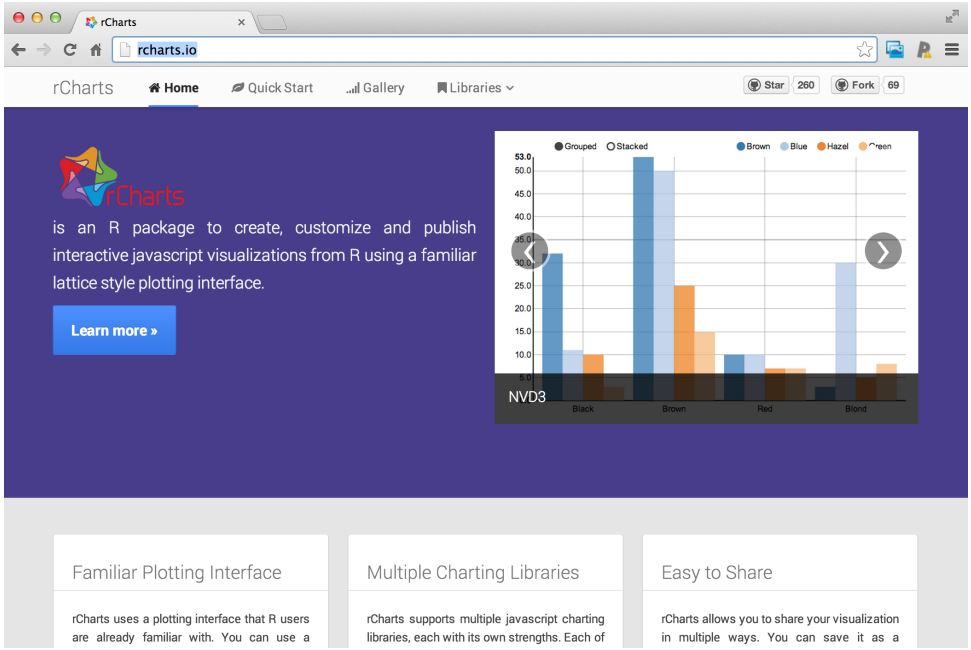
- R packages
  - devtools
  - roxygen
  - testthat
- rCharts
- Slidify
- Shiny

R packages - for the engineers



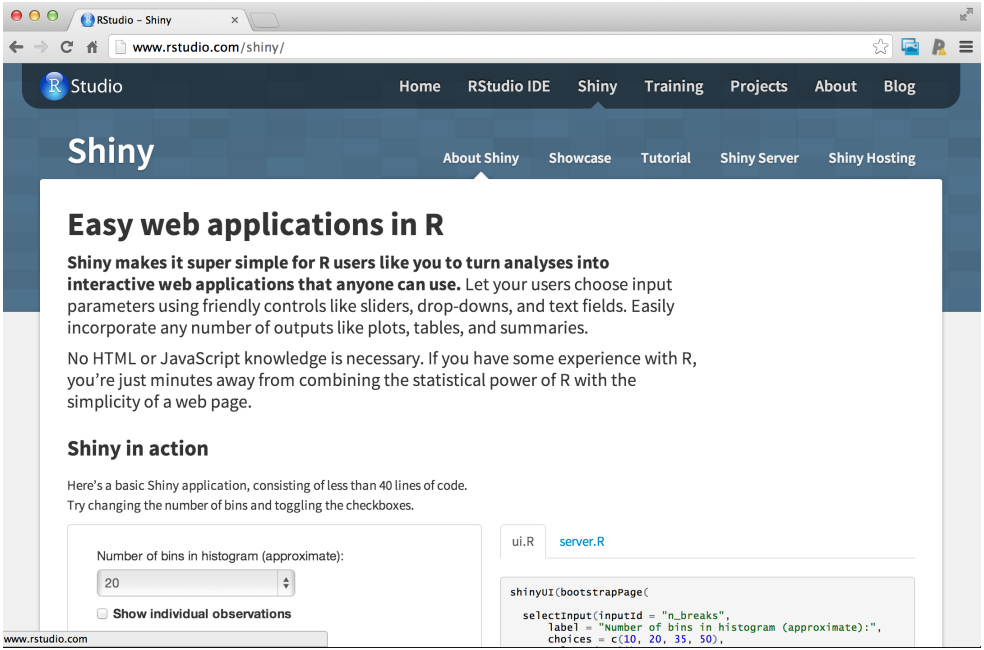
http://cran.r-project.org/web/packages/ http://www.rstudio.com/ide/docs/packages/overview

rCharts - for marketing



http://rcharts.io/ http://ramnathv.github.io/rChartsNYT/

Shiny - for your users



http://www.rstudio.com/shiny/ http://www.rstudio.com/shiny/showcase/

Introduction to the Command Line Interface

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

What is the Command Line Interface?

Nearly ever computer comes with a CLI

- Windows: Git Bash (See "Introduction to Git")
- Mac/Linux: Terminal

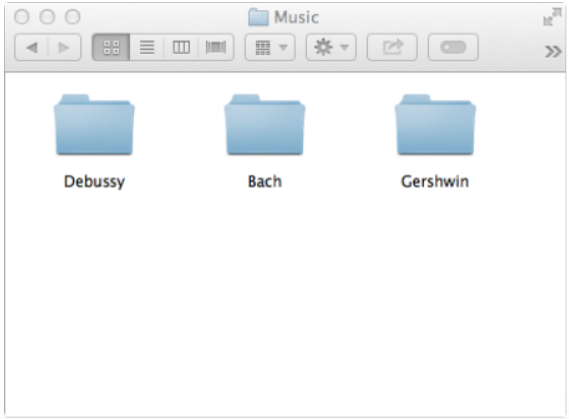
What can the CLI do?

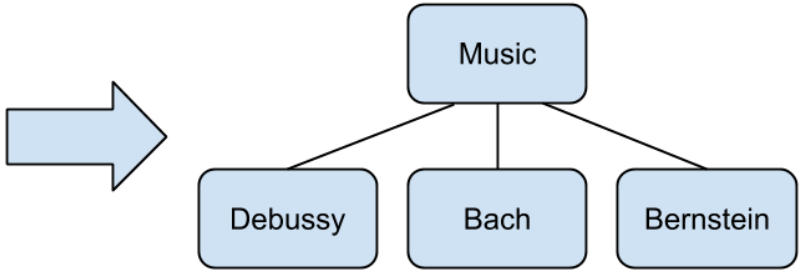
The CLI can help you:

- Navigate folders
- Create files, folders, and programs
- Edit files, folders, and programs
- Run computer programs

Basics of Directories

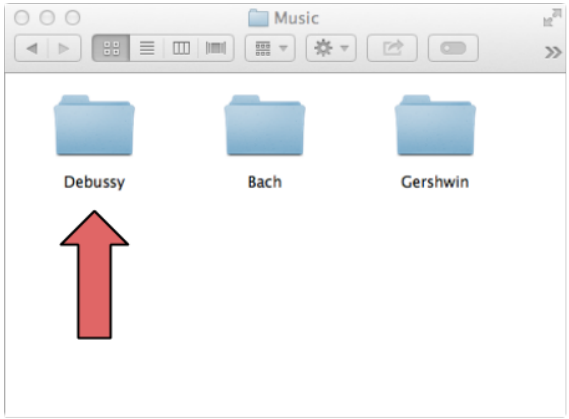
- "Directory" is just another name for folder
- Directories on your computer are organized like a tree
- Directories can be inside other directories
- We can navigate directories using the CLI

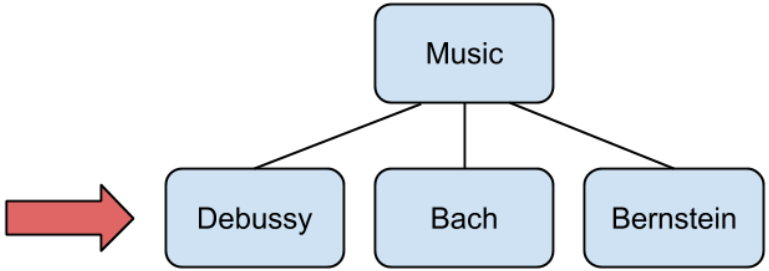




Basics of Directories

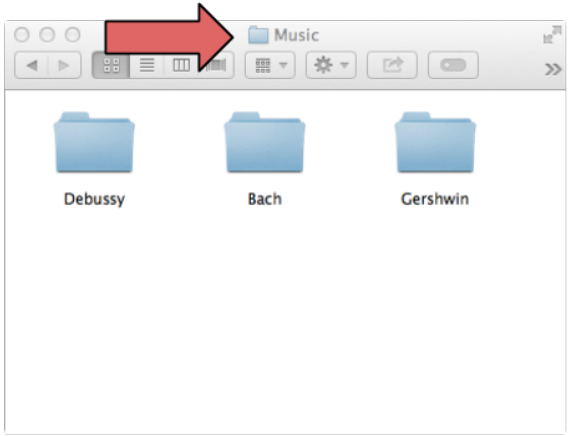
- My "Debussy" directory is contained inside of my "Music" directory

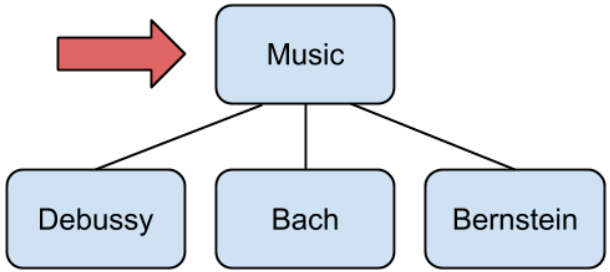




Basics of Directories

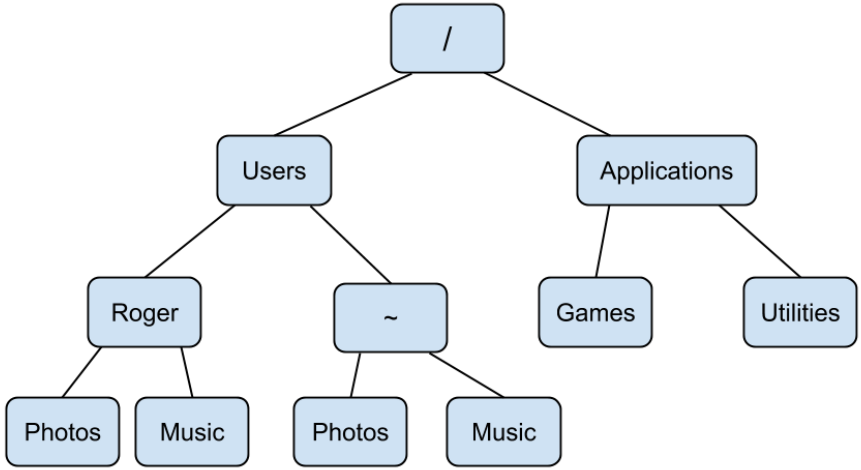
- One directory "up" from my Debussy directory is my Music directory





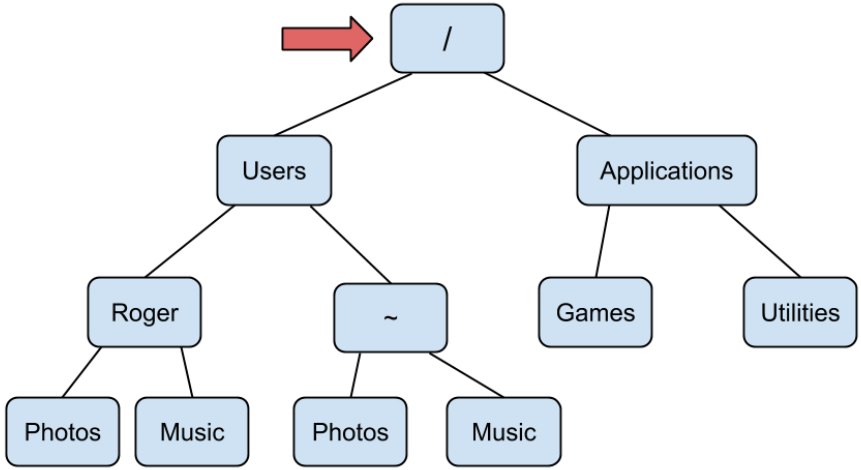
Your computer's directory structure

- The directory structure on your computer looks something like this



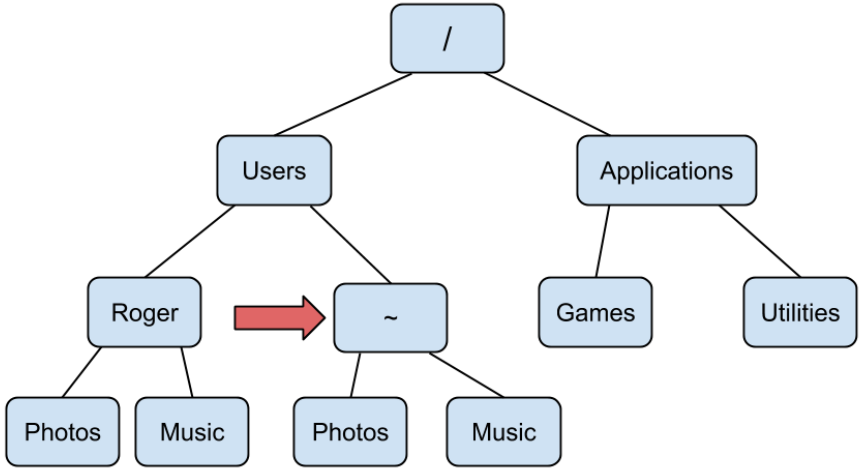
Special directories: root

- The directory at the top of the tree is called the root directory
- The root directory contains all other directories
- The name of this directory is represented by a slash: `/`



Special directories: home

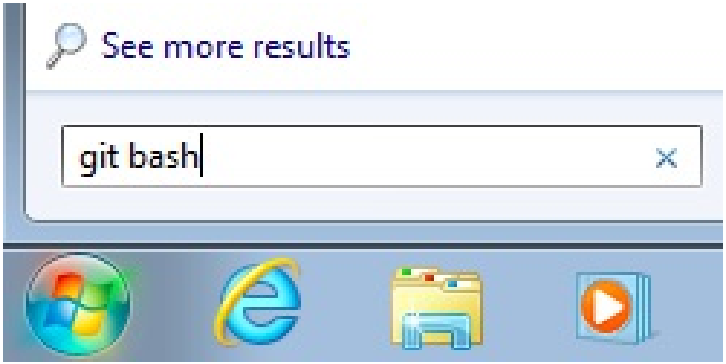
- Your home directory is represented by a tilde: ~
- Your home directory usually contains most of your personal files, pictures, music, etc.
- The name of your home directory is usually the name you use to log into your computer



Navigating directories with the CLI

Windows users:

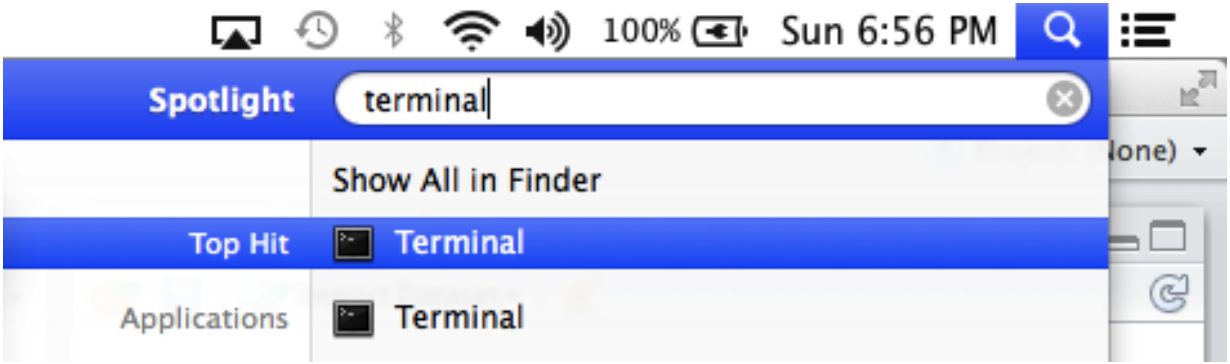
- Open the start menu
- Search for Git Bash
- Open Git Bash



Navigating directories with the CLI

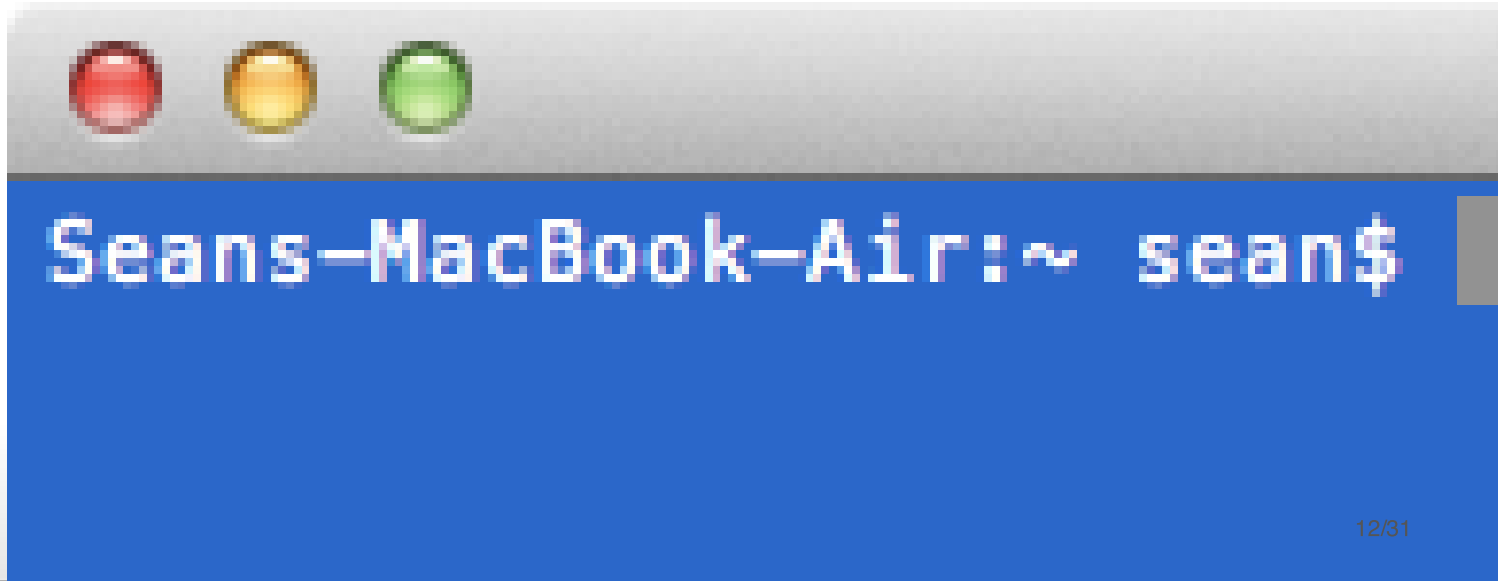
Mac users:

- Open Spotlight
- Search Terminal
- Open Terminal



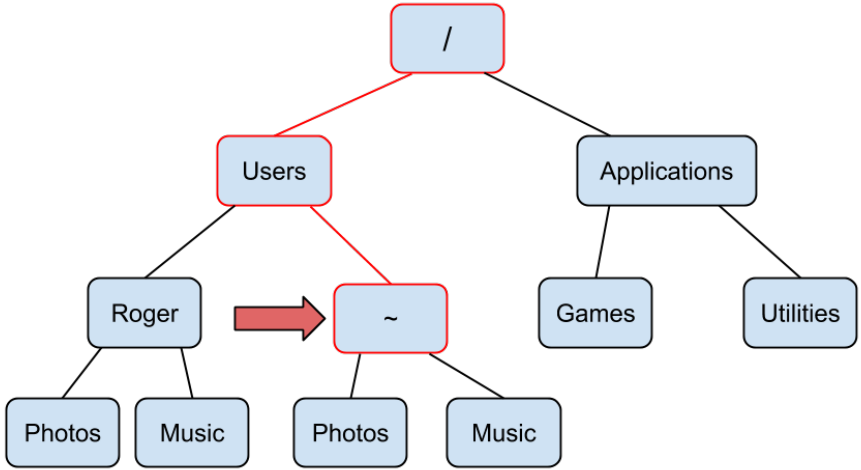
CLI Basics

- When you open your CLI you will see your prompt, which will looks something like the name of your computer, followed by your username, followed by a `$`
- When you open your CLI you start in your home directory.
- Whatever directory directory you're currently working with in your CLI is called the "working directory"



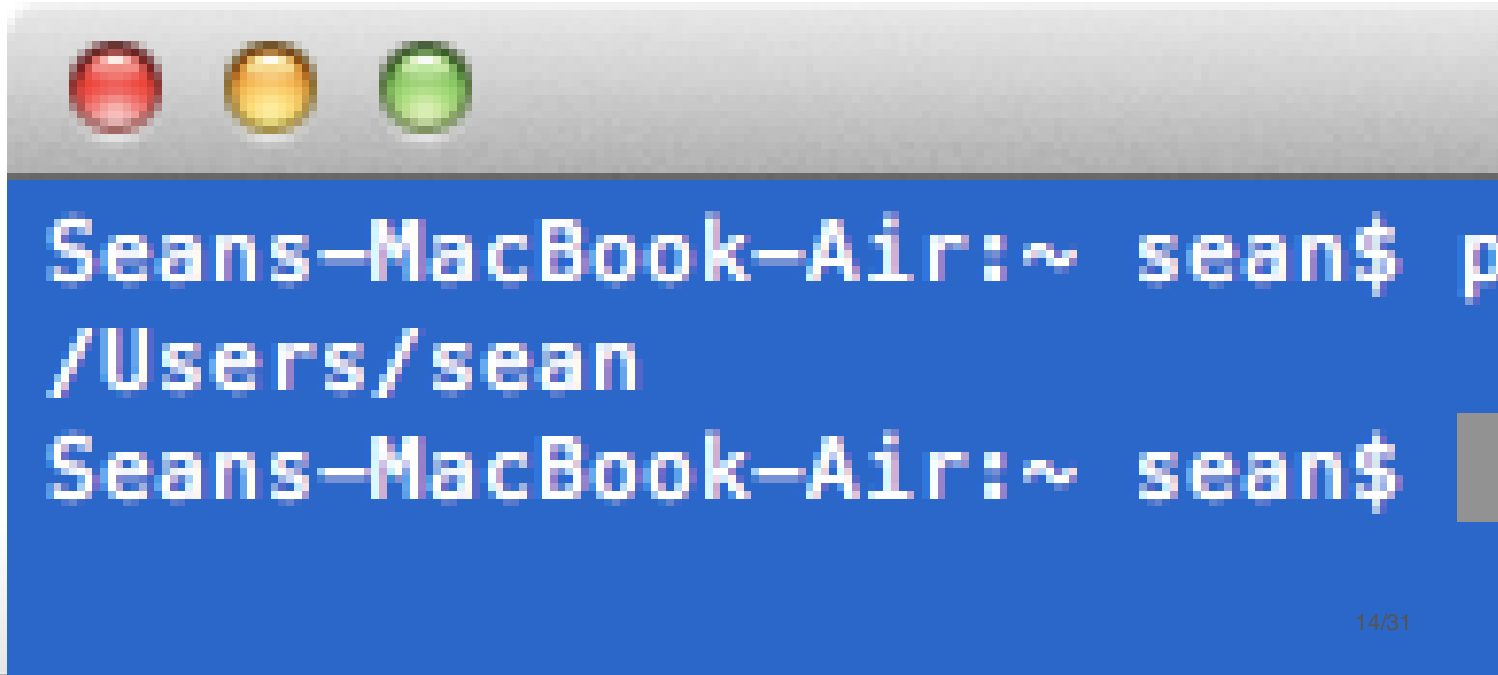
CLI Basics

- You can imagine tracing all of the directories from your root directory to the directory you're currently in.
- This is called the "path" to your working directory.



CLI Basics

- In your CLI prompt, type `pwd` and press enter.
- This will display the path to you're working directory.
- As you can see we get the prompt back after entering a command.



CLI Commands

- You use the CLI prompt by typing in a command and pressing enter.
- `pwd` can be used at any time to display the path to your working directory (`pwd` is an abbreviation for "print working directory")

CLI Commands

- CLI commands follow this recipe: ***command flags arguments***
- ***command*** is the CLI command which does a specific task
- ***flags*** are options we give to the ***command*** to trigger certain behaviors, preceded by a **-**
- ***arguments*** can be what the ***command*** is going to modify, or other options for the ***command***
- Depending on the ***command***, there can be zero or more ***flags*** and ***arguments***
- For example `pwd` is a ***command*** that requires no ***flags*** or ***arguments***

CLI Commands

- `pwd` displays the path to the current working directory

```
jeff$ pwd
/Users/jeff
jeff$
```

CLI Commands

- `clear` will clear out the commands in your current CLI window

```
jeff$ pwd
/Users/jeff
jeff$ clear
```

```
jeff$
```

CLI Commands

- `ls` lists files and folders in the current directory
- `ls -a` lists hidden and unhidden files and folders
- `ls -al` lists details for hidden and unhidden files and folders
- Notice that `-a` and `-l` are flags (they're preceded by a `-`)
- They can be combined into the flag: `-al`

```
jeff$ ls
Desktop   Photos   Music
jeff$ ls -a
Desktop   Photos   Music   .Trash   .DS_Store
jeff$
```

CLI Commands

- `cd` stands for "change directory"
- `cd` takes as an argument the directory you want to visit
- `cd` with no argument takes you to your home directory
- `cd ..` allows you to chnage directory to one level above your current directory

```
jeff$ cd Music/Debussy
jeff$ pwd
/Users/jeff/Music/Debussy
jeff$ cd ..
jeff$ pwd
/Users/jeff/Music
jeff$ cd
jeff$ pwd
/Users/jeff
jeff$
```

CLI Commands

- `mkdir` stands for "make directory"
- Just like: right click -> create new folder
- `mkdir` takes as an argument the name of the directory you're creating

```
jeff$ mkdir Documents
jeff$ ls
Desktop   Photos  Music   Documents
jeff$ cd Documents
jeff$ pwd
/Users/jeff/Documents
jeff$ cd
jeff$
```

CLI Commands

- `touch` creates an empty file

```
jeff$ touch test_file
jeff$ ls
Desktop   Photos  Music   Documents   test_file
jeff$
```

CLI Commands

- `cp` stands for "copy"
- `cp` takes as its first argument a file, and as its second argument the path to where you want the file to be copied

```
jeff$ cp test_file Documents
jeff$ cd Documents
jeff$ ls
test_file
jeff$ cd ..
jeff$
```

CLI Commands

- `cp` can also be used for copying the contents of directories, but you must use the `-r` flag
- The line: `cp -r Documents More_docs` copies the contents of `Documents` into `More_docs`

```
jeff$ mkdir More_docs
jeff$ cp -r Documents More_docs
jeff$ cd More_docs
jeff$ ls
test_file
jeff$ cd ..
jeff$
```

CLI Commands

- `rm` stands for "remove"
- `rm` takes the name of a file you wish to remove as its argument

```
jeff$ ls
Desktop   Photos  Music   Documents   More_docs   test_file
jeff$ rm test_file
jeff$ ls
Desktop   Photos  Music   Documents   More_docs
jeff$
```

CLI Commands

- You can also use `rm` to delete entire directories and their contents by using the `-r` flag
- **Be very careful when you do this, there is no was to undo an `rm`**

```
jeff$ ls
Desktop   Photos  Music   Documents   More_docs
jeff$ rm -r More_docs
jeff$ ls
Desktop   Photos  Music   Documents
jeff$
```

CLI Commands

- `mv` stands for "move"
- With `mv` you can move files between directories

```
jeff$ touch new_file
jeff$ mv new_file Documents
jeff$ ls
Desktop   Photos  Music   Documents
jeff$ cd Documents
jeff$ ls
test_file   new_file
jeff$
```

CLI Commands

- You can also use `mv` to rename files

```
jeff$ ls
test_file   new_file
jeff$ mv new_file renamed_file
jeff$ ls
test_file renamed_file
jeff$
```

CLI Commands

- `echo` will print whatever arguments you provide

```
jeff$ echo Hello World!
Hello World!
jeff$
```

CLI Commands

- `date` will print today's date

```
jeff$ date
Mon Nov  4 20:48:03 EST 2013
jeff$
```

Summary of Commands

- `pwd`
- `clear`
- `ls`
- `cd`
- `mkdir`
- `touch`
- `cp`
- `rm`
- `mv`
- `date`
- `echo`

Introduction to Git

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

What is Version Control?

> Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

http://git-scm.com/book/en/Getting-Started-About-Version-Control

What is Version Control?

- Many of us constantly create something, save it, change it, then save it again
- Version (or revision) control is a means of managing this process in a reliable and efficient way
- Especially important when collaborating with others

http://en.wikipedia.org/wiki/Revision_control

What is Git?

> “Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.”

http://git-scm.com/

What is Git?

- Created by the same people who developed Linux
- The most popular implementation of version control today
- Everything is stored in local repositories on your computer
- Operated from the command line

http://git-scm.com/book/en/Getting-Started-A-Short-History-of-Git

Download Git

- Go to the following website and click on the download link for your operating system (Mac, Windows, Linux, etc):

http://git-scm.com/downloads

Install Git

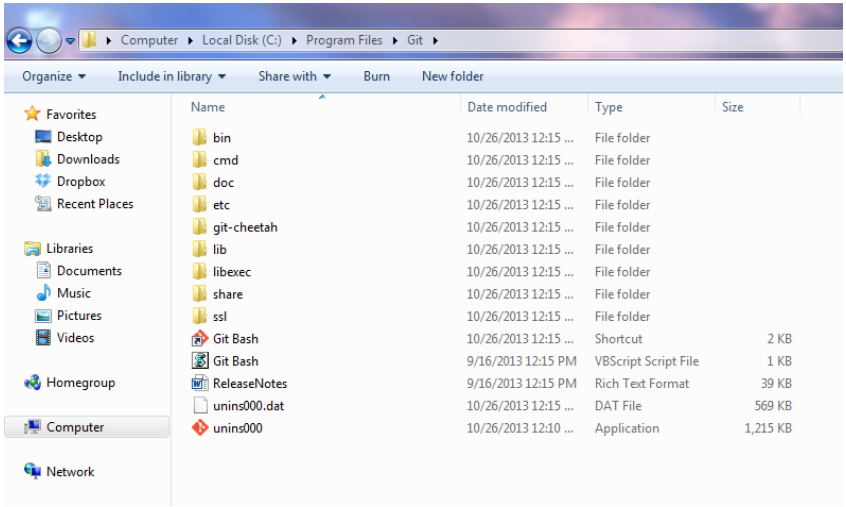
- Once the file is done downloading, open it up to begin the Git installation

Install Git

- Unless you really know what you are doing, just go with the default options at each step of the installation
- Once the install is complete, hit the "Finish" button (you may want to uncheck the box next to "Review ReleaseNotes.rtf")

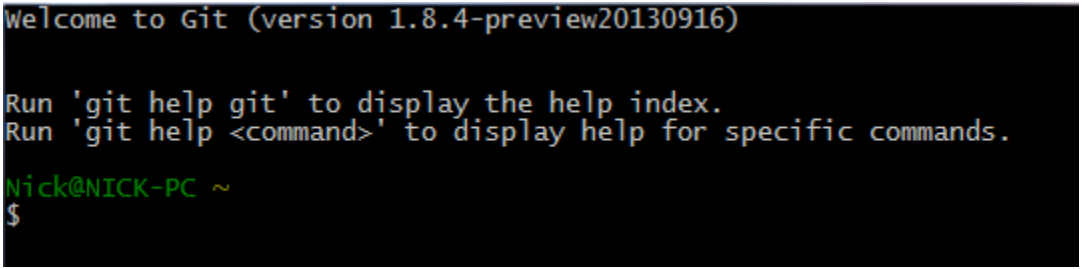
Open Git Bash

- Find a program called Git Bash, which is the command line environment for interacting with Git
- It should be located in the directory into which Git was installed (or, for Windows users, in the Start Menu)



Open Git Bash

- Once Git Bash opens, you'll see a short welcome message followed by the name of your computer and a dollar sign on the next line
- The dollar sign means that it's your turn to type a command



Configure Username and Email

- Each commit to a Git repository will be "tagged" with the username of the person who made the commit
- Enter the following commands in Git Bash, one at a time, to set your username and email:

```
$ git config --global user.name "Your Name Here"
$ git config --global user.email "your_email@example.com"
```

- You'll only have to do this once, but you can always change these down the road using the same commands

Configure Username and Email

- Now type the following to confirm your changes (they may be listed toward the bottom):

```
$ git config --list
```

What's Next?

- Go ahead and close Git Bash with following command:

```
$ exit
```

- Now that Git is set up on your computer, we're ready to move on to GitHub, which is a web-based platform that lets you do some pretty cool stuff
- Once GitHub is up and running, we'll show you how to start using these tools to your benefit

Introduction to GitHub

Jeffrey Leek, Assistant Professor of Biostatistics
Johns Hopkins Bloomberg School of Public Health

What is GitHub?

> GitHub is a web-based hosting service for software development projects that use the Git revision control system.

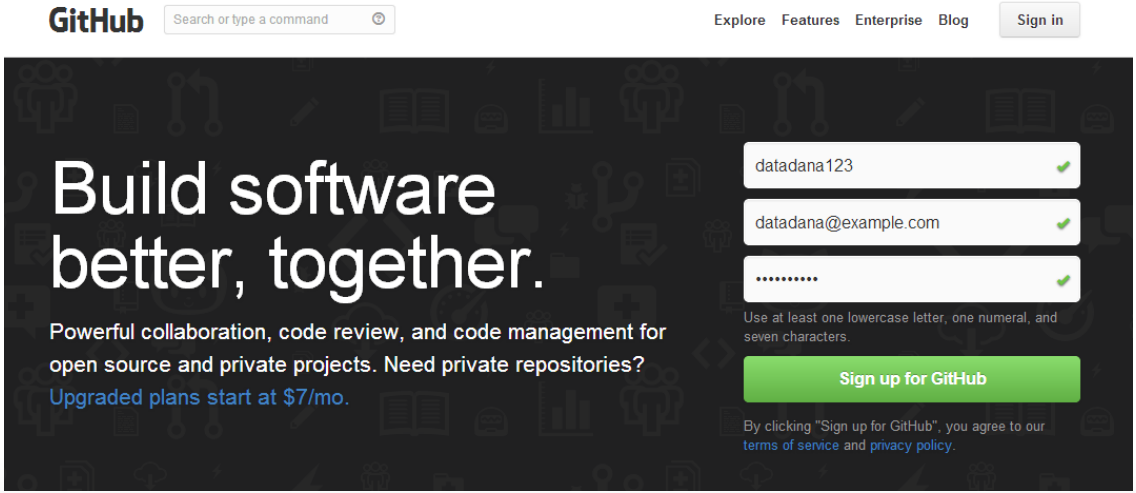
http://en.wikipedia.org/wiki/GitHub

What is GitHub?

- Allows users to "push" and "pull" their local repositories to and from remote repositories on the web
- Provides users with a homepage that displays their public repositories
- Users' repositories are backed up on the GitHub server in case something happens to the local copies
- Social aspect allows users to follow one another and share projects

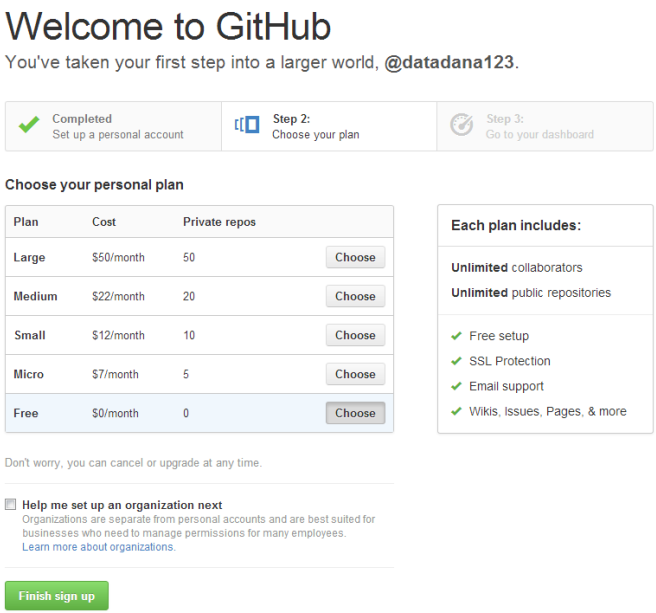
Set Up a GitHub Account

- Go to the GitHub homepage at https://github.com/
- Enter a username, email, and password and click "Sign up for GitHub"
- **NOTE: You should use the same email address that you used when setting up Git in the previous lecture**



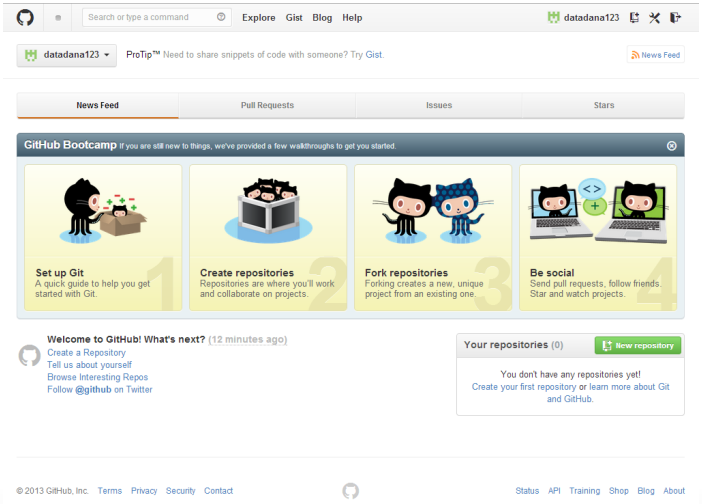
Set Up a GitHub Account

- On the next screen, select the free plan and click "Finish sign up"



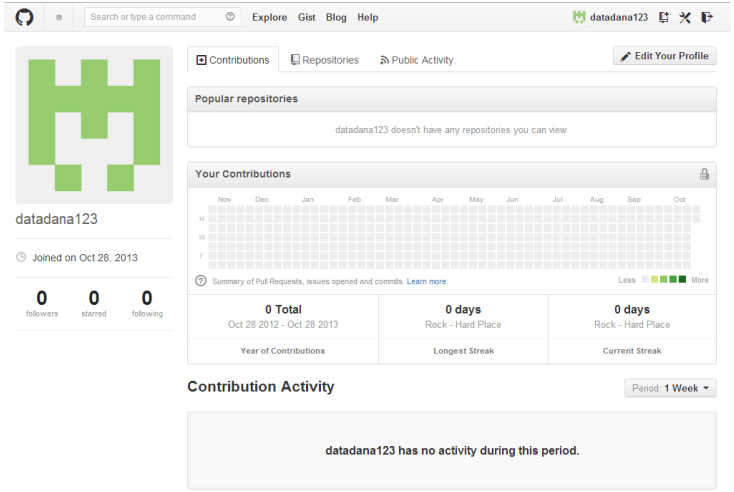
Navigating GitHub

- After signing up, you will find yourself on this page, which has several helpful resources for learning more about Git and GitHub
- Try clicking on your username in the upper righthand corner of the screen to view your GitHub profile



Your GitHub Profile

- Your profile is where all of your activity on GitHub is displayed
- Allows you to show other people who you are and what you are working on
- As you work on more and more projects, your profile becomes a portfolio of your work



Your GitHub Profile

- Finally, if you click on "Edit Your Profile" in the top righthand portion of the screen you can add some basic information about yourself to your profile
- This is totally optional, but if you do good work, you ought to take some credit for it!
- In the next lecture, we'll get you started by walking you through two ways of creating a repository
- In the meantime, feel free to explore the GitHub site for interesting projects that others are working on

Creating a GitHub Repository

Jeffrey Leek, Assistant Professor of Biostatistics
Johns Hopkins Bloomberg School of Public Health

Recap: Git vs. GitHub

- You don't need GitHub to use Git
- Git = Local (on your computer); GitHub = Remote (on the web)
- GitHub allows you to:
  1. Share your repositories with others
  2. Access other users' repositories
  3. Store remote copies of your repositories (on GitHub's server) in case something happens to your local copies (on your computer)

Creating a GitHub Repository

- Two methods of creating a GitHub repository:
  1. Start a repository from scratch
  2. "Fork" another user's repository
- We'll start with the first method
- *NOTE: A repository is often referred to as a "repo"*

Start a Repository From Scratch

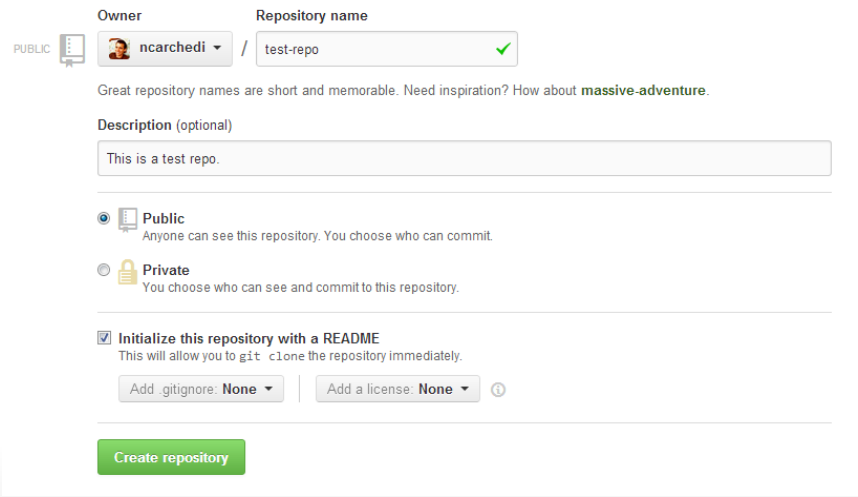
- Either go to your profile page (https://github.com/yourUserNameHere/) and click on "Create a new repo" in the upper righthand corner of the page

...OR...

- Go directly to https://github.com/new (you'll need to log into your GitHub account if you haven't already done so)

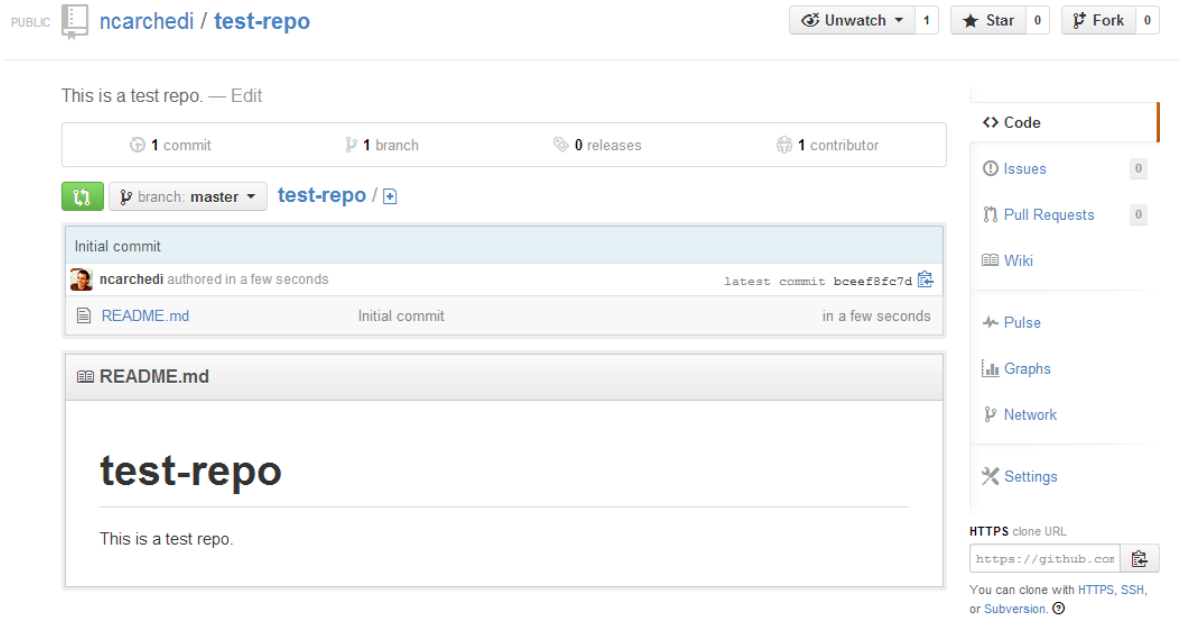
Start a Repository From Scratch

- Create a name for your repo and type a brief description of it
- Select "Public" (Private repos require a paid [or education] account)
- Check the box next to "Initialize this repository with a README"
- Click the "Create repository" button



Start a Repository From Scratch

- Congratulations! You've created a GitHub repository.



Creating a Local Copy

- Now you need to create a copy of this repo on your computer so that you can make changes to it
- Open Git Bash
- Create a directory on your computer where you will store your copy of the repo:

```
$ mkdir ~/test-repo
```

- Navigate to this new directory using the following command:

```
$ cd ~/test-repo
```

Creating a Local Copy

- Initialize a local Git repository in this directory

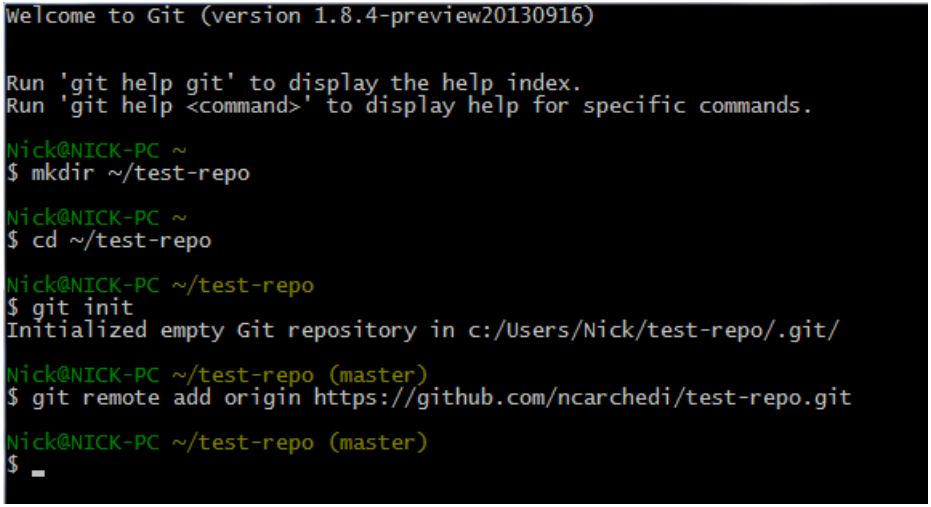
```
$ git init
```

- Point your local repository at the remote repository you just created on the GitHub server

```
$ git remote add origin https://github.com/yourUserNameHere/test-repo.git
```

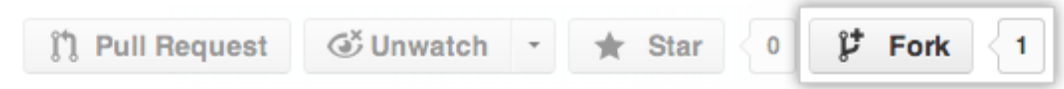
Creating a Local Copy

- Here's what this process looks like in action:



Fork a Another User's Repository

- The second method of creating a respository is to make a copy of someone else's
- This process is called "forking" and is an important aspect of open-source software development
- Begin by navigating to the desired repository on the GitHub website and click the "Fork" button shown below



https://help.github.com/articles/fork-a-repo

Clone the Repo

- You now have a copy of the desired respository on your GitHub account
- Need to make a local copy of the repo on your computer
- This process is called "cloning" and can be done using the following command:

```
$ git clone https://github.com/yourUserNameHere/repoNameHere.git
```

- *NOTE: This will clone the repository into your current directory.*

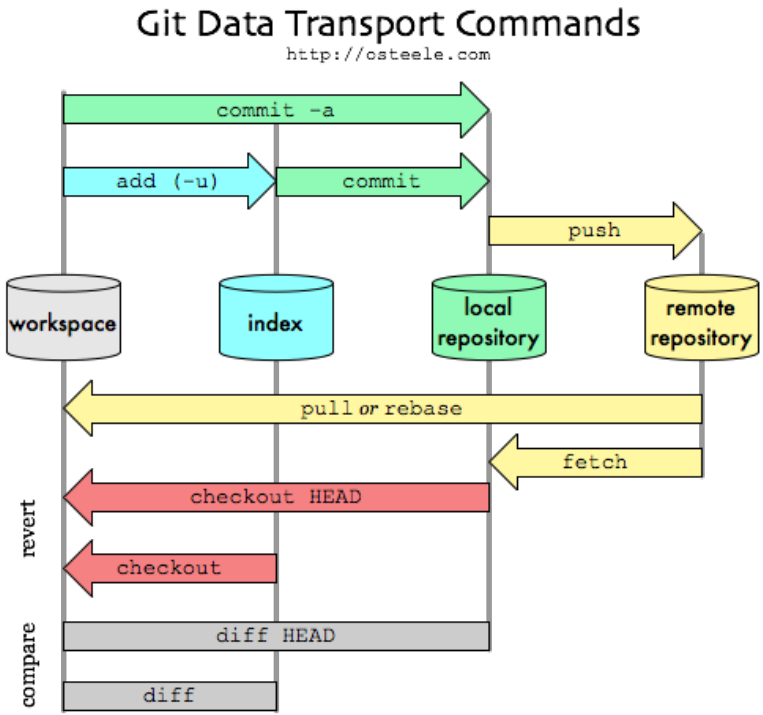
What Else?

- If you make changes to your local copy of the repo, you'll probably want to push your changes to GitHub at some point
- You also may be interested in staying current with any changes made to the original repository from which you forked your copy
- We will cover some more Git/GitHub basics in coming lectures, but in the meantime, here are some great resources:
  - https://help.github.com/articles/fork-a-repo
  - http://git-scm.com/book/en/Git-Basics-Getting-a-Git-Repository

Basic Git Commands

Jeffrey Leek, Assistant Professor of Biostatistics
Johns Hopkins Bloomberg School of Public Health

Pushing and pulling



http://gitready.com/beginner/2009/01/21/pushing-and-pulling.html

Adding

- Suppose you add new files to a local repository under version control
- You need to let Git know that they need to be tracked
  - `git add .` adds all new files
  - `git add -u` updates tracking for files that changed names or were deleted
  - `git add -A` does both of the previous
- You should do this before committing

Committing

- You have changes you want to commit to be saved as an intermediate version
- You type the command
  - `git commit -m "message"` where message is a useful description of what you did
- This only updates your local repo, not the remote repo on Github

Pushing

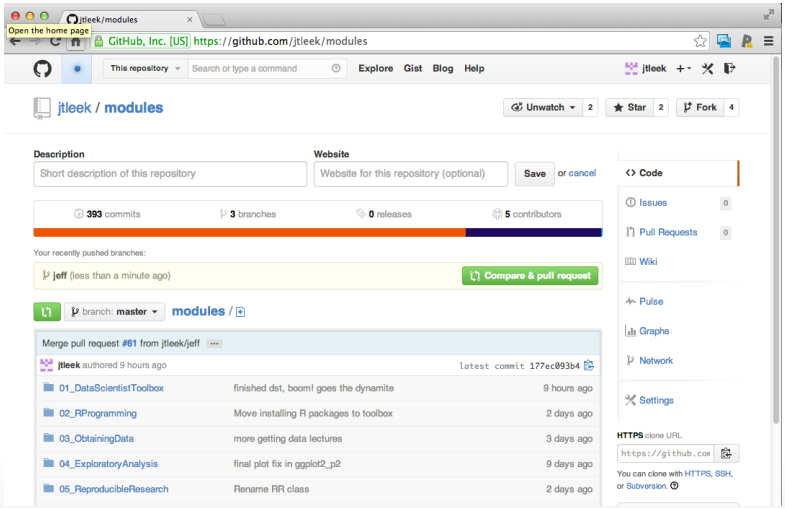
- You have saved local commits you would like to update on the remote (Github)
- You type the command
  - `git push`

Branches

- Sometimes you are working on a project with a version being used by many people
- You may not want to edit that version
- So you can create a branch with the command
  - `git checkout -b branchname`
- To see what branch you are on type:
  - `git branch`
- To switch back to the master branch type
  - `git checkout master`

Pull requests

- If you fork someone's repo or have multiple branches you will both be working seperately
- Sometimes you want to merge in your changes into the other branch/repo
- To do so you need to send a pull request.
- This is a feature of Github.



Time to be a hacker!

- Git documentation http://git-scm.com/doc
- Github help https://help.github.com/
- Google/Stack Overflow are great for Github

Basic markdown

Jeffrey Leek, Assistant Professor of Biostatistics
Johns Hopkins Bloomberg School of Public Health

Markdown Syntax

Headings

```
## This is a secondary heading
### This is a tertiary heading
```

This is a secondary heading

This is a tertiary heading

Markdown Syntax

Unordered Lists

```
* first item in list
* second item in list
* third item in list
```

- first item in list
- second item in list
- third item in list

Getting markdown help

- An introduction to markdown http://daringfireball.net/projects/markdown/
- Click the MD button in Rstudio for a quick guide
- R markdown http://www.rstudio.com/ide/docs/authoring/using_markdown (you don't need this until Reproducible Research)

Installing R Packages

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

R Packages

- When you download R from the Comprehensive R Archive Network (CRAN), you get that ``base'' R system
- The base R system comes with basic functionality; implements the R language
- One reason R is so useful is the large collection of packages that extend the basic functionality of R
- R packages are developed and published by the larger R community

Obtaining R Packages

- The primary location for obtaining R packages is CRAN
- For biological applications, many packages are available from the Bioconductor Project
- You can obtain information about the available packages on CRAN with the `available.packages()` function

```
a <- available.packages()
head(rownames(a), 3)  ## Show the names of the first few packages
```

```
## [1] "A3"       "abc"      "abcdeFBA"
```

- There are approximately 5200 packages on CRAN covering a wide range of topics
- A list of some topics is available through the Task Views link, which groups together many R packages related to a given topic

Installing an R Package

- Packages can be installed with the `install.packages()` function in R
- To install a single package, pass the name of the lecture to the `install.packages()` function as the first argument
- The following the code installs the **slidify** package from CRAN

```
install.packages("slidify")
```

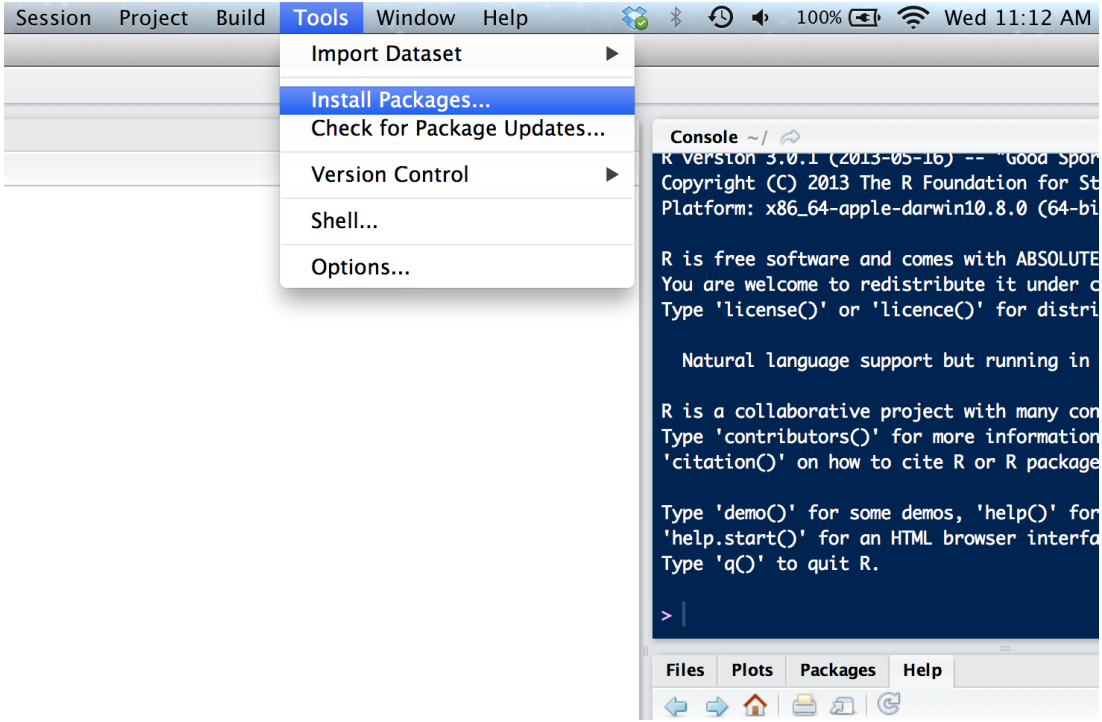
- This command downloads the **slidify** package from CRAN and installs it on your computer
- Any packages on which this package depends will also be downloaded and installed

Installing an R Package

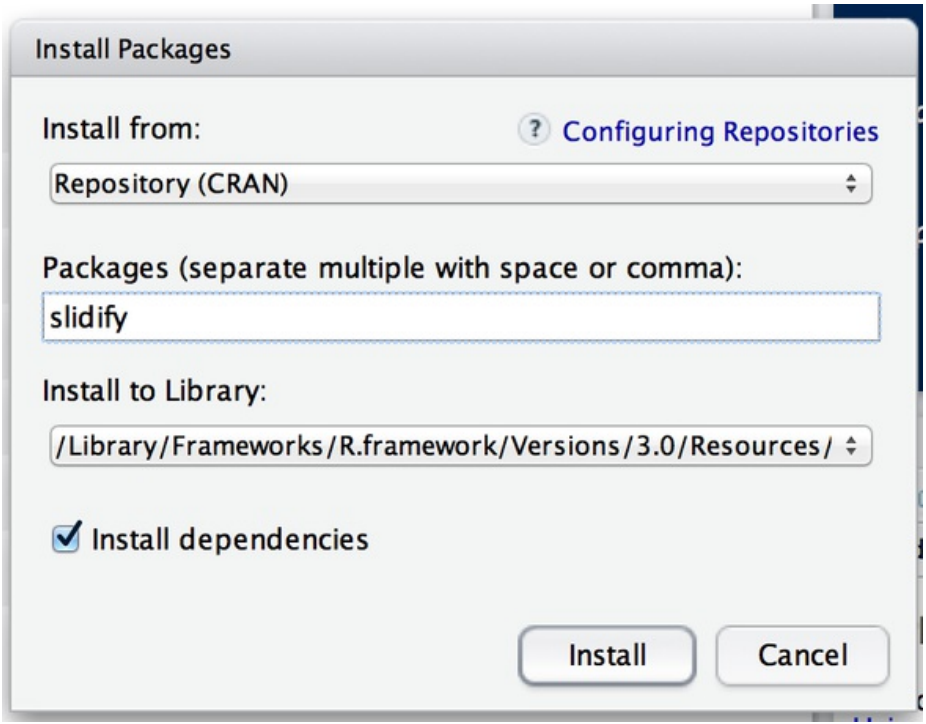
- You can install multiple R packages at once with a single call to `install.packages()`
- Place the names of the R packages in a character vector

```
install.packages(c("slidify", "ggplot2", "devtools"))
```

Installing an R Package in RStudio



Installing an R Package in RStudio



Installing an R Package from Bioconductor

- To get the basic installer and basic set of R packages (warning, will install multiple packages)

```
source("http://bioconductor.org/biocLite.R")
biocLite()
```

- Place the names of the R packages in a character vector

```
biocLite(c("GenomicFeatures", "AnnotationDbi"))
```

http://www.bioconductor.org/install/

Loading R Packages

- Installing a package does not make it immediately available to you in R; you must load the package
- The `library()` function is used to **load** packages into R
- The following code is used to load the **ggplot2** package into R

```
library(ggplot2)
```

- Any packages that need to be loaded as dependencies will be loaded first, before the named package is loaded
- NOTE: Do not put the package name in quotes!
- Some packages produce messages when they are loaded (but some don't)

Loading R Packages

After loading a package, the functions exported by that package will be attached to the top of the `search()` list (after the workspace)

```
library(ggplot2)
search()
```

```
##  [1] ".GlobalEnv"        "package:kernlab"   "package:caret"
##  [4] "package:lattice"   "package:ggplot2"   "package:makeslides"
##  [7] "package:knitr"     "package:slidify"   "tools:rstudio"
## [10] "package:stats"     "package:graphics"  "package:grDevices"
## [13] "package:utils"     "package:datasets"  "package:methods"
## [16] "Autoloads"         "package:base"
```

Summary

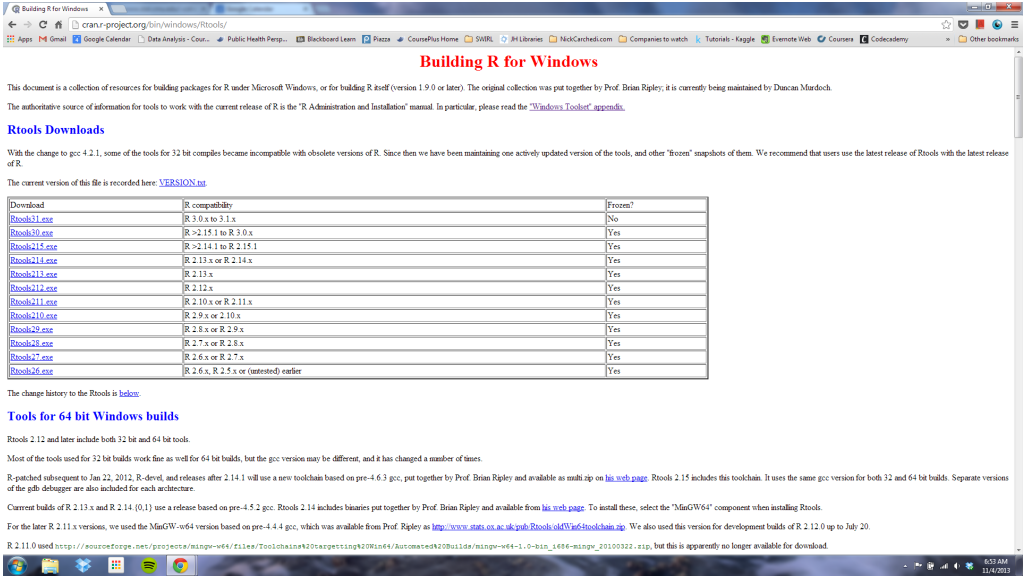
- R packages provide a powerful mechanism for extending the functionality of R
- R packages can be obtained from CRAN or other repositories
- The `install.packages()` can be used to install packages at the R console
- The `library()` function loads packages that have been installed so that you may access the functionality in the package

Installing Rtools

Jeffrey Leek, Assistant Professor of Biostatistics
Johns Hopkins Bloomberg School of Public Health

What is Rtools?

- A collection of tools necessary for building R packages in Windows
- Available for download at http://cran.r-project.org/bin/windows/Rtools/



Building R for Windows

cran.r-project.org/bin/windows/Rtools/

Building R for Windows

This document is a collection of resources for building packages for R under Microsoft Windows, or for building R itself (version 1.9.0 or later). The original collection was put together by Prof. Brian Ripley; it is currently being maintained by Duncan Murdoch.

The authoritative source of information for tools to work with the current release of R is the "R Administration and Installation" manual. In particular, please read the "Windows Toolset" appendix.

Rtools Downloads

With the change to gcc 4.2.1, some of the tools for 32 bit compiles became incompatible with obsolete versions of R. Since then we have been maintaining one actively updated version of the tools, and other "frozen" snapshots of them. We recommend that users use the latest release of Rtools with the latest release of R.

The current version of this file is recorded here: VERSION.txt.

Download	R compatibility	Frozen?
Rtools21.exe	R 3.0.x to 3.1.x	No
Rtools20.exe	R >2.15.1 to R 3.0.x	Yes
Rtools215.exe	R >2.14.1 to R 2.15.1	Yes
Rtools214.exe	R 2.13.x or R 2.14.x	Yes
Rtools213.exe	R 2.13.x	Yes
Rtools212.exe	R 2.12.x	Yes
Rtools211.exe	R 2.10.x or R 2.11.x	Yes
Rtools210.exe	R 2.9.x or 2.10.x	Yes
Rtools29.exe	R 2.8.x or R 2.9.x	Yes
Rtools28.exe	R 2.7.x or R 2.8.x	Yes
Rtools27.exe	R 2.6.x or R 2.7.x	Yes
Rtools26.exe	R 2.6.x, R 2.5.x or (untested) earlier	Yes

The change history to the Rtools is below.

Tools for 64 bit Windows builds

Rtools 2.12 and later include both 32 bit and 64 bit tools.

Most of the tools used for 32 bit builds work fine as well for 64 bit builds, but the gcc version may be different, and it has changed a number of times.

R-patched subsequent to Jan 22, 2012, R-devel, and releases after 2.14.1 will use a new toolchain based on pre-4.6.3 gcc, put together by Prof. Brian Ripley and available as multi zip on his web page. Rtools 2.15 includes this toolchain. It uses the same gcc version for both 32 and 64 bit builds. Separate versions of the gdb debugger are also included for each architecture.

Current builds of R 2.13.x and R 2.14.{0,1} use a release based on pre-4.5.2 gcc. Rtools 2.14 includes binaries put together by Prof. Brian Ripley and available from his web page. To install these, select the "MinGW64" component when installing Rtools.

For the later R 2.11.x versions, we used the MinGW-w64 version based on pre-4.4.4 gcc, which was available from Prof. Ripley as http://www.stats.ox.ac.uk/pub/Rtools/oldWin6-ltoolchain.zip. We also used this version for development builds of R 2.12.0 up to July 20.

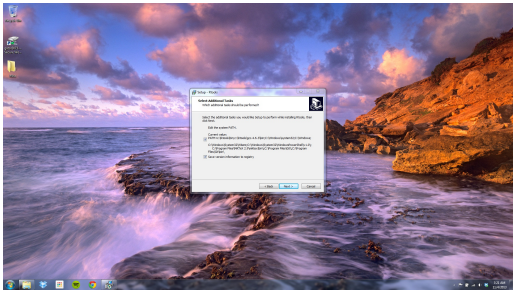
R 2.11.0 used http://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win64/Automated%20Builds/mingw-w64-1.0-bin_i686-mingw_20100322.zip, but this is apparently no longer available for download.

Download Rtools

- Select the .exe download link from the table that corresponds to your version of R
  - Note: If you're not sure what version of R you have, open or restart R and it's the first thing that comes up in the console
- If you have the most recent version of R, you should select the most recent Rtools download (at the top of the chart)
- Once the download completes, open the .exe file to begin the installation

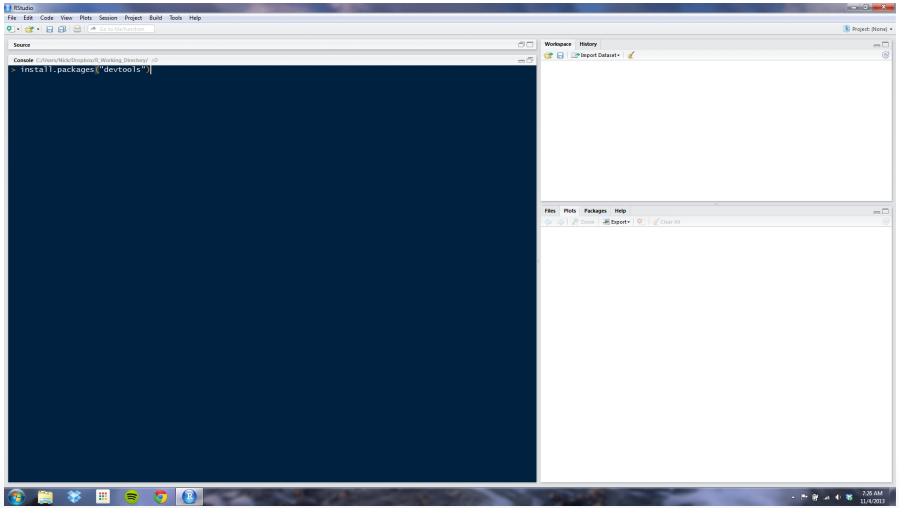
Install Rtools

- Unless you really know what you are doing, you should just go with the default selections at each step of the installation
- There are only two exceptions worth noting:
  - If you already have Cygwin installed on your machine, you should follow the instructions given during installation (and linked to here: http://cran.r-project.org/bin/windows/Rtools/Rtools.txt)
  - *IMPORTANT: You should make sure that the box is checked to have the installer edit your PATH (see below).*



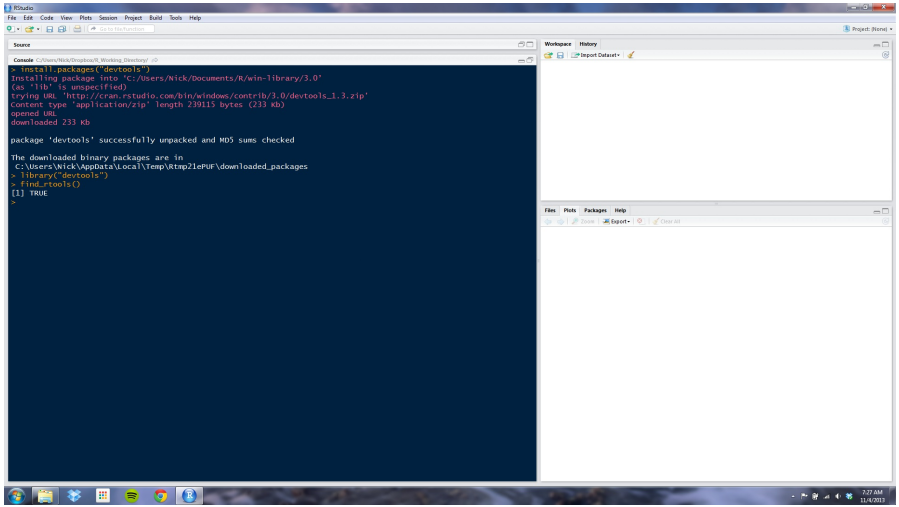
Install devtools

- Once the Rtools installation completes, open RStudio
- Install the devtools R package if you have not previously done so
  - If you aren't sure, enter `find.package("devtools")` in the console
- To install devtools, use `install.packages("devtools")`



Verify Rtools installation

- After devtools is done installing, load it using `library(devtools)`
- Then type `find_rtools()` as shown below
- This should return `TRUE` in the console if your Rtools installation worked properly



Types of Data Science Questions

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Types of Data Science Questions

In approximate order of difficulty

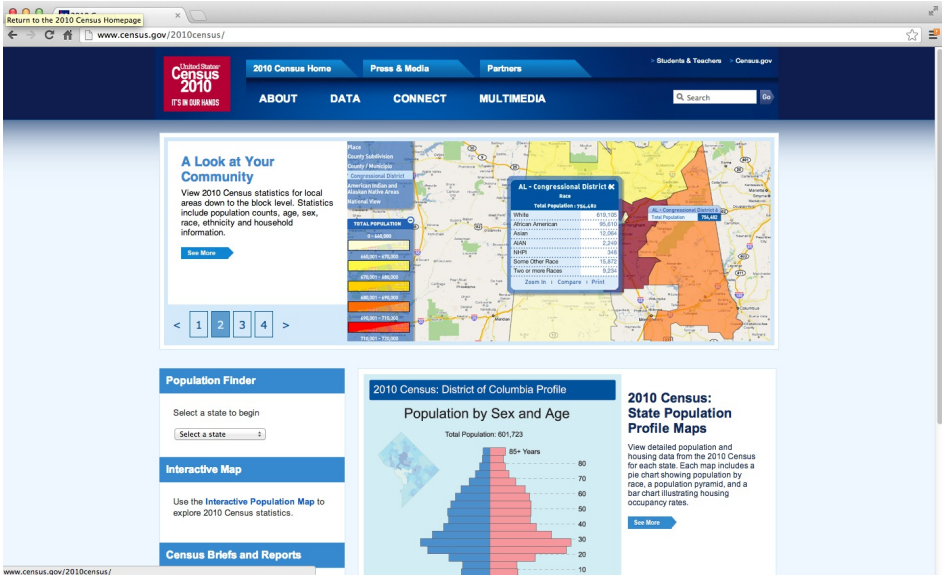
- Descriptive
- Exploratory
- Inferential
- Predictive
- Causal
- Mechanistic

About descriptive analyses

Goal: Describe a set of data

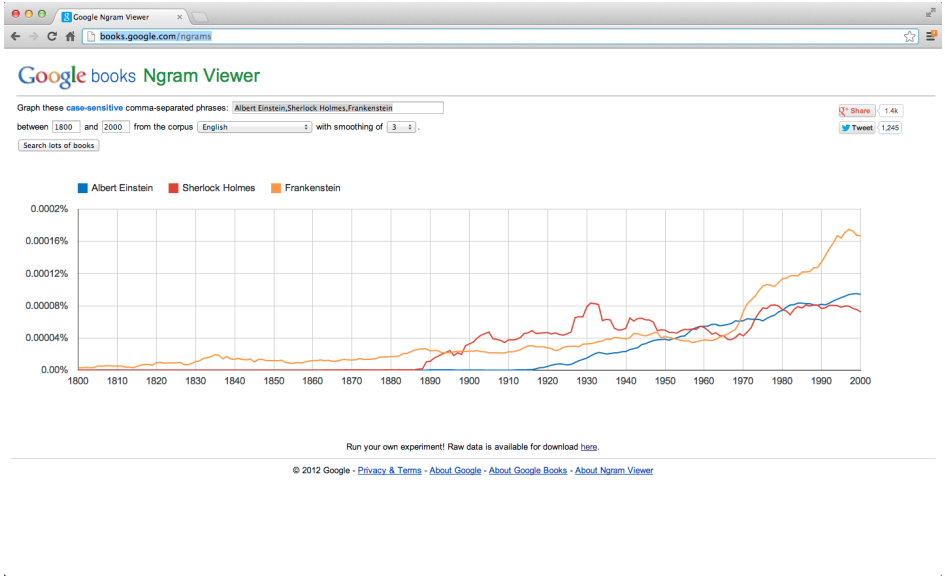
- The first kind of data analysis performed
- Commonly applied to census data
- The description and interpretation are different steps
- Descriptions can usually not be generalized without additional statistical modeling

Descriptive analysis



http://www.census.gov/2010census/

Descriptive analysis



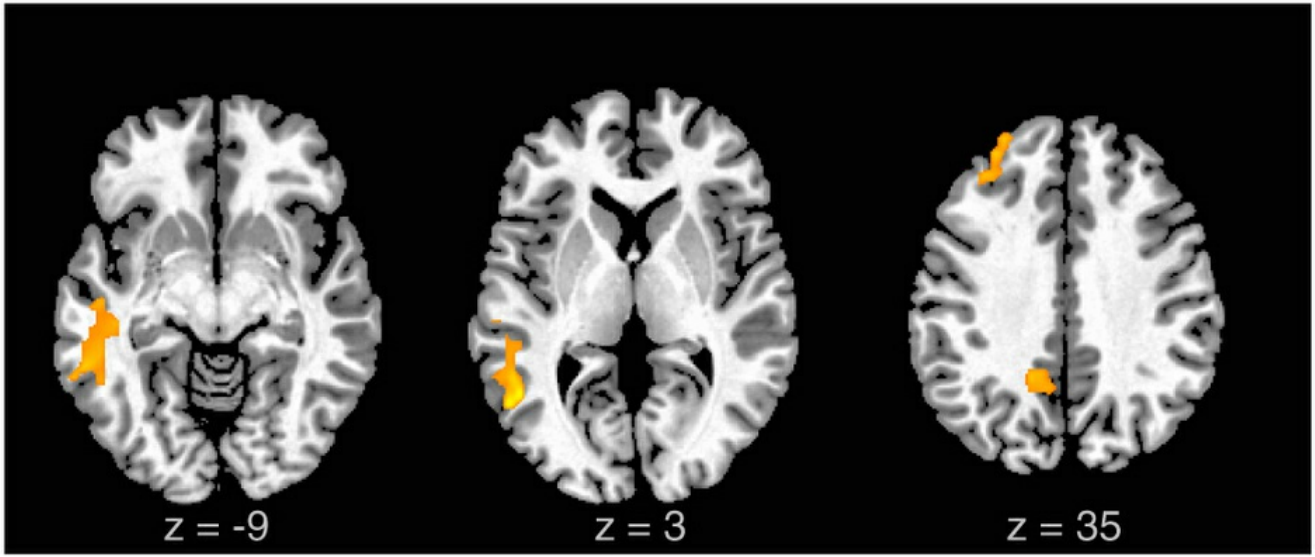
http://books.google.com/ngrams

About exploratory analysis

Goal: Find relationships you didn't know about

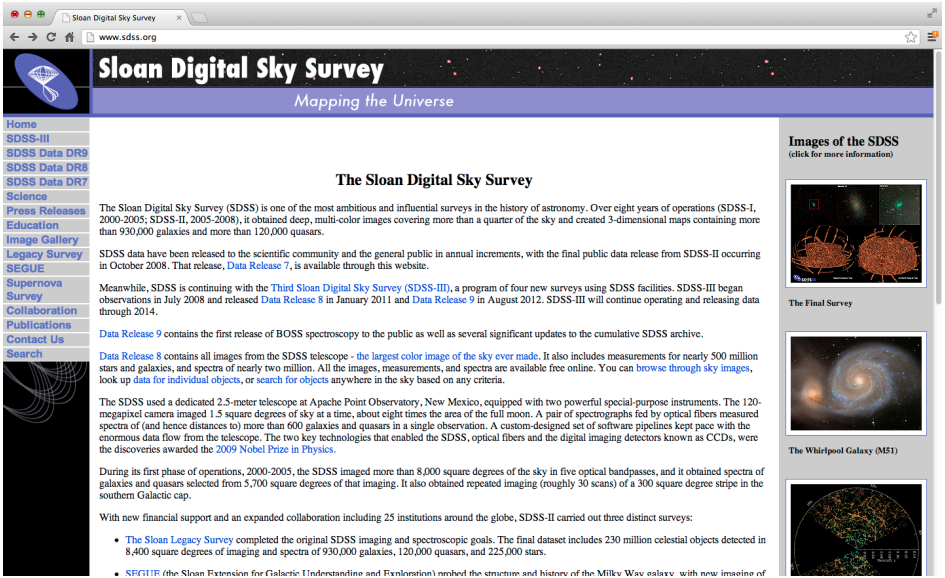
- Exploratory models are good for discovering new connections
- They are also useful for defining future studies
- Exploratory analyses are usually not the final say
- Exploratory analyses alone should not be used for generalizing/predicting
- Correlation does not imply causation

Exploratory analysis



Liu et al. (2012) Scientific Reports

Exploratory analysis



http://www.sdss.org/

About inferential analysis

Goal: Use a relatively small sample of data to say something about a bigger population

- Inference is commonly the goal of statistical models
- Inference involves estimating both the quantity you care about and your uncertainty about your estimate
- Inference depends heavily on both the population and the sampling scheme

Inferential analysis

< Previous Article | Next Article >

Epidemiology:
January 2013 - Volume 24 - Issue 1 - p 23–31
doi: 10.1097/EDE.0b013e3182770237
Air Pollution

Effect of Air Pollution Control on Life Expectancy in the United States: An Analysis of 545 U.S. Counties for the Period from 2000 to 2007

Correia, Andrew W.[a]; Pope, C. Arden III[b]; Dockery, Douglas W.[c]; Wang, Yun[a]; Ezzati, Majid[d]; Dominici, Francesca[a]

FREE SDC

Article Outline

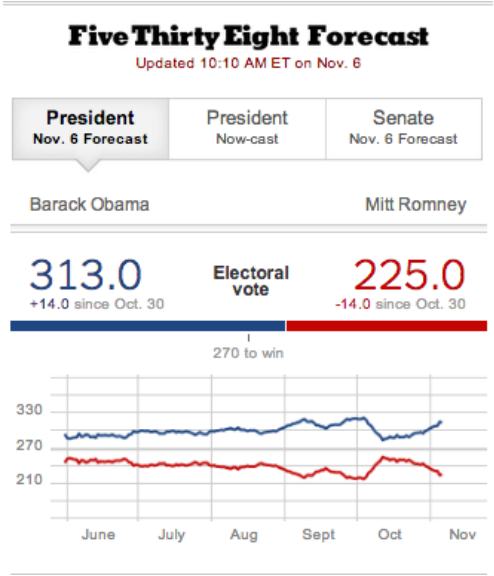
Correia et al. (2013) Epidemiology

About predictive analysis

Goal: To use the data on some objects to predict values for another object

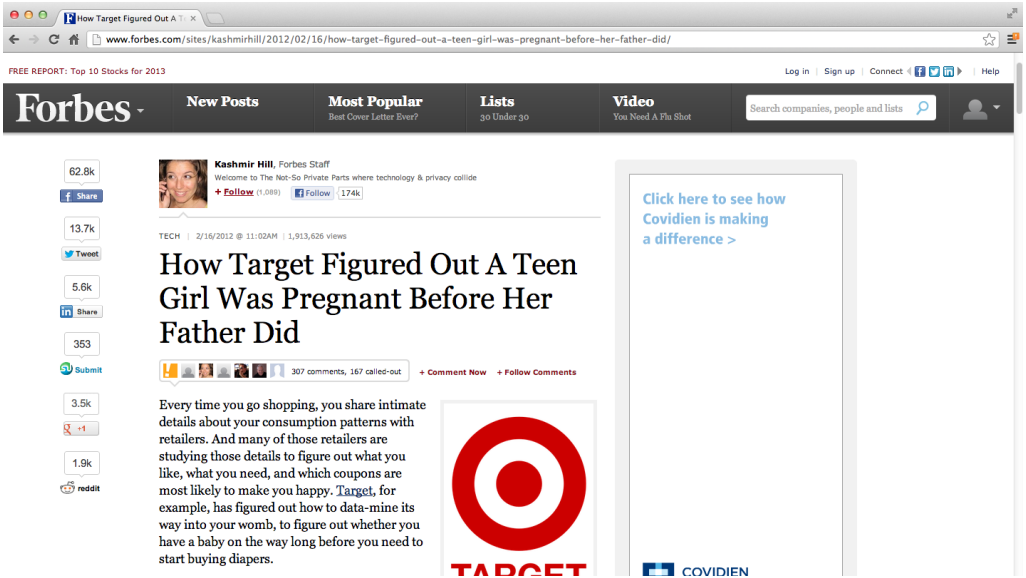
- If X predicts Y it does not mean that X causes Y
- Accurate prediction depends heavily on measuring the right variables
- Although there are better and worse prediction models, more data and a simple model works really well
- Prediction is very hard, especially about the future references

Predictive analysis



http://fivethirtyeight.blogs.nytimes.com/

Predictive analysis



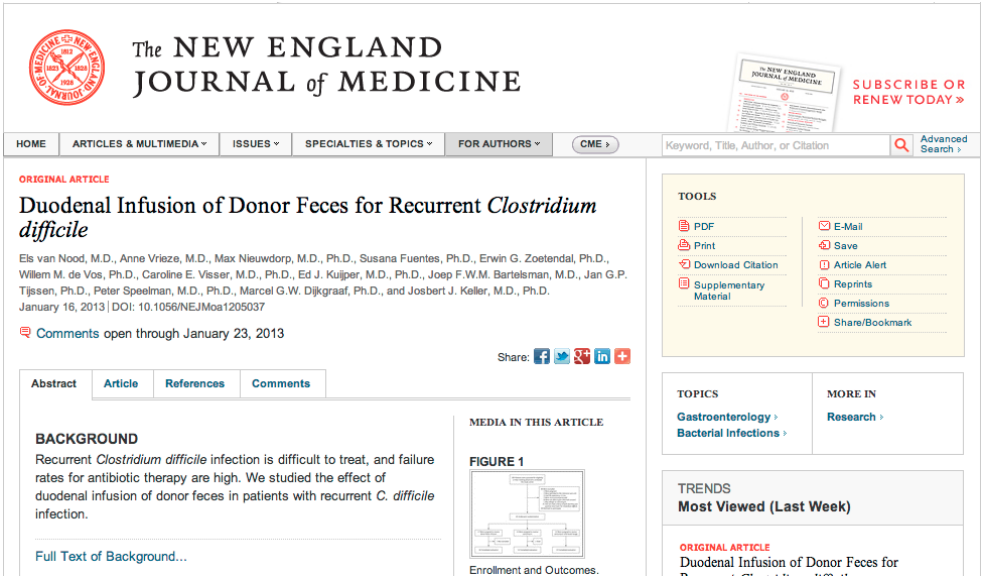
http://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/

About causal analysis

Goal: To find out what happens to one variable when you make another variable change.

- Usually randomized studies are required to identify causation
- There are approaches to inferring causation in non-randomized studies, but they are complicated and sensitive to assumptions
- Causal relationships are usually identified as average effects, but may not apply to every individual
- Causal models are usually the "gold standard" for data analysis

Causal analysis



The NEW ENGLAND JOURNAL of MEDICINE

SUBSCRIBE OR RENEW TODAY »

HOME | ARTICLES & MULTIMEDIA | ISSUES | SPECIALTIES & TOPICS | FOR AUTHORS | CME | Keyword, Title, Author, or Citation | Advanced Search

ORIGINAL ARTICLE

Duodenal Infusion of Donor Feces for Recurrent *Clostridium difficile*

Els van Nood, M.D., Anne Vrieze, M.D., Max Nieuwdorp, M.D., Ph.D., Susana Fuentes, Ph.D., Erwin G. Zoetendal, Ph.D., Willem M. de Vos, Ph.D., Caroline E. Visser, M.D., Ph.D., Ed J. Kuijper, M.D., Ph.D., Joep F.W.M. Bartelsman, M.D., Jan G.P. Tijssen, Ph.D., Peter Speelman, M.D., Ph.D., Marcel G.W. Dijkgraaf, Ph.D., and Josbert J. Keller, M.D., Ph.D.
January 16, 2013 | DOI: 10.1056/NEJMoa1205037

Comments open through January 23, 2013

Share:

Abstract | Article | References | Comments

BACKGROUND

Recurrent *Clostridium difficile* infection is difficult to treat, and failure rates for antibiotic therapy are high. We studied the effect of duodenal infusion of donor feces in patients with recurrent *C. difficile* infection.

Full Text of Background...

MEDIA IN THIS ARTICLE

FIGURE 1

Enrollment and Outcomes.

TOOLS

PDF | E-Mail | Print | Save | Download Citation | Article Alert | Supplementary Material | Reprints | Permissions | Share/Bookmark

TOPICS

Gastroenterology | Bacterial Infections

MORE IN

Research

TRENDS

Most Viewed (Last Week)

ORIGINAL ARTICLE

Duodenal Infusion of Donor Feces for

van Nood et al. (2013) NEJM

About mechanistic analysis

Goal: Understand the exact changes in variables that lead to changes in other variables for individual objects.

- Incredibly hard to infer, except in simple situations
- Usually modeled by a deterministic set of equations (physical/engineering science)
- Generally the random component of the data is measurement error
- If the equations are known but the parameters are not, they may be inferred with data analysis

Mechanistic analysis

U.S. Department of Transportation
Federal Highway Administration

Federal Highway Administration
RESOURCE CENTER

PAVEMENT AND MATERIALS

Mechanistic – Empirical Pavement Design

Problem: Empirical Design Process Restrict Performance Prediction

Accurately predicting performance and durability is critical to improving the design of new and existing pavements. Poor performance increases traffic congestion, compromises public safety, and raises maintenance costs due to frequent repairs. Each year, transportation agencies spend more than $20 billion in Federal funds to improve the Nation's pavements. Existing design procedures are based upon the 1950's AASHO Road Test and use empirical relationships. Presently, pavement designs often exceed the data limits and conditions used in the AASHTO Road Test have been exceeded. Pavement with expected traffic as much as 30 times greater are

Deployment Process:

The Federal Highway Administration (FHWA) organized the Design Guide Implementation Team (DGIT) to inform the FHWA division offices, State highway agencies, industry members, and other organizations and experts about the upcoming guide and to help potential users prepare for it. To introduce the guide and to discuss implementation issues, the DGIT has developed a one-day workshop. Seven of these workshops will be held across the Nation, starting on May 25, 2004, in Biloxi, MS. Other workshops will be held in Vancouver, WA (June); Indianapolis, IN (July); Hawaii (July); Mystic, CT (August); Kansas City, KS (September); and Phoenix, AZ (October).

http://www.fhwa.dot.gov/resourcecenter/teams/pavement/pave_3pdg.pdf

What is data?

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Definition of data

> “Data are values of qualitative or quantitative variables, belonging to a set of items.”

http://en.wikipedia.org/wiki/Data

Definition of data

> “Data are values of qualitative or quantitative variables, belonging to a set of items.”

http://en.wikipedia.org/wiki/Data

Set of items: Sometimes called the population; the set of objects you are interested in

Definition of data

> “Data are values of qualitative or quantitative variables, belonging to a set of items.”

http://en.wikipedia.org/wiki/Data

Variables: A measurement or characteristic of an item.

Definition of data

> “Data are values of qualitative or quantitative variables, belonging to a set of items.”

http://en.wikipedia.org/wiki/Data

Qualitative: Country of origin, sex, treatment

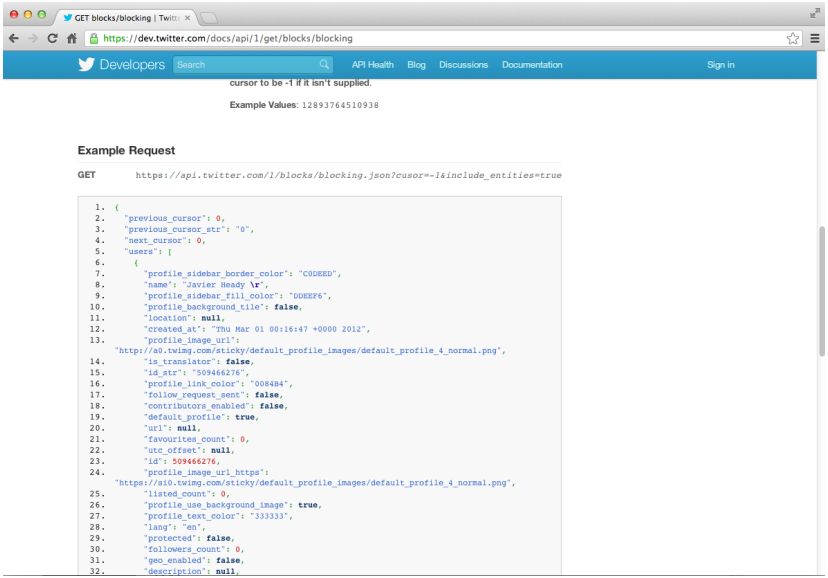
Quantitative: Height, weight, blood pressure

What do data look like?

```
@HWI-EAS121:4:100:1783:550#0/1
CGTTACGAGATCGGAAGAGCGGTTCAGCAGGAATGCCGAGACGGATCTCGTATGCGGTCTGCTGCGTGACAAGACAGGGG
+HWI-EAS121:4:100:1783:550#0/1
aaaaa`b_aa`aa`YaX]aZ`aZM^Z]YRa]YSG[[ZREQLHESDHNDDHNMEEDDMPENITKFLFEEDDDHEJQMEDDD
@HWI-EAS121:4:100:1783:1611#0/1
GGGTGGGCATTTCCACTCGCAGTATGGGTTGCCGCACGACAGGCAGCGGTCAGCCTGCGCTTTGGCCTGGCCTTCGGAAA
+HWI-EAS121:4:100:1783:1611#0/1
a```^\__`_```^a``a`^a_^__]a_]\]`a_______`_^^`]X]_]XTV_\]]NX_XVX]]_TTTTG[VTHPN]VFDZ
@HWI-EAS121:4:100:1783:322#0/1
CGTTTATGTTTTTGAATATGTCTTATCTTAACGGTTATATTTTAGATGTTGGTCTTATTCTAACGGTCATATATTTTCTA
+HWI-EAS121:4:100:1783:322#0/1
abaa`^aaaaabbbaababbbbbb`bbbb_bbbbbbbb`bbbaV^_a``a``]``aT]a__V\]]_]^a`]a_abbaV__
@HWI-EAS121:4:100:1783:1394#0/1
GGGTCTTTATTGGTCTGGTGATCCCCCATATTCTCCGGTTGTGTGGTTTAACCGATCATCGCGCATTACTTCCCGGCTGC
+HWI-EAS121:4:100:1783:1394#0/1
```[aa\b^^[]aabbb][`a_abbb`a``bbbbbabaabaaaab_VZa_^___bab_X`[a\HV_[_]_[^_X\T_VQQ
@HWI-EAS121:4:100:1783:207#0/1
CCCTGGGAGATCGGAAGAGCGGTTCAGCAGGAATGCCGAGACCGATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAACA
+HWI-EAS121:4:100:1783:207#0/1
abba`Xa\^\\`aa]ba__bba[a_O_a`aa`aa`a]^V]X_a^YS\R_\H_[]\ZTDUZZUSOPX]]POP\GS\WSHHD
@HWI-EAS121:4:100:1783:455#0/1
GGGTAATTCAGGGACAATGTAATGGCTGCACAAAAAAATACATCTTTCATGTTCCATTGCACCATTGACAAATACATATT
+HWI-EAS121:4:100:1783:455#0/1
abb_babbabaabbbbbbbbbbbbbbbba\`b`\abbbabbbbabbbbbbaabbbbb`bb`ab_O_bab_Q_bbabaa_a
```

http://brianknaus.com/software/srtoolbox/s_4_1_sequence80.txt
```

What do data look like?



https://dev.twitter.com/docs/api/1/get/blocks/blocking

What do data look like?

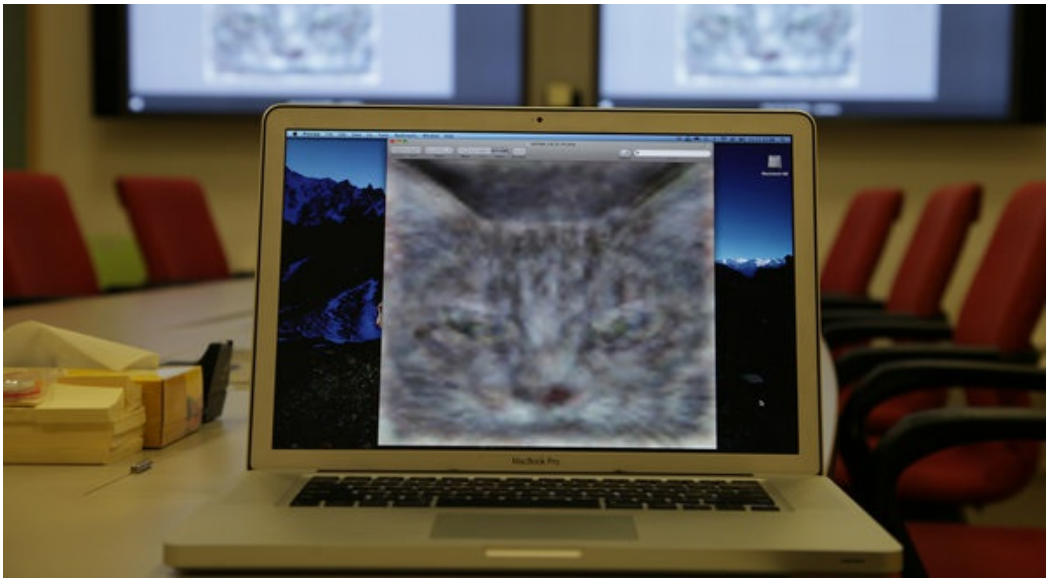
```
-------------------------  ALLERGIES  -------------------------      -------------------------  MEDICATION HISTORY  ----------------

ast Updated: 01 Dec 2011 @ 0851                                     Last Updated: 11 Apr 2011 @ 1737

                                                                    Medication: AMLODIPINE BESYLATE 10MG TAB
llergy Name:        TRIMETHOPRIM                                    Instructions: TAKE ONE TABLET BY MOUTH TAKE ONE-HALF TABLET FOR
ocation:            DAYT29                                          GRAPEFRUIT JUICE--
ate Entered:        09 Mar 2011                                     Status: Active
eaction:                                                            Refills Remaining: 3
llergy Type:        DRUG                                            Last Filled On: 20 Aug 2010
A Drug Class:       ANTI-INFECTIVES,OTHER                           Initially Ordered On: 13 Aug 2010
bserved/Historical: HISTORICAL                                      Quantity: 45
omments:            The reaction to this allergy was MILD (NO SQUELAE)  Days Supply: 90
                                                                    Pharmacy: DAYTON
llergy Name:        TRAMADOL                                        Prescription Number: 2718953
ocation:            DAYT29
ate Entered:        09 Mar 2011                                     Medication: IBUPROFEN 600MG TAB
eaction:            URINARY RETENTION                               Instructions: TAKE ONE TABLET BY MOUTH FOUR TIMES A DAY WITH FOO
llergy Type:        DRUG                                            Status: Active
A Drug Class:       NON-OPIOID ANALGESICS                           Refills Remaining: 3
bserved/Historical: HISTORICAL                                      Last Filled On: 20 Aug 2010
omments:            gradually worsening difficulty emptying bladder  Initially Ordered On: 01 Jul 2010
```

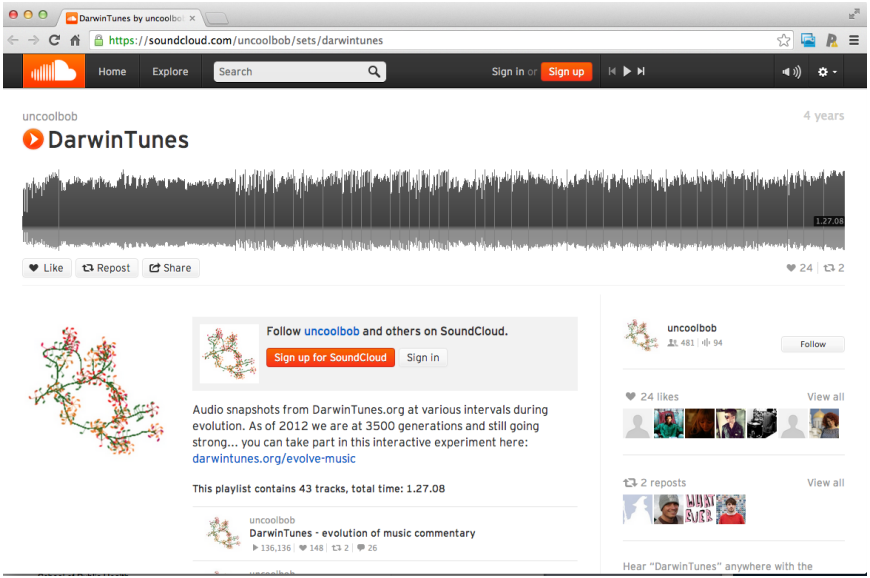
http://blue-button.github.com/challenge/

What do data look like?



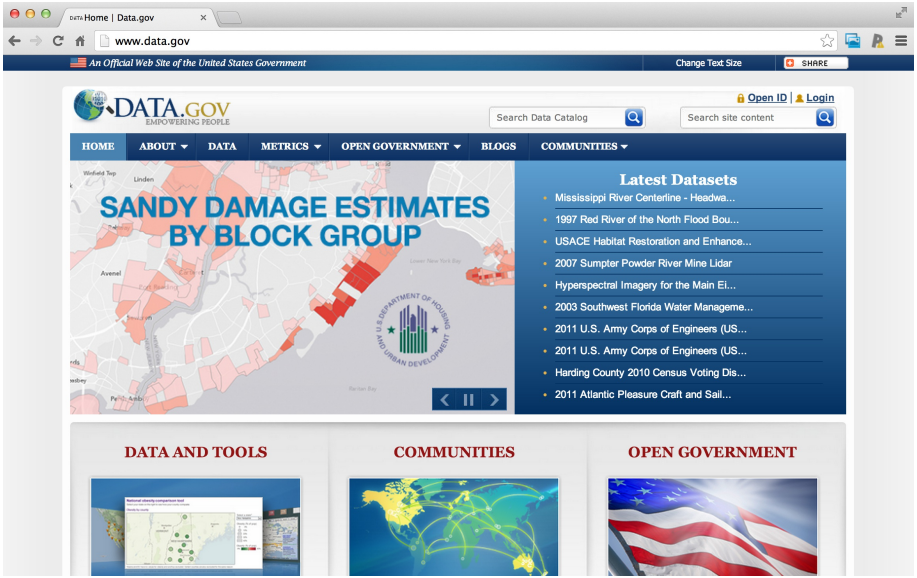
http://www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html?pagewanted=all&_r=0

What do data look like?



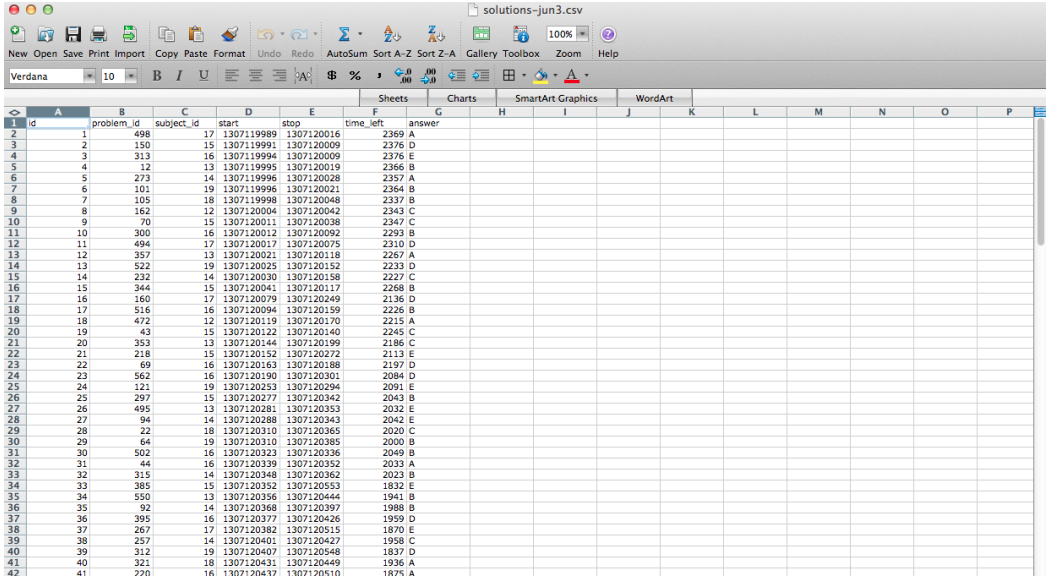
http://www.pnas.org/content/109/30/12081.full https://soundcloud.com/uncoolbob/sets/darwintunes

What do data look like?



http://www.data.gov/

What do data look like? Rarely



	A	B	C	D	E	F	G
1	id	problem_id	subject_id	start	stop	time_left	answer
2	1	498	17	1307119989	1307120016	2369	A
3	2	150	15	1307119991	1307120009	2376	D
4	3	313	16	1307119994	1307120009	2376	E
5	4	12	13	1307119995	1307120019	2366	B
6	5	273	14	1307119996	1307120028	2357	A
7	6	101	19	1307119996	1307120021	2364	B
8	7	105	18	1307119998	1307120048	2337	B
9	8	162	12	1307120004	1307120042	2343	C
10	9	70	15	1307120011	1307120038	2347	C
11	10	300	16	1307120012	1307120092	2293	B
12	11	494	17	1307120017	1307120075	2310	D
13	12	357	13	1307120021	1307120118	2267	A
14	13	522	19	1307120025	1307120152	2233	D
15	14	232	14	1307120030	1307120158	2227	C
16	15	344	15	1307120041	1307120117	2268	B
17	16	160	17	1307120079	1307120249	2136	D
18	17	516	16	1307120094	1307120159	2226	B
19	18	472	12	1307120119	1307120170	2215	A
20	19	43	15	1307120122	1307120140	2245	C
21	20	353	13	1307120144	1307120199	2186	C
22	21	218	15	1307120152	1307120272	2113	E
23	22	69	16	1307120163	1307120188	2197	D
24	23	562	16	1307120190	1307120301	2084	D
25	24	121	19	1307120253	1307120294	2091	E
26	25	297	15	1307120277	1307120342	2043	B
27	26	495	13	1307120281	1307120353	2032	E
28	27	94	14	1307120288	1307120343	2042	E
29	28	22	18	1307120310	1307120365	2020	C
30	29	64	19	1307120310	1307120385	2000	B
31	30	502	16	1307120323	1307120336	2049	B
32	31	44	16	1307120339	1307120352	2033	A
33	32	315	14	1307120348	1307120362	2023	B
34	33	385	15	1307120352	1307120553	1832	E
35	34	550	13	1307120356	1307120444	1941	B
36	35	92	14	1307120368	1307120397	1988	B
37	36	395	16	1307120377	1307120426	1959	D
38	37	267	17	1307120382	1307120515	1870	E
39	38	257	14	1307120401	1307120427	1958	C
40	39	312	19	1307120407	1307120548	1837	D
41	40	321	18	1307120431	1307120449	1936	A
42	41	220	16	1307120437	1307120510	1875	A

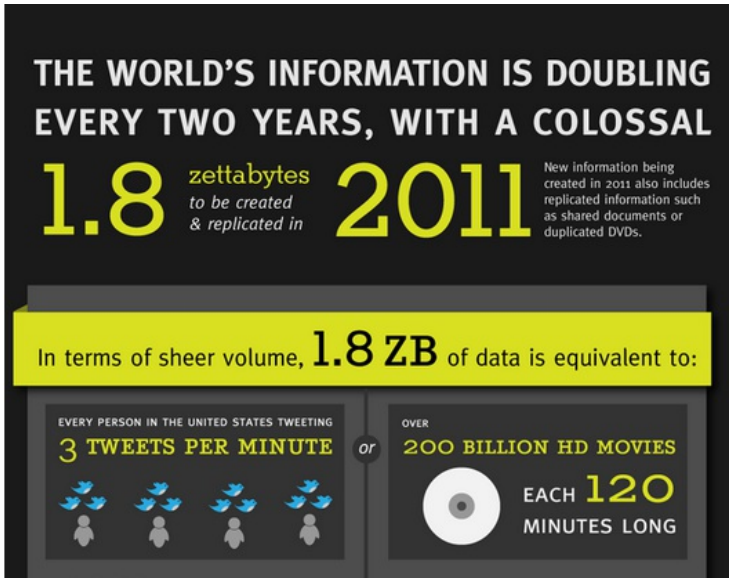
The data is the second most important thing

- The most important thing in data science is the question
- The second most important is the data
- Often the data will limit or enable the questions
- But having data can't save you if you don't have a question

What about big data?

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

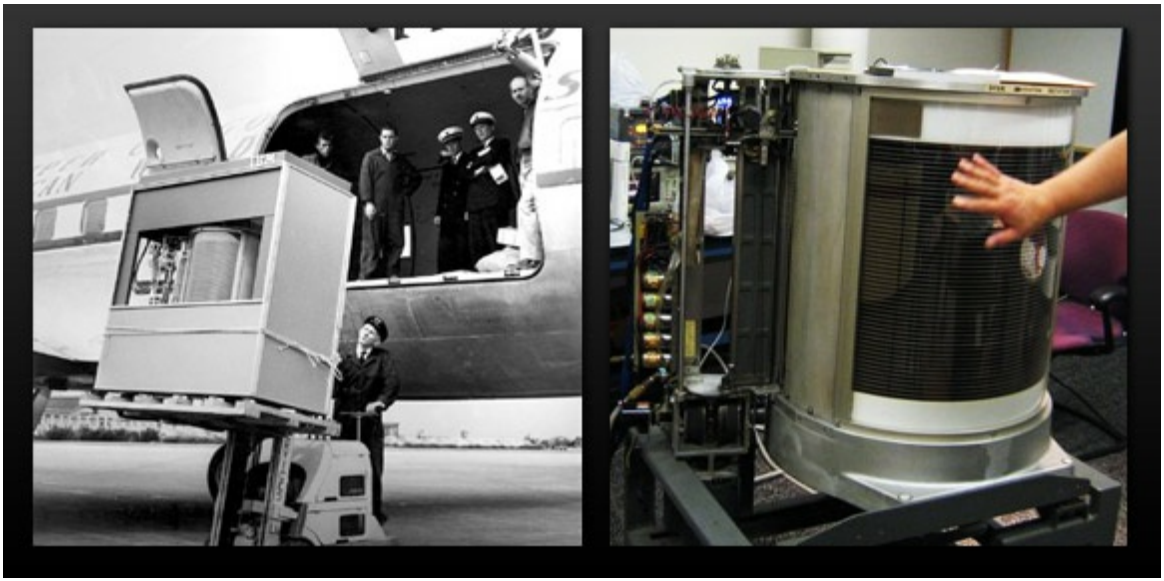
How much is there?



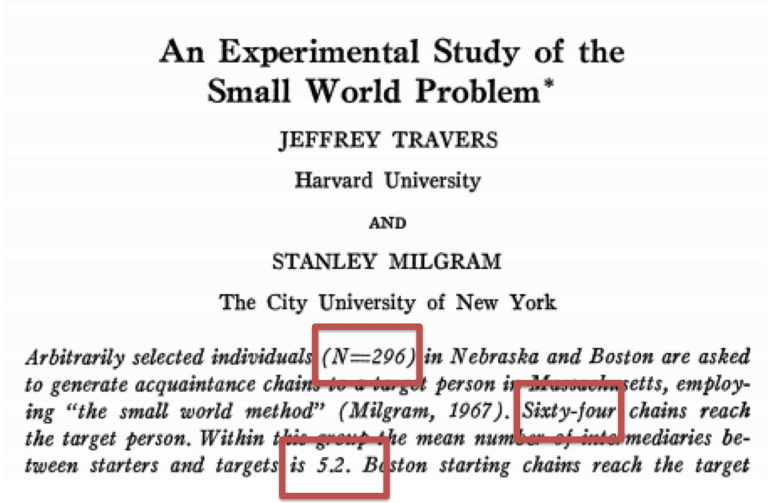
http://mashable.com/2011/06/28/data-infographic/

So what about big data?

Depends on your perspective



Why big data now?



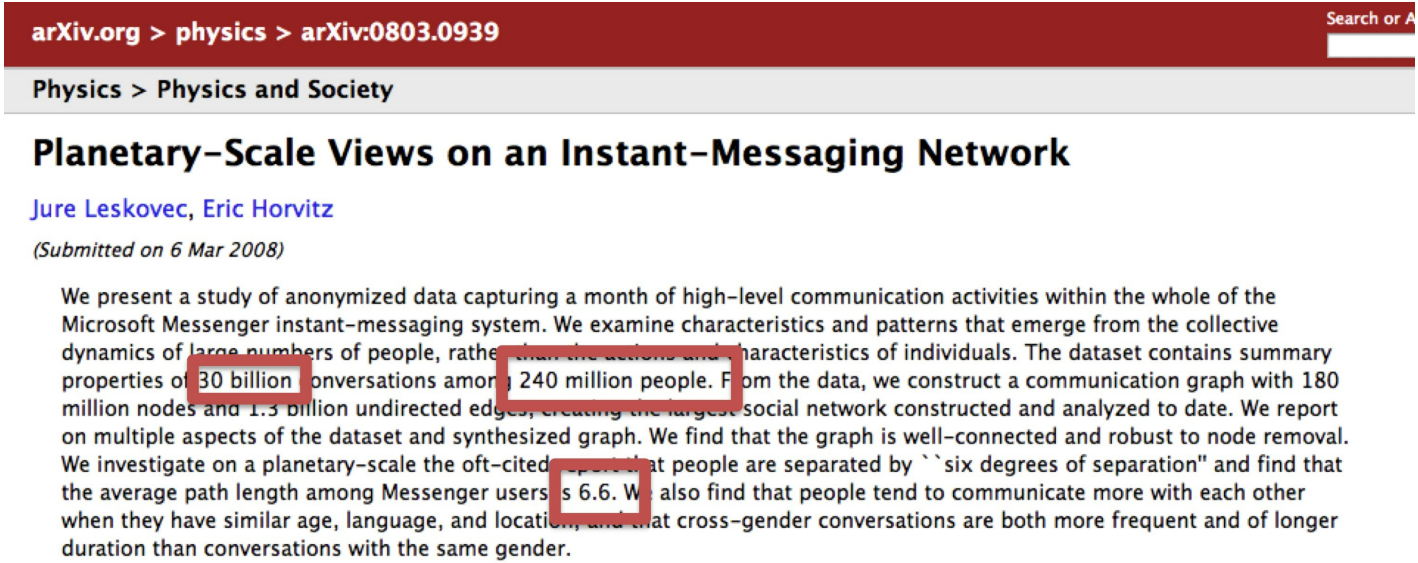
An Experimental Study of the Small World Problem*

JEFFREY TRAVERS
Harvard University
AND
STANLEY MILGRAM
The City University of New York

Arbitrarily selected individuals (N=296) in Nebraska and Boston are asked to generate acquaintance chains to a target person in Massachusetts, employing "the small world method" (Milgram, 1967). Sixty-four chains reach the target person. Within this group the mean number of intermediaries between starters and targets is 5.2. Boston starting chains reach the target

Travers and Milgram (1969) Sociometry

Why big data now?



arXiv.org > physics > arXiv:0803.0939

Search or A

Physics > Physics and Society

Planetary-Scale Views on an Instant-Messaging Network

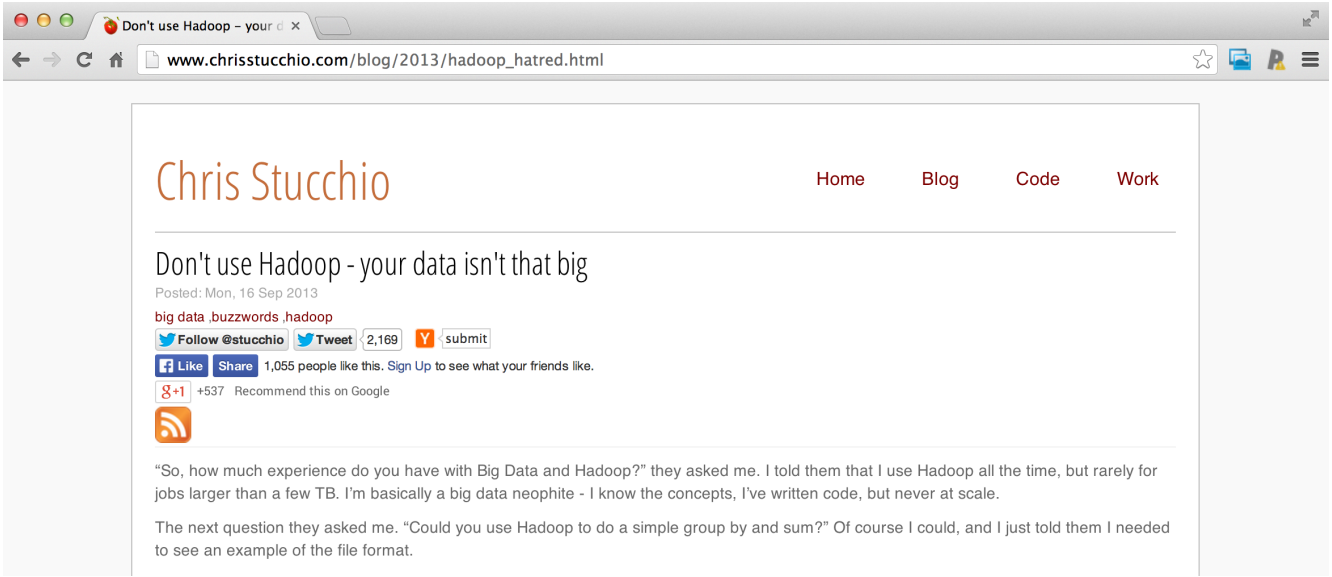
Jure Leskovec, Eric Horvitz

(Submitted on 6 Mar 2008)

We present a study of anonymized data capturing a month of high-level communication activities within the whole of the Microsoft Messenger instant-messaging system. We examine characteristics and patterns that emerge from the collective dynamics of large numbers of people, rathe haracteristics of individuals. The dataset contains summary properties of 30 billion onversations amon 240 million people. F om the data, we construct a communication graph with 180 million nodes and 1.3 billion undirected edge social network constructed and analyzed to date. We report on multiple aspects of the dataset and synthesized graph. We find that the graph is well-connected and robust to node removal. We investigate on a planetary-scale the oft-cited t people are separated by ``six degrees of separation'' and find that the average path length among Messenger users s 6.6. W also find that people tend to communicate more with each other when they have similar age, language, and locatio t cross-gender conversations are both more frequent and of longer duration than conversations with the same gender.

Leskovec and Horvitz WWW '08

Big or small - you need the right data



http://www.chrisstucchio.com/blog/2013/hadoop_hatred.html

Big or small - you need the right data

> “The data may not contain the answer. The combination of some data and an aching desire for an answer does not ensure that a reasonable answer can be extracted from a given body of data...”

Tukey

Big or small - you need the right data

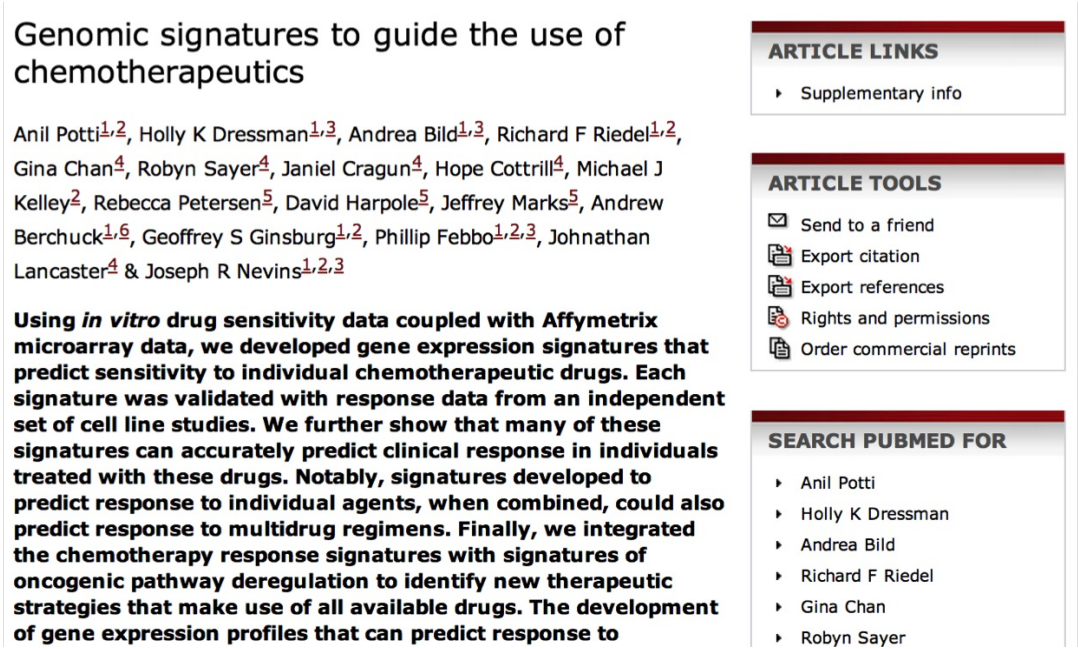
> “...no matter how big the data are.”

Leek

Experimental design

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Why you should care - an exciting result!



Genomic signatures to guide the use of chemotherapeutics

Anil Potti[1,2], Holly K Dressman[1,3], Andrea Bild[1,3], Richard F Riedel[1,2], Gina Chan[4], Robyn Sayer[4], Janiel Cragun[4], Hope Cottrill[4], Michael J Kelley[2], Rebecca Petersen[5], David Harpole[5], Jeffrey Marks[5], Andrew Berchuck[1,6], Geoffrey S Ginsburg[1,2], Phillip Febbo[1,2,3], Johnathan Lancaster[4] & Joseph R Nevins[1,2,3]

Using *in vitro* drug sensitivity data coupled with Affymetrix microarray data, we developed gene expression signatures that predict sensitivity to individual chemotherapeutic drugs. Each signature was validated with response data from an independent set of cell line studies. We further show that many of these signatures can accurately predict clinical response in individuals treated with these drugs. Notably, signatures developed to predict response to individual agents, when combined, could also predict response to multidrug regimens. Finally, we integrated the chemotherapy response signatures with signatures of oncogenic pathway deregulation to identify new therapeutic strategies that make use of all available drugs. The development of gene expression profiles that can predict response to

ARTICLE LINKS

- Supplementary info

ARTICLE TOOLS

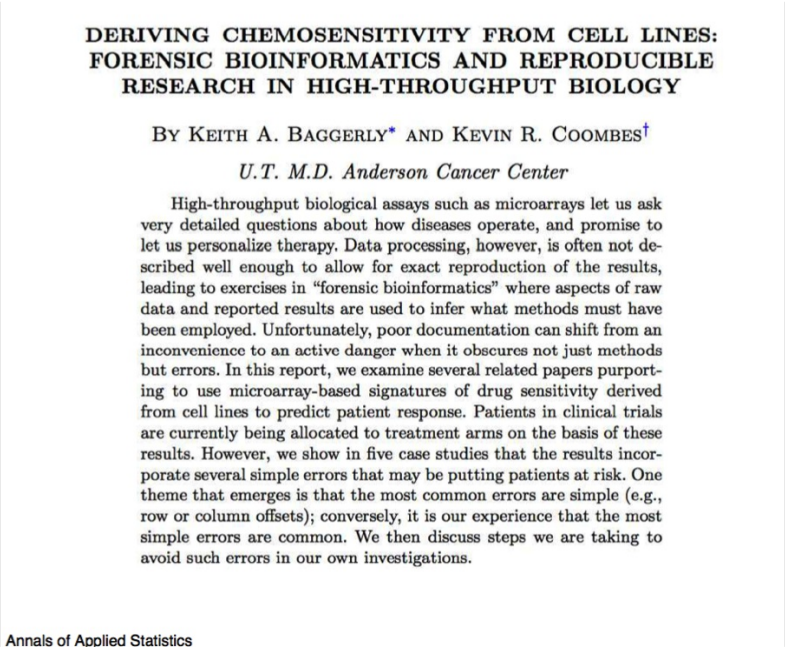
- Send to a friend
- Export citation
- Export references
- Rights and permissions
- Order commercial reprints

SEARCH PUBMED FOR

- Anil Potti
- Holly K Dressman
- Andrea Bild
- Richard F Riedel
- Gina Chan
- Robyn Sayer

http://www.nature.com/nm/journal/v12/n11/full/nm1491.html

Why you should care - uh oh!



DERIVING CHEMOSENSITIVITY FROM CELL LINES: FORENSIC BIOINFORMATICS AND REPRODUCIBLE RESEARCH IN HIGH-THROUGHPUT BIOLOGY

By Keith A. Baggerly* and Kevin R. Coombes†

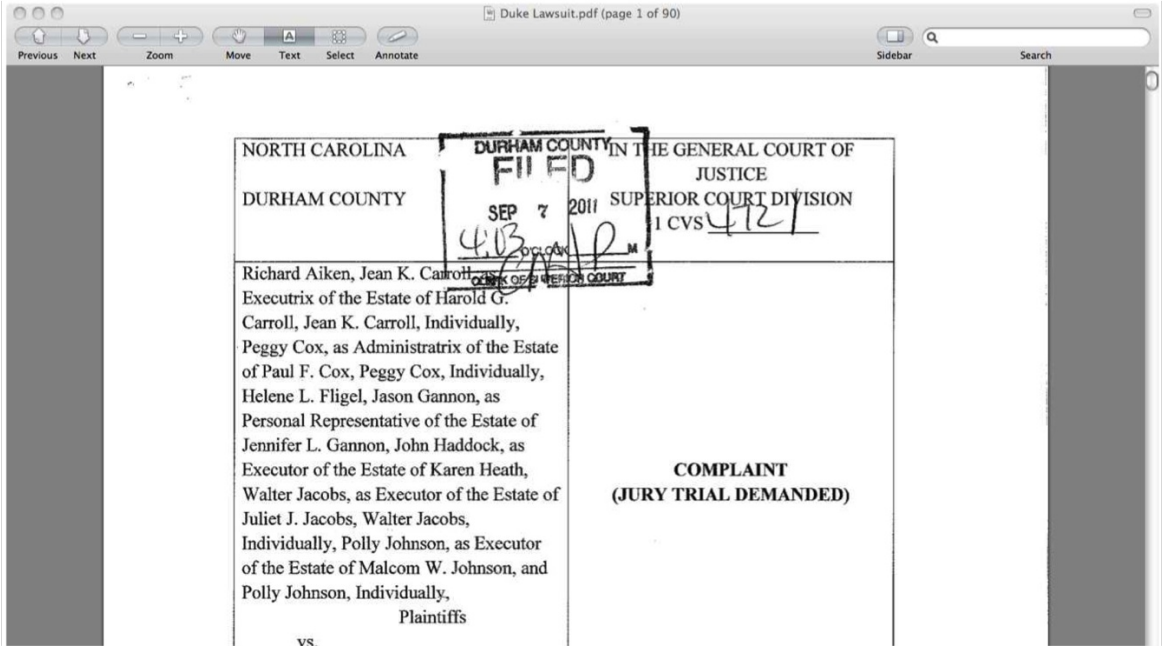
U.T. M.D. Anderson Cancer Center

High-throughput biological assays such as microarrays let us ask very detailed questions about how diseases operate, and promise to let us personalize therapy. Data processing, however, is often not described well enough to allow for exact reproduction of the results, leading to exercises in "forensic bioinformatics" where aspects of raw data and reported results are used to infer what methods must have been employed. Unfortunately, poor documentation can shift from an inconvenience to an active danger when it obscures not just methods but errors. In this report, we examine several related papers purporting to use microarray-based signatures of drug sensitivity derived from cell lines to predict patient response. Patients in clinical trials are currently being allocated to treatment arms on the basis of these results. However, we show in five case studies that the results incorporate several simple errors that may be putting patients at risk. One theme that emerges is that the most common errors are simple (e.g., row or column offsets); conversely, it is our experience that the most simple errors are common. We then discuss steps we are taking to avoid such errors in our own investigations.

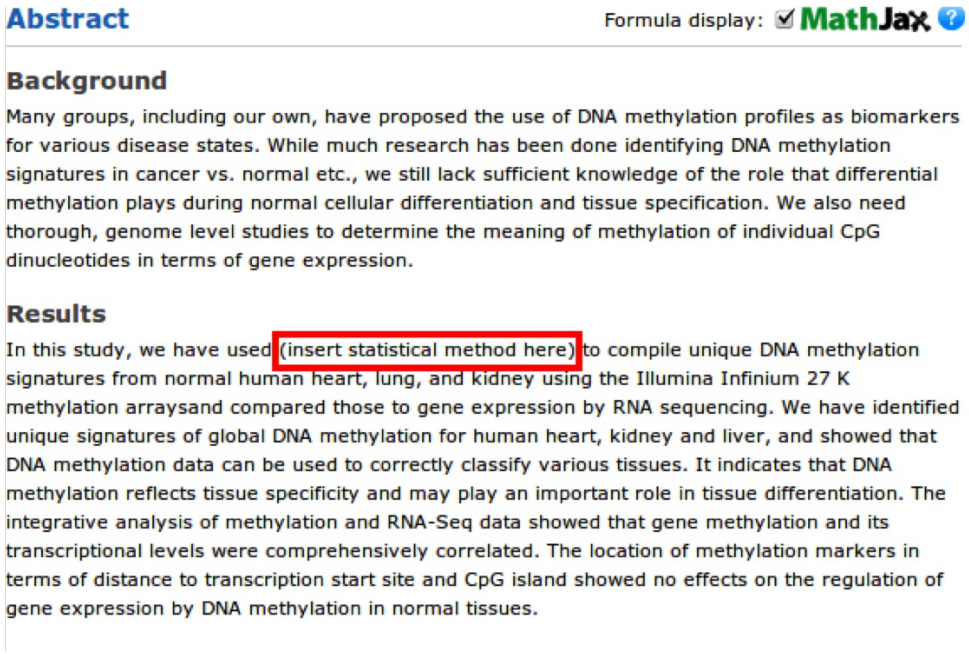
Annals of Applied Statistics

http://arxiv.org/pdf/1010.1092.pdf

Why you should care - serious trouble



Know and care about the analysis plan!



Abstract

Formula display: MathJax

Background

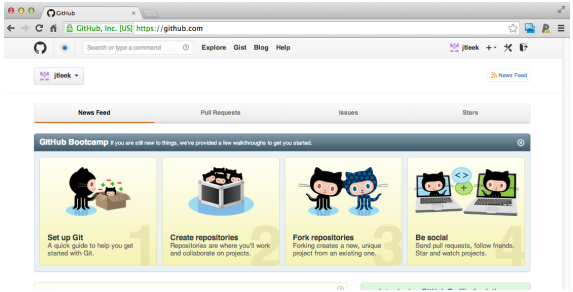
Many groups, including our own, have proposed the use of DNA methylation profiles as biomarkers for various disease states. While much research has been done identifying DNA methylation signatures in cancer vs. normal etc., we still lack sufficient knowledge of the role that differential methylation plays during normal cellular differentiation and tissue specification. We also need thorough, genome level studies to determine the meaning of methylation of individual CpG dinucleotides in terms of gene expression.

Results

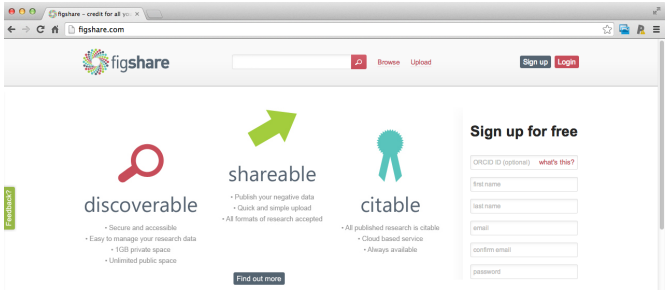
In this study, we have used (insert statistical method here) to compile unique DNA methylation signatures from normal human heart, lung, and kidney using the Illumina Infinium 27 K methylation arraysand compared those to gene expression by RNA sequencing. We have identified unique signatures of global DNA methylation for human heart, kidney and liver, and showed that DNA methylation data can be used to correctly classify various tissues. It indicates that DNA methylation reflects tissue specificity and may play an important role in tissue differentiation. The integrative analysis of methylation and RNA-Seq data showed that gene methylation and its transcriptional levels were comprehensively correlated. The location of methylation markers in terms of distance to transcription start site and CpG island showed no effects on the regulation of gene expression by DNA methylation in normal tissues.

http://nsaunders.wordpress.com/2012/07/23/we-really-dont-care-what-statistical-method-you-used/

Have a plan for data and code sharing

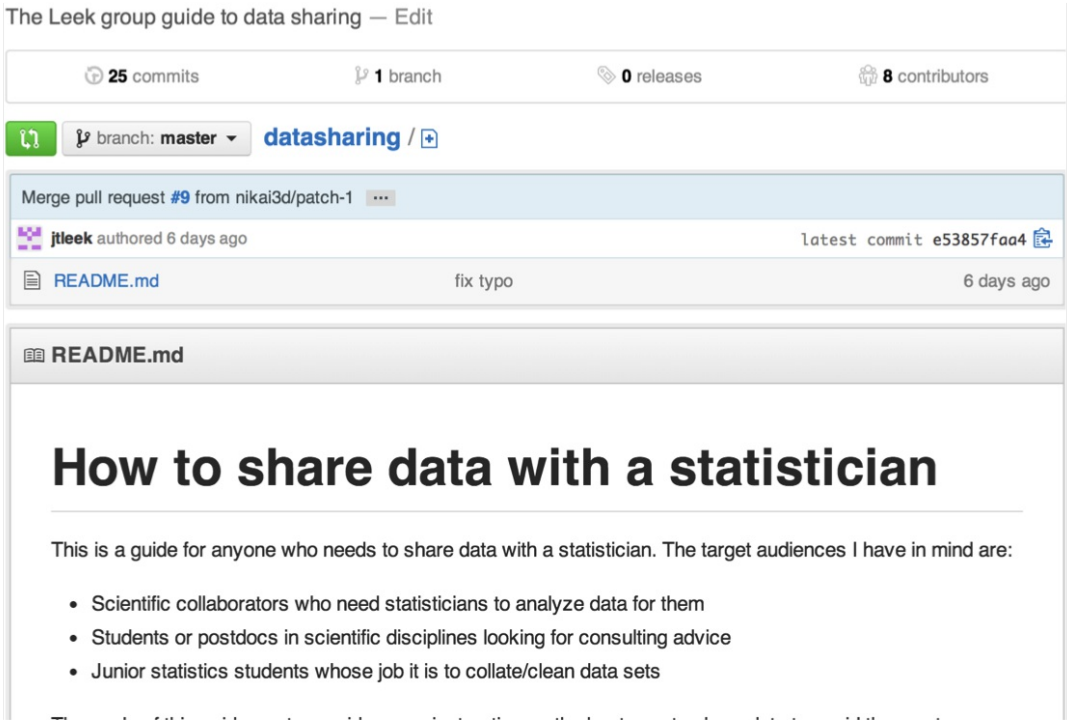


https://github.com/



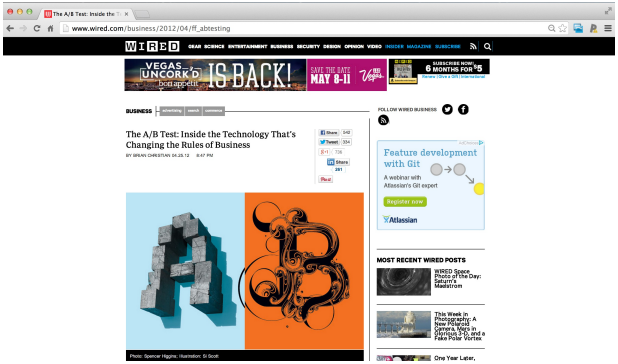
http://figshare.com/

May I recommend?



https://github.com/jtleek/datasharing

Formulate your question in advance



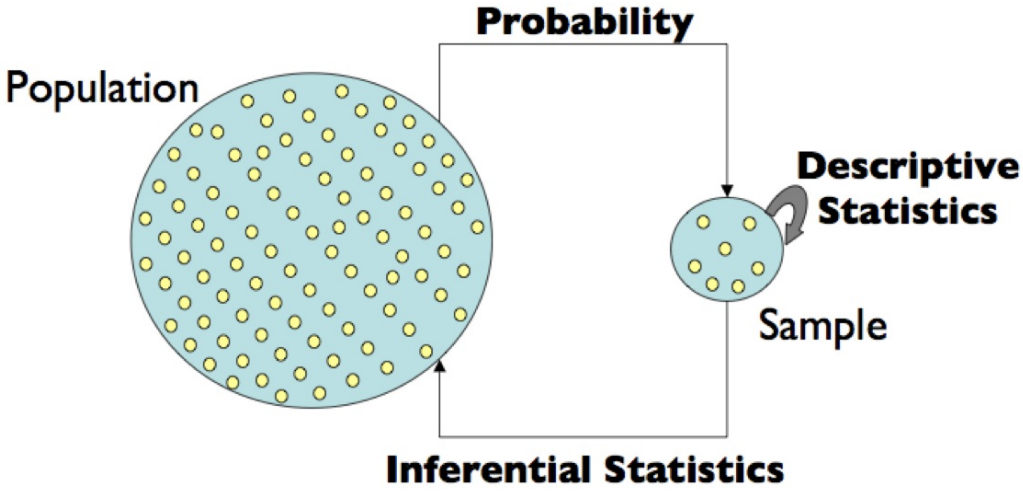
Question: Does changing the text on your website improve donations?

Experiment:

1. Randomly show visitors one version or the other
2. Measure how much they donate
3. Determine which is better

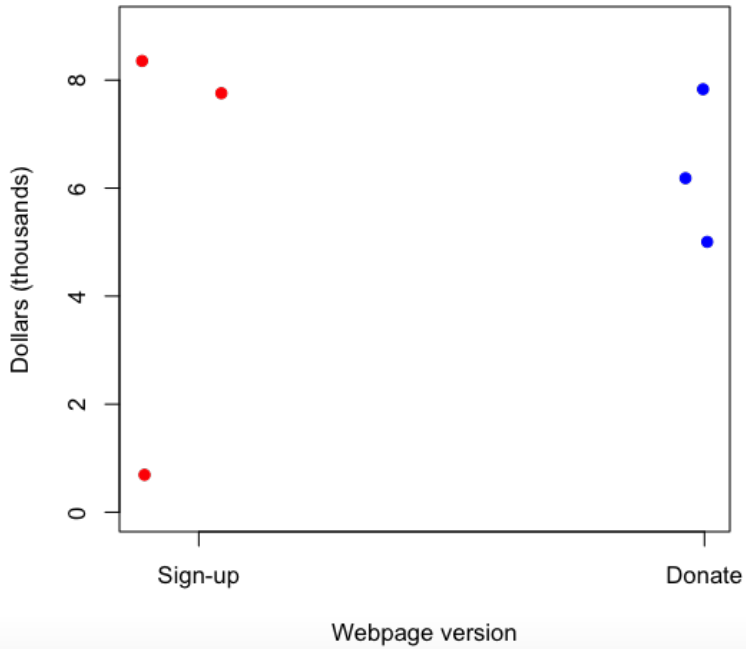
http://www.wired.com/business/2012/04/ff_abtesting

Statistical inference

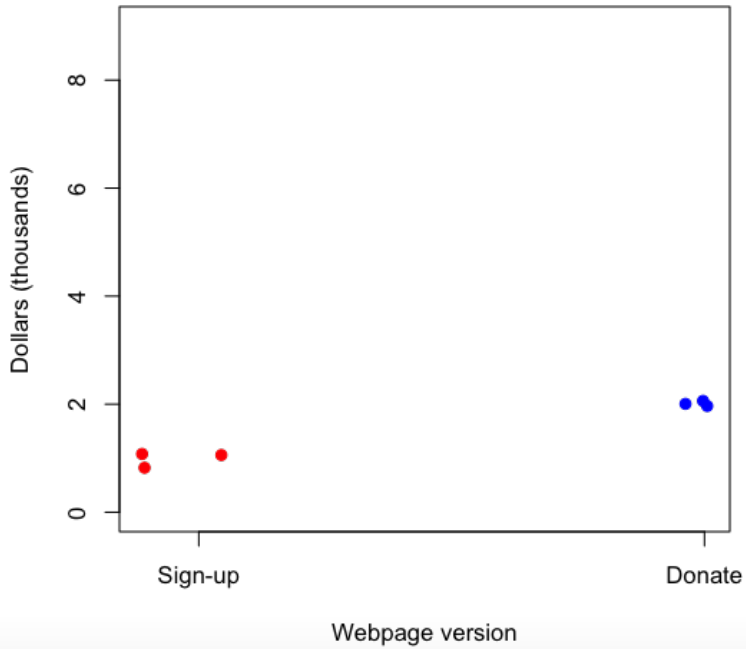


http://www.gs.washington.edu/academics/courses/akey/56008/lecture/lecture2.pdf

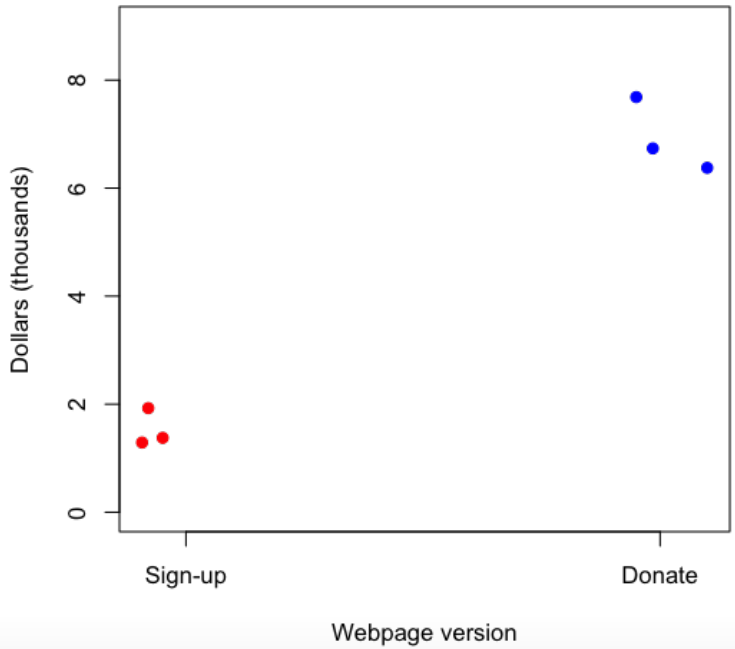
Variability - Scenario 1



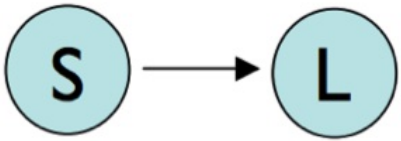
Variability - Scenario 2

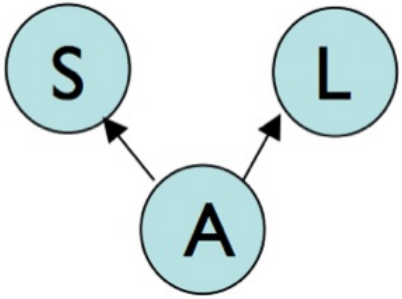


Variability - Scenario 3

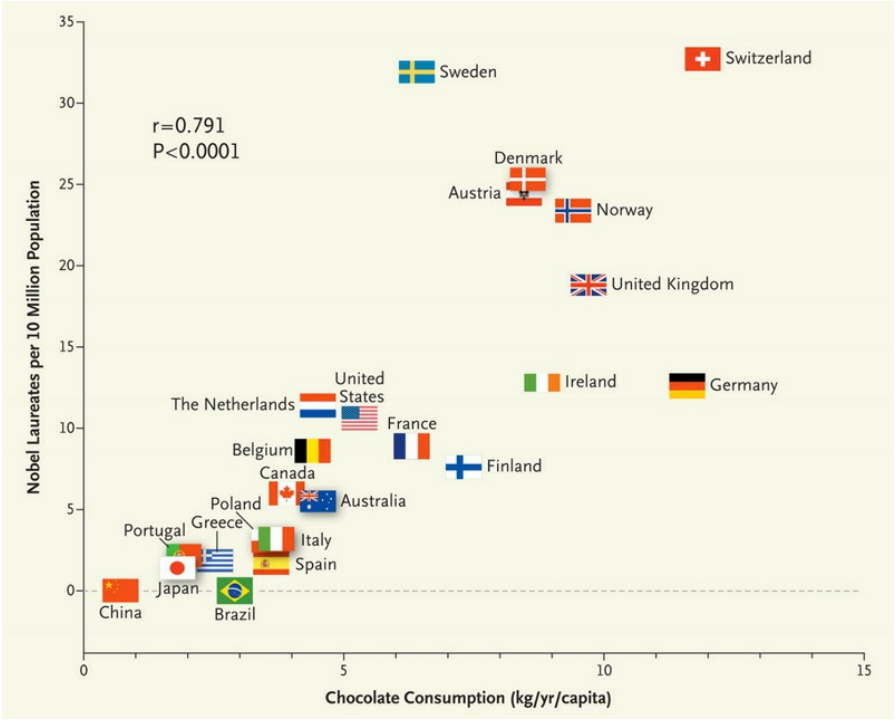


Confounding





Correlation is not causation*



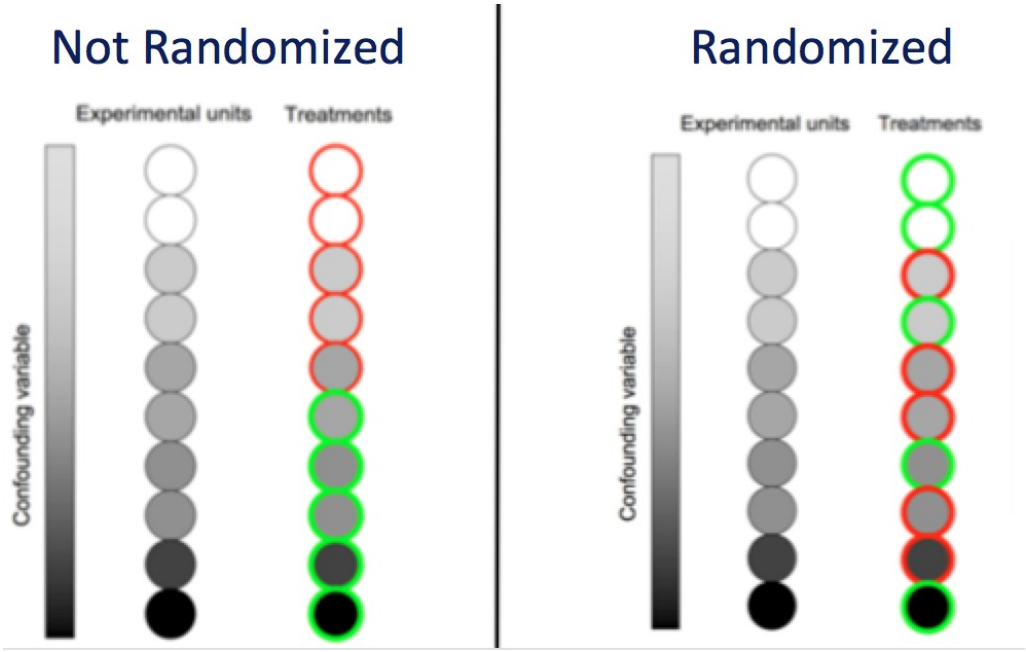
http://www.nejm.org/doi/full/10.1056/NEJMon1211064

*Sometimes called spurious correlation**

Randomization and blocking

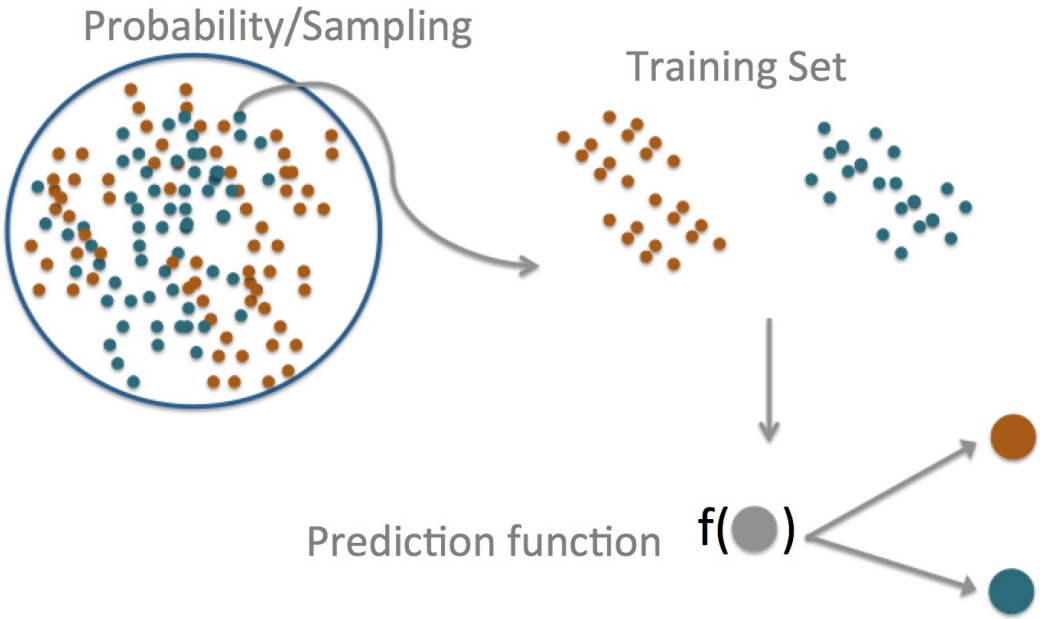
- If you can (and want to) fix a variable
  - Website always says Obama 2014 on it
- If you don't fix a variable, stratify it
  - If you are testing sign up phrases and have two website colors, use both phrases equally on both.
- If you can't fix a variable, randomize it

Why does randomization help?

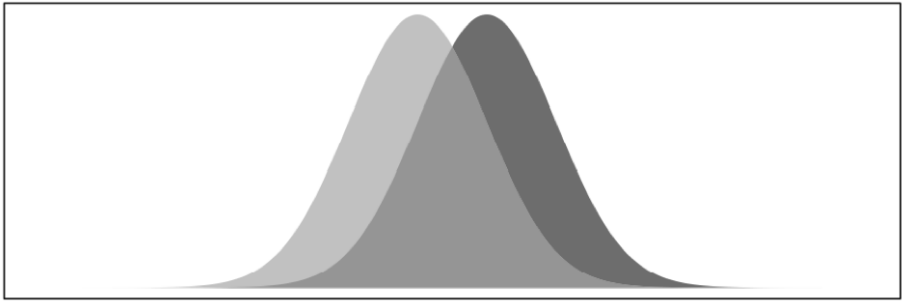


http://www.gs.washington.edu/academics/courses/akey/56008/lecture/lecture1.pdf

Prediction

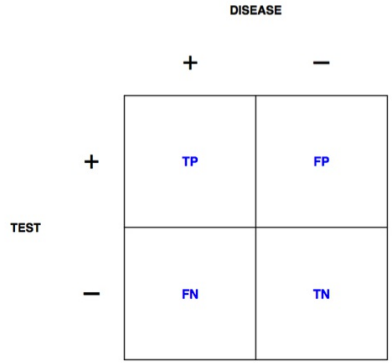


Prediction versus inference



http://www.biostat.jhsph.edu/~iruczins/teaching/140.615/

Prediction key quantities



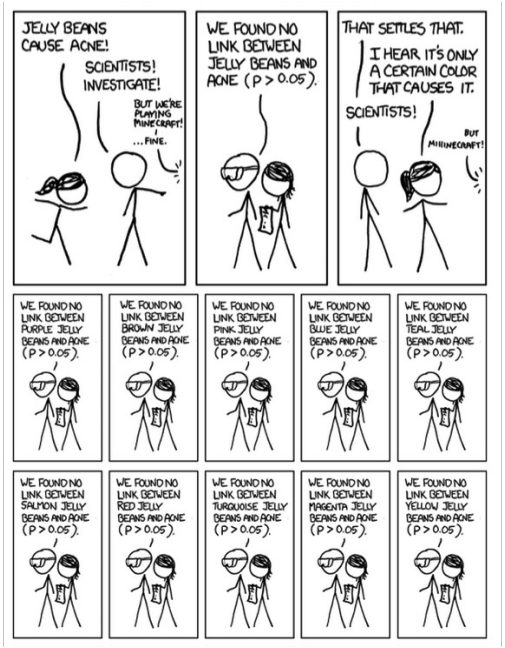
Sensitivity	→	Pr (positive test | disease)
Specificity	→	Pr (negative test | no disease)
Positive Predictive Value	→	Pr (disease | positive test)
Negative Predictive Value	→	Pr (no disease | negative test)
Accuracy	→	Pr (correct outcome)

http://www.biostat.jhsph.edu/~iruczins/teaching/140.615/

Beware data dredging

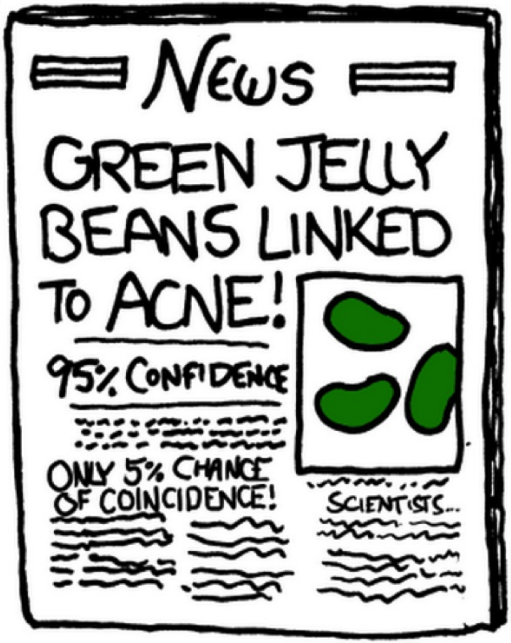
http://xkcd.com/882/

Beware data dredging



http://xkcd.com/882/

Beware data dredging



http://xkcd.com/882/

Summary

- Good experiments
  - Have replication
  - Meaure variability
  - Generalize to the problem you care about
  - Are transparent
- Prediction is not inference
  - Both can be important
- Beware data dredging