

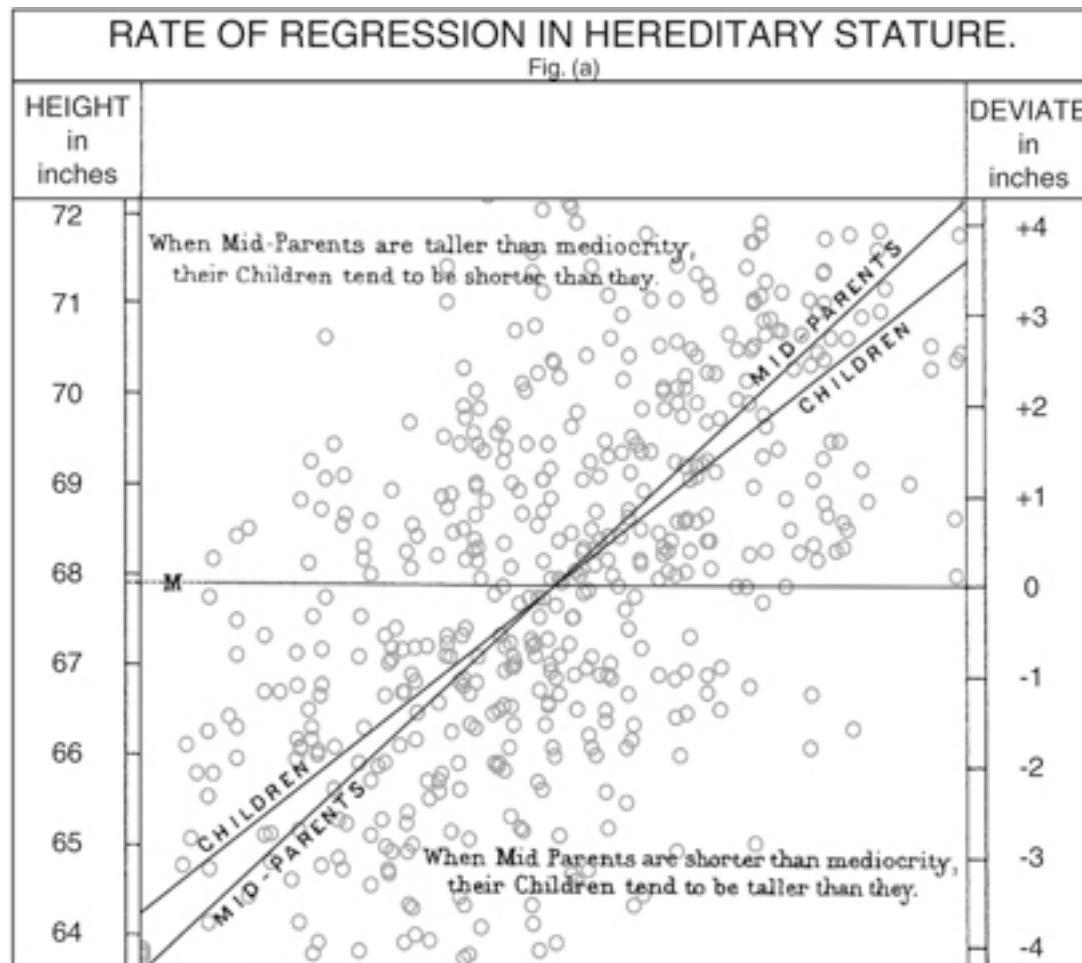
# Basic least squares

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

# Goals of statistical modeling

- Describe the distribution of variables
- Describe the relationship between variables
- Make inferences about distributions or relationships

# Example: Average parent and child heights



<http://www.nature.com/ejhg/journal/v17/n8/full/ejhg20095a.html>

3/19

# Still relevant

## Article

---

*European Journal of Human Genetics* (2009) **17**, 1070–1075; doi:10.1038/ejhg.2009.5; published online 18 February 2009

### Predicting human height by Victorian and genomic methods

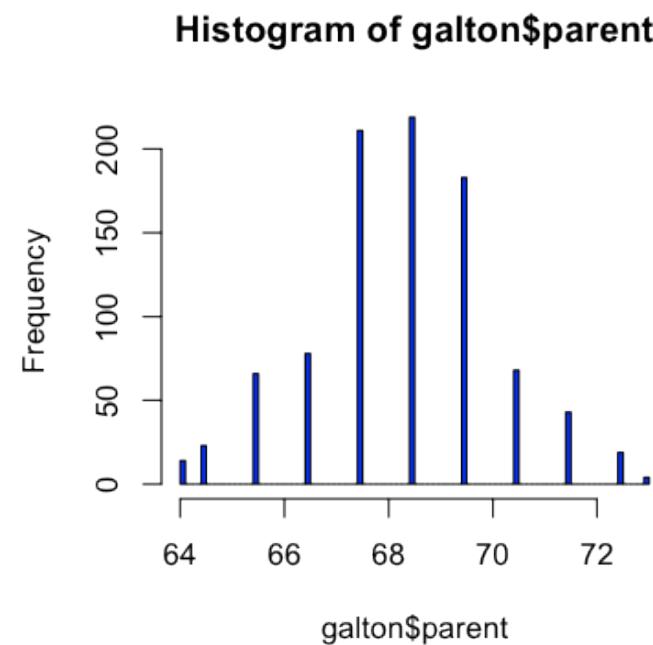
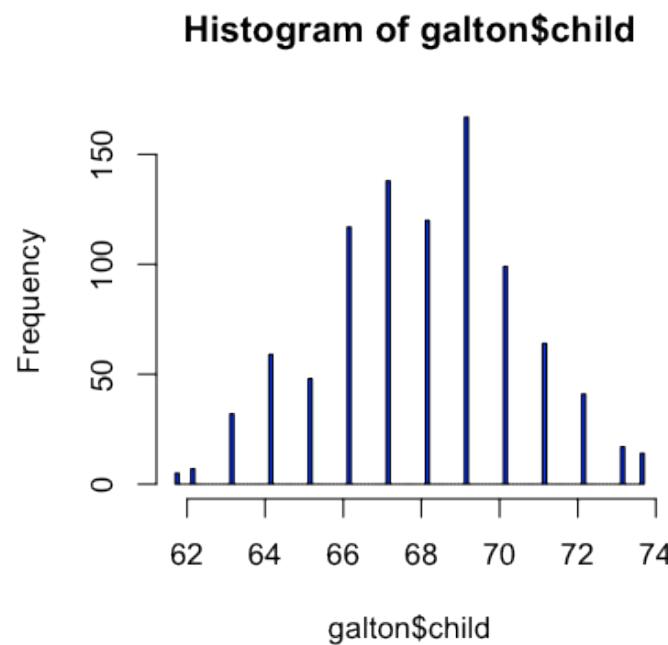
Yurii S Aulchenko<sup>1,2,7</sup>, Maksim V Struchalin<sup>1,3,7</sup>, Nadezhda M Belonogova<sup>2,4</sup>, Tatiana I Axenovich<sup>2</sup>, Michael N Weedon<sup>5</sup>, Albert Hofman<sup>1</sup>, Andre G Uitterlinden<sup>6</sup>, Manfred Kayser<sup>3</sup>, Ben A Oostra<sup>1</sup>, Cornelia M van Duijn<sup>1</sup>, A Cecile J W Janssens<sup>1</sup> and Pavel M Borodin<sup>2,4</sup>

<http://www.nature.com/ejhg/journal/v17/n8/full/ejhg20095a.html>

Predicting height: the Victorian approach beats modern genomics

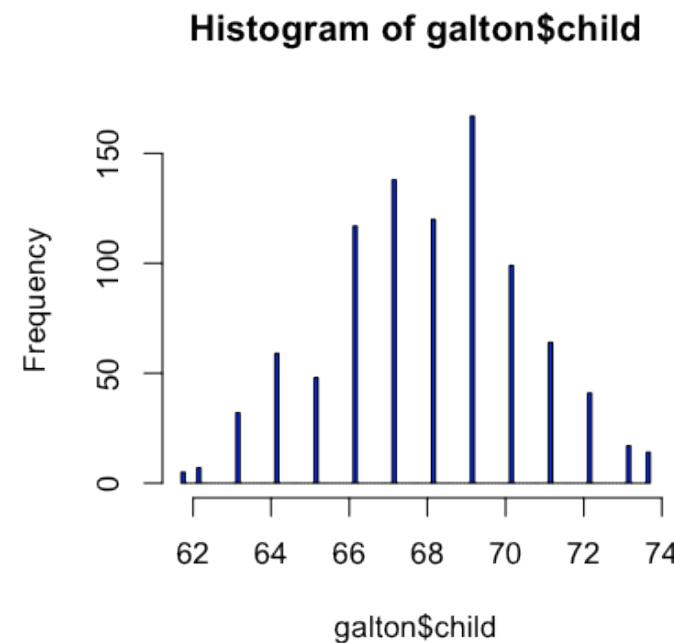
# Load Galton Data

```
library(UsingR); data(galton)
par(mfrow=c(1,2))
hist(galton$child,col="blue",breaks=100)
hist(galton$parent,col="blue",breaks=100)
```



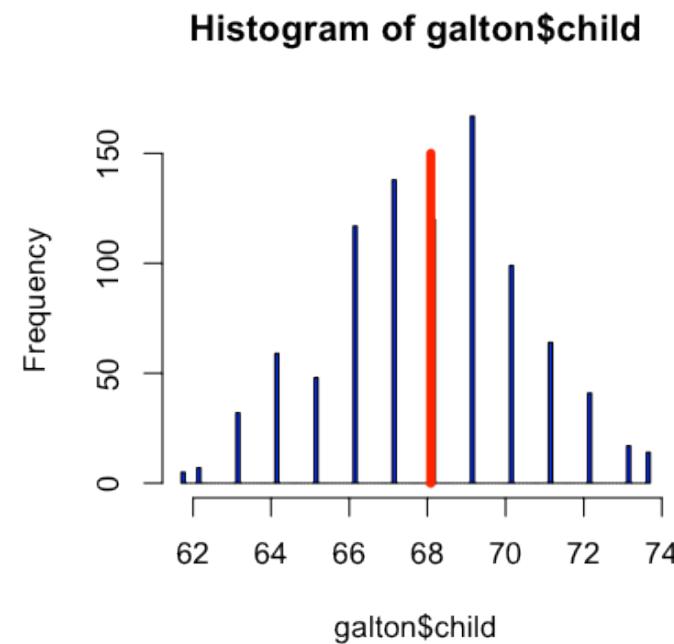
# The distribution of child heights

```
hist(galton$child,col="blue",breaks=100)
```



# Only know the child - average height

```
hist(galton$child,col="blue",breaks=100)
meanChild <- mean(galton$child)
lines(rep(meanChild,100),seq(0,150,length=100),col="red",lwd=5)
```



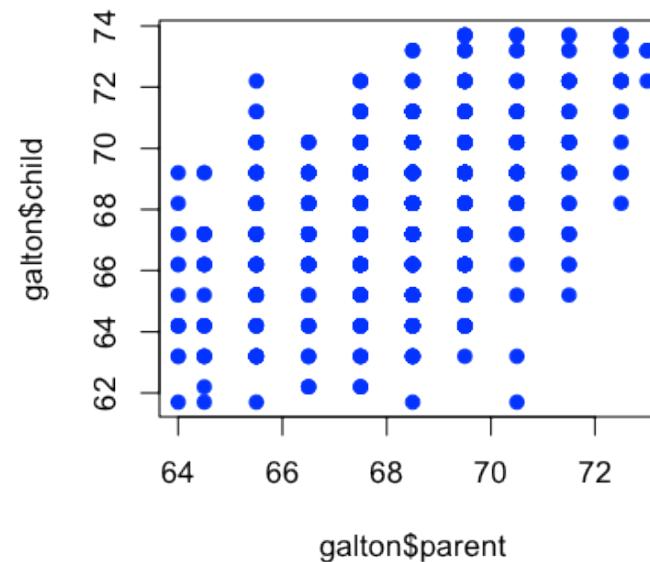
# Only know the child - why average?

If  $C_i$  is the height of child  $i$  then the average is the value of  $\mu$  that minimizes:

$$\sum_{i=1}^{928} (C_i - \mu)^2$$

# What if we plot child versus average parent

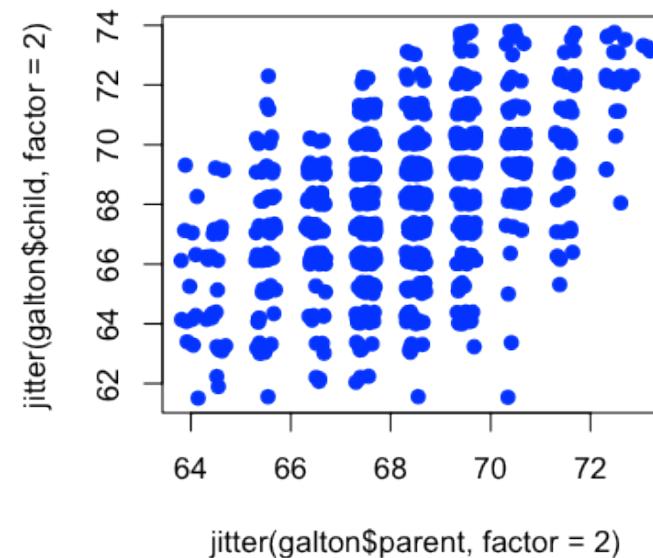
```
plot(galton$parent, galton$child, pch=19, col="blue")
```



9/19

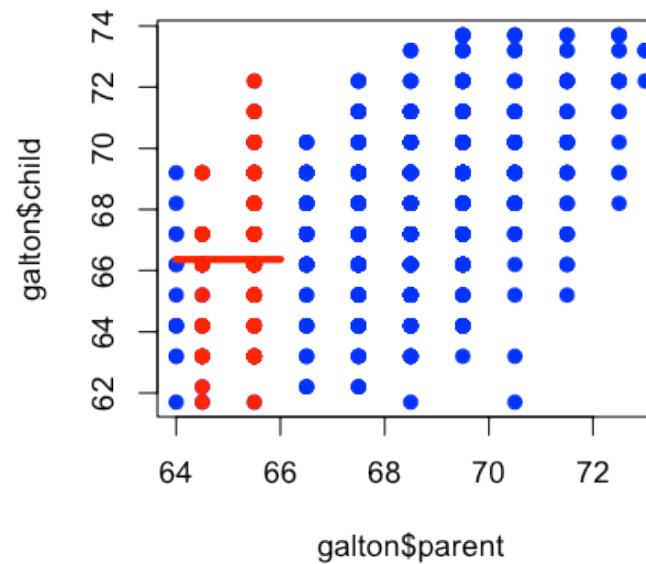
# Jittered plot

```
set.seed(1234)
plot(jitter(galton$parent,factor=2),jitter(galton$child,factor=2),pch=19,col="blue")
```



# Average parent = 65 inches tall

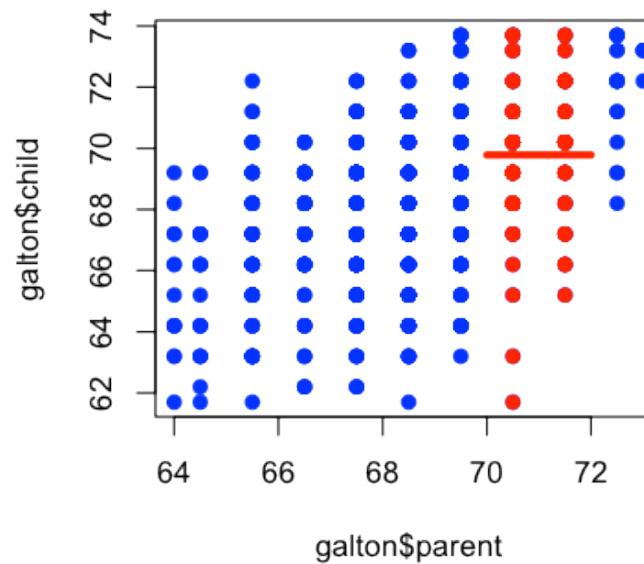
```
plot(galton$parent, galton$child, pch=19, col="blue")
near65 <- galton[abs(galton$parent - 65)<1, ]
points(near65$parent, near65$child, pch=19, col="red")
lines(seq(64,66, length=100), rep(mean(near65$child), 100), col="red", lwd=4)
```



11/19

# Average parent = 71 inches tall

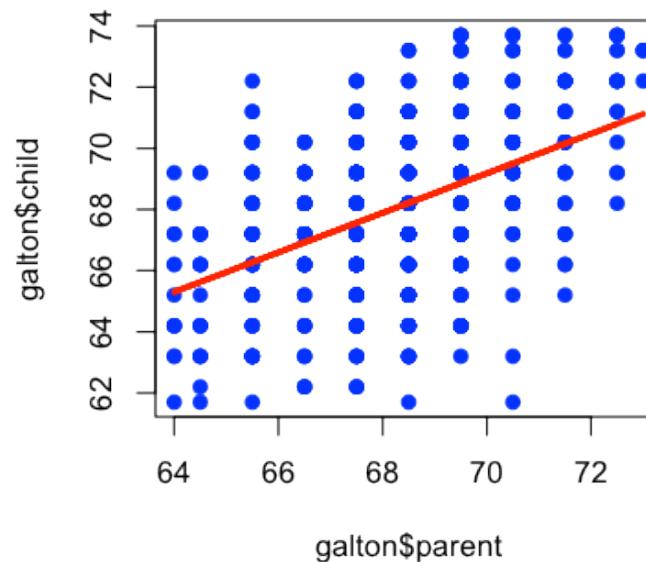
```
plot(galton$parent, galton$child, pch=19, col="blue")
near71 <- galton[abs(galton$parent - 71)<1, ]
points(near71$parent, near71$child, pch=19, col="red")
lines(seq(70,72, length=100), rep(mean(near71$child), 100), col="red", lwd=4)
```



12/19

# Fitting a line

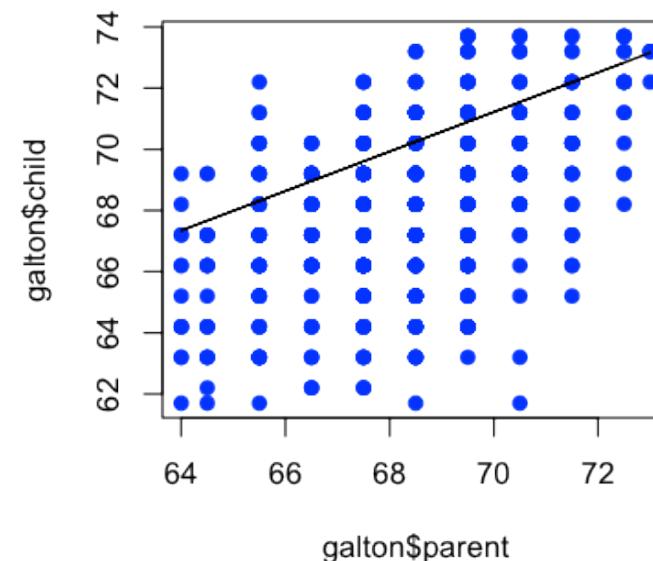
```
plot(galton$parent, galton$child, pch=19, col="blue")
lm1 <- lm(galton$child ~ galton$parent)
lines(galton$parent, lm1$fitted, col="red", lwd=3)
```



13/19

# Why not this line?

```
plot(galton$parent, galton$child, pch=19, col="blue")
lines(galton$parent, 26 + 0.646*galton$parent)
```



14/19

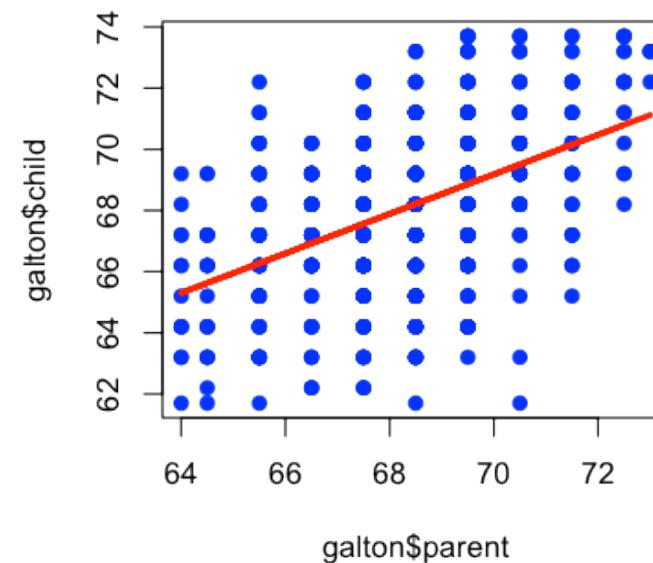
# The equation for a line

If  $C_i$  is the height of child  $i$  and  $P_i$  is the height of the average parent, then we can imagine writing the equation for a line

$$C_i = b_0 + b_1 P_i$$

# Not all points are on the line

```
plot(galton$parent, galton$child, pch=19, col="blue")
lines(galton$parent, lm1$fitted, col="red", lwd=3)
```



16/19

# Allowing for variation

If  $C_i$  is the height of child  $i$  and  $P_i$  is the height of the average parent, then we can imagine writing the equation for a line

$$C_i = b_0 + b_1 P_i + e_i$$

$e_i$  is everything we didn't measure (how much they eat, where they live, do they stretch in the morning...)

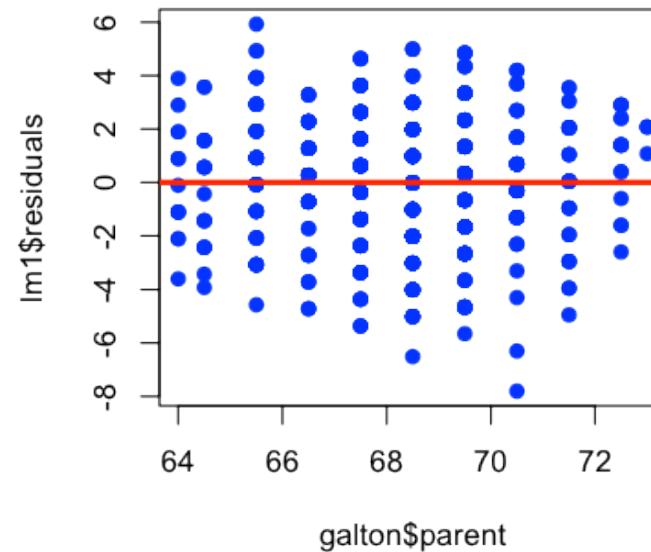
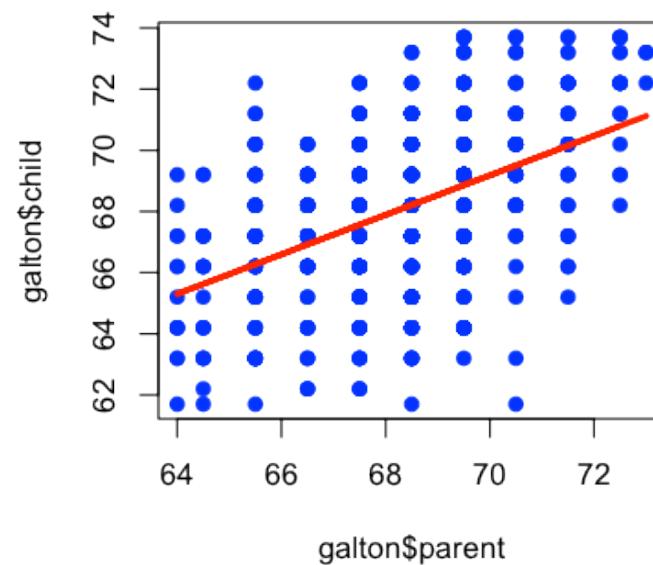
# How do we pick best?

If  $C_i$  is the height of child  $i$  and  $P_i$  is the height of the average parent, pick the line that makes the child values  $C_i$  and our guesses

$$\sum_{i=1}^{928} (C_i - (b_0 + b_1 P_i))^2$$

# Plot what is leftover

```
par(mfrow=c(1,2))
plot(galton$parent,galton$child,pch=19,col="blue")
lines(galton$parent,lm1$fitted,col="red",lwd=3)
plot(galton$parent,lm1$residuals,col="blue",pch=19)
abline(c(0,0),col="red",lwd=3)
```



19/19

# Binary outcomes

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

# Key ideas

- Frequently we care about outcomes that have two values
  - Alive/dead
  - Win/loss
  - Success/Failure
  - etc
- Called binary outcomes or 0/1 outcomes
- Linear regression (like we've seen) may not be the best

# Example: Baltimore Ravens

**Baltimore Ravens** Sign in to personalize

AFC North 

[Clubhouse](#) [Stats](#) [Schedule](#) [Roster](#) [Splits](#) [Depth Chart](#) [Transactions](#) [Rankings](#) [Photos](#) [Stadium](#) [News](#) [Forum](#) [Tickets](#) [Shop](#)

| Sun  | Feb 3               | Sun  | Feb 3                            | 2012 Season  |
|--|---------------------|--|----------------------------------|--|
| @  W<br>34-31 | San Francisco 49ers | Final @  Baltimore (10-6) | Superbowl San Francisco (11-4-1) | Record:<br>Overall: 10-6<br>vs AFC North: 4-2<br>vs AFC: 8-4                         |
| Pass: Kaepernick 302 yds<br>Rush: Gore 110 yds<br>Rec: Crabtree 109 yds                        |                     | 1 2 3 4 T<br>BAL 7 14 7 6 34<br>SF 3 3 17 8 31   |                                  | Team leaders:<br>Pass: Flacco 3817 yds<br>Rush: Rice 1143 yds<br>Rec: Boldin 921 yds |
| <a href="#">Recap »</a>  |                     | <a href="#">Recap »</a>  | <a href="#">Box Score »</a>      |  |

**2012 OVERALL NFL RANKINGS**

| PASSING YDS | RUSHING YDS | OPP PASSING YDS | OPP RUSHING YDS |
|-------------|-------------|-----------------|-----------------|
| 100.0       | 100.0       | 100.0           | 100.0           |

**BALTIMORE TEAMS**




[http://espn.go.com/nfl/team/\\_/name/bal/baltimore-ravens](http://espn.go.com/nfl/team/_/name/bal/baltimore-ravens)

3/22

# Ravens Data

```
download.file("https://dl.dropbox.com/u/7710864/data/ravensData.rda",
              destfile=".~/data/ravensData.rda",method="curl")
load("./data/ravensData.rda")
head(ravensData)
```

|   | ravenWinNum | ravenWin | ravenScore | opponentScore |
|---|-------------|----------|------------|---------------|
| 1 | 1           | W        | 24         | 9             |
| 2 | 1           | W        | 38         | 35            |
| 3 | 1           | W        | 28         | 13            |
| 4 | 1           | W        | 34         | 31            |
| 5 | 1           | W        | 44         | 13            |
| 6 | 0           | L        | 23         | 24            |

# Linear regression

$$RW_i = b_0 + b_1 RS_i + e_i$$

$RW_i$  - 1 if a Ravens win, 0 if not

$RS_i$  - Number of points Ravens scored

$b_0$  - probability of a Ravens win if they score 0 points

$b_1$  - increase in probability of a Ravens win for each additional point

$e_i$  - variation due to everything we didn't measure

# Linear regression in R

```
lmRavens <- lm(ravensData$ravenWinNum ~ ravensData$ravenScore)
summary(lmRavens)
```

Call:

```
lm(formula = ravensData$ravenWinNum ~ ravensData$ravenScore)
```

Residuals:

| Min    | 1Q     | Median | 3Q    | Max   |
|--------|--------|--------|-------|-------|
| -0.730 | -0.508 | 0.182  | 0.322 | 0.572 |

Coefficients:

|                        | Estimate | Std. Error | t value | Pr(> t ) |
|------------------------|----------|------------|---------|----------|
| (Intercept)            | 0.28503  | 0.25664    | 1.11    | 0.281    |
| ravensData\$ravenScore | 0.01590  | 0.00906    | 1.76    | 0.096 .  |

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

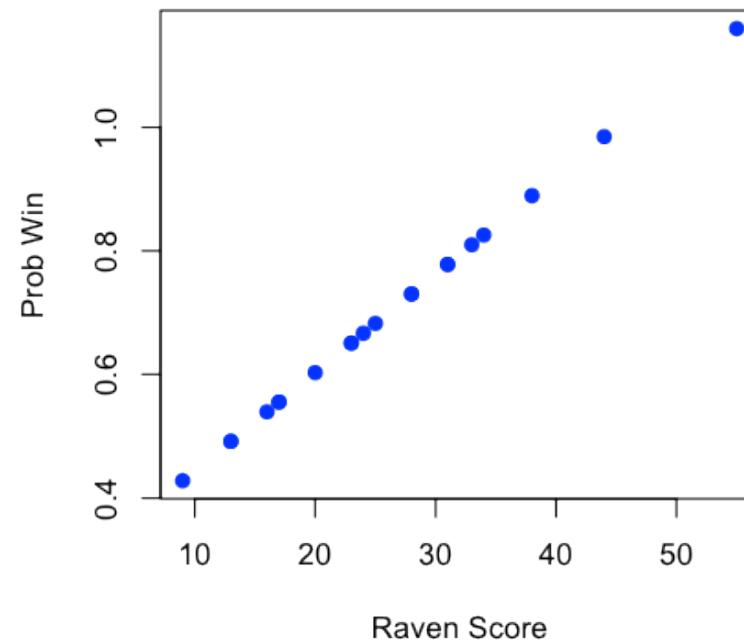
Residual standard error: 0.446 on 18 degrees of freedom

Multiple R-squared: 0.146, Adjusted R-squared: 0.0987

6/22

# Linear regression

```
plot(ravensData$ravenScore, lmRavens$fitted, pch=19, col="blue", ylab="Prob Win", xlab="Raven Score")
```



7/22

# Odds

**Binary Outcome 0/1**

$$RW_i$$

**Probability (0,1)**

$$\Pr(RW_i|RS_i, b_0, b_1)$$

**Odds**  $(0, \infty)$

$$\frac{\Pr(RW_i|RS_i, b_0, b_1)}{1 - \Pr(RW_i|RS_i, b_0, b_1)}$$

**Log odds**  $(-\infty, \infty)$

$$\log\left(\frac{\Pr(RW_i|RS_i, b_0, b_1)}{1 - \Pr(RW_i|RS_i, b_0, b_1)}\right)$$

# Linear vs. logistic regression

## Linear

$$RW_i = b_0 + b_1 RS_i + e_i$$

or

$$E[RW_i | RS_i, b_0, b_1] = b_0 + b_1 RS_i$$

## Logistic

$$\Pr(RW_i | RS_i, b_0, b_1) = \frac{\exp(b_0 + b_1 RS_i)}{1 + \exp(b_0 + b_1 RS_i)}$$

or

$$\log\left(\frac{\Pr(RW_i | RS_i, b_0, b_1)}{1 - \Pr(RW_i | RS_i, b_0, b_1)}\right) = b_0 + b_1 RS_i$$

# Interpreting Logistic Regression

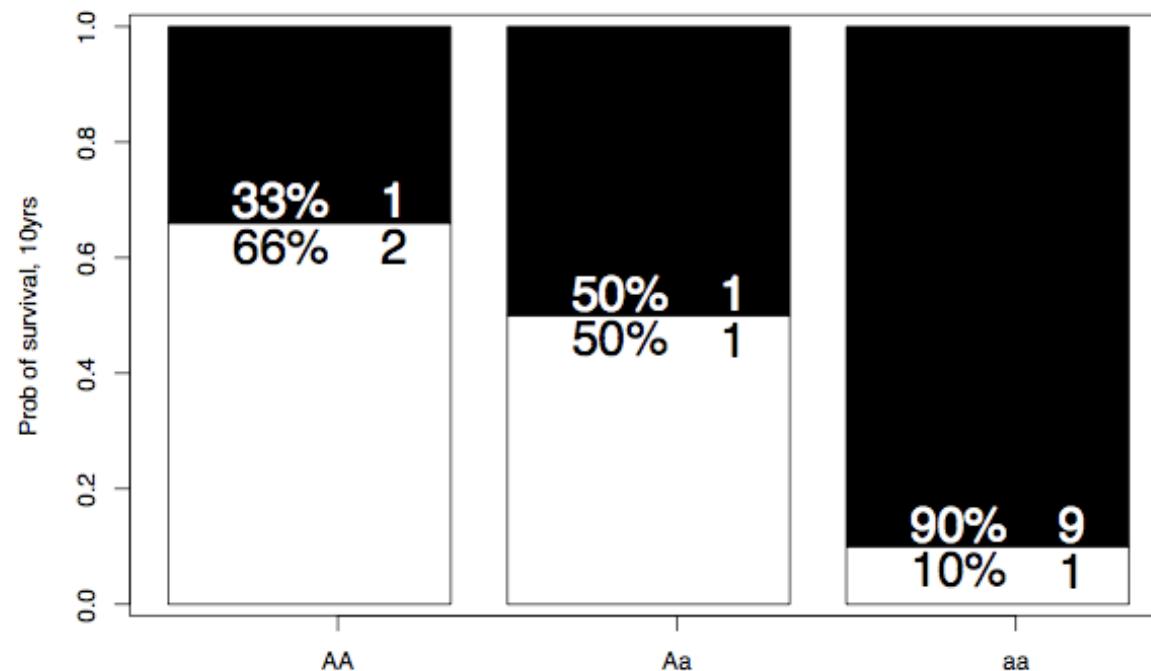
$$\log\left(\frac{\Pr(\text{RW}_i|\text{RS}_i, b_0, b_1)}{1 - \Pr(\text{RW}_i|\text{RS}_i, b_0, b_1)}\right) = b_0 + b_1 \text{RS}_i$$

$b_0$  - Log odds of a Ravens win if they score zero points

$b_1$  - Log odds ratio of win probability for each point scored (compared to zero points)

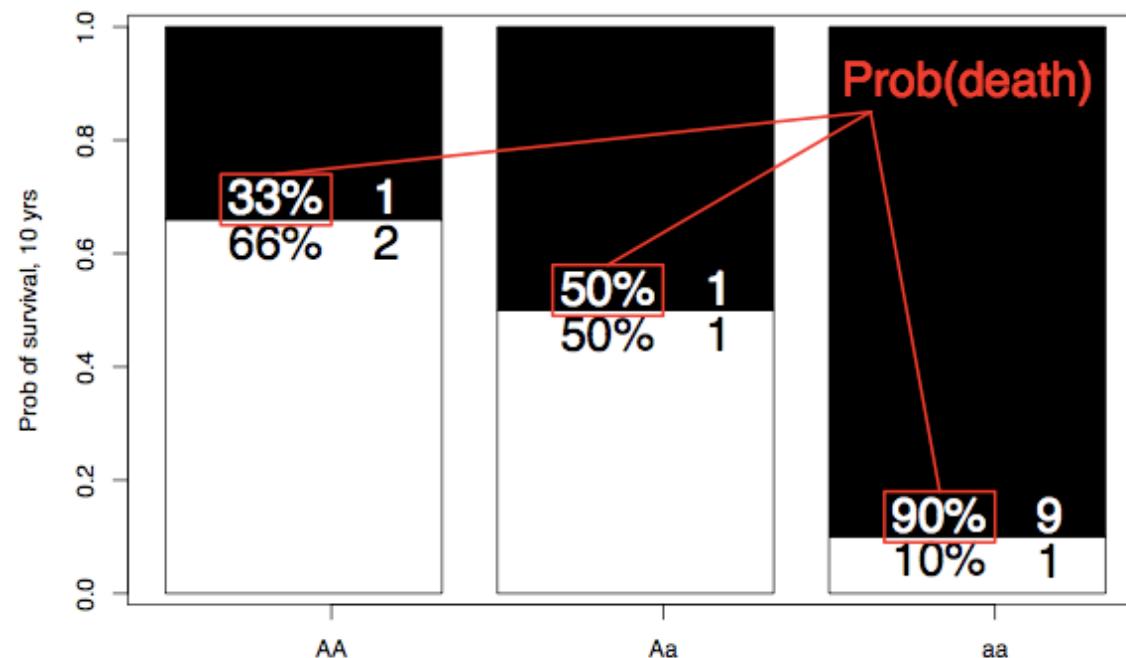
$\exp(b_1)$  - Odds ratio of win probability for each point scored (compared to zero points)

# Explaining Odds



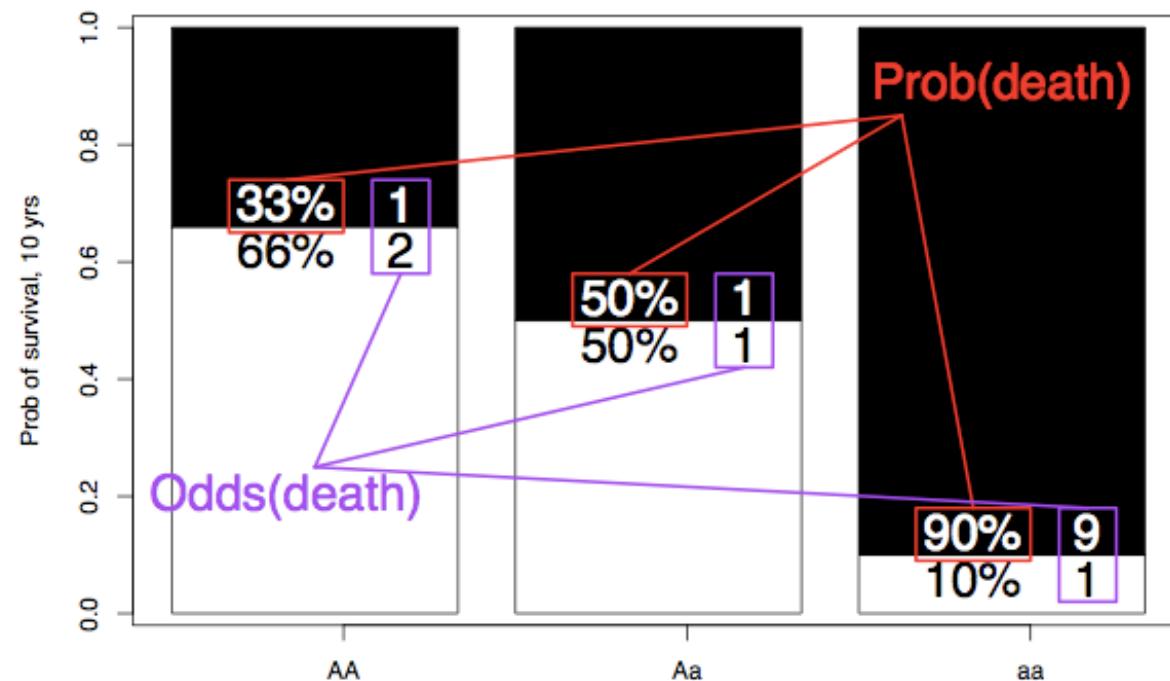
via Ken Rice

# Probability of Death



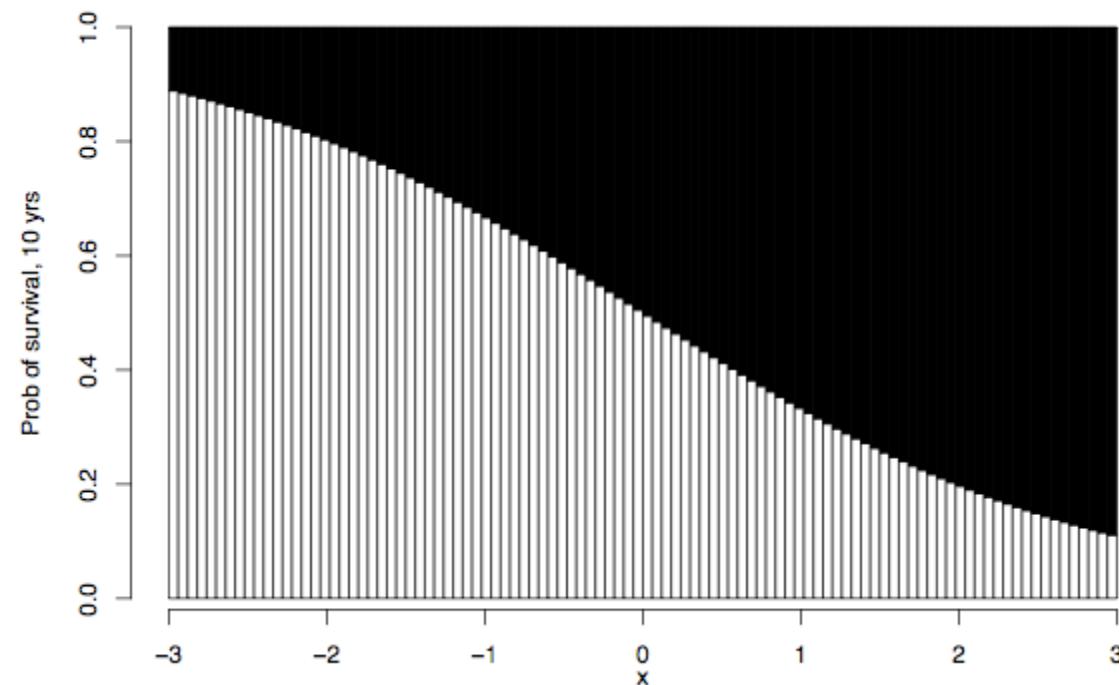
via Ken Rice

# Odds of Death



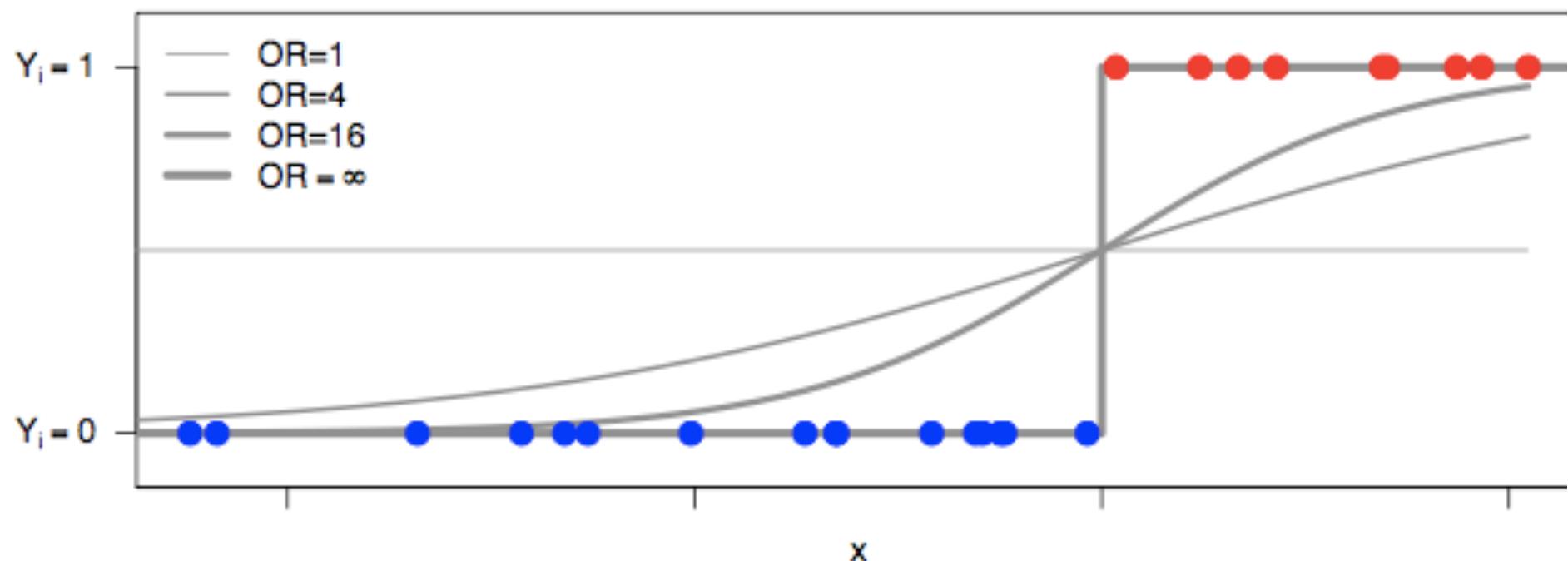
[via Ken Rice](#)

# Odds Ratio = 1, Continuous Covariate



via Ken Rice

# Different odds ratios



via Ken Rice

# Ravens logistic regression

```
logRegRavens <- glm(ravensData$ravenWinNum ~ ravensData$ravenScore, family="binomial")
summary(logRegRavens)
```

Call:

```
glm(formula = ravensData$ravenWinNum ~ ravensData$ravenScore,
family = "binomial")
```

Deviance Residuals:

| Min    | 1Q     | Median | 3Q    | Max   |
|--------|--------|--------|-------|-------|
| -1.758 | -1.100 | 0.530  | 0.806 | 1.495 |

Coefficients:

|                        | Estimate | Std. Error | z value | Pr(> z ) |
|------------------------|----------|------------|---------|----------|
| (Intercept)            | -1.6800  | 1.5541     | -1.08   | 0.28     |
| ravensData\$ravenScore | 0.1066   | 0.0667     | 1.60    | 0.11     |

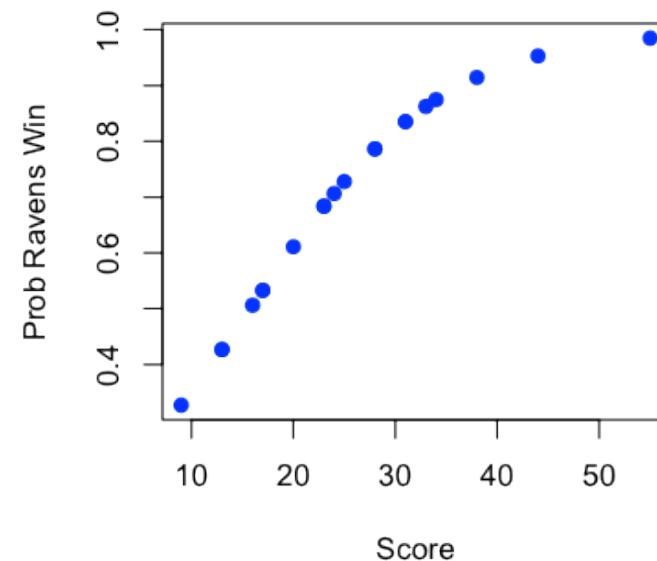
(Dispersion parameter for binomial family taken to be 1)

Null deviance: 24.435 on 19 degrees of freedom

16/22

# Ravens fitted values

```
plot(ravensData$ravenScore, logRegRavens$fitted, pch=19, col="blue", xlab="Score", ylab="Prob Ravens Win")
```



# Odds ratios and confidence intervals

```
exp(logRegRavens$coeff)
```

```
(Intercept) ravensData$ravenScore  
0.1864      1.1125
```

```
exp(confint(logRegRavens))
```

```
2.5 % 97.5 %  
(Intercept) 0.005675 3.106  
ravensData$ravenScore 0.996230 1.303
```

# ANOVA for logistic regression

```
anova(logRegRavens,test="Chisq")
```

Analysis of Deviance Table

Model: binomial, link: logit

Response: ravensData\$ravenWinNum

Terms added sequentially (first to last)

|   | Df | Deviance | Resid. Df | Resid. Dev | Pr(>Chi) |
|---|----|----------|-----------|------------|----------|
| NULL  |    |          | 19        | 24.4       |          |
| ravensData\$ravenScore  | 1  | 3.54     | 18        | 20.9       | 0.06 .   |
| <hr/>   |    |          |           |            |          |
| Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 |    |          |           |            |          |

# Simpson's paradox

|              | Treatment A                     | Treatment B                     |
|--------------|---------------------------------|---------------------------------|
| Small Stones | <i>Group 1</i><br>93% (81/87)   | <i>Group 2</i><br>87% (234/270) |
| Large Stones | <i>Group 3</i><br>73% (192/263) | <i>Group 4</i><br>69% (55/80)   |
| Both         | 78% (273/350)                   | 83% (289/350)                   |

[http://en.wikipedia.org/wiki/Simpson's\\_paradox](http://en.wikipedia.org/wiki/Simpson's_paradox)

# Interpreting Odds Ratios

- Not probabilities
- Odds ratio of 1 = no difference in odds
- Log odds ratio of 0 = no difference in odds
- Odds ratio < 0.5 or > 2 commonly a "moderate effect"
- Relative risk  $\frac{\Pr(\text{RW}_i | \text{RS}_i=1)}{\Pr(\text{RW}_i | \text{RS}_i=0)}$  often easier to interpret, harder to estimate
- For small probabilities RR  $\approx$  OR but **they are not the same!**

[Wikipedia on Odds Ratio](#)

# Further resources

- [Wikipedia on Logistic Regression](#)
- [Logistic regression and glms in R](#)
- Brian Caffo's lecture notes on: [Simpson's paradox](#), [Case-control studies](#)
- [Open Intro Chapter on Logistic Regression](#)

22/22

# The bootstrap

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

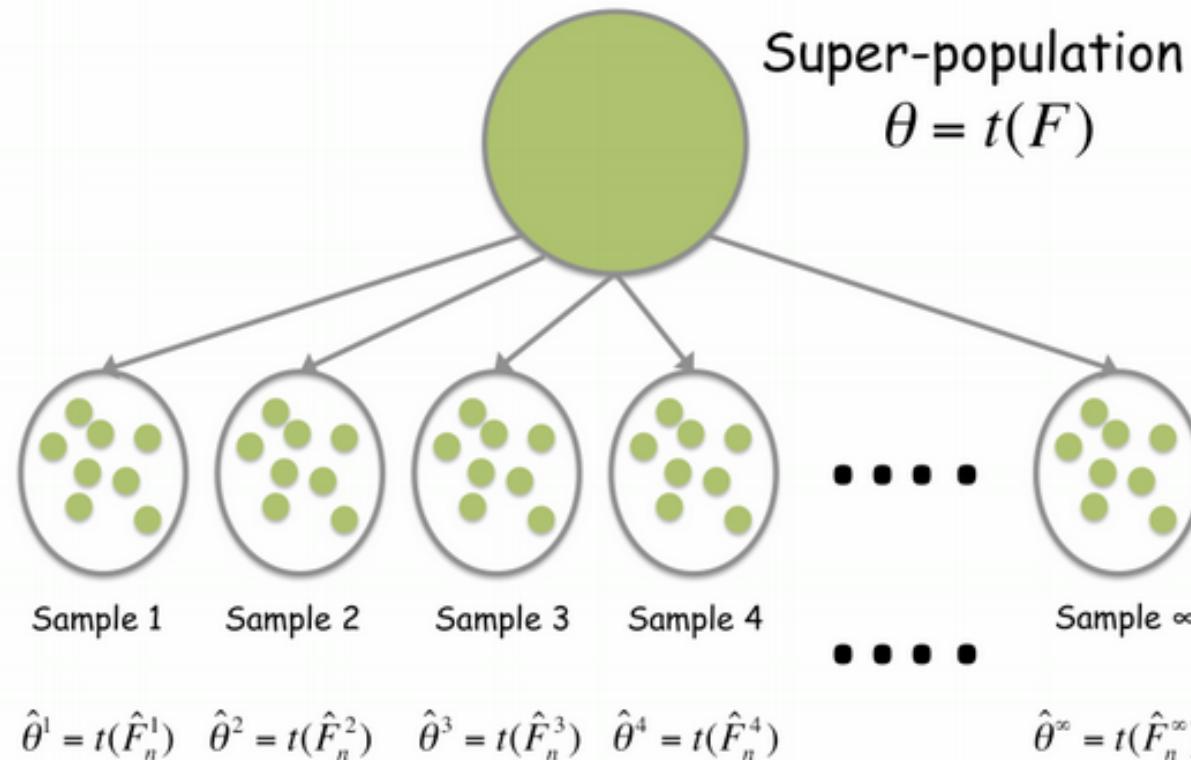
# Key ideas

- Treat the sample as if it were the population

## What it is good for:

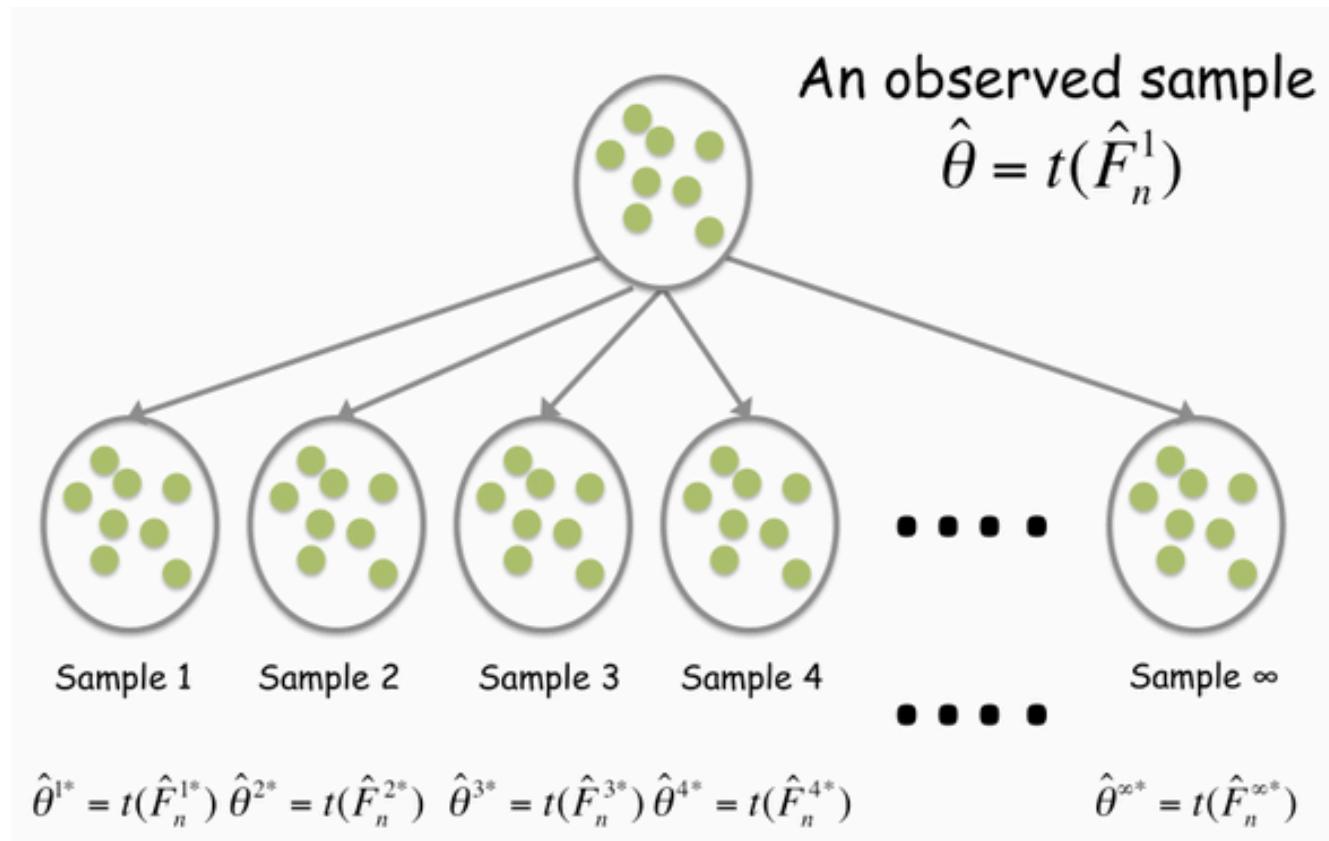
- Calculating standard errors
- Forming confidence intervals
- Performing hypothesis tests
- Improving predictors

# The "Central Dogma" of statistics



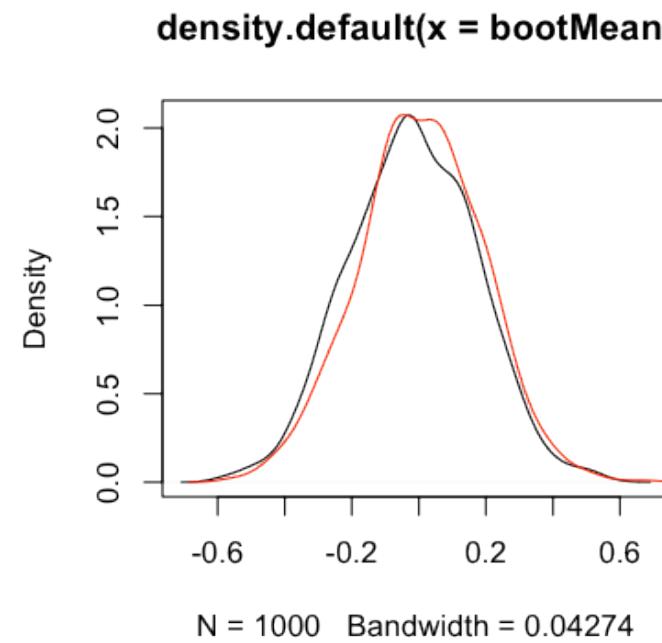
<http://www.gs.washington.edu/academics/courses/akey/56008/lecture/lecture5.pdf>

# The bootstrap



# Example

```
set.seed(333); x <- rnorm(30)
bootMean <- rep(NA,1000); sampledMean <- rep(NA,1000)
for(i in 1:1000){bootMean[i] <- mean(sample(x,replace=TRUE))} 
for(i in 1:1000){sampledMean[i] <- mean(rnorm(30))} 
plot(density(bootMean)); lines(density(sampledMean),col="red")
```



# Example with boot package

```
set.seed(333); x <- rnorm(30); sampledMean <- rep(NA,1000)
for(i in 1:1000){sampledMean[i] <- mean(rnorm(30))}

meanFunc <- function(x,i){mean(x[i])}
bootMean <- boot(x,meanFunc,1000)
bootMean
```

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

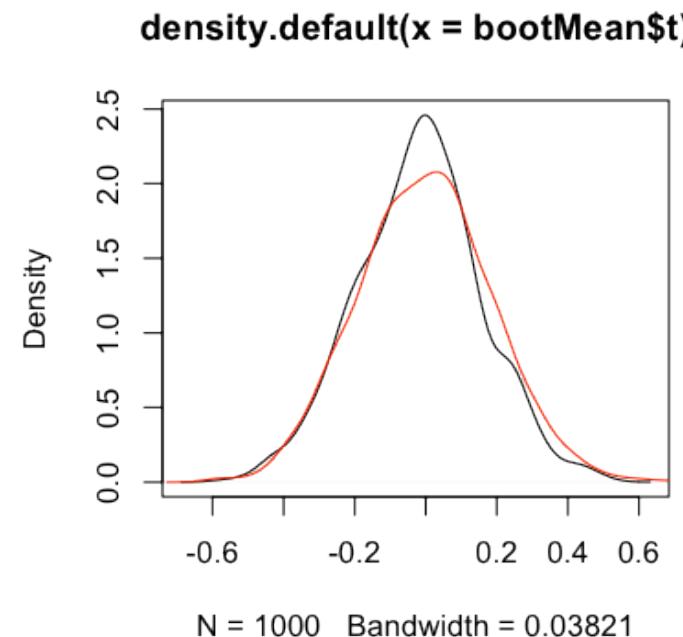
```
boot(data = x, statistic = meanFunc, R = 1000)
```

Bootstrap Statistics :

|     | original | bias      | std. error |
|-----|----------|-----------|------------|
| t1* | -0.01942 | 0.0006377 | 0.175      |

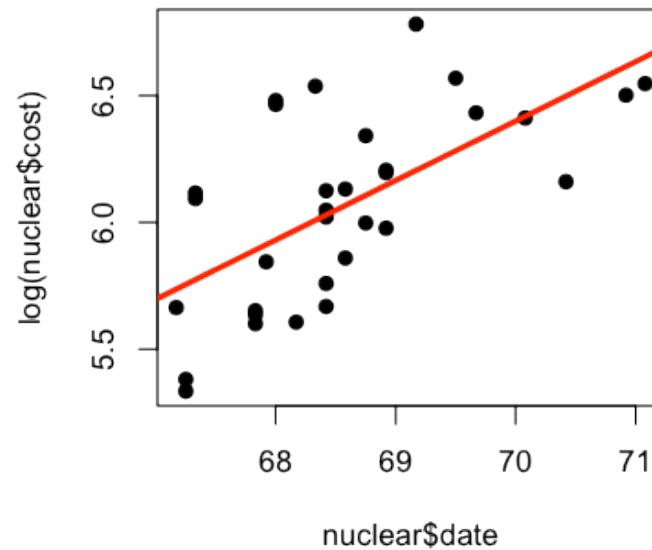
# Plotting boot package example

```
plot(density(bootMean$t)); lines(density(sampledMean), col="red")
```



# Nuclear costs

```
library(boot); data(nuclear)
nuke.lm <- lm(log(cost) ~ date,data=nuclear)
plot(nuclear$date,log(nuclear$cost),pch=19)
abline(nuke.lm,col="red",lwd=3)
```



# Nuclear costs

```
par(mfrow=c(1,3))
for(i in 1:3){
  nuclear0 <- nuclear[sample(1:dim(nuclear)[1],replace=TRUE),]
  nuke.lm0 <- lm(log(cost) ~ date,data=nuclear0)
  plot(nuclear0$date,log(nuclear0$cost),pch=19)
  abline(nuke.lm0,col="red",lwd=3)
}
```

9/17

# Bootstrap distribution

```
bs <- function(data, indices, formula) {  
  d <- data[indices,];fit <- lm(formula, data=d);return(coef(fit))  
}  
results <- boot(data=nuclear, statistic=bs, R=1000, formula=log(cost) ~ date)  
plot(density(results$t[,2]), col="red", lwd=3)  
lines(rep(nuke.lm$coeff[2], 10), seq(0, 8, length=10), col="blue", lwd=3)
```

10/17

# Bootstrap confidence intervals

```
boot.ci(results)
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 1000 bootstrap replicates

CALL :

```
boot.ci(boot.out = results)
```

Intervals :

| Level | Normal | Basic | Studentized |
|-------|--------|-------|-------------|
|-------|--------|-------|-------------|

|     |                    |                    |                    |
|-----|--------------------|--------------------|--------------------|
| 95% | (-16.481, -3.130 ) | (-15.746, -2.553 ) | (-17.153, -3.842 ) |
|-----|--------------------|--------------------|--------------------|

| Level | Percentile | BCa |
|-------|------------|-----|
|-------|------------|-----|

|     |                    |                    |
|-----|--------------------|--------------------|
| 95% | (-17.435, -4.242 ) | (-17.475, -4.249 ) |
|-----|--------------------|--------------------|

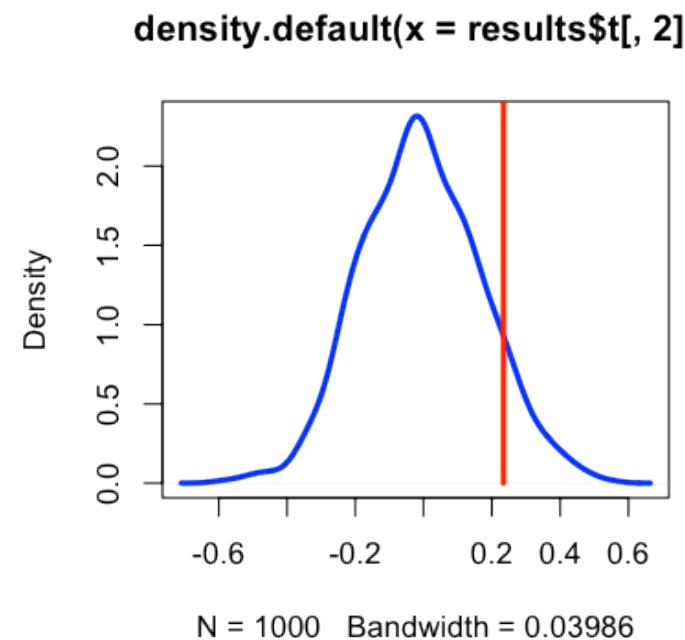
Calculations and Intervals on Original Scale

# Bootstrapping from a model

```
resid <- rstudent(nuke.lm)
fit0 <- fitted(lm(log(cost) ~ 1,data=nuclear))
newNuc <- cbind(nuclear,resid=resid,fit0=fit0)
bs <- function(data, indices) {
  return(coef(glm(data$fit0 + data$resid[indices] ~ data$date,data=data)))
}
results <- boot(data=newNuc, statistic=bs, R=1000)
```

# Results

```
plot(density(results$t[, 2]), lwd=3, col="blue")
lines(rep(coef(nuke.lm)[2], 10), seq(0, 3, length=10), col="red", lwd=3)
```



# An empirical p-value

$$\hat{p} = \frac{1 + \sum_{b=1}^B |t_b^0| > |t|}{B + 1}$$

```
B <- dim(results$t)[1]  
(1 + sum((abs(results$t[,2]) > abs(coef(nuke.lm)[2]))))/(B+1)
```

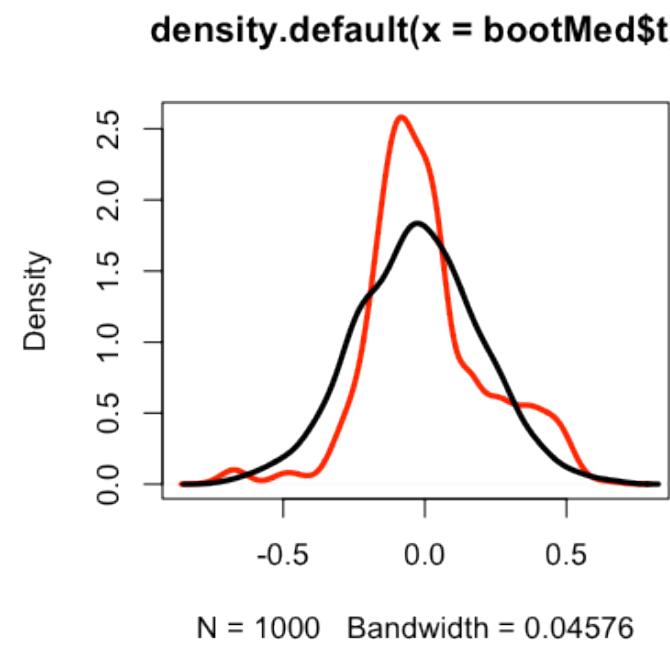
```
[1] 0.1838
```

14/17

# Bootstrapping non-linear statistics

```
set.seed(555); x <- rnorm(30); sampledMed <- rep(NA,1000)
for(i in 1:1000){sampledMed[i] <- median(rnorm(30))}

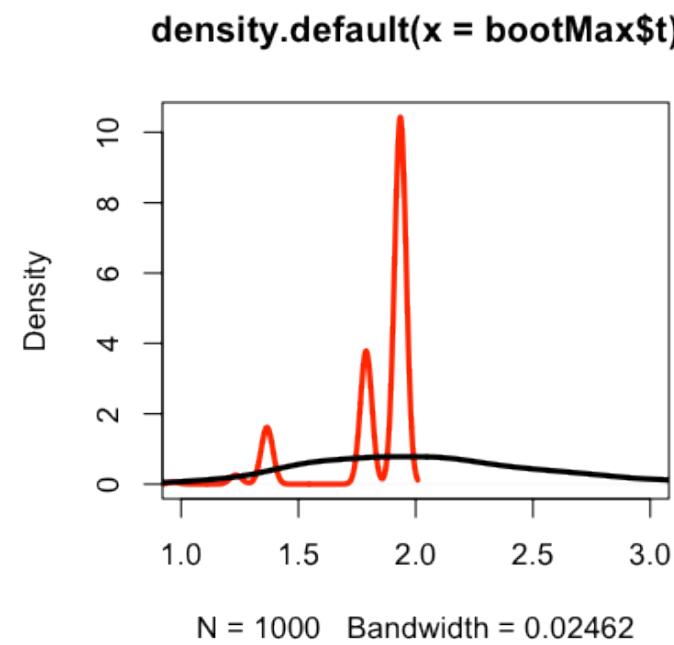
medFunc <- function(x,i){median(x[i])}; bootMed <- boot(x,medFunc,1000)
plot(density(bootMed$t),col="red",lwd=3)
lines(density(sampledMed),lwd=3)
```



# Things you can't bootstrap (max)

```
set.seed(333); x <- rnorm(30); sampledMax <- rep(NA,1000)
for(i in 1:1000){sampledMax[i] <- max(rnorm(30))}

maxFunc <- function(x,i){max(x[i])}; bootMax <- boot(x,maxFunc,1000)
plot(density(bootMax$t),col="red",lwd=3,xlim=c(1,3))
lines(density(sampledMax),lwd=3)
```



16/17

# Notes and further resources

## Notes:

- Can be useful for complicated statistics
- Be careful near the boundaries
- Be careful with non-linear functions

## Further resources:

- [Brian Caffo's bootstrap notes](#)
- [Nice basic intro to boot package](#)
- [Another basic boot tutorial](#)
- [An introduction to the bootstrap](#)
- [Confidence limits on phylogenies](#)

# Bootstrapping for prediction

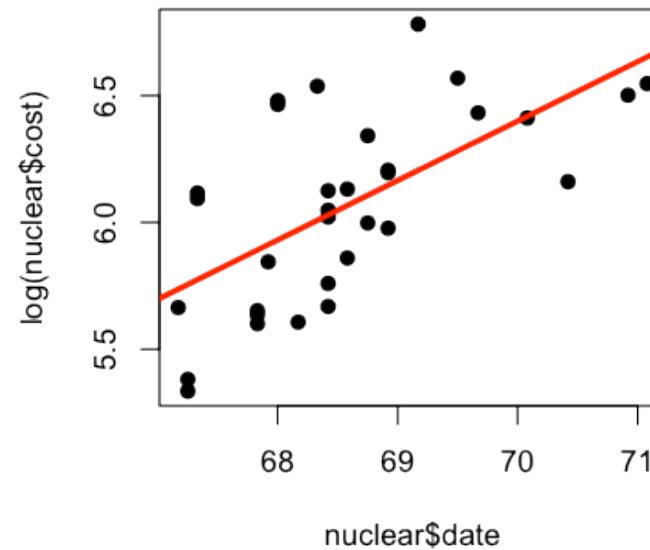
Jeffrey Leek, Assistant Professor of Biostatistics  
Johns Hopkins Bloomberg School of Public Health

# Key ideas

- Bootstrapping can be used for
  - Cross-validation type error rates
  - Prediction errors in regression models
  - Improving prediction

# Bootstrapping prediction errors

```
library(boot); data(nuclear)
nuke.lm <- lm(log(cost) ~ date,data=nuclear)
plot(nuclear$date,log(nuclear$cost),pch=19)
abline(nuke.lm,col="red",lwd=3)
```



# Bootstrapping prediction errors

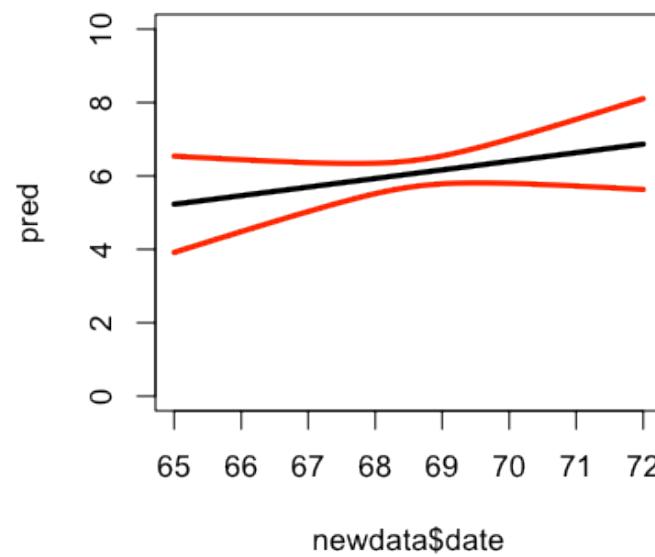
```
newdata <- data.frame(date = seq(65,72,length=100))
nuclear <- cbind(nuclear,resid=rstudent(nuke.lm),fit=fitted(nuke.lm))
nuke.fun <- function(data,inds,newdata){
  lm.b <- lm(fit + resid[inds] ~ date,data=data)
  pred.b <- predict(lm.b,newdata)
  return(pred.b)
}
nuke.boot <- boot(nuclear,nuke.fun,R=1000,newdata=newdata)
head(nuke.boot$t)
```

```
[,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15]
[1,] 4.565 4.597 4.629 4.661 4.693 4.725 4.757 4.789 4.821 4.853 4.885 4.917 4.950 4.982 5.014
[2,] 6.453 6.450 6.447 6.444 6.441 6.438 6.435 6.432 6.429 6.426 6.423 6.420 6.417 6.414 6.411
[3,] 5.168 5.183 5.198 5.213 5.228 5.243 5.258 5.273 5.288 5.303 5.318 5.333 5.348 5.363 5.378
[4,] 5.401 5.413 5.425 5.437 5.449 5.461 5.473 5.485 5.497 5.509 5.521 5.533 5.545 5.557 5.569
[5,] 4.013 4.047 4.081 4.115 4.149 4.183 4.217 4.251 4.285 4.319 4.353 4.387 4.421 4.454 4.488
[6,] 6.263 6.261 6.259 6.258 6.256 6.254 6.252 6.250 6.248 6.246 6.245 6.243 6.241 6.239 6.237
[,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29] [,30]
[1,] 5.046 5.078 5.110 5.142 5.174 5.206 5.238 5.270 5.303 5.335 5.367 5.399 5.431 5.463 5.495
[2,] 6.408 6.405 6.402 6.399 6.396 6.393 6.390 6.387 6.384 6.381 6.377 6.374 6.371 6.368 6.365
```

4/22

# Bootstrapping prediction errors

```
pred <- predict(nuke.lm,newdata)
predSds <- apply(nuke.boot$t,2,SD)
plot(newdata$date,pred,col="black",type="l",lwd=3,ylim=c(0,10))
lines(newdata$date,pred + 1.96*predSds,col="red",lwd=3)
lines(newdata$date,pred - 1.96*predSds,col="red",lwd=3)
```



# Bootstrap aggregating (bagging)

## Basic idea:

1. Resample cases and recalculate predictions
2. Average or majority vote

## Notes:

- Similar bias
- Reduced variance
- More useful for non-linear functions

# Bagged loess

```
library(ElemStatLearn); data(ozone, package="ElemStatLearn")
ozone <- ozone[order(ozone$ozone), ]
head(ozone)
```

|     | ozone | radiation | temperature | wind |
|-----|-------|-----------|-------------|------|
| 17  | 1     | 8         | 59          | 9.7  |
| 19  | 4     | 25        | 61          | 9.7  |
| 14  | 6     | 78        | 57          | 18.4 |
| 45  | 7     | 48        | 80          | 14.3 |
| 106 | 7     | 49        | 69          | 10.3 |
| 7   | 8     | 19        | 61          | 20.1 |

[http://en.wikipedia.org/wiki/Bootstrap\\_aggregating](http://en.wikipedia.org/wiki/Bootstrap_aggregating)

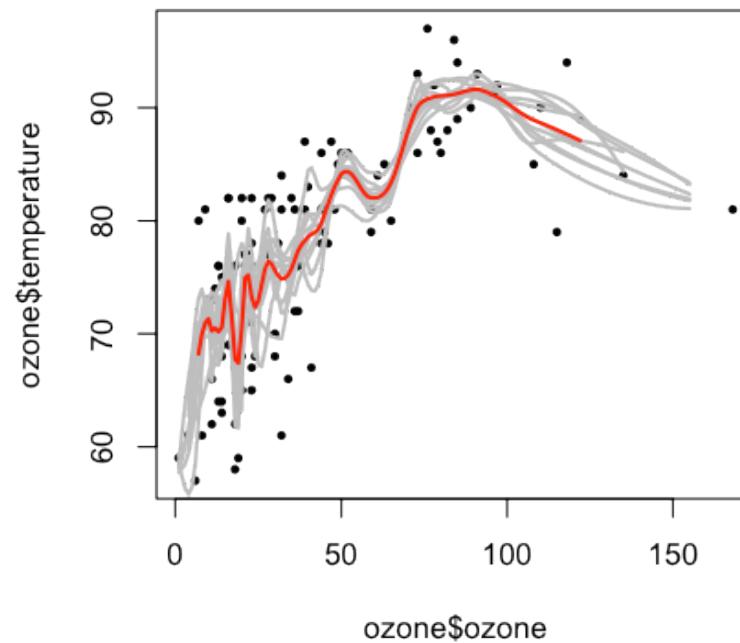
7/22

# Bagged loess

```
ll <- matrix(NA,nrow=10,ncol=155)
for(i in 1:10){
  ss <- sample(1:dim(ozone)[1],replace=T)
  ozone0 <- ozone[ss,]; ozone0 <- ozone0[order(ozone0$ozone),]
  loess0 <- loess(temperature ~ ozone,data=ozone0,span=0.2)
  ll[i,] <- predict(loess0,newdata=data.frame(ozone=1:155))
}
```

# Bagged loess

```
plot(ozone$ozone,ozone$temperature,pch=19,cex=0.5)
for(i in 1:10){lines(1:155,ll[i,],col="grey",lwd=2)}
lines(1:155,apply(ll,2,mean),col="red",lwd=2)
```



9/22

# Bagged trees

**Basic idea:**

1. Resample data
2. Recalculate tree
3. Average/mode) of predictors

**Notes:**

1. More stable
2. May not be as good as random forests

10/22

# Iris data

```
data(iris)  
head(iris)
```

|   | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|---|--------------|-------------|--------------|-------------|---------|
| 1 | 5.1          | 3.5         | 1.4          | 0.2         | setosa  |
| 2 | 4.9          | 3.0         | 1.4          | 0.2         | setosa  |
| 3 | 4.7          | 3.2         | 1.3          | 0.2         | setosa  |
| 4 | 4.6          | 3.1         | 1.5          | 0.2         | setosa  |
| 5 | 5.0          | 3.6         | 1.4          | 0.2         | setosa  |
| 6 | 5.4          | 3.9         | 1.7          | 0.4         | setosa  |

# Bagging a tree

```
library(ipred)
bagTree <- bagging(Species ~ ., data=iris, coob=TRUE)
print(bagTree)
```

Bagging classification trees with 25 bootstrap replications

Call: bagging.data.frame(formula = Species ~ ., data = iris, coob = TRUE)

Out-of-bag estimate of misclassification error: 0.06

# Looking at bagged tree one

```
bagTree$mtrees[[1]]$btree
```

```
n= 150
```

```
node), split, n, loss, yval, (yprob)
 * denotes terminal node
```

```
1) root 150 98 virginica (0.33333 0.32000 0.34667)
  2) Petal.Length< 2.5 50  0 setosa (1.00000 0.00000 0.00000) *
  3) Petal.Length>=2.5 100 48 virginica (0.00000 0.48000 0.52000)
    6) Petal.Width< 1.75 52  5 versicolor (0.00000 0.90385 0.09615)
      12) Petal.Length< 4.9 46  2 versicolor (0.00000 0.95652 0.04348)
        24) Petal.Width< 1.65 44  0 versicolor (0.00000 1.00000 0.00000) *
        25) Petal.Width>=1.65 2  0 virginica (0.00000 0.00000 1.00000) *
    13) Petal.Length>=4.9 6  3 versicolor (0.00000 0.50000 0.50000)
      26) Sepal.Width>=2.65 3  0 versicolor (0.00000 1.00000 0.00000) *
      27) Sepal.Width< 2.65 3  0 virginica (0.00000 0.00000 1.00000) *
  7) Petal.Width>=1.75 48  1 virginica (0.00000 0.02083 0.97917)
  14) Petal.Length< 4.85 3  1 virginica (0.00000 0.33333 0.66667)
    28) Sepal.Length< 5.95 1  0 versicolor (0.00000 1.00000 0.00000) *
```

13/22

# Looking at bagged tree two

```
bagTree$mtrees[[2]]$btree
```

```
n= 150
```

```
node), split, n, loss, yval, (yprob)
 * denotes terminal node
```

```
1) root 150 98 versicolor (0.33333 0.34667 0.32000)
  2) Petal.Length< 2.6 50  0 setosa (1.00000 0.00000 0.00000) *
  3) Petal.Length>=2.6 100 48 versicolor (0.00000 0.52000 0.48000)
     6) Petal.Length< 4.85 51  3 versicolor (0.00000 0.94118 0.05882)
        12) Petal.Width< 1.65 45  0 versicolor (0.00000 1.00000 0.00000) *
        13) Petal.Width>=1.65 6  3 versicolor (0.00000 0.50000 0.50000)
           26) Sepal.Width>=3.1 3  0 versicolor (0.00000 1.00000 0.00000) *
           27) Sepal.Width< 3.1 3  0 virginica (0.00000 0.00000 1.00000) *
  7) Petal.Length>=4.85 49  4 virginica (0.00000 0.08163 0.91837)
  14) Petal.Width< 1.75 8  4 versicolor (0.00000 0.50000 0.50000)
     28) Petal.Length< 4.95 2  0 versicolor (0.00000 1.00000 0.00000) *
     29) Petal.Length>=4.95 6  2 virginica (0.00000 0.33333 0.66667)
        58) Petal.Width>=1.55 2  0 versicolor (0.00000 1.00000 0.00000) *
```

14/22

# Random forests

1. Bootstrap samples
2. At each split, bootstrap variables
3. Grow multiple trees and vote

## Pros:

1. Accuracy

## Cons:

1. Speed
2. Interpretability
3. Overfitting

# Random forests

```
library(randomForest)
forestIris <- randomForest(Species~ Petal.Width + Petal.Length,data=iris,prox=TRUE)
forestIris
```

Call:

```
randomForest(formula = Species ~ Petal.Width + Petal.Length,      data = iris, prox = TRUE)
```

Type of random forest: classification

Number of trees: 500

No. of variables tried at each split: 1

OOB estimate of error rate: 3.33%

Confusion matrix:

|            | setosa | versicolor | virginica | class.error |
|------------|--------|------------|-----------|-------------|
| setosa     | 50     | 0          | 0         | 0.00        |
| versicolor | 0      | 47         | 3         | 0.06        |
| virginica  | 0      | 2          | 48        | 0.04        |

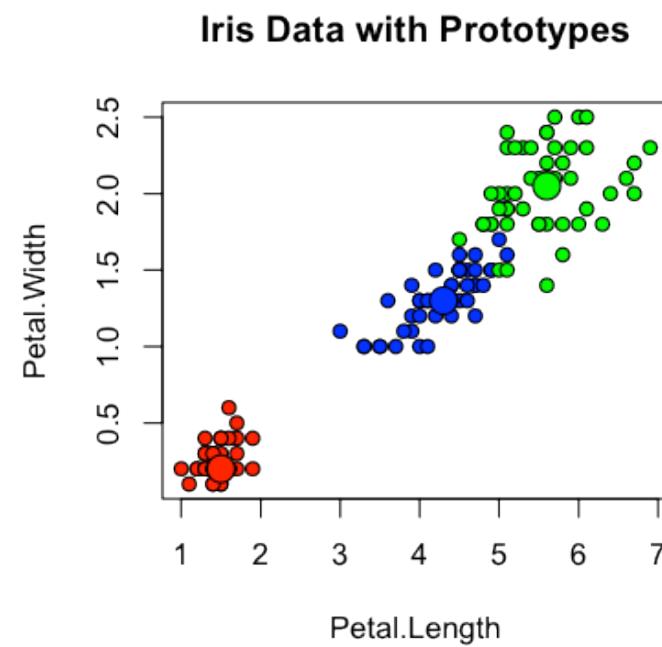
# Getting a single tree

```
getTree(forestIris,k=2)
```

|    | left | daughter | right | daughter | split | var | split | point | status | prediction |
|----|------|----------|-------|----------|-------|-----|-------|-------|--------|------------|
| 1  |      | 2        |       | 3        |       | 2   |       | 2.45  | 1      | 0          |
| 2  |      | 0        |       | 0        |       | 0   |       | 0.00  | -1     | 1          |
| 3  |      | 4        |       | 5        |       | 1   |       | 1.70  | 1      | 0          |
| 4  |      | 6        |       | 7        |       | 1   |       | 1.55  | 1      | 0          |
| 5  |      | 0        |       | 0        |       | 0   |       | 0.00  | -1     | 3          |
| 6  |      | 8        |       | 9        |       | 1   |       | 1.35  | 1      | 0          |
| 7  |      | 0        |       | 0        |       | 0   |       | 0.00  | -1     | 2          |
| 8  |      | 0        |       | 0        |       | 0   |       | 0.00  | -1     | 2          |
| 9  |      | 10       |       | 11       |       | 1   |       | 1.45  | 1      | 0          |
| 10 |      | 12       |       | 13       |       | 2   |       | 5.20  | 1      | 0          |
| 11 |      | 0        |       | 0        |       | 0   |       | 0.00  | -1     | 2          |
| 12 |      | 0        |       | 0        |       | 0   |       | 0.00  | -1     | 2          |
| 13 |      | 0        |       | 0        |       | 0   |       | 0.00  | -1     | 3          |

# Class "centers"

```
iris.p <- classCenter(iris[,c(3,4)], iris$Species, forestIris$prox)
plot(iris[,3], iris[,4], pch=21, xlab=names(iris)[3], ylab=names(iris)[4],
bg=c("red", "blue", "green")[as.numeric(factor(iris$Species))],
main="Iris Data with Prototypes")
points(iris.p[,1], iris.p[,2], pch=21, cex=2, bg=c("red", "blue", "green"))
```



18/22

# Combining random forests

```
forestIris1 <- randomForest(Species~Petal.Width + Petal.Length,data=iris,prox=TRUE,ntree=50)
forestIris2 <- randomForest(Species~Petal.Width + Petal.Length,data=iris,prox=TRUE,ntree=50)
forestIris3 <- randomForest(Species~Petal.Width + Petal.Length,data=iris,prox=TRUE,nrtee=50)
combine(forestIris1,forestIris2,forestIris3)
```

Call:

```
randomForest(formula = Species ~ Petal.Width + Petal.Length,           data = iris, prox = TRUE, ntree
              Type of random forest: classification
```

Number of trees: 600

No. of variables tried at each split: 1

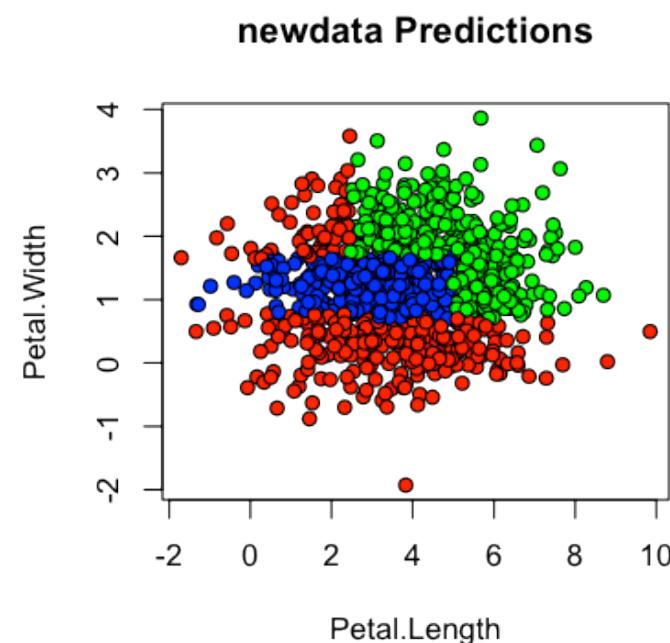
# Predicting new values

```
newdata <- data.frame(Sepal.Length<- rnorm(1000,mean(iris$Sepal.Length),  
                      sd(iris$Sepal.Length)),  
                      Sepal.Width <- rnorm(1000,mean(iris$Sepal.Width),  
                      sd(iris$Sepal.Width)),  
                      Petal.Width <- rnorm(1000,mean(iris$Petal.Width),  
                      sd(iris$Petal.Width)),  
                      Petal.Length <- rnorm(1000,mean(iris$Petal.Length),  
                      sd(iris$Petal.Length)))  
  
pred <- predict(forestIris,newdata)
```

20/22

# Predicting new values

```
plot(newdata[,4], newdata[,3], pch=21, xlab="Petal.Length", ylab="Petal.Width",
bg=c("red", "blue", "green")[as.numeric(pred)], main="newdata Predictions")
```



# Notes and further resources

## Notes:

- Bootstrapping is useful for nonlinear models
- Care should be taken to avoid overfitting (see [rfcv](#) function)
- Out of bag estimates are efficient estimates of test error

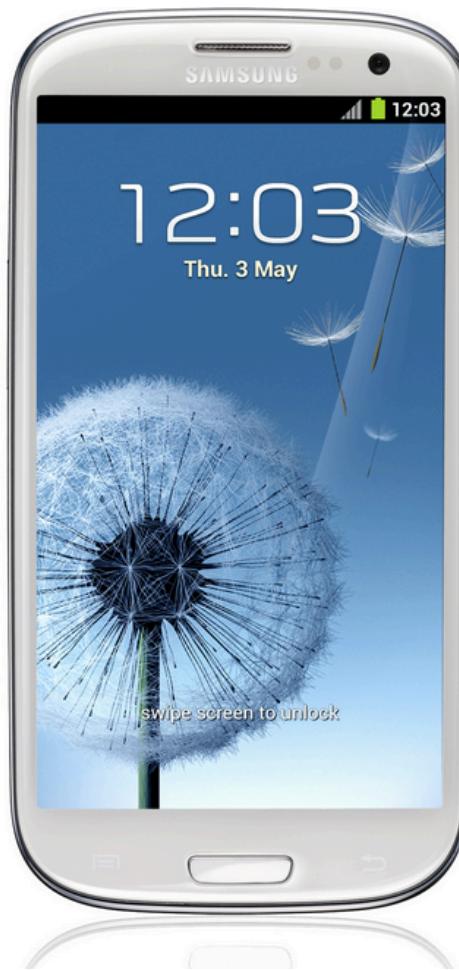
## Further resources:

- [Random forests](#)
- [Random forest Wikipedia](#)
- [Bagging](#)
- [Bagging and boosting](#)

# Clustering example

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

# Samsung Galaxy S3



<http://www.samsung.com/global/galaxys3/>

2/18

# Samsung Data

The screenshot shows a web browser window for the UCI Machine Learning Repository. The URL in the address bar is [archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones](http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones). The page features the UCI logo and a stylized antechinus illustration. The main title is "Human Activity Recognition Using Smartphones Data Set". Below it are download links for "Data Folder" and "Data Set Description". An abstract describes the database as built from recordings of 30 subjects performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. A table provides data set characteristics:

| Data Set Characteristics:  | Multivariate, Time-Series  | Number of Instances:  | 10299 | Area:               | Computer   |
|----------------------------|----------------------------|-----------------------|-------|---------------------|------------|
| Attribute Characteristics: | N/A                        | Number of Attributes: | 561   | Date Donated:       | 2012-12-10 |
| Associated Tasks:          | Classification, Clustering | Missing Values?       | N/A   | Number of Web Hits: | 5485       |

**Source:**  
Jorge L. Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto.  
Smartlab - Non Linear Complex Systems Laboratory  
DITEN - Università degli Studi di Genova, Genoa I-16145, Italy.  
[activityrecognition@smartlab.ws](mailto:activityrecognition@smartlab.ws)  
[www.smartlab.ws](http://www.smartlab.ws)

<http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>

# Slightly processed data

```
download.file("https://dl.dropbox.com/u/7710864/courseraPublic/samsungData.rda"
              ,destfile="./data/samsungData.rda",method="curl")
load("./data/samsungData.rda")
names(samsungData)[1:12]
```

```
[1] "tBodyAcc-mean()-X" "tBodyAcc-mean()-Y" "tBodyAcc-mean()-Z" "tBodyAcc-std()-X"
[5] "tBodyAcc-std()-Y"   "tBodyAcc-std()-Z"   "tBodyAcc-mad()-X"  "tBodyAcc-mad()-Y"
[9] "tBodyAcc-mad()-Z"  "tBodyAcc-max()-X"  "tBodyAcc-max()-Y"  "tBodyAcc-max()-Z"
```

```
table(samsungData$activity)
```

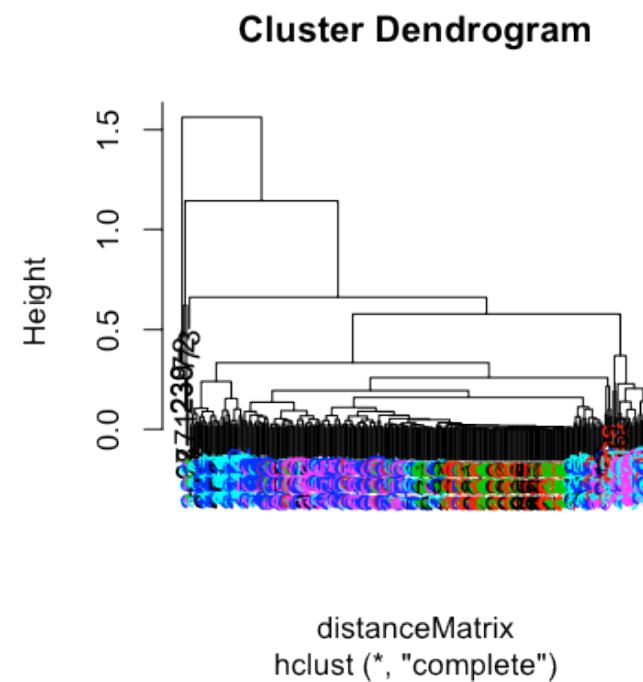
|  | laying | sitting | standing | walk | walkdown | walkup |
|--|--------|---------|----------|------|----------|--------|
|  | 1407   | 1286    | 1374     | 1226 | 986      | 1073   |

# Plotting average acceleration for first subject

```
par(mfrow=c(1,2))
numericActivity <- as.numeric(as.factor(samsungData$activity))[samsungData$subject==1]
plot(samsungData[samsungData$subject==1,1],pch=19,col=numericActivity,ylab=names(samsungData)[1])
plot(samsungData[samsungData$subject==1,2],pch=19,col=numericActivity,ylab=names(samsungData)[2])
legend(150,-0.1,legend=unique(samsungData$activity),col=unique(numericActivity),pch=19)
```

# Clustering based just on average acceleration

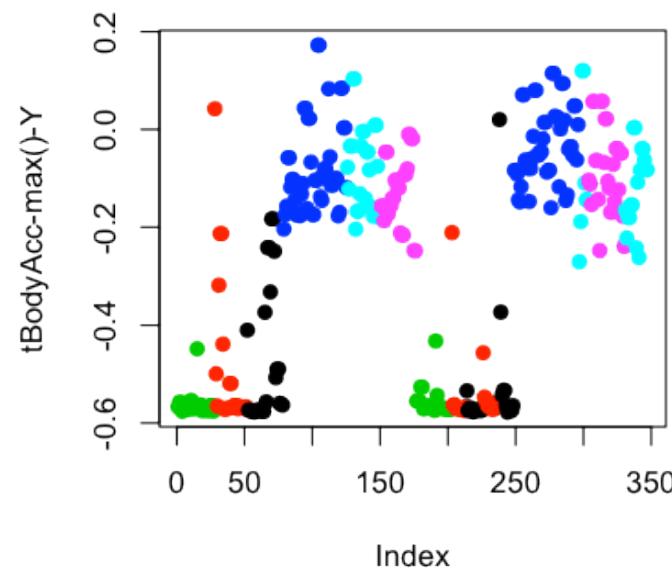
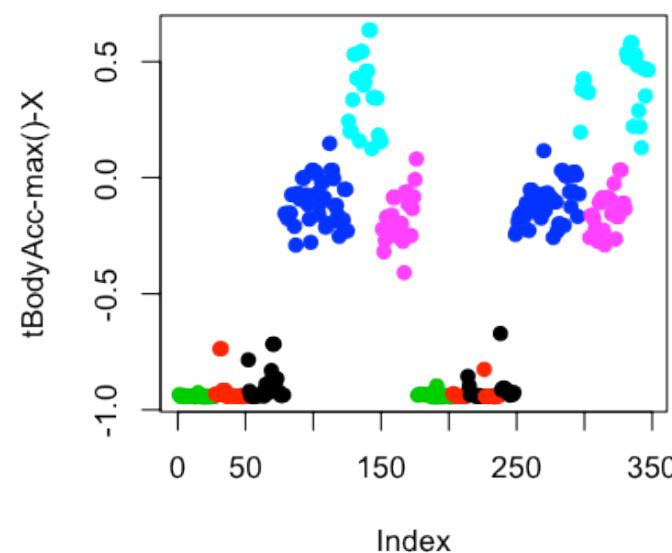
```
source("http://dl.dropbox.com/u/7710864/courseraPublic/myplclust.R")
distanceMatrix <- dist(samsungData[samsungData$subject==1,1:3])
hclustering <- hclust(distanceMatrix)
myplclust(hclustering,lab.col=numericActivity)
```



6/18

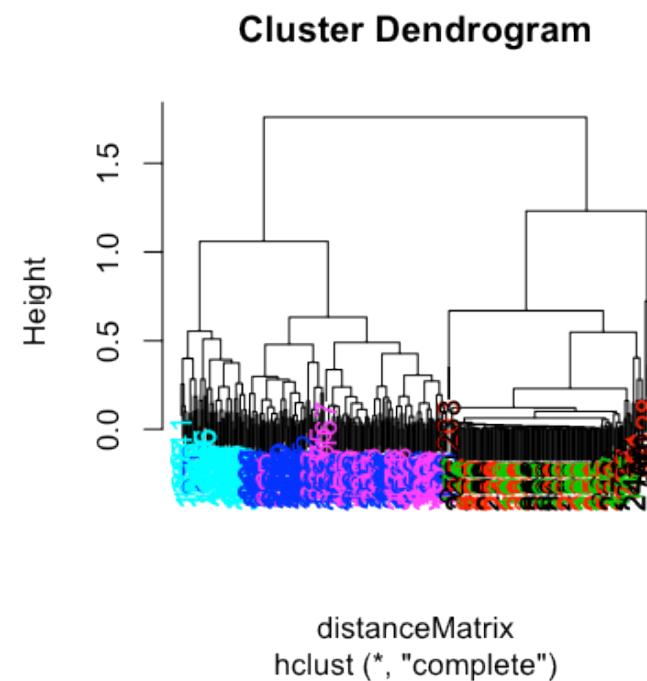
# Plotting max acceleration for the first subject

```
par(mfrow=c(1,2))
plot(samsungData[samsungData$subject==1,10],pch=19,col=numericActivity,ylab=names(samsungData)[10])
plot(samsungData[samsungData$subject==1,11],pch=19,col=numericActivity,ylab=names(samsungData)[11])
```



# Clustering based on maximum acceleration

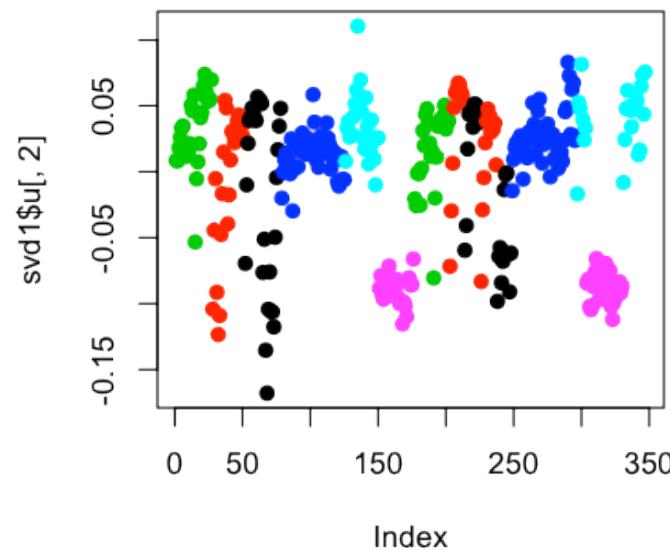
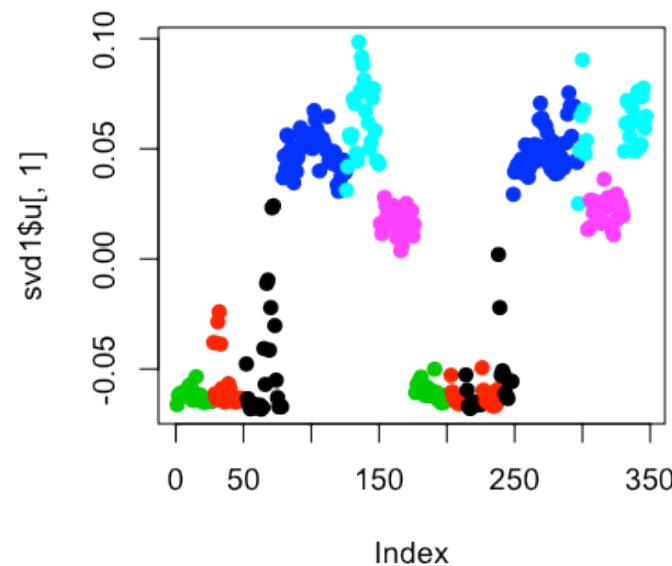
```
source("http://dl.dropbox.com/u/7710864/courseraPublic/myplclust.R")
distanceMatrix <- dist(samsungData[samsungData$subject==1,10:12])
hclustering <- hclust(distanceMatrix)
myplclust(hclustering,lab.col=numericActivity)
```



8/18

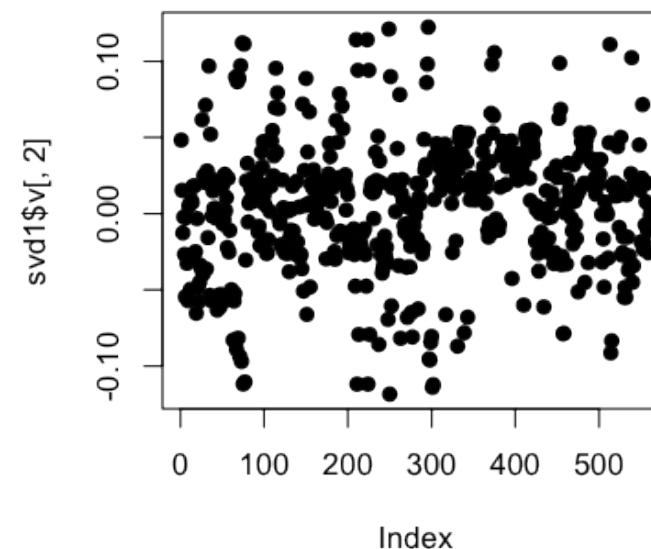
# Singular value decomposition

```
svd1 = svd(scale(samsungData[samsungData$subject==1,-c(562,563)]))  
par(mfrow=c(1,2))  
plot(svd1$u[,1],col=numericActivity,pch=19)  
plot(svd1$u[,2],col=numericActivity,pch=19)
```



# Find maximum contributor

```
plot(svd1$v[,2],pch=19)
```



10/18

# New clustering with maximum contributer

```
maxContrib <- which.max(svd1$v[,2])
distanceMatrix <- dist(samsungData[samsungData$subject==1,c(10:12,maxContrib)])
hclustering <- hclust(distanceMatrix)
myplclust(hclustering,lab.col=numericActivity)
```

11/18

# New clustering with maximum contributer

```
names(samsungData)[maxContrib]
```

```
[1] "fBodyAcc-meanFreq() -Z"
```

12/18

# K-means clustering (nstart=1, first try)

```
kClust <- kmeans(samsungData[samsungData$subject==1,-c(562,563)],centers=6)
table(kClust$cluster,samsungData$activity[samsungData$subject==1])
```

|   | laying | sitting | standing | walk | walkdown | walkup |
|---|--------|---------|----------|------|----------|--------|
| 1 | 42     | 45      | 53       | 0    | 0        | 0      |
| 2 | 0      | 0       | 0        | 0    | 26       | 0      |
| 3 | 0      | 0       | 0        | 45   | 0        | 0      |
| 4 | 0      | 0       | 0        | 50   | 0        | 0      |
| 5 | 0      | 0       | 0        | 0    | 23       | 0      |
| 6 | 8      | 2       | 0        | 0    | 0        | 53     |

# K-means clustering (nstart=1, second try)

```
kClust <- kmeans(samsungData[samsungData$subject==1,-c(562,563)],centers=6,nstart=1)
table(kClust$cluster,samsungData$activity[samsungData$subject==1])
```

|   | laying | sitting | standing | walk | walkdown | walkup |
|---|--------|---------|----------|------|----------|--------|
| 1 | 0      | 0       | 0        | 27   | 1        | 0      |
| 2 | 0      | 0       | 0        | 46   | 0        | 0      |
| 3 | 0      | 0       | 0        | 22   | 0        | 0      |
| 4 | 8      | 2       | 0        | 0    | 0        | 53     |
| 5 | 0      | 0       | 0        | 0    | 48       | 0      |
| 6 | 42     | 45      | 53       | 0    | 0        | 0      |

# K-means clustering (nstart=100, first try)

```
kClust <- kmeans(samsungData[samsungData$subject==1,-c(562,563)],centers=6,nstart=100)
table(kClust$cluster,samsungData$activity[samsungData$subject==1])
```

|   | laying | sitting | standing | walk | walkdown | walkup |
|---|--------|---------|----------|------|----------|--------|
| 1 | 0      | 37      | 51       | 0    | 0        | 0      |
| 2 | 18     | 10      | 2        | 0    | 0        | 0      |
| 3 | 0      | 0       | 0        | 95   | 0        | 0      |
| 4 | 0      | 0       | 0        | 0    | 49       | 0      |
| 5 | 29     | 0       | 0        | 0    | 0        | 0      |
| 6 | 3      | 0       | 0        | 0    | 0        | 53     |

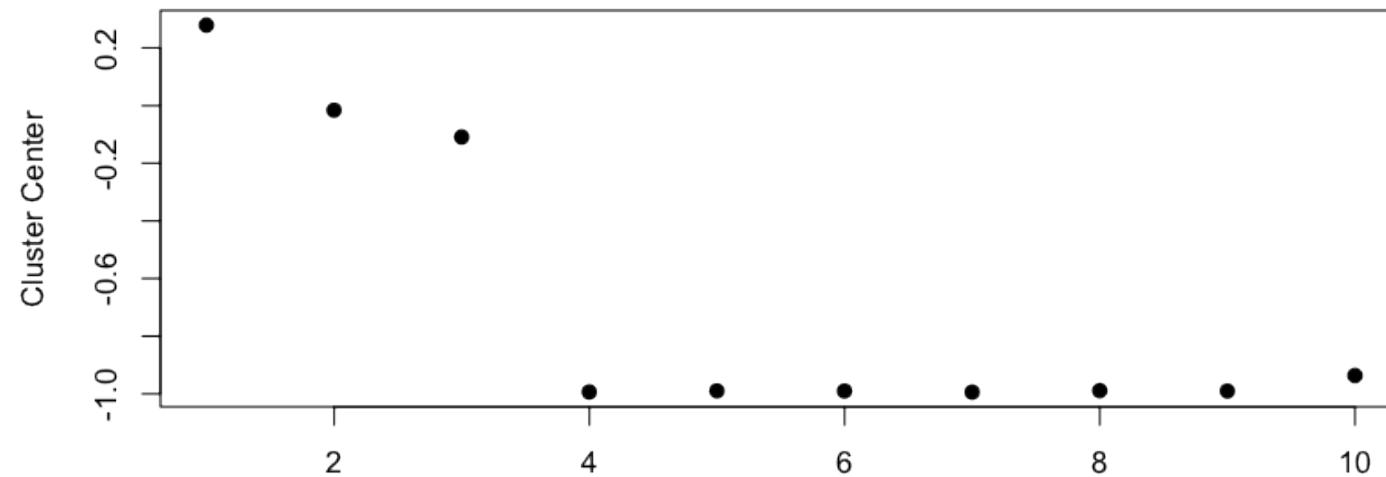
# K-means clustering (nstart=100, second try)

```
kClust <- kmeans(samsungData[samsungData$subject==1,-c(562,563)],centers=6,nstart=100)
table(kClust$cluster,samsungData$activity[samsungData$subject==1])
```

|   | laying | sitting | standing | walk | walkdown | walkup |
|---|--------|---------|----------|------|----------|--------|
| 1 | 29     | 0       | 0        | 0    | 0        | 0      |
| 2 | 0      | 0       | 0        | 0    | 49       | 0      |
| 3 | 0      | 0       | 0        | 95   | 0        | 0      |
| 4 | 18     | 10      | 2        | 0    | 0        | 0      |
| 5 | 0      | 37      | 51       | 0    | 0        | 0      |
| 6 | 3      | 0       | 0        | 0    | 0        | 53     |

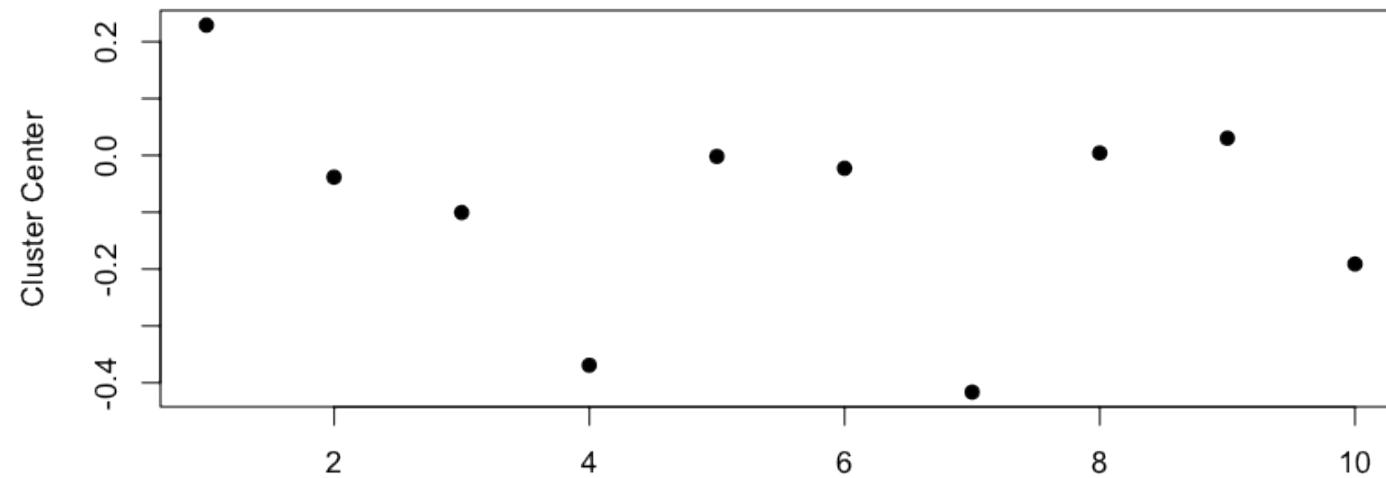
# Cluster 1 Variable Centers (Laying)

```
plot(kClust$center[1,1:10],pch=19,ylab="Cluster Center",xlab="")
```



# Cluster 2 Variable Centers (Walking)

```
plot(kClust$center[6,1:10],pch=19,ylab="Cluster Center",xlab="")
```



# Combining predictors

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

# Key ideas

- You can combine classifiers by averaging/voting
- Combining classifiers improves accuracy
- Combining classifiers reduces interpretability

2/13

# Netflix prize

BellKor = Combination of 107 predictors

The screenshot shows a web browser displaying the Netflix Prize Leaderboard at [www.netflixprize.com//leaderboard](http://www.netflixprize.com//leaderboard). The page features a prominent yellow banner with the words "Netflix Prize" and "COMPLETED". Below the banner, there's a navigation menu with links for Home, Rules, Leaderboard, and Update. The main section is titled "Leaderboard" and displays a table of top teams. A note says "Showing Test Score. [Click here to show quiz score](#)". The table has columns for Rank, Team Name, Best Test Score, % Improvement, and Best Submit Time. The top team is "BellKor's Pragmatic Chaos" with an RMSE of 0.8567. The table includes the following data:

| Rank | Team Name   | Best Test Score | % Improvement | Best Submit Time    |
|------|---|-----------------|---------------|---------------------|
| 1    | <a href="#">BellKor's Pragmatic Chaos</a>           | 0.8567          | 10.06         | 2009-07-26 18:18:28 |
| 2    | <a href="#">The Ensemble</a>                        | 0.8567          | 10.06         | 2009-07-26 18:38:22 |
| 3    | <a href="#">Grand Prize Team</a>                    | 0.8582          | 9.90          | 2009-07-10 21:24:40 |
| 4    | <a href="#">Opera Solutions and Vandelay United</a> | 0.8588          | 9.84          | 2009-07-10 01:12:31 |
| 5    | <a href="#">Vandelay Industries!</a>                | 0.8591          | 9.81          | 2009-07-10 00:32:20 |
| 6    | <a href="#">PragmaticTheory</a>                     | 0.8594          | 9.77          | 2009-06-24 12:06:56 |
| 7    | <a href="#">BellKor in BigChaos</a>                 | 0.8601          | 9.70          | 2009-05-13 08:14:09 |
| 8    | <a href="#">Dace</a>                                | 0.8612          | 9.59          | 2009-07-24 17:18:43 |
| 9    | <a href="#">Feeds2</a>                              | 0.8622          | 9.48          | 2009-07-12 13:11:51 |
| 10   | <a href="#">BigChaos</a>                            | 0.8623          | 9.47          | 2009-04-07 12:33:59 |

<http://www.netflixprize.com//leaderboard>

3/13

# Heritage health prize - Progress Prize 1

## 2. *Predictive Modelling*

Predictive models were built utilising the data sets created in Step 1. Numerous mathematical techniques were used to generate a set of candidate solutions.

## 3. *Ensembling*

The individual solutions produced in Step 2 were combined to create a single solution that was more accurate than any of its components.

## Market Makers

## 1 Introduction

My milestone 1 solution to the Heritage Health Prize with a RMSLE score of 0.457239 on the leaderboard consists of a linear blend of 21 result. These are mostly generated by relatively simple models which are all trained using stochastic gradient descent. First in section 2 I provide a description of the way the data is organized and the features that were used. Then in section 3 the training method and the post-processing steps are described. In section 4 each individual model is briefly described, all the relevant meta-parameter settings can be found in appendix Parameter settings. Finally the weights in the final blend are given in section 5.

## Mestrom

4/13

# Basic intuition - majority vote

Suppose we have 5 completely independent classifiers

If accuracy is 70% for each:

- $10 \times (0.7)^3(0.3)^2 + 5 \times (0.7)^4(0.3)^2 + (0.7)^5$
- 83.7% majority vote accuracy

With 101 independent classifiers

- 99.9% majority vote accuracy

# Approaches for combining classifiers

1. Bagging (see previous lecture)
2. Boosting
3. Combining different classifiers

6/13

# Example

```
#library(devtools)
#install_github("medley","mewo2")
library(medley)
set.seed(453234)
y <- rnorm(1000)
x1 <- (y > 0); x2 <- y*rnorm(1000)
x3 <- rnorm(1000,mean=y,sd=1); x4 <- (y > 0) & (y < 3)
x5 <- rbinom(1000,size=4,prob=exp(y)/(1+exp(y)))
x6 <- (y < -2) | (y > 2)
data <- data.frame(y=y,x1=x1,x2=x2,x3=x3,x4=x4,x5=x5,x6=x6)
train <- sample(1:1000,size=500)
trainData <- data[train,]; testData <- data[-train,]
```

# Basic models

```
library(tree)
lm1 <- lm(y ~.,data=trainData)
rmse(predict(lm1,data=testData), testData$y)
```

```
[1] 1.294
```

```
tree1 <- tree(y ~.,data=trainData)
rmse(predict(tree1,data=testData), testData$y)
```

```
[1] 1.299
```

```
tree2 <- tree(y~.,data=trainData[sample(1:dim(trainData)[1]),])
```

# Combining models

```
combine1 <- predict(lm1,data=testData)/2 + predict(tree1,data=testData)/2  
rmse(combine1,testData$y)
```

```
[1] 1.281
```

```
combine2 <- (predict(lm1,data=testData)/3 + predict(tree1,data=testData)/3  
+ predict(tree2,data=testData)/3)  
rmse(combine2,testData$y)
```

```
[1] 1.175
```

# Medley package

```
#library(devtools)
#install_github("medley","mewo2")
library(medley)
library(e1071)
library(randomForests)
x <- trainData[, -1]
y <- trainData$y
newx <- testData[, -1]
```

<http://www.kaggle.com/users/10748/martin-o-leary>

10/13

# Blending models (part 1)

```
m <- create.medley(x, y, errfunc=rmse);
for (g in 1:10) {
  m <- add.medley(m, svm, list(gamma=1e-3 * g));
}
```

```
CV model 1 svm (gamma = 0.001) time: 0.362 error: 0.5557
CV model 2 svm (gamma = 0.002) time: 0.373 error: 0.5367
CV model 3 svm (gamma = 0.003) time: 0.38 error: 0.5345
CV model 4 svm (gamma = 0.004) time: 0.376 error: 0.5333
CV model 5 svm (gamma = 0.005) time: 0.364 error: 0.5301
CV model 6 svm (gamma = 0.006) time: 0.355 error: 0.5265
CV model 7 svm (gamma = 0.007) time: 0.365 error: 0.5197
CV model 8 svm (gamma = 0.008) time: 0.359 error: 0.5115
CV model 9 svm (gamma = 0.009) time: 0.369 error: 0.5026
CV model 10 svm (gamma = 0.01) time: 0.355 error: 0.4946
```

# Blending models (part 2)

```
for (mt in 1:2) {  
  m <- add.medley(m, randomForest, list(mtry=mt));  
}
```

```
CV model 11 randomForest (mtry = 1) time: 2.015 error: 0.4668  
CV model 12 randomForest (mtry = 2) time: 3.532 error: 0.4135
```

```
m <- prune.medley(m, 0.8);  
rmse(predict(m,newx),testData$y)
```

```
Sampled... 96.00 %: 3 svm (gamma = 0.01)  
1.00 %: 4 svm (gamma = 0.009)  
1.00 %: 5 svm (gamma = 0.008)  
1.00 %: 6 svm (gamma = 0.007)  
1.00 %: 7 svm (gamma = 0.006)  
CV error: 0.4953
```

12/13

# Notes and further resources

## Notes:

- Even simple blending can be useful
- Majority vote is typical model for binary/multiclass data
- Makes models hard to interpret

## Further resources:

- [Bayesian model averaging](#)
- [Heritage health prize](#)
- [Netflix model blending](#)

# Count outcomes

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

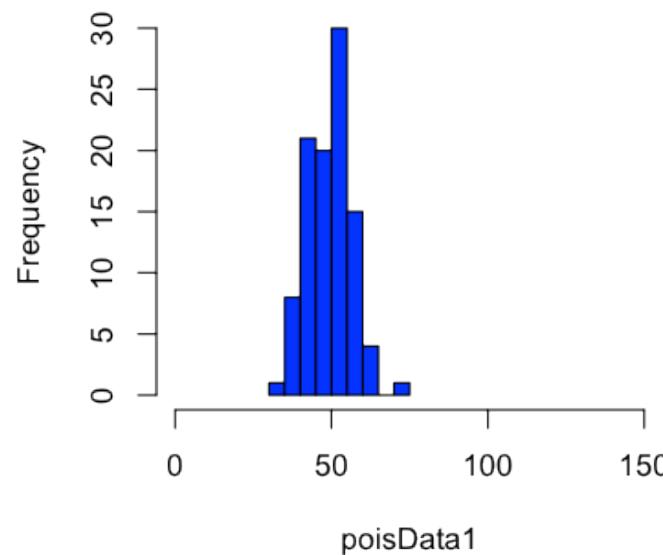
# Key ideas

- Many data take the form of counts
  - Calls to a call center
  - Number of flu cases in an area
  - Number of cars that cross a bridge
- Data may also be in the form of rates
  - Percent of children passing a test
  - Percent of hits to a website from a country
- Linear regression with transformation is an option

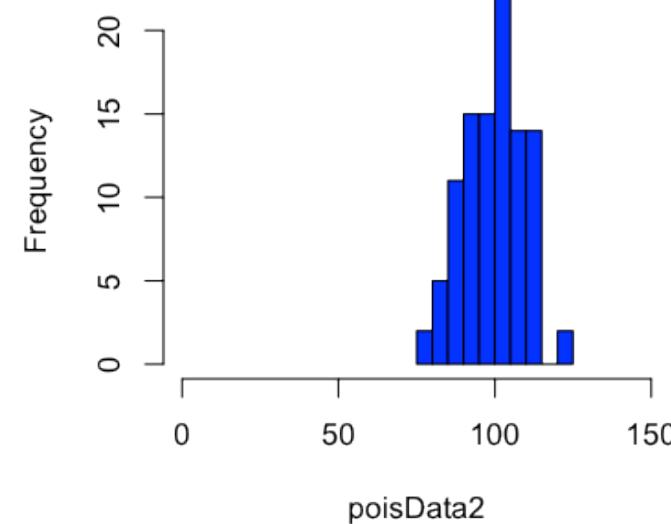
# Poisson distribution

```
set.seed(3433); par(mfrow=c(1,2))
poisData2 <- rpois(100,lambda=100); poisData1 <- rpois(100,lambda=50)
hist(poisData1,col="blue",xlim=c(0,150)); hist(poisData2,col="blue",xlim=c(0,150))
```

Histogram of poisData1



Histogram of poisData2



# Poisson distribution

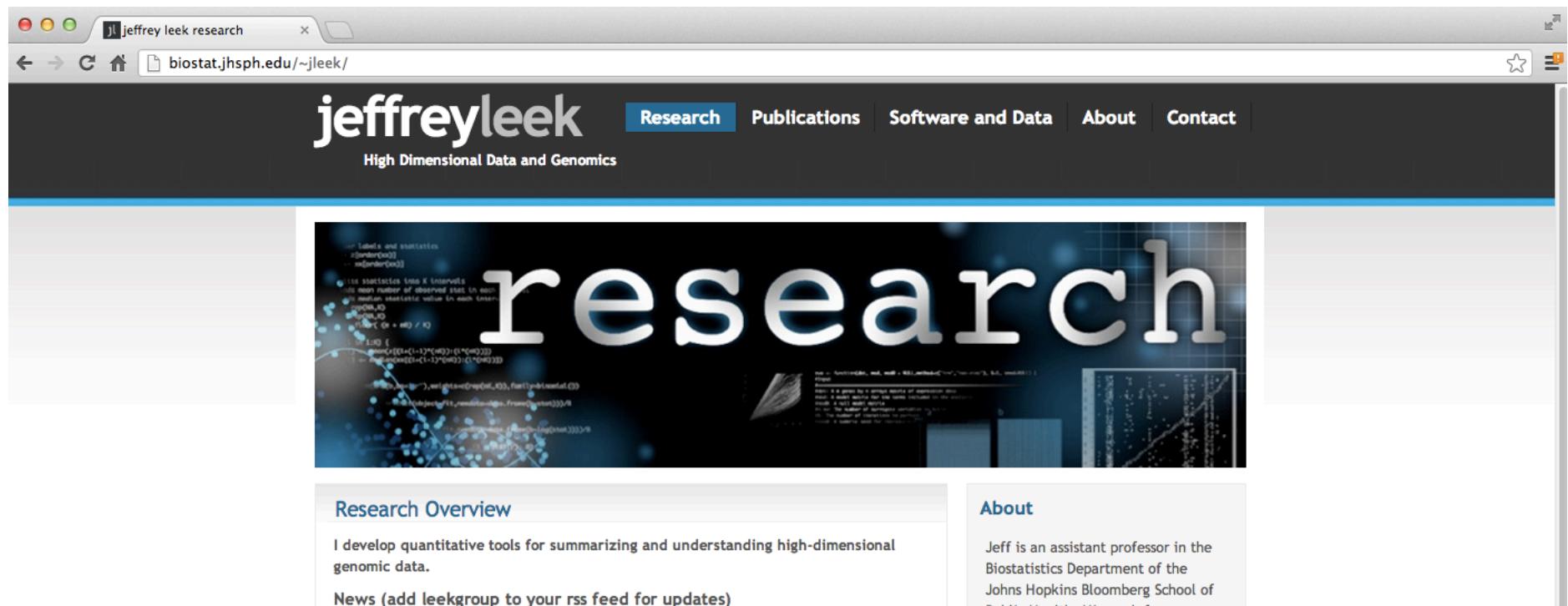
```
c(mean(poisData1),var(poisData1))
```

```
[1] 49.85 49.38
```

```
c(mean(poisData2),var(poisData2))
```

```
[1] 100.12 95.26
```

# Example: Leek Group Website Traffic



A screenshot of a web browser showing the homepage of the Jeffrey Leek research website. The URL in the address bar is <http://biostat.jhsph.edu/~jleek/>. The page features a dark header with the logo "jeffreyleek" and the subtitle "High Dimensional Data and Genomics". Below the header is a large banner image containing the word "research" in a large, glowing white font against a dark blue background. To the left of the banner, there is some small, illegible code or mathematical text. On the right side of the banner, there is a small image of a scatter plot. Below the banner, there are two sidebar boxes: "Research Overview" on the left and "About" on the right. The "Research Overview" box contains text about developing quantitative tools for genomic data and a link to add the RSS feed. The "About" box contains text about Jeff being an assistant professor at Johns Hopkins Bloomberg School of Public Health.

<http://biostat.jhsph.edu/~jleek/>

5/19

# Website data

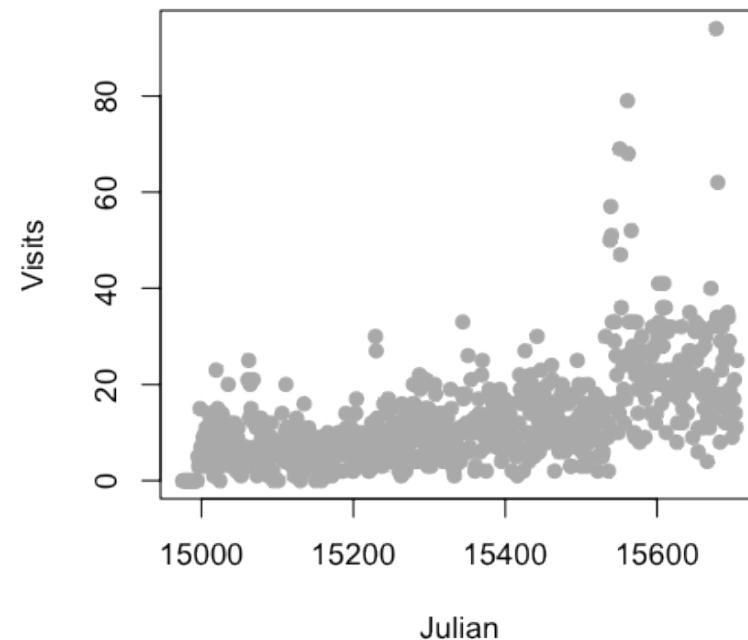
```
download.file("https://dl.dropbox.com/u/7710864/data/gaData.rda", destfile=".~/data/gaData.rda", method="curl")
load("./data/gaData.rda")
gaData$ julian <- julian(gaData$date)
head(gaData)
```

|   | date       | visits | simplystats | julian |
|---|------------|--------|-------------|--------|
| 1 | 2011-01-01 | 0      | 0           | 14975  |
| 2 | 2011-01-02 | 0      | 0           | 14976  |
| 3 | 2011-01-03 | 0      | 0           | 14977  |
| 4 | 2011-01-04 | 0      | 0           | 14978  |
| 5 | 2011-01-05 | 0      | 0           | 14979  |
| 6 | 2011-01-06 | 0      | 0           | 14980  |

<http://skardhamar.github.com/rga/>

# Plot data

```
plot(gaData$julian,gaData$visits,pch=19,col="darkgrey",xlab="Julian",ylab="Visits")
```



7/19

# Linear regression

$$NH_i = b_0 + b_1 JD_i + e_i$$

$NH_i$  - number of hits to the website

$JD_i$  - day of the year (Julian day)

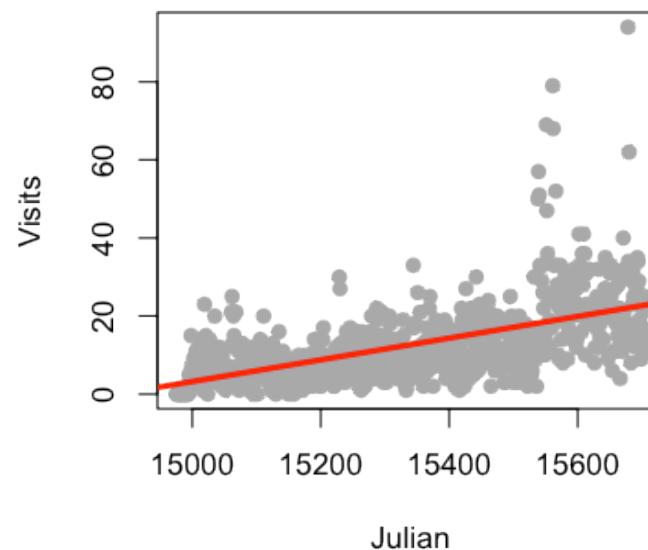
$b_0$  - number of hits on Julian day 0 (1970-01-01)

$b_1$  - increase in number of hits per unit day

$e_i$  - variation due to everything we didn't measure

# Linear regression line

```
plot(gaData$julian,gaData$visits,pch=19,col="darkgrey",xlab="Julian",ylab="Visits")
lm1 <- lm(gaData$visits ~ gaData$julian)
abline(lm1,col="red",lwd=3)
```



9/19

# Linear vs. Poisson regression

## Linear

$$NH_i = b_0 + b_1 JD_i + e_i$$

or

$$E[NH_i|JD_i, b_0, b_1] = b_0 + b_1 JD_i$$

## Poisson/log-linear

$$\log(E[NH_i|JD_i, b_0, b_1]) = b_0 + b_1 JD_i$$

or

$$E[NH_i|JD_i, b_0, b_1] = \exp(b_0 + b_1 JD_i)$$

# Multiplicative differences

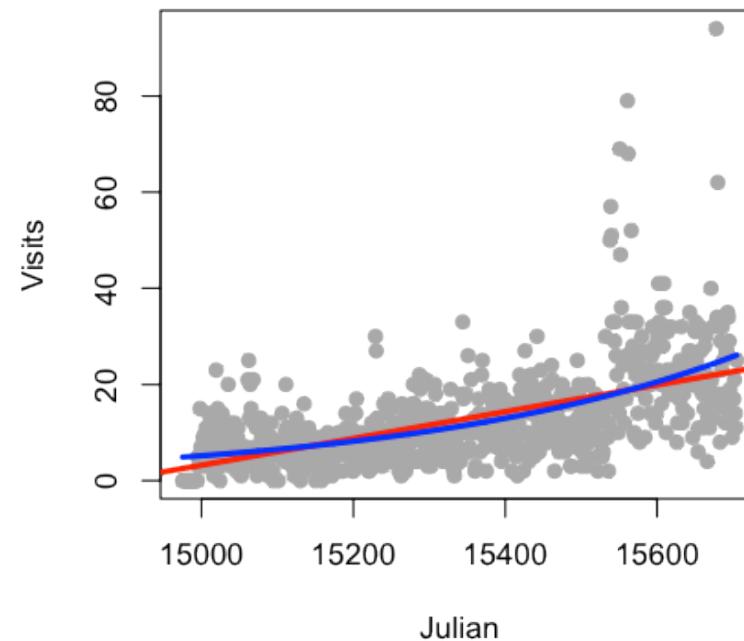
$$E[NH_i|JD_i, b_0, b_1] = \exp(b_0 + b_1 JD_i)$$

$$E[NH_i|JD_i, b_0, b_1] = \exp(b_0) \exp(b_1 JD_i)$$

If  $JD_i$  is increased by one unit,  $E[NH_i|JD_i, b_0, b_1]$  is multiplied by  $\exp(b_1)$

# Poisson regression in R

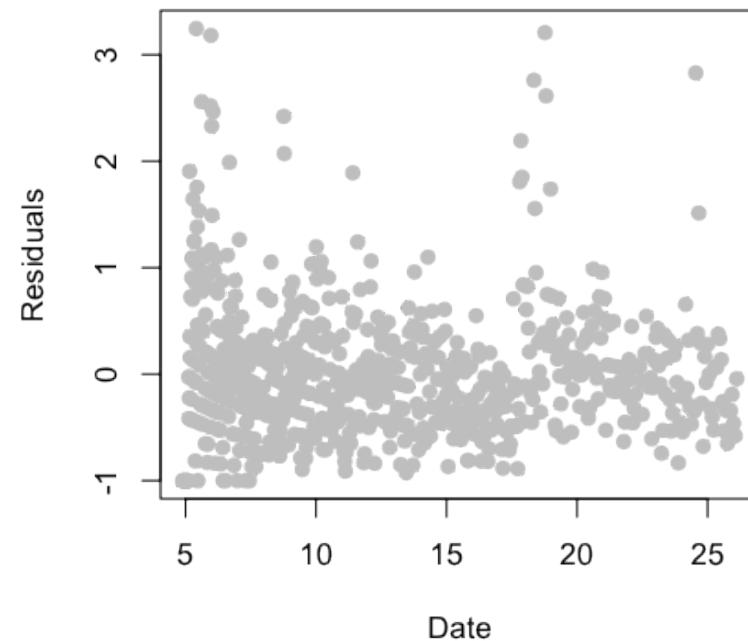
```
plot(gaData$julian,gaData$visits,pch=19,col="darkgrey",xlab="Julian",ylab="Visits")
glm1 <- glm(gaData$visits ~ gaData$julian,family="poisson")
abline(lm1,col="red",lwd=3); lines(gaData$julian,glm1$fitted,col="blue",lwd=3)
```



12/19

# Mean-variance relationship?

```
plot(glm1$fitted,glm1$residuals,pch=19,col="grey",ylab="Residuals",xlab="Date")
```



13/19

# Model agnostic standard errors

```
library(sandwich)
confint.agnostic <- function (object, parm, level = 0.95, ...)
{
  cf <- coef(object); pnames <- names(cf)
  if (missing(parm))
    parm <- pnames
  else if (is.numeric(parm))
    parm <- pnames[parm]
  a <- (1 - level)/2; a <- c(a, 1 - a)
  pct <- stats:::format.perc(a, 3)
  fac <- qnorm(a)
  ci <- array(NA, dim = c(length(parm), 2L), dimnames = list(parm,
    pct))
  ses <- sqrt(diag(sandwich::vcovHC(object)))[parm]
  ci[] <- cf[parm] + ses %o% fac
  ci
}
```

<http://stackoverflow.com/questions/3817182/vcovhc-and-confidence-interval>

14/19

# Estimating confidence intervals

```
confint(glm1)
```

```
        2.5 %      97.5 %
(Intercept) -34.34658 -31.159716
gaData$julian  0.00219  0.002396
```

```
confint.agnostic(glm1)
```

15/19

# Rates

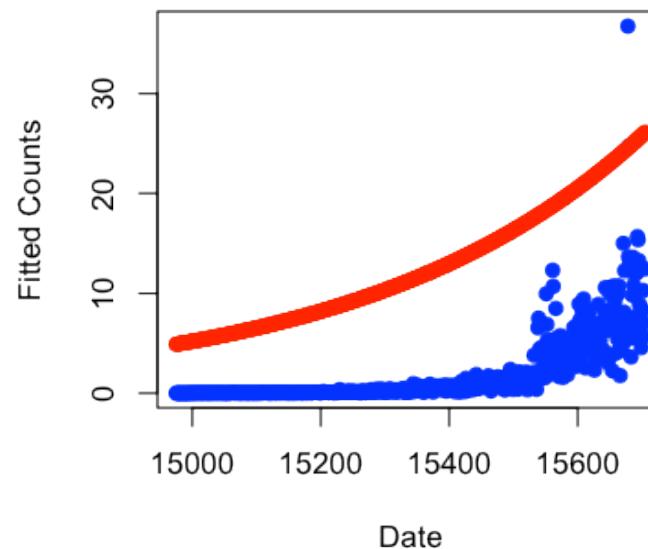
$$E[NHSS_i|JD_i, b_0, b_1]/NH_i = \exp(b_0 + b_1 JD_i)$$

$$\log(E[NHSS_i|JD_i, b_0, b_1]) - \log(NH_i) = b_0 + b_1 JD_i$$

$$\log(E[NHSS_i|JD_i, b_0, b_1]) = \log(NH_i) + b_0 + b_1 JD_i$$

# Fitting rates in R

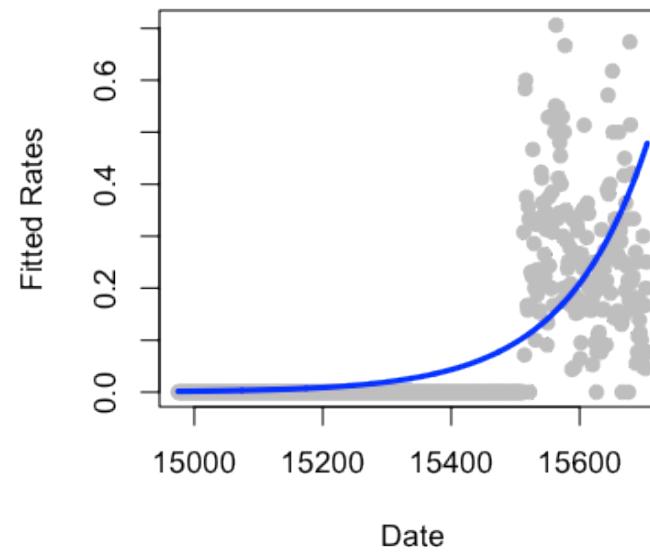
```
glm2 <- glm(gaData$simplystats ~ julian(gaData$date), offset=log(visits+1),
             family="poisson", data=gaData)
plot(julian(gaData$date), glm2$fitted, col="blue", pch=19, xlab="Date", ylab="Fitted Counts")
points(julian(gaData$date), glm1$fitted, col="red", pch=19)
```



17/19

# Fitting rates in R

```
glm2 <- glm(gaData$simplystats ~ julian(gaData$date), offset=log(visits+1),
             family="poisson", data=gaData)
plot(julian(gaData$date), gaData$simplystats/(gaData$visits+1), col="grey", xlab="Date",
      ylab="Fitted Rates", pch=19)
lines(julian(gaData$date), glm2$fitted/(gaData$visits+1), col="blue", lwd=3)
```



18/19

# More information

- [Log-linear models and multiway tables](#)
- [Wikipedia on Poisson regression](#), [Wikipedia on overdispersion](#)
- [Regression models for count data in R](#)
- [pscl package - the function \*zeroInfl\* fits zero inflated models.](#)

# Course wrap-up

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

# Why we do applied statistics

"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 applied statistics

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

3/16

# Why applied statistics?



4/16

# Why applied statistics?



## 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

5/16

# Why applied statistics?

McKinsey Global Institute

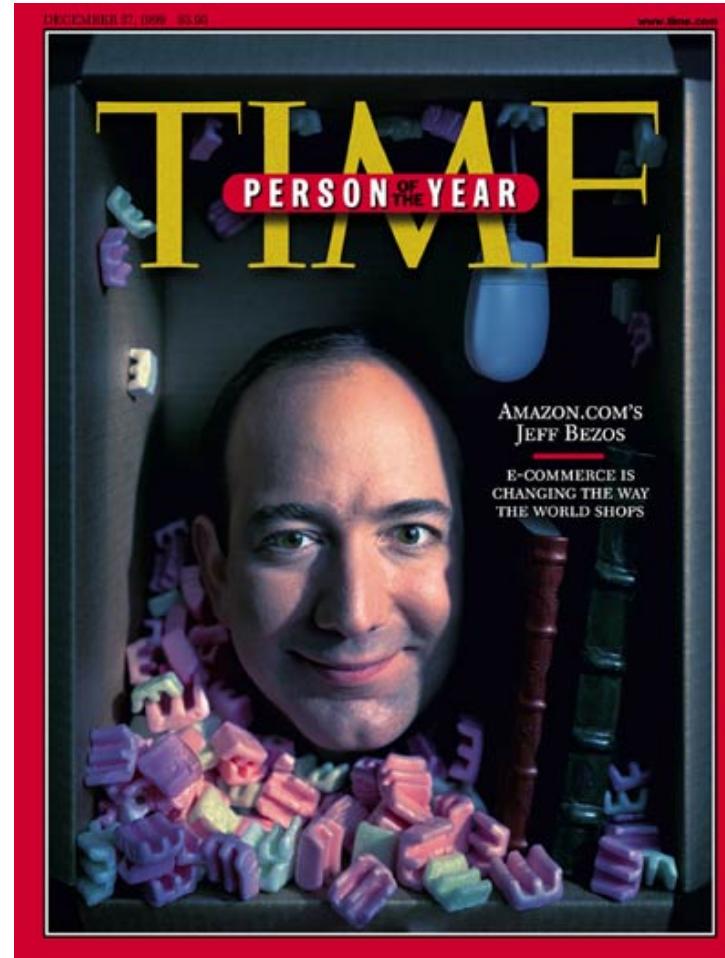


June 2011

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

6/16

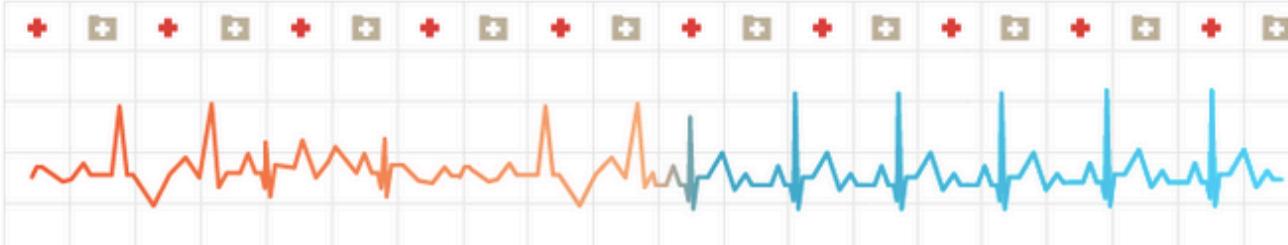
# Why are you lucky?



7/16

# Why are you lucky

[Competition Details](#) » [Get the Data](#) » [Make a submission](#)



## Improve Healthcare, Win \$3,000,000.

**Identify patients who will be admitted to a hospital  
within the next year using historical claims data. (Enter  
by 06:59:59 UTC Oct 4 2012)**

Please note: Deadline is 06:59:59 UTC on October 4, 2012 for new registrations and team mergers.

-----

- [Description](#)
- [Evaluation](#)
- [Rules](#)
- [Dos and Don'ts](#)
- [FAQ](#)
- [Milestone Winners](#)
- [Timeline](#)

[Heritage Health Prize](#)

8/16

# New data drives new statistical ideas

- How do we make better beer?
  - **Data:** Measures of beer quality
  - **Statistic:** The t-statistic
- What characteristics of field lead to better crops?
  - **Data:** Field characteristics, crop yields
  - **Statistic:** Analysis of variance (ANOVA)
- How long do people live?
  - **Data:** Survival times of people (censored)
  - **Statistic:** Kaplan-Meier Estimator
- What movies will you like?
  - **Data:** Lots of other peoples movie ratings
  - **Statistic(s):** Recommender systems

# Who is an applied statistician?

[Daryl Morey](#)



[Hilary Mason](#)



[Daphne Koller](#)

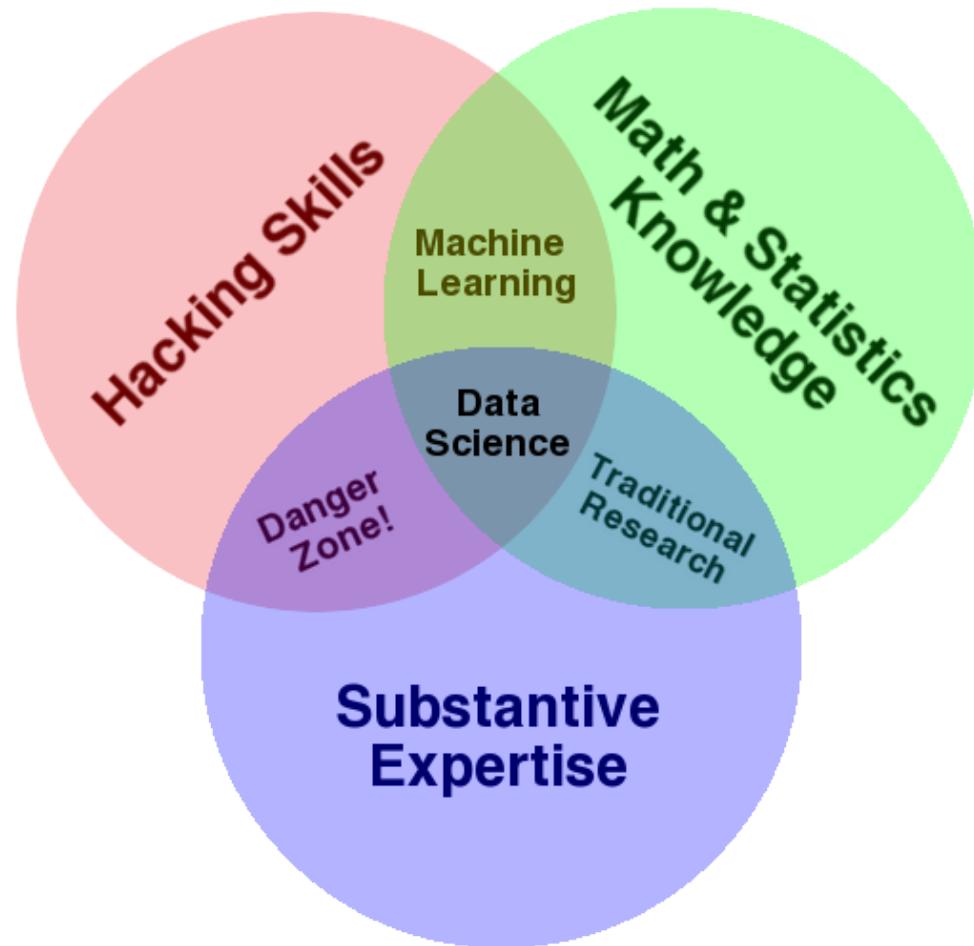


[Nate Silver](#)



10/16

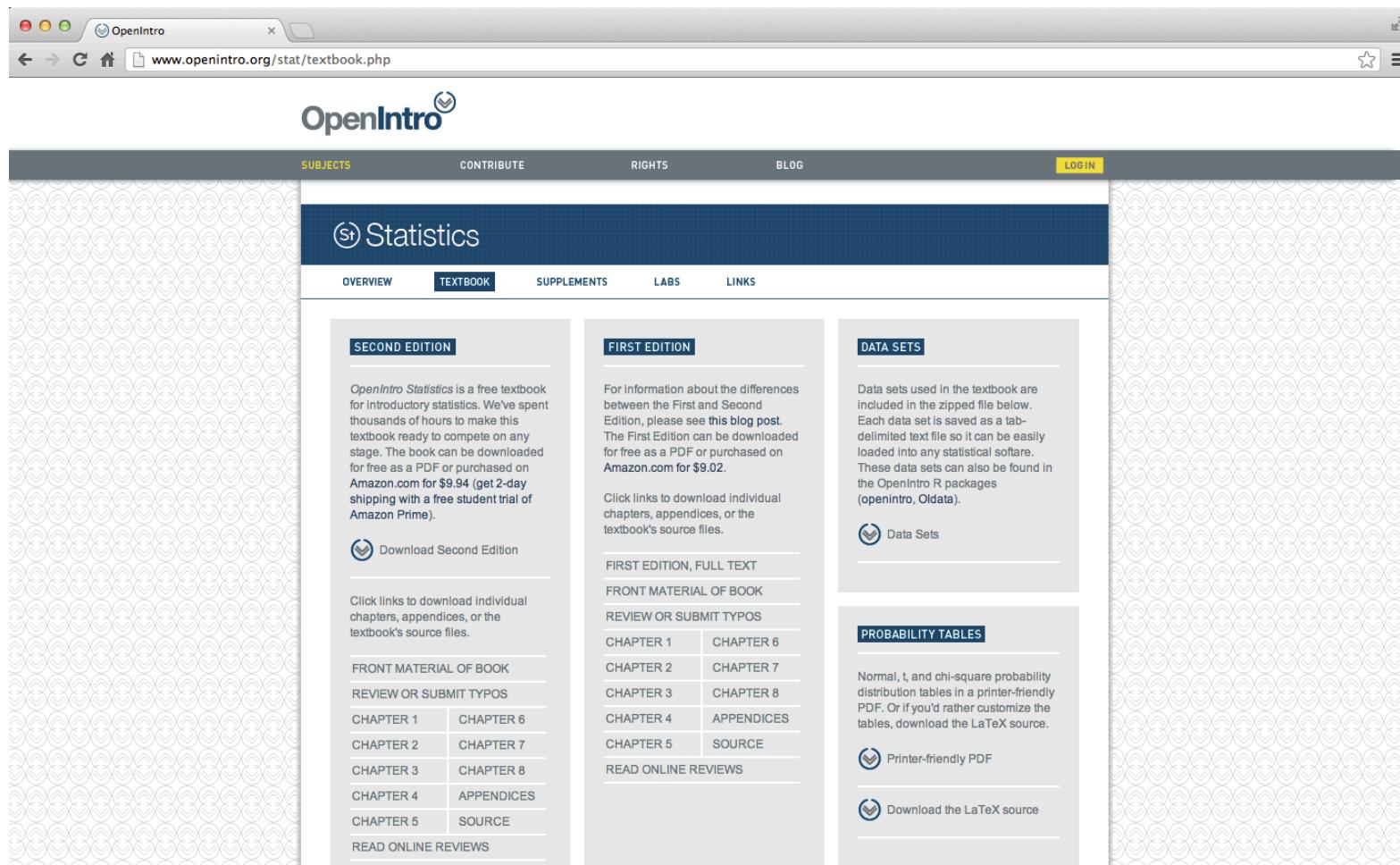
# An important goal



Drew Conway

11/16

# These might be useful

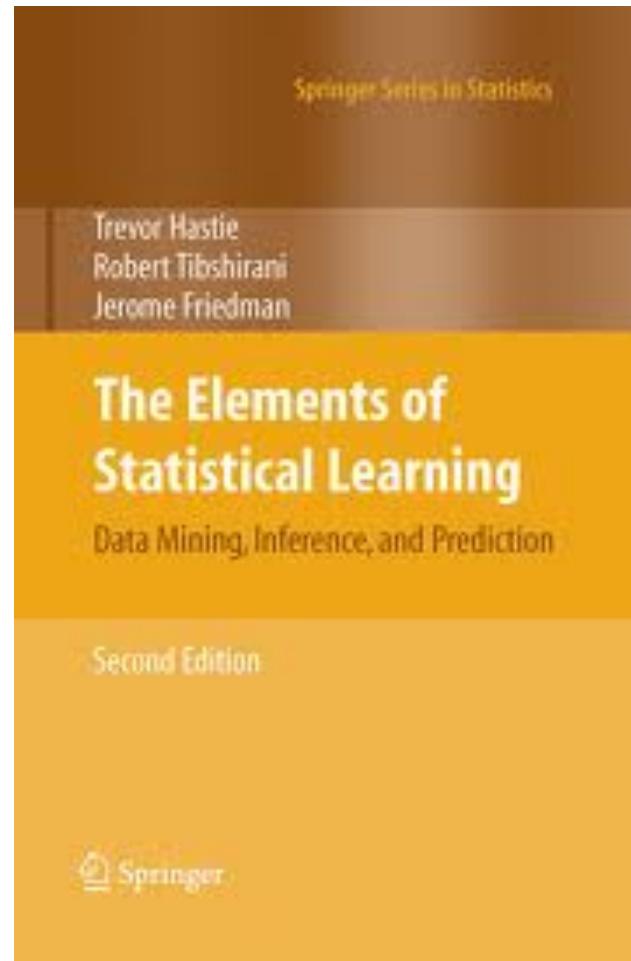


The screenshot shows a web browser displaying the OpenIntro Statistics textbook page at [www.openintro.org/stat/textbook.php](http://www.openintro.org/stat/textbook.php). The page has a dark blue header with the OpenIntro logo and navigation links for SUBJECTS, CONTRIBUTE, RIGHTS, BLOG, and LOGIN. Below the header, there's a large title "Statistics" with a circular icon containing "St". A navigation bar below the title includes OVERVIEW, TEXTBOOK (which is selected), SUPPLEMENTS, LABS, and LINKS. The main content is divided into three columns: "SECOND EDITION" (describing the free textbook available as a PDF or purchase on Amazon), "FIRST EDITION" (linking to download individual chapters, appendices, or the full text), and "DATA SETS" (describing data sets included in the zipped file). There are also sections for "FRONT MATERIAL OF BOOK", "REVIEW OR SUBMIT TYPOs", and "PROBABILITY TABLES". Each section contains links to specific files or sources.

<http://www.openintro.org/>

12/16

# These might be useful



<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>

13/16

# These might be useful

## Advanced Data Analysis from an Elementary Point of View

Cosma Rohilla Shalizi

Spring 2013  
Last L<sup>A</sup>T<sub>E</sub>X'd Friday 18<sup>th</sup> January, 2013

[Advanced Data Analysis from An Elementary Point of View](#)

14/16

# Also check out

- [Andrew Gelman's blog](#)
- [Larry Wasserman's blog](#)
- [Statsblogs](#)
- [Flowing Data](#)
- [junkcharts](#)
- [Hilary Mason's Blog](#) and [@hmason](#)
- [Cosma Shalizi's Blog](#)
- [Some other guys' blog](#)

[The top Biostatistics Department in the World](#) - No bias here :-)

# It has been my exteme pleasure

Thank you so much for all of your dedication, time, and enthusiasm. This has been a wonderful experience for me and I hope it has been for you too.

16/16

# Cross validation

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

# Key ideas

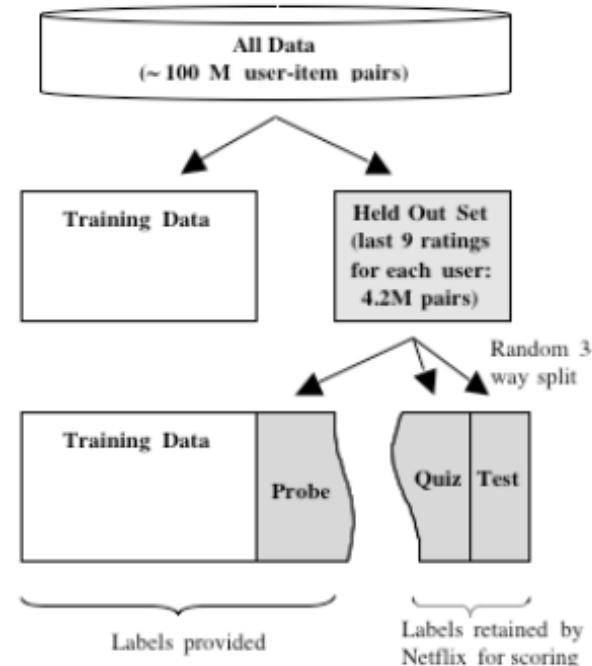
- Sub-sampling the training data
- Avoiding overfitting
- Making predictions generalizable

2/14

# Steps in building a prediction

1. Find the right data
2. Define your error rate
3. Split data into:
  - Training
  - Testing
  - Validation (optional)
4. On the training set pick features
5. On the training set pick prediction function
6. On the training set cross-validate
7. If no validation - apply 1x to test set
8. If validation - apply to test set and refine
9. If validation - apply 1x to validation

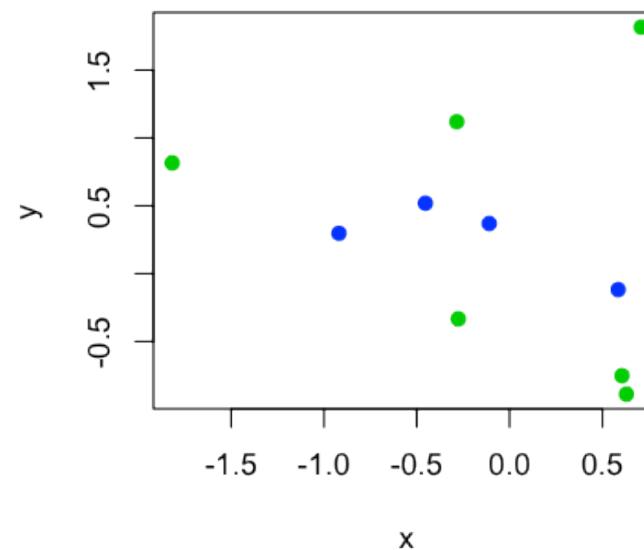
# Study design



<http://www2.research.att.com/~volinsky/papers/ASASStatComp.pdf>

# Overfitting

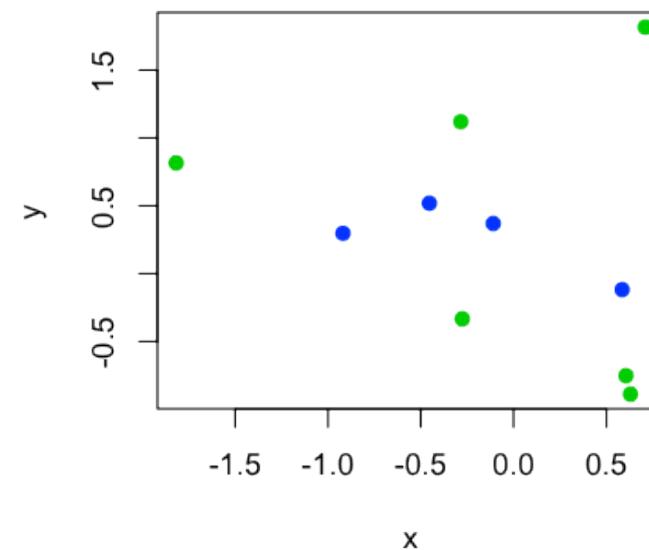
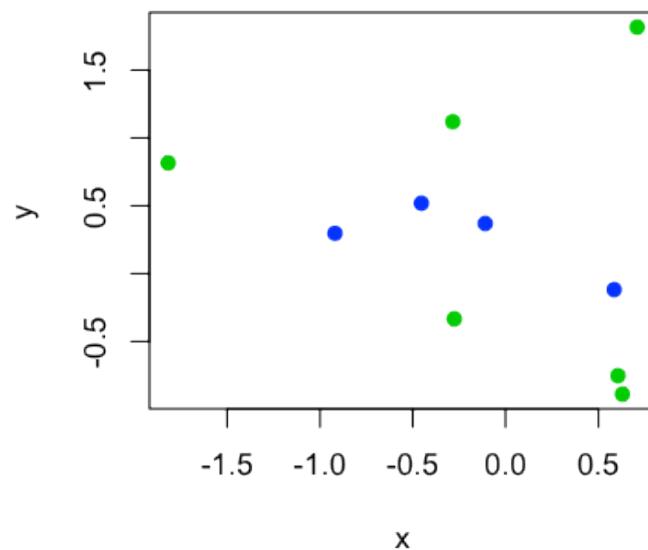
```
set.seed(12345)
x <- rnorm(10); y <- rnorm(10); z <- rbinom(10,size=1,prob=0.5)
plot(x,y,pch=19,col=(z+3))
```



# Classifier

If  $-0.2 < y < 0.6$  call blue, otherwise green

```
par(mfrow=c(1,2))
zhat <- (-0.2 < y) & (y < 0.6)
plot(x,y,pch=19,col=(z+3)); plot(x,y,pch=19,col=(zhat+3))
```



# New data

If  $-0.2 < y < 0.6$  call blue, otherwise green

```
set.seed(1233)
xnew <- rnorm(10); ynew <- rnorm(10); znew <- rbinom(10, size=1, prob=0.5)
par(mfrow=c(1, 2)); zhatnew <- (-0.2 < ynew) & (ynew < 0.6)
plot(xnew, ynew, pch=19, col=(z+3)); plot(xnew, ynew, pch=19, col=(zhatnew+3))
```

# Key idea

1. Accuracy on the training set (resubstitution accuracy) is optimistic
2. A better estimate comes from an independent set (test set accuracy)
3. But we can't use the test set when building the model or it becomes part of the training set
4. So we estimate the test set accuracy with the training set.

# Cross-validation

*Approach:*

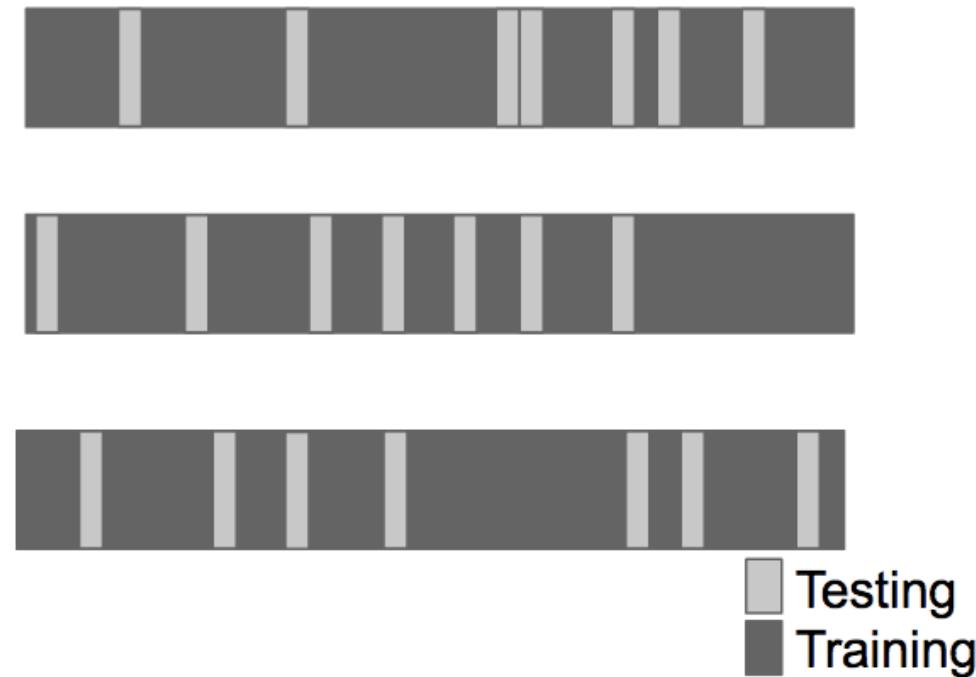
1. Use the training set
2. Split it into training/test sets
3. Build a model on the training set
4. Evaluate on the test set
5. Repeat and average the estimated errors

*Used for:*

1. Picking variables to include in a model
2. Picking the type of prediction function to use
3. Picking the parameters in the prediction function
4. Comparing different predictors

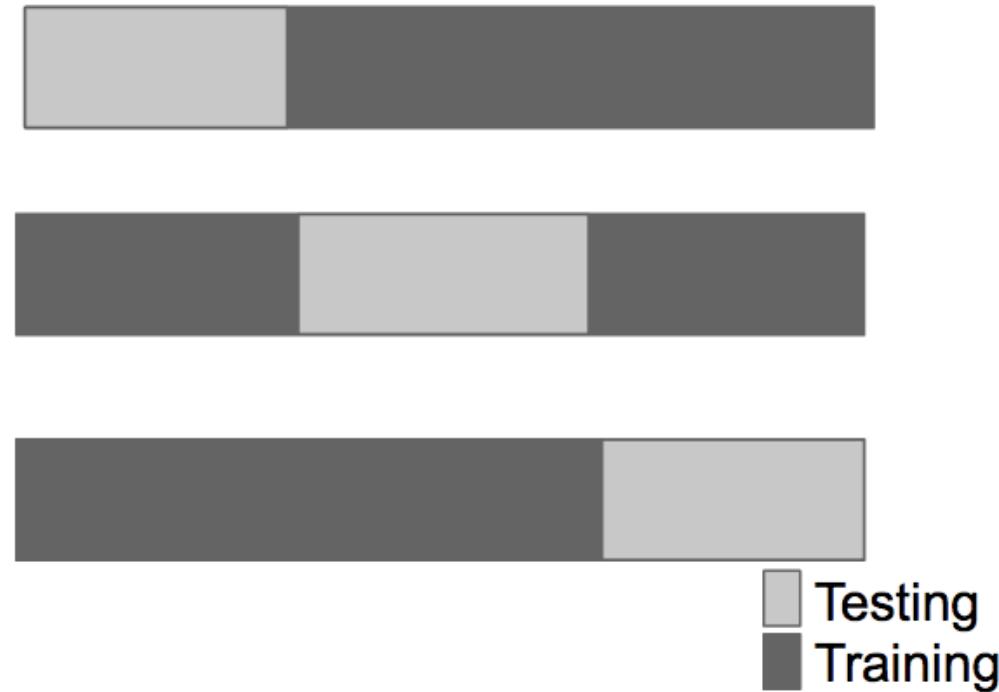
9/14

# Random subsampling



10/14

# K-fold



11/14

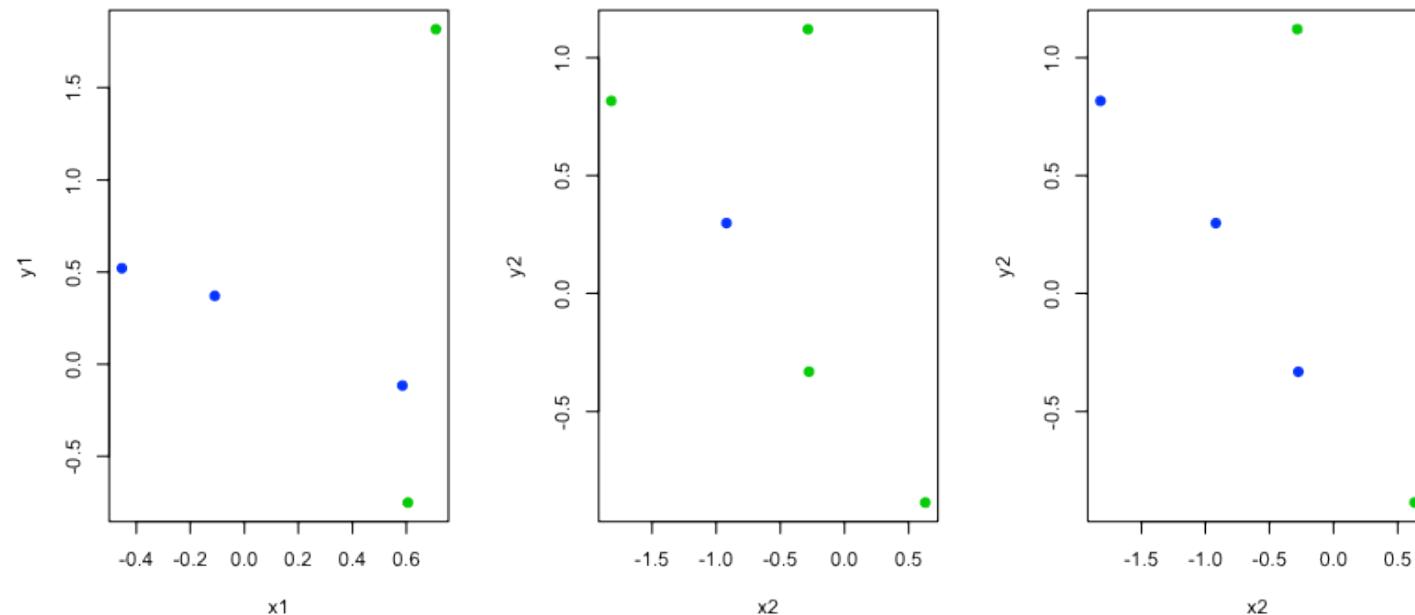
# Leave one out



12/14

# Example

```
y1 <- y[1:5]; x1 <- x[1:5]; z1 <- z[1:5]
y2 <- y[6:10]; x2 <- x[6:10]; z2 <- z[6:10];
zhat2 <- (y2 < 1) & (y2 > -0.5)
par(mfrow=c(1,3))
plot(x1,y1,col=(z1+3),pch=19); plot(x2,y2,col=(z2+3),pch=19); plot(x2,y2,col=(zhat2+3),pch=19)
```



13/14

# Notes and further resources

- The training and test sets must come from the same population.
- Sampling should be designed to mimic real patterns (e.g., sampling time chunks for time series)
- Cross validation estimates have variance - it is difficult to estimate how much
- [Cross validation in R](#)
- [cvTools](#)
- [boot](#)

# Data munging basics

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

# Recall Tidy Data

|    | A  | B          | C          | D          | E          | F         | G      | H | I | J | K | L | M | N | O | P |
|----|----|------------|------------|------------|------------|-----------|--------|---|---|---|---|---|---|---|---|---|
| #  | id | problem_id | subject_id | start      | end        | time_left | answer |   |   |   |   |   |   |   |   |   |
| 1  | 1  | 498        | 17         | 1307119989 | 1307120016 | 2369      | A      |   |   |   |   |   |   |   |   |   |
| 2  | 2  | 150        | 15         | 1307119991 | 1307120009 | 2376      | D      |   |   |   |   |   |   |   |   |   |
| 3  | 3  | 313        | 16         | 1307119994 | 1307120009 | 2376      | E      |   |   |   |   |   |   |   |   |   |
| 4  | 4  | 12         | 13         | 1307119995 | 1307120019 | 2366      | B      |   |   |   |   |   |   |   |   |   |
| 5  | 5  | 273        | 14         | 1307119996 | 1307120013 | 2357      | A      |   |   |   |   |   |   |   |   |   |
| 6  | 6  | 101        | 19         | 1307119997 | 1307120021 | 2356      | B      |   |   |   |   |   |   |   |   |   |
| 7  | 7  | 105        | 18         | 1307119998 | 1307120048 | 2337      | B      |   |   |   |   |   |   |   |   |   |
| 8  | 8  | 162        | 12         | 1307120004 | 1307120042 | 2343      | C      |   |   |   |   |   |   |   |   |   |
| 9  | 9  | 70         | 15         | 1307120011 | 1307120038 | 2347      | C      |   |   |   |   |   |   |   |   |   |
| 10 | 10 | 300        | 16         | 1307120012 | 1307120092 | 2293      | B      |   |   |   |   |   |   |   |   |   |
| 11 | 11 | 494        | 17         | 1307120021 | 1307120118 | 2310      | D      |   |   |   |   |   |   |   |   |   |
| 12 | 12 | 557        | 13         | 1307120021 | 1307120118 | 2357      | A      |   |   |   |   |   |   |   |   |   |
| 13 | 13 | 522        | 19         | 1307120025 | 1307120152 | 2233      | D      |   |   |   |   |   |   |   |   |   |
| 14 | 14 | 232        | 14         | 1307120030 | 1307120158 | 2227      | C      |   |   |   |   |   |   |   |   |   |
| 15 | 15 | 344        | 15         | 1307120041 | 1307120117 | 2268      | B      |   |   |   |   |   |   |   |   |   |
| 16 | 16 | 160        | 17         | 1307120079 | 1307120249 | 2136      | D      |   |   |   |   |   |   |   |   |   |
| 17 | 17 | 516        | 16         | 1307120080 | 1307120249 | 2216      | B      |   |   |   |   |   |   |   |   |   |
| 18 | 18 | 472        | 12         | 1307120119 | 1307120170 | 2115      | A      |   |   |   |   |   |   |   |   |   |
| 19 | 19 | 43         | 15         | 1307120122 | 1307120140 | 2245      | C      |   |   |   |   |   |   |   |   |   |
| 20 | 20 | 353        | 13         | 1307120144 | 1307120199 | 2186      | C      |   |   |   |   |   |   |   |   |   |
| 21 | 21 | 218        | 15         | 1307120152 | 1307120272 | 2113      | E      |   |   |   |   |   |   |   |   |   |
| 22 | 22 | 69         | 16         | 1307120153 | 1307120188 | 2197      | D      |   |   |   |   |   |   |   |   |   |
| 23 | 23 | 656        | 16         | 1307120154 | 1307120184 | 2080      | D      |   |   |   |   |   |   |   |   |   |
| 24 | 24 | 121        | 19         | 1307120253 | 1307120294 | 2091      | E      |   |   |   |   |   |   |   |   |   |
| 25 | 25 | 297        | 15         | 1307120277 | 1307120342 | 2043      | B      |   |   |   |   |   |   |   |   |   |
| 26 | 26 | 495        | 13         | 1307120281 | 1307120353 | 2032      | E      |   |   |   |   |   |   |   |   |   |
| 27 | 27 | 94         | 14         | 1307120286 | 1307120343 | 2042      | E      |   |   |   |   |   |   |   |   |   |
| 28 | 28 | 22         | 18         | 1307120313 | 1307120353 | 2020      | C      |   |   |   |   |   |   |   |   |   |
| 29 | 29 | 54         | 19         | 1307120310 | 1307120386 | 2000      | B      |   |   |   |   |   |   |   |   |   |
| 30 | 30 | 502        | 16         | 1307120323 | 1307120336 | 2049      | B      |   |   |   |   |   |   |   |   |   |
| 31 | 31 | 44         | 16         | 1307120339 | 1307120352 | 2033      | A      |   |   |   |   |   |   |   |   |   |
| 32 | 32 | 315        | 14         | 1307120348 | 1307120362 | 2023      | B      |   |   |   |   |   |   |   |   |   |
| 33 | 33 | 385        | 15         | 1307120352 | 1307120383 | 1832      | E      |   |   |   |   |   |   |   |   |   |
| 34 | 34 | 550        | 13         | 1307120354 | 1307120444 | 1910      | B      |   |   |   |   |   |   |   |   |   |
| 35 | 35 | 92         | 14         | 1307120368 | 1307120397 | 1988      | B      |   |   |   |   |   |   |   |   |   |
| 36 | 36 | 395        | 16         | 1307120377 | 1307120426 | 1959      | D      |   |   |   |   |   |   |   |   |   |
| 37 | 37 | 267        | 17         | 1307120382 | 1307120515 | 1870      | E      |   |   |   |   |   |   |   |   |   |
| 38 | 38 | 257        | 14         | 1307120401 | 1307120427 | 1958      | C      |   |   |   |   |   |   |   |   |   |
| 39 | 39 | 312        | 19         | 1307120407 | 1307120548 | 1837      | D      |   |   |   |   |   |   |   |   |   |
| 40 | 40 | 321        | 18         | 1307120431 | 1307120449 | 1936      | A      |   |   |   |   |   |   |   |   |   |
| 41 | 41 | 270        | 16         | 1307120437 | 1307120410 | 1877      | A      |   |   |   |   |   |   |   |   |   |

1. Each variable forms a column
2. Each observation forms a row
3. Each table/file stores data about one kind of observation (e.g. people/hospitals).

<http://vita.had.co.nz/papers/tidy-data.pdf>

[Leek, Taub, and Pineda 2011 PLoS One](#)

2/26

# Where we would like to be

- Tidy data refers to the shape of the data
  - Variables in columns
  - Observations in rows
  - Tables holding elements of only one kind
- Plus
  - Column names are easy to use and informative
  - Row names are easy to use and informative
  - Obvious mistakes in the data have been removed
  - Variable values are internally consistent
  - Appropriate transformed variables have been added

# A partial list of munging operations

- Fix variable names
- Create new variables
- Merge data sets
- Reshape data sets
- Deal with missing data
- Take transforms of variables
- Check on and remove inconsistent values

**These steps must be recorded**

**90% of your effort will often be spent here**

# A partial list of munging operations

- Fix variable names
- Create new variables
- Merge data sets
- Reshape data sets
- Deal with missing data
- Take transforms of variables
- Check on and remove inconsistent values

# Fixing character vectors - tolower(), toupper()

```
cameraData <- read.csv("./data/cameras.csv")
names(cameraData)

[1] "address"      "direction"     "street"       "crossStreet"
[5] "intersection" "Location.1"
```

```
tolower(names(cameraData))
```

```
[1] "address"      "direction"     "street"       "crossstreet"
[5] "intersection" "location.1"
```

# Fixing character vectors - strsplit()

- Good for automatically splitting variable names
- Important parameters:  $x$ ,  $split$

```
splitNames = strsplit(names(cameraData), "\\.")  
splitNames[[5]]
```

```
[1] "intersection"
```

```
splitNames[[6]]
```

```
[1] "Location" "1"
```

7/26

# Fixing character vectors - sapply()

- Applies a function to each element in a vector or list
- Important parameters:  $X, FUN$

```
splitNames[[6]][1]
```

```
[1] "Location"
```

```
firstElement <- function(x){x[1]}  
sapply(splitNames, firstElement)
```

```
[1] "address"        "direction"      "street"        "crossStreet"  
[5] "intersection"  "Location"
```

# Peer review experiment data

- Data on submissions/reviews in an experiment

The screenshot shows a PLOS ONE article page. At the top, there is an advertisement for 'Simplify your research with automatic and continuous dosing' featuring medical syringes and capsules. The main header reads 'PLOS ONE'. Below it, there are tabs for 'Articles', 'For Authors', and 'About Us', along with a search bar and user account links ('plos.org', 'create account', 'sign in'). The article title is 'Cooperation between Referees and Authors Increases Peer Review Accuracy' by Jeffrey T. Leek, Margaret A. Taub, and Fernando J. Pineda. It is marked as an 'OPEN ACCESS' and 'PEER-REVIEWED' research article. To the right of the title, metrics are displayed: 6,497 views, 2 citations, 61 academic bookmarks, and 108 social shares. Below the title, there is a section titled 'Article' with a dropdown arrow, followed by 'About the Authors', 'Metrics', 'Comments', and 'Related Content'. The 'Metrics' section contains several small figures, including a diagram comparing 'Closed/Private' and 'Open/Public' review types, and three panels showing network graphs of referee interactions. The 'Comments' section includes a link to 'Media Coverage of This Article' posted by 'PLoS\_ONE\_Group'. On the right side, there are buttons for 'Download', 'Print', and 'Share'.

<http://www.plosone.org/article/info:doi/10.1371/journal.pone.0026895>

9/26

# Peer review data

```
fileUrl1 <- "https://dl.dropbox.com/u/7710864/data/reviews-apr29.csv"
fileUrl2 <- "https://dl.dropbox.com/u/7710864/data/solutions-apr29.csv"
download.file(fileUrl1, destfile = "./data/reviews.csv", method = "curl")
download.file(fileUrl2, destfile = "./data/solutions.csv", method = "curl")
reviews <- read.csv("./data/reviews.csv"); solutions <- read.csv("./data/solutions.csv")
head(reviews, 2)
```

```
  id solution_id reviewer_id      start      stop time_left accept
1  1          3    1304095698 1304095758      1754       1
2  2          4    1304095188 1304095206      2306       1
```

```
head(solutions, 2)
```

```
  id problem_id subject_id      start      stop time_left answer
1  1        156    1304095119 1304095169      2343       B
2  2        269    1304095119 1304095183      2329       C
```

10/26

# Fixing character vectors - sub(), gsub()

- Important parameters: *pattern, replacement, x*

```
names(reviews)
```

```
[1] "id"           "solution_id" "reviewer_id" "start"  
[5] "stop"         "time_left"   "accept"
```

```
sub("_","",names(reviews),)
```

```
[1] "id"           "solutionid" "reviewerid" "start"       "stop"  
[6] "timeleft"    "accept"
```

# Fixing character vectors - sub(), gsub()

```
testName <- "this_is_a_test"  
sub("_","",testName)
```

```
[1] "thisis_a_test"
```

```
gsub("_","",testName)
```

```
[1] "thisisatest"
```

12/26

# Quantitative variables in ranges - - cut()

- Important parameters:  $x, breaks$

```
reviews$time_left[1:10]
```

```
[1] 1754 2306 2192 2089 2043 1999 2130    NA 1899 2024
```

```
timeRanges <- cut(reviews$time_left, seq(0, 3600, by=600))  
timeRanges[1:10]
```

```
[1] (1.2e+03,1.8e+03] (1.8e+03,2.4e+03] (1.8e+03,2.4e+03]  
[4] (1.8e+03,2.4e+03] (1.8e+03,2.4e+03] (1.8e+03,2.4e+03]  
[7] (1.8e+03,2.4e+03] <NA>                 (1.8e+03,2.4e+03]  
[10] (1.8e+03,2.4e+03]  
6 Levels: (0,600] (600,1.2e+03] ... (3e+03,3.6e+03]
```

# Quantitative variables in ranges - - cut()

```
class(timeRanges)
```

```
[1] "factor"
```

```
table(timeRanges,useNA="ifany")
```

```
timeRanges
```

|                   | (0,600] | (600,1.2e+03]   | (1.2e+03,1.8e+03] |    |
|-------------------|---------|-----------------|-------------------|----|
|                   | 30      | 32              | 25                |    |
| (1.8e+03,2.4e+03] |         | (2.4e+03,3e+03] | (3e+03,3.6e+03]   |    |
|                   | 28      | 0               | 0                 |    |
| <NA>              |         |                 |                   | 84 |

# Quantitative variables in ranges - cut2() {Hmisc}

```
library(Hmisc)
timeRanges<- cut2(reviews$time_left,g=6)
table(timeRanges,useNA="ifany")

timeRanges
[ 22, 384) [ 384, 759) [ 759,1150) [1150,1496) [1496,1909)
      20          19          19          19          19
[1909,2306]       <NA>
      19          84
```

# Adding an extra variable

```
timeRanges<- cut2(reviews$time_left,g=6)  
reviews$timeRanges <- timeRanges  
head(reviews,2)
```

|   | id | solution_id | reviewer_id | start | stop       | time_left  | accept |   |
|---|----|-------------|-------------|-------|------------|------------|--------|---|
| 1 | 1  |             | 3           | 27    | 1304095698 | 1304095758 | 1754   | 1 |
| 2 | 2  |             | 4           | 22    | 1304095188 | 1304095206 | 2306   | 1 |

```
timeRanges  
1 [1496,1909)  
2 [1909,2306]
```

16/26

# Merging data - merge()

- Merges data frames
- Important parameters: *x,y,by,by.x,by.y,all*

```
names(reviews)
```

```
[1] "id"          "solution_id" "reviewer_id" "start"  
[5] "stop"        "time_left"   "accept"      "timeRanges"
```

```
names(solutions)
```

```
[1] "id"          "problem_id"  "subject_id" "start"       "stop"  
[6] "time_left"   "answer"
```

# Merging data - merge()

```
mergedData <- merge(reviews,solutions,all=TRUE)  
head(mergedData)
```

|   |   | id          | start      | stop       | time_left | solution_id | reviewer_id | accept |
|---|---|-------------|------------|------------|-----------|-------------|-------------|--------|
| 1 | 1 | 1304095119  | 1304095169 |            | 2343      | NA          | NA          | NA     |
| 2 | 1 | 1304095698  | 1304095758 |            | 1754      | 3           | 27          | 1      |
| 3 | 2 | 1304095119  | 1304095183 |            | 2329      | NA          | NA          | NA     |
| 4 | 2 | 1304095188  | 1304095206 |            | 2306      | 4           | 22          | 1      |
| 5 | 3 | 1304095127  | 1304095146 |            | 2366      | NA          | NA          | NA     |
| 6 | 3 | 1304095276  | 1304095320 |            | 2192      | 5           | 28          | 1      |
|   |   | timeRanges  | problem_id | subject_id | answer    |             |             |        |
| 1 |   | <NA>        | 156        | 29         | B         |             |             |        |
| 2 |   | [1496,1909) | NA         | NA         | <NA>      |             |             |        |
| 3 |   | <NA>        | 269        | 25         | C         |             |             |        |
| 4 |   | [1909,2306] | NA         | NA         | <NA>      |             |             |        |
| 5 |   | <NA>        | 34         | 22         | C         |             |             |        |
| 6 |   | [1909,2306] | NA         | NA         | <NA>      |             |             |        |

# Merging data - merge()

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

|   | solution_id | id | reviewer_id | start.x | stop.x     | time_left.x |      |
|---|-------------|----|-------------|---------|------------|-------------|------|
| 1 |             | 1  | 4           | 26      | 1304095267 | 1304095423  | 2089 |
| 2 |             | 2  | 6           | 29      | 1304095471 | 1304095513  | 1999 |
| 3 |             | 3  | 1           | 27      | 1304095698 | 1304095758  | 1754 |

```
reviews[1,1:6]
```

|   | id | solution_id | reviewer_id | start | stop       | time_left  |      |
|---|----|-------------|-------------|-------|------------|------------|------|
| 1 | 1  |             | 3           | 27    | 1304095698 | 1304095758 | 1754 |

# Sorting values - sort()

- Important parameters:  $x$ , *decreasing*

```
mergedData2$reviewer_id[1:10]
```

```
[1] 26 29 27 22 28 22 29 23 25 29
```

```
sort(mergedData2$reviewer_id)[1:10]
```

```
[1] 22 22 22 22 22 22 22 22 22 22
```

20/26

# Ordering values - `order()`

- Important parameters: *list of variables to order, na.last, decreasing*

```
mergedData2$reviewer_id[1:10]
```

```
[1] 26 29 27 22 28 22 29 23 25 29
```

```
order(mergedData2$reviewer_id)[1:10]
```

```
[1] 4 6 14 22 23 24 27 32 37 39
```

```
mergedData2$reviewer_id[order(mergedData2$reviewer_id)]
```

```
[1] 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22  
[22] 22 22 22 22 22 22 22 23 23 23 23 23 23 23 23 23 23 23 23  
[43] 23 23 23 23 23 23 23 23 23 23 23 23 23 23 24 24 24 24 24
```

21/26

# Reordering a data frame

```
head(mergedData2[,1:6],3)
```

```
  solution_id id reviewer_id      start.x      stop.x time_left.x
1          1   4           26 1304095267 1304095423        2089
2          2   6           29 1304095471 1304095513        1999
3          3   1           27 1304095698 1304095758        1754
```

```
sortedData <- mergedData2[order(mergedData2$reviewer_id),]
head(sortedData[,1:6],3)
```

```
  solution_id id reviewer_id      start.x      stop.x time_left.x
4          4   2           22 1304095188 1304095206        2306
6          6  16           22 1304095303 1304095471        2041
14         14  12           22 1304095280 1304095301        2211
```

# Reordering by multiple variables

```
head(mergedData2[,1:6],3)
```

|   | solution_id | id | reviewer_id | start.x | stop.x     | time_left.x |      |
|---|-------------|----|-------------|---------|------------|-------------|------|
| 1 |             | 1  | 4           | 26      | 1304095267 | 1304095423  | 2089 |
| 2 |             | 2  | 6           | 29      | 1304095471 | 1304095513  | 1999 |
| 3 |             | 3  | 1           | 27      | 1304095698 | 1304095758  | 1754 |

```
sortedData <- mergedData2[order(mergedData2$reviewer_id,mergedData2$id),]  
head(sortedData[,1:6],3)
```

|    | solution_id | id | reviewer_id | start.x | stop.x     | time_left.x |      |
|----|-------------|----|-------------|---------|------------|-------------|------|
| 4  |             | 4  | 2           | 22      | 1304095188 | 1304095206  | 2306 |
| 14 |             | 14 | 12          | 22      | 1304095280 | 1304095301  | 2211 |
| 6  |             | 6  | 16          | 22      | 1304095303 | 1304095471  | 2041 |

# Reshaping data - example

```
misShaped <- as.data.frame(matrix(c(NA,5,1,4,2,3),byrow=TRUE,nrow=3))
names(misShaped) <- c("treatmentA","treatmentB")
misShaped$people <- c("John","Jane","Mary")
misShaped
```

|   | treatmentA | treatmentB | people |
|---|------------|------------|--------|
| 1 | NA         | 5          | John   |
| 2 | 1          | 4          | Jane   |
| 3 | 2          | 3          | Mary   |

<http://vita.had.co.nz/papers/tidy-data.pdf>

# Reshaping data - melt()

- Important parameters: *id.vars*, *measure.vars*, *variable.name*

```
melt(misShaped,id.vars="people",variable.name="treatment",value.name="value")
```

|   | people | treatment  | value |
|---|--------|------------|-------|
| 1 | John   | treatmentA | NA    |
| 2 | Jane   | treatmentA | 1     |
| 3 | Mary   | treatmentA | 2     |
| 4 | John   | treatmentB | 5     |
| 5 | Jane   | treatmentB | 4     |
| 6 | Mary   | treatmentB | 3     |

# More resources

- [Tidy data and tidy tools](#)
- Andrew Jaffe's [Data Cleaning Lecture](#)
- Hadley Wickham on [regular expressions](#)
- Long, painful experience :-)

# Data Resources

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

# Open Government Data (U.S.)

The screenshot shows the Data.gov homepage. At the top, there's a banner for the American Community Survey (2007-2011) featuring a diverse group of people. To the right, a sidebar lists "Latest Datasets" with links to various federal campaign datasets. Below the banner, there are three main sections: "DATA AND TOOLS" (with a map tool thumbnail), "COMMUNITIES" (with a world map thumbnail), and "OPEN GOVERNMENT DATA" (with an American flag thumbnail). Each section has a brief description and a list of related datasets.

**Latest Datasets**

- Combined Federal Campaign, CFC, 2009
- Combined Federal Campaign, CFC, 2010
- Combined Federal Campaign, CFC, 2009-...
- Combined Federal Campaign, CFC, 2011
- Gravesite locations of Veterans and...

**DATA AND TOOLS**

- 378,529 raw and geospatial datasets
- 1,264 data tools
- 236 citizen-developed data tools

**COMMUNITIES**

Come explore, discuss, meet others in the same field, and develop the data and apps in the community that you care about. Join in the

**OPEN GOVERNMENT DATA**

First open source code released for the Open Government Platform delivered by the governments of India and the U.S. [Find out more](#) and then

<http://www.data.gov/>

2/11

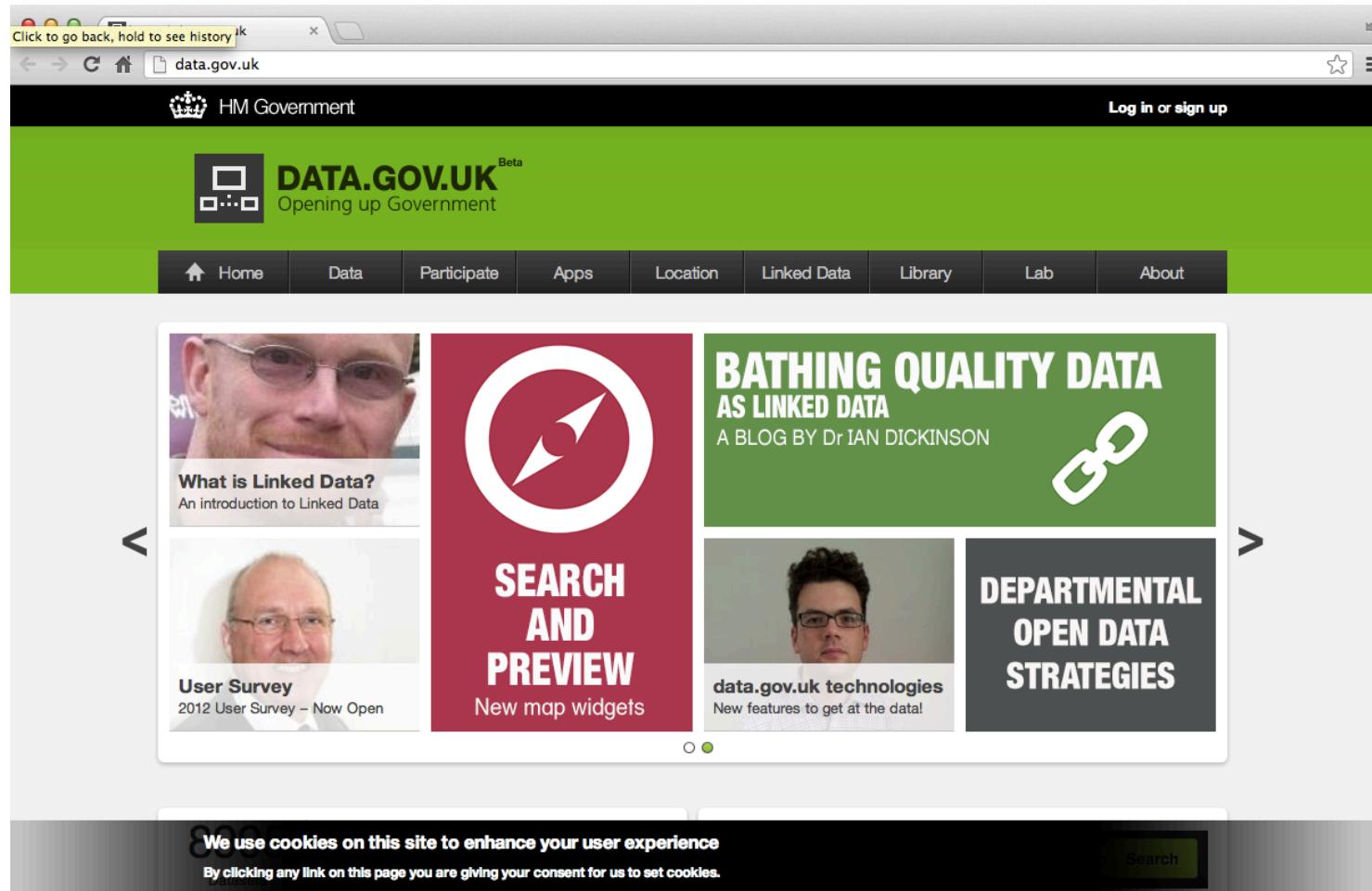
# Open Government Data (France)

The screenshot shows the homepage of data.gouv.fr. At the top, there is a navigation bar with links for ACCUEIL, DONNÉES, PRODUCTEURS, ARTICLES, LICENCE OUVERTE, COMMUNAUTÉ, S'IDENTIFIER, and A propos. The main title "data.gouv.fr" is displayed with "BETA" in red. Below the title, there is a logo for the French Republic and a link to "Premier ministre". A search bar contains the placeholder "Rechercher une donnée" and a "RECHERCHER" button. Below the search bar is a "RECHERCHE AVANCÉE" link. The main content area features a large image of a laptop displaying various charts and graphs. Text on the image reads "LA MISE À DISPOSITION DES DONNÉES PUBLIQUES SUR LA PLATEFORME DATA.GOUV.FR" and "Etalab, mission interministérielle placée sous l'autorité du Premier ministre, coordonne depuis [...]" with a [...] icon. Below the image are several small square icons. To the right of the image is a sidebar with sections for "SUGGESTION DE RECHERCHE" (including "LES PLUS RECHERCHÉS" like "Principales infractions en vigueur", "Résultats des élections européennes", etc.) and "REPÈRES" (listing "353 226 jeux de données publiques et plus sur data.gouv.fr"). There is also a section for "ETALAB" with a "Suivez notre actualité sur le blog" link and a Twitter feed for "@etalab" with 4 012 abonnés.

<http://www.data.gouv.fr/>

3/11

# Open Government Data (UK)



<http://data.gov.uk/>

4/11

# Gapminder

The screenshot shows a web browser window displaying the Gapminder website at [www.gapminder.org/data/](http://www.gapminder.org/data/). The page title is "DATA" and the main heading is "Data in Gapminder World". Below the heading is a sub-headline: "The table below lists all indicators displayed in Gapminder World. Click the name of the indicator or the data provider to access information about the indicator and a link to the data provider. Indicators labeled 'Various sources' are compiled by Gapminder. They can be reused freely but please attribute Gapminder." A sidebar on the right has a button labeled "Ask a question". The main content area contains a table titled "List of indicators in Gapminder World" with columns for Indicator name, Data provider, Category, Subcategory, Download, View, and Visualize.

| Indicator name                               | Data provider                     | Category   | Subcategory                | Download | View | Visualize |
|--|-----------------------------------|------------|----------------------------|----------|------|-----------|
| Adults with HIV (%), age 15-49               | Based on UNAIDS                   | Health     | HIV                        |          |      |           |
| Age at 1st marriage (women)                  | Various sources                   | Population |                            |          |      |           |
| Aged 15+ employment rate (%)                 | International Labour Organization | Work       | Employment rate            |          |      |           |
| Aged 15+ labour force participation rate (%) | International Labour Organization | Work       | Labour force participation |          |      |           |
| Aged 15+ unemployment rate (%)               | International Labour Organization | Work       | Unemployment               |          |      |           |
| Aged 15-24 employment rate (%)               | International Labour Organization | Work       | Employment rate            |          |      |           |

<http://www.gapminder.org/>

5/11

# More open government data (possibly overlapping)

- <http://opengovernmentdata.org/data/catalogues/>
- <http://wiki.civiccommons.org/Initiatives>
- [List of cities/states with open data](#)

6/11

# Survey data from the United States

The screenshot shows a web browser window with the title bar "asdfree by anthony damico". The address bar displays "www.asdfree.com". The main content area has a header "analyze survey data for free". Below the header is a navigation bar with links: "about / faq", "main code repository", "latest releases", "rss", "ajdamico@gmail.com", "twotorials", and "r-bloggers".

The main content includes:

- A section titled "reproducible survey analysis syntax from a website that's easy to type." containing a link to "analyze the health and retirement study (hrs) with r".
- A section titled "AVAILABLE DATA" listing various survey datasets:
  - american community survey (acs)
  - area resource file (arf)
  - basic stand alone medicare claims public use files (baspubfs)
  - behavioral risk factor surveillance system (brfss)
  - consumer expenditure survey (ce)
  - current population survey (cps)
  - general social survey (gss)
  - health and retirement study (hrs)
  - medical expenditure panel survey (meps)
  - national health and nutrition examination survey (nhanes)
  - national health interview survey (nhis)
  - national study on drug use and health (nsduh)
- A section titled "METHODS" with a link to "why and how to install monetdb with r on windows".

At the bottom of the page, there is a footer with the text "enter your email address for updates:" followed by a text input field containing "www.asdfree.com/2013/01/analyze-health-and-retirement-study-hrs.html". Below the input field is the text "1992 - 2010 download HRS microdata.R".

<http://www.asdfree.com/>

7/11

# Infochimps Marketplace

The screenshot shows a web browser displaying the Infochimps Data Marketplace. The page has a dark header with the Infochimps logo, navigation links for Home, Solutions, Platform, Resources, Community, Company, Blog, and a prominent orange 'Request a Demo' button. Below the header is a search bar with the placeholder 'Search for data' and a magnifying glass icon, along with links for 'API Documentation' and 'Log in'. The main content area features a large title 'Data Marketplace' and a subtext: 'With thousands of public and proprietary data sets, our data marketplace is a great resource of experimental and sample data for data scientists and developers.' Two sections are highlighted: 'Geo APIs' (with a globe icon) and 'Social APIs' (with a speech bubble icon). Each section includes a brief description and a 'Learn more' link. Below these sections is a 'Top Tags' section containing a grid of 25 tags: locations, list, business, geo, location, firebase, places, phone, stores, hours, business-information, phone-numbers, place-names, store-hours, agadata, population, census, government, demographics, america, health, maps, state, care, toxic, facilities, geonames, facility, release, school, community, and a 'More...' link.

<http://www.infochimps.com/marketplace>

8/11

# Kaggle

The screenshot shows the Kaggle homepage. At the top, there's a navigation bar with links for Sign In, Sign Up, About, Hosting Center, All Competitions, Users, Forums, Wiki, Blog, and Data Science Jobs. Below the navigation is a main heading "What can data science do for you?". There are two main sections: "Participate in competitions" and "Create a competition". The "Participate in competitions" section features a "Join as a participant" button and a link for "[Need convincing?]" The "Create a competition" section has a "Learn more about hosting" button. To the right, there's a sidebar titled "Host a competition for..." with options for Analytics, Data Exploration, Recruitment, and Education. Below that is a box for the GE Quests competition, specifically the GE Flight Quest and GE Hospital Quest. The GE Flight Quest has a deadline of 18 days from now, 131 teams, and a prize of \$250,000. The GE Hospital Quest has a deadline of 20 days from now, 131 teams, and a prize of \$100,000. At the bottom, there's a "Featured Competitions" section.

<http://www.kaggle.com/>

9/11

# More specialized collections

- [Hilary Mason's research data](#)
- [Stanford Large Network Data](#)
- [UCI Machine Learning](#)
- [KDD Nuggets Datasets](#)
- [CMU Statlib](#)
- [Gene expression omnibus](#)
- [ArXiv Data](#)

# Some API's

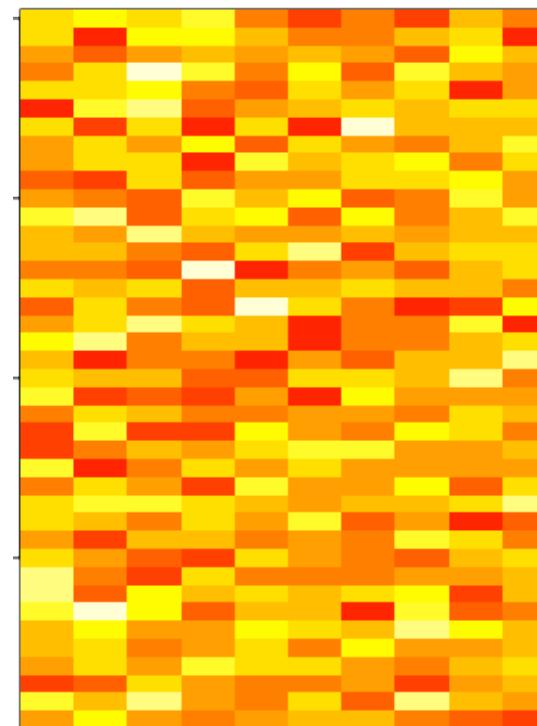
- [twitter](#) and [twitteR](#) package
- [figshare](#) and [rfigshare](#)
- [PLoS](#) and [rplos](#)
- [rOpenSci](#)

# Principal components analysis and singular value decomposition

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

# Matrix data

```
set.seed(12345); par(mar=rep(0.2,4))
dataMatrix <- matrix(rnorm(400), nrow=40)
image(1:10, 1:40, t(dataMatrix)[, nrow(dataMatrix):1])
```



2/25

# Cluster the data

```
par(mar=rep(0.2,4))  
heatmap(dataMatrix)
```

3/25

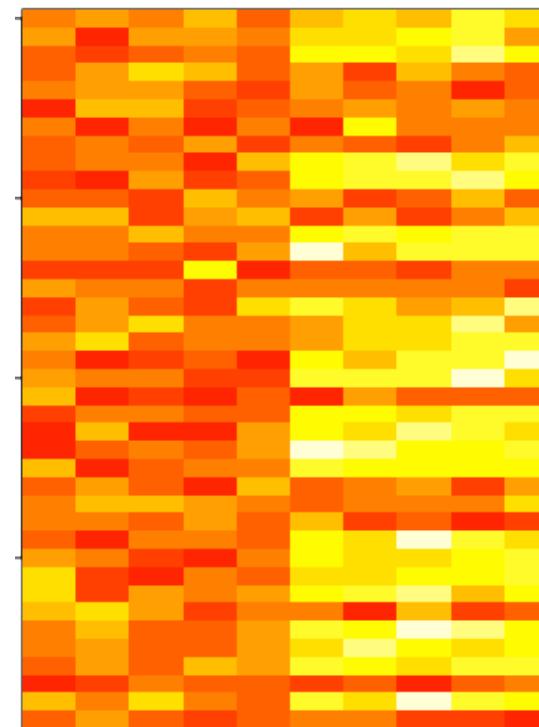
# What if we add a pattern?

```
set.seed(678910)
for(i in 1:40){
  # flip a coin
  coinFlip <- rbinom(1,size=1,prob=0.5)
  # if coin is heads add a common pattern to that row
  if(coinFlip){
    dataMatrix[i,] <- dataMatrix[i,] + rep(c(0,3),each=5)
  }
}
```

4/25

# What if we add a pattern? - the data

```
par(mar=rep(0.2,4))  
image(1:10,1:40,t(dataMatrix)[,nrow(dataMatrix):1])
```



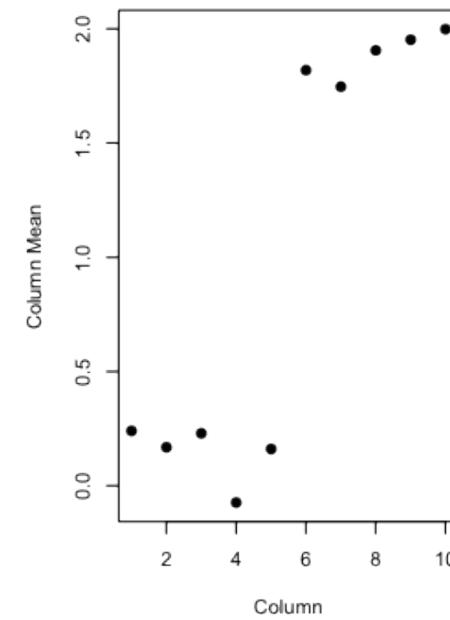
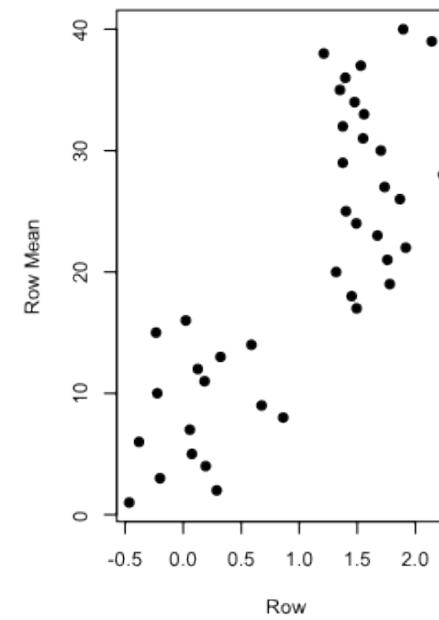
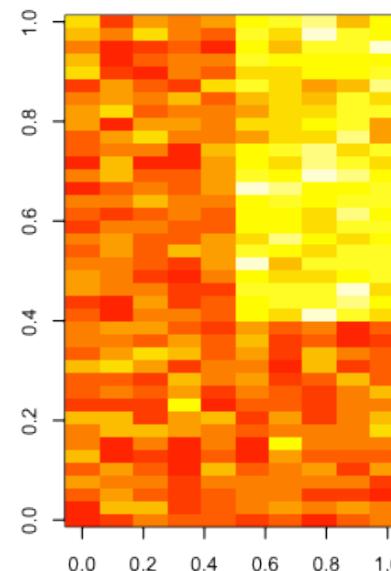
# What if we add a pattern? - the clustered data

```
par(mar=rep(0.2,4))  
heatmap(dataMatrix)
```

6/25

# Patterns in rows and columns

```
hh <- hclust(dist(dataMatrix)); dataMatrixOrdered <- dataMatrix[hh$order, ]
par(mfrow=c(1,3))
image(t(dataMatrixOrdered)[,nrow(dataMatrixOrdered):1])
plot(rowMeans(dataMatrixOrdered), 40:1, xlab="Row", ylab="Row Mean", pch=19)
plot(colMeans(dataMatrixOrdered), xlab="Column", ylab="Column Mean", pch=19)
```



7/25

# Related problems

You have multivariate variables  $X_1, \dots, X_n$  so  $X_1 = (X_{11}, \dots, X_{1m})$

- Find a new set of multivariate variables that are uncorrelated and explain as much variance as possible.
- If you put all the variables together in one matrix, find the best matrix created with fewer variables (lower rank) that explains the original data.

The first goal is **statistical** and the second goal is **data compression**.

# Related solutions - PCA/SVD

## SVD

If  $X$  is a matrix with each variable in a column and each observation in a row then the SVD is a "matrix decomposition"

$$X = UDV^T$$

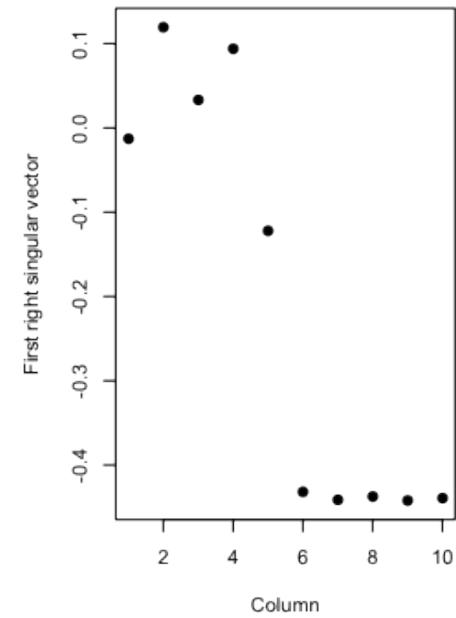
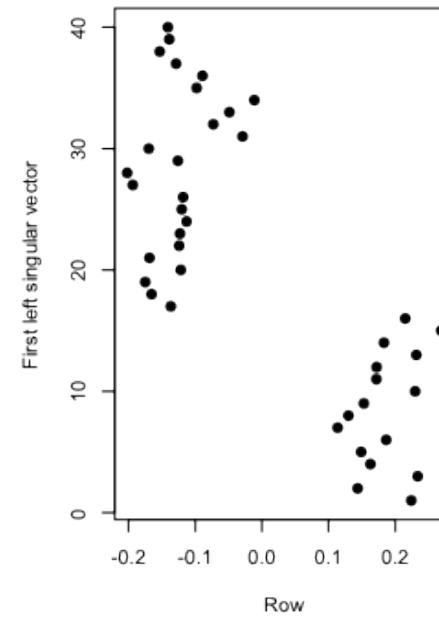
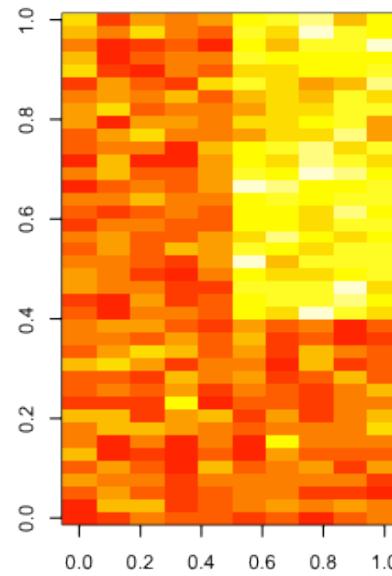
where the columns of  $U$  are orthogonal (left singular vectors), the columns of  $V$  are orthogonal (right singular vectors) and  $D$  is a diagonal matrix (singular values).

## PCA

The principal components are equal to the right singular values if you first scale (subtract the mean, divide by the standard deviation) the variables.

# Components of the SVD - u and v

```
svd1 <- svd(scale(dataMatrixOrdered))
par(mfrow=c(1,3))
image(t(dataMatrixOrdered)[,nrow(dataMatrixOrdered):1])
plot(svd1$u[,1],xlab="Row",ylab="First left singular vector",pch=19)
plot(svd1$v[,1],xlab="Column",ylab="First right singular vector",pch=19)
```



10/25

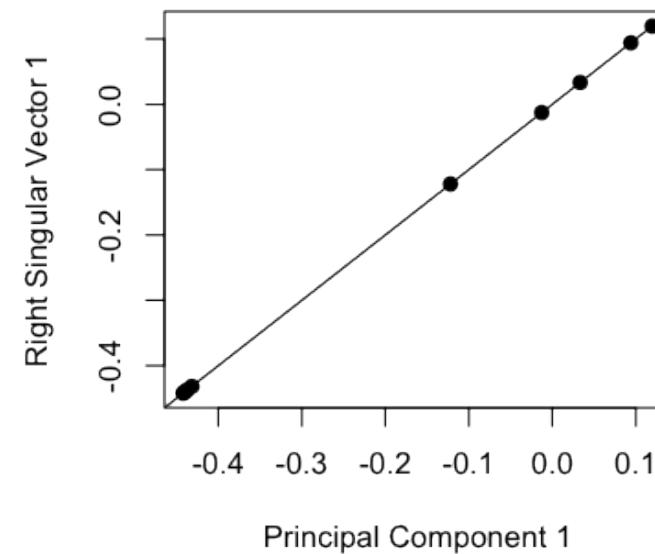
# Components of the SVD - d and variance explained

```
svd1 <- svd(scale(dataMatrixOrdered))
par(mfrow=c(1,2))
plot(svd1$d,xlab="Column",ylab="Singluar value",pch=19)
plot(svd1$d^2/sum(svd1$d^2),xlab="Column",ylab="Percent of variance explained",pch=19)
```

11/25

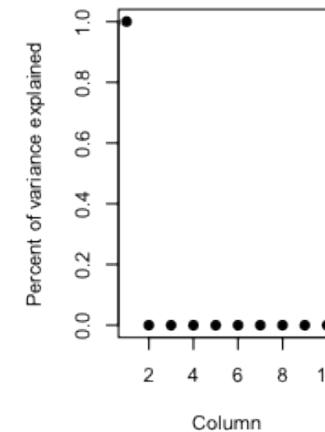
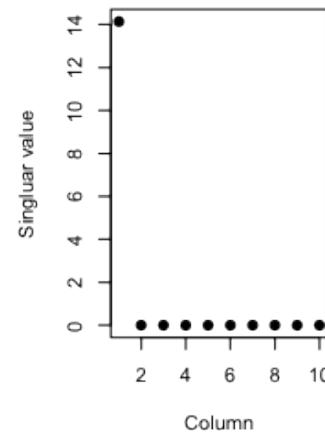
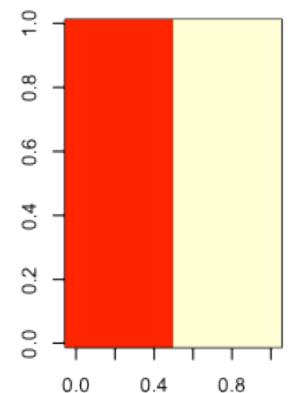
# Relationship to principal components

```
svd1 <- svd(scale(dataMatrixOrdered))
pca1 <- prcomp(dataMatrixOrdered,scale=TRUE)
plot(pca1$rotation[,1],svd1$v[,1],pch=19,xlab="Principal Component 1",ylab="Right Singular Vector 1"
abline(c(0,1))
```



# Components of the SVD - variance explained

```
constantMatrix <- dataMatrixOrdered*0
for(i in 1:dim(dataMatrixOrdered)[1]) {constantMatrix[i,] <- rep(c(0,1),each=5)}
svd1 <- svd(constantMatrix)
par(mfrow=c(1,3))
image(t(constantMatrix)[,nrow(constantMatrix):1])
plot(svd1$d,xlab="Column",ylab="Singluar value",pch=19)
plot(svd1$d^2/sum(svd1$d^2),xlab="Column",ylab="Percent of variance explained",pch=19)
```

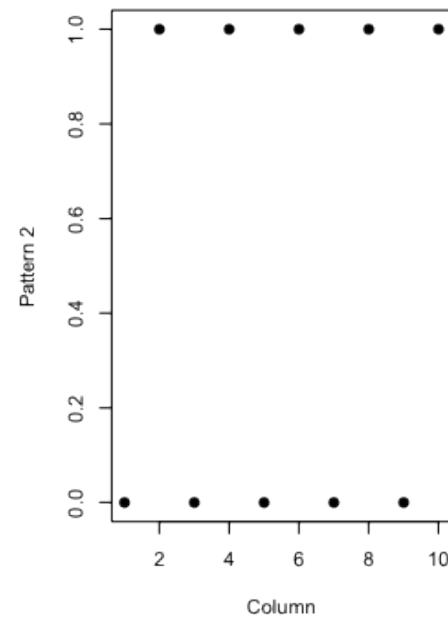
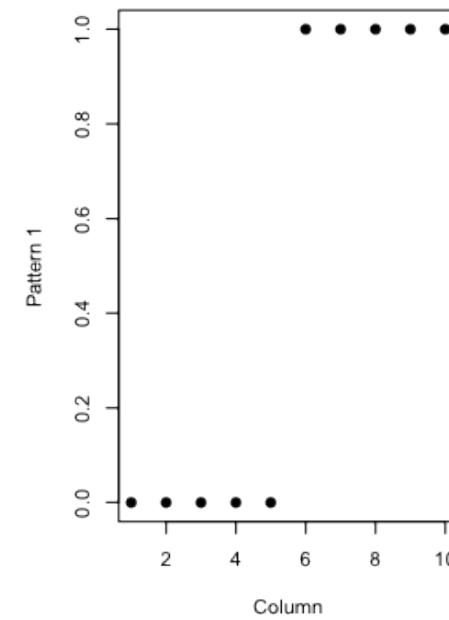
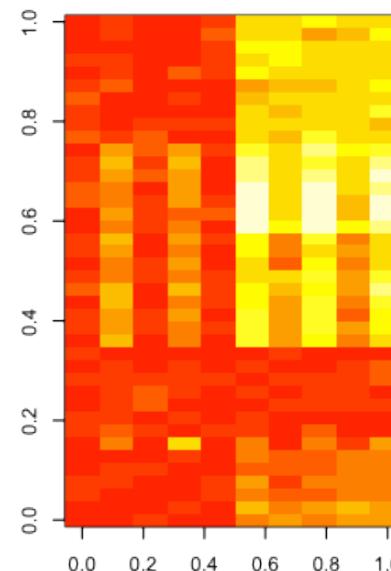


# What if we add a second pattern?

```
set.seed(678910)
for(i in 1:40){
  # flip a coin
  coinFlip1 <- rbinom(1,size=1,prob=0.5)
  coinFlip2 <- rbinom(1,size=1,prob=0.5)
  # if coin is heads add a common pattern to that row
  if(coinFlip1){
    dataMatrix[i,] <- dataMatrix[i,] + rep(c(0,5),each=5)
  }
  if(coinFlip2){
    dataMatrix[i,] <- dataMatrix[i,] + rep(c(0,5),5)
  }
}
hh <- hclust(dist(dataMatrix)); dataMatrixOrdered <- dataMatrix[hh$order, ]
```

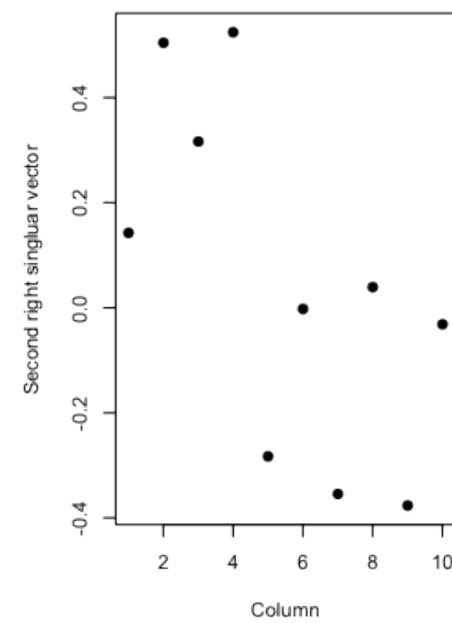
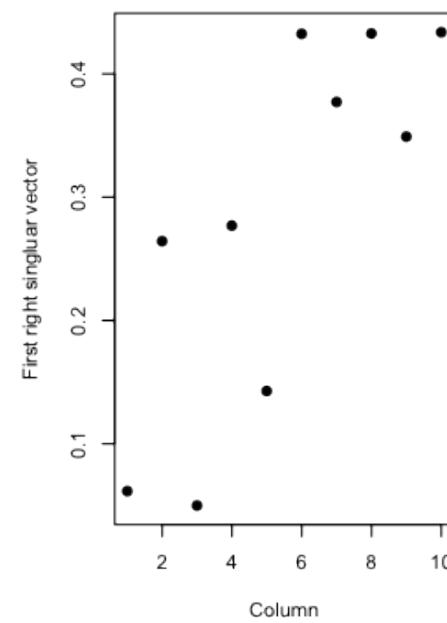
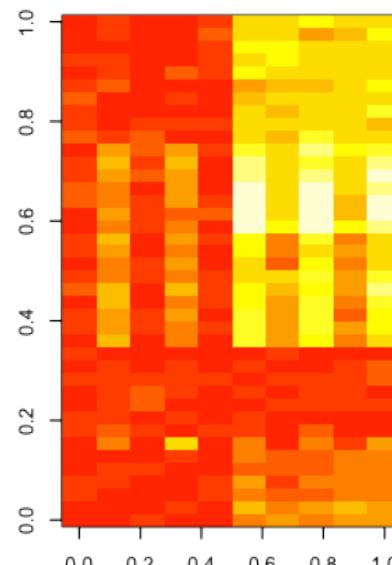
# Singular value decomposition - true patterns

```
svd2 <- svd(scale(dataMatrixOrdered))
par(mfrow=c(1,3))
image(t(dataMatrixOrdered)[,nrow(dataMatrixOrdered):1])
plot(rep(c(0,1),each=5),pch=19,xlab="Column",ylab="Pattern 1")
plot(rep(c(0,1),5),pch=19,xlab="Column",ylab="Pattern 2")
```



# v and patterns of variance in rows

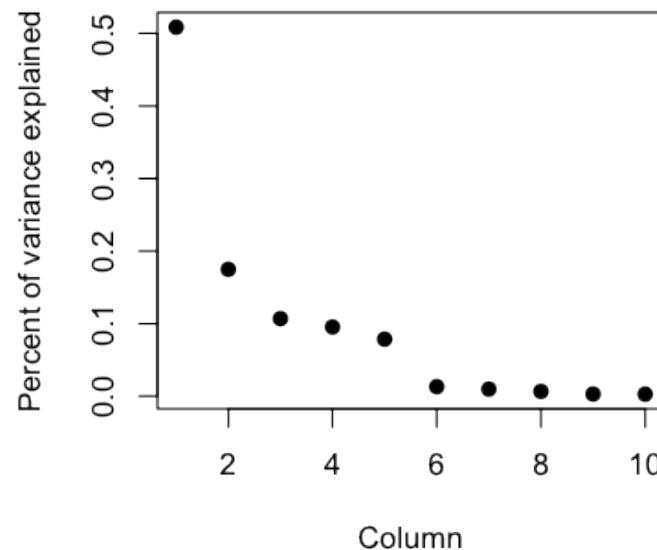
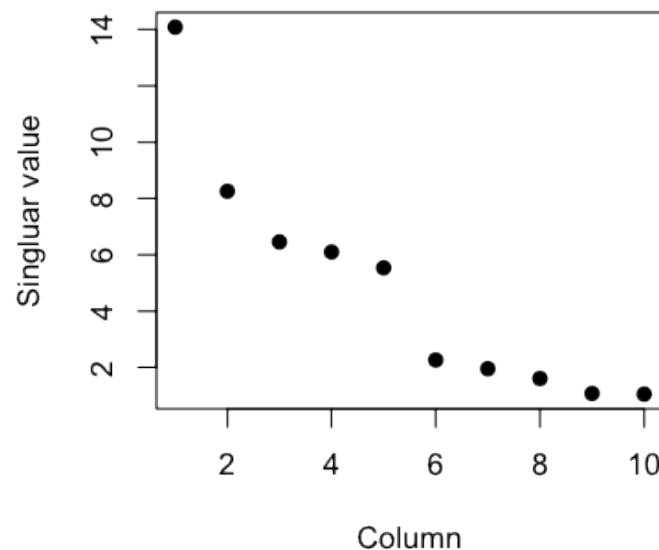
```
svd2 <- svd(scale(dataMatrixOrdered))
par(mfrow=c(1,3))
image(t(dataMatrixOrdered)[,nrow(dataMatrixOrdered):1])
plot(svd2$v[,1],pch=19,xlab="Column",ylab="First right singular vector")
plot(svd2$v[,2],pch=19,xlab="Column",ylab="Second right singular vector")
```



16/25

# d and variance explained

```
svd1 <- svd(scale(dataMatrixOrdered))
par(mfrow=c(1,2))
plot(svd1$d,xlab="Column",ylab="Singluar value",pch=19)
plot(svd1$d^2/sum(svd1$d^2),xlab="Column",ylab="Percent of variance explained",pch=19)
```



# fast.svd function {corpcor}

Important parameters:  $m, tol$

```
bigMatrix <- matrix(rnorm(1e4*40), nrow=1e4)
system.time(svd(scale(bigMatrix)))
```

```
user  system elapsed
0.151   0.013   0.166
```

```
system.time(fast.svd(scale(bigMatrix), tol=0))
```

```
user  system elapsed
0.115   0.009   0.127
```

# Missing values

```
dataMatrix2 <- dataMatrixOrdered  
dataMatrix2[sample(1:100, size=40, replace=F)] <- NA  
svd1 <- svd(scale(dataMatrix2))
```

```
Error: infinite or missing values in 'x'
```

19/25

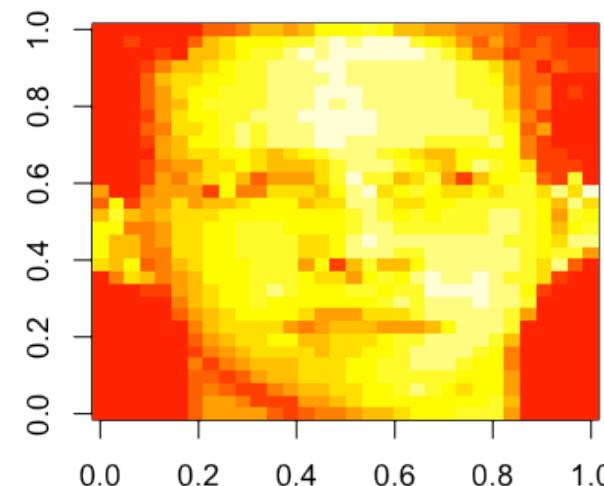
# Imputing {impute}

```
library(impute)
dataMatrix2 <- dataMatrixOrdered
dataMatrix2[sample(1:100,size=40,replace=F)] <- NA
dataMatrix2 <- impute.knn(dataMatrix2)$data
svd1 <- svd(scale(dataMatrixOrdered)); svd2 <- svd(scale(dataMatrix2))
par(mfrow=c(1,2)); plot(svd1$v[,1],pch=19); plot(svd2$v[,1],pch=19)
```

20/25

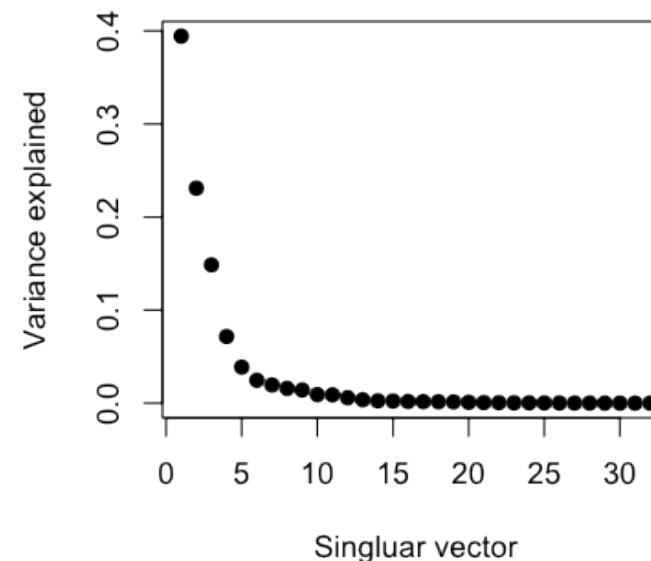
# Face example

```
download.file("https://spark-public.s3.amazonaws.com/dataanalysis/face.rda", destfile="./data/face.rda")
load("./data/face.rda")
image(t(faceData)[,nrow(faceData):1])
```



# Face example - variance explained

```
svd1 <- svd(scale(faceData))
plot(svd1$d^2/sum(svd1$d^2),pch=19,xlab="Singluar vector",ylab="Variance explained")
```



# Face example - create approximations

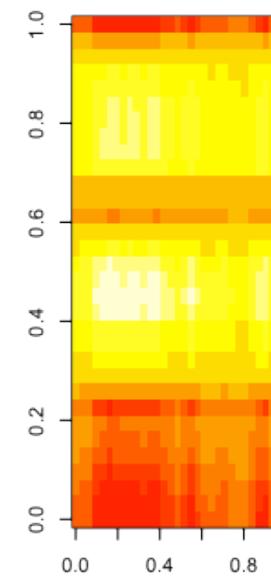
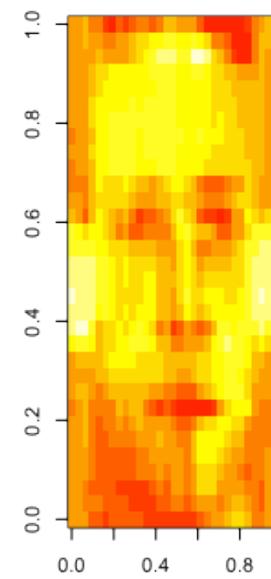
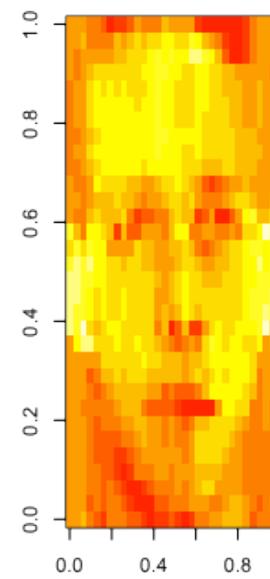
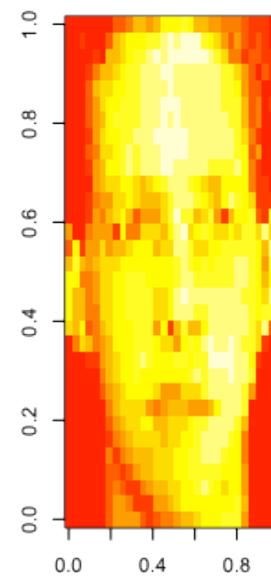
```
svd1 <- svd(scale(faceData))
# %*% is matrix multiplication

# Here svd1$d[1] is a constant
approx1 <- svd1$u[,1] %*% t(svd1$v[,1]) * svd1$d[1]

# In these examples we need to make the diagonal matrix out of d
approx5 <- svd1$u[,1:5] %*% diag(svd1$d[1:5])%*% t(svd1$v[,1:5])
approx10 <- svd1$u[,1:10] %*% diag(svd1$d[1:10])%*% t(svd1$v[,1:10])
```

# Face example - plot approximations

```
par(mfrow=c(1,4))
image(t(faceData)[,nrow(faceData):1])
image(t(approx10)[,nrow(approx10):1])
image(t(approx5)[,nrow(approx5):1])
image(t(approx1)[,nrow(approx1):1])
```



24/25

# Notes and further resources

- Scale matters
- PC's/SV's may mix real patterns
- Can be computationally intensive
- [Advanced data analysis from an elementary point of view](#)
- [Elements of statistical learning](#)
- Alternatives
  - [Factor analysis](#)
  - [Independent components analysis](#)
  - [Latent semantic analysis](#)

# Exploratory graphs

## Part 1

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

# Why do we use graphs in data analysis?

- To understand data properties
- To find patterns in data
- To suggest modeling strategies
- To "debug" analyses
- To communicate results

# Exploratory graphs

- To understand data properties
- To find patterns in data
- To suggest modeling strategies
- To "debug" analyses
- To communicate results

3/20

# Characteristics of exploratory graphs

- They are made quickly
- A large number are made
- The goal is for personal understanding
- Axes/legends are generally cleaned up
- Color/size are primarily used for information

# Background - perceptual tasks

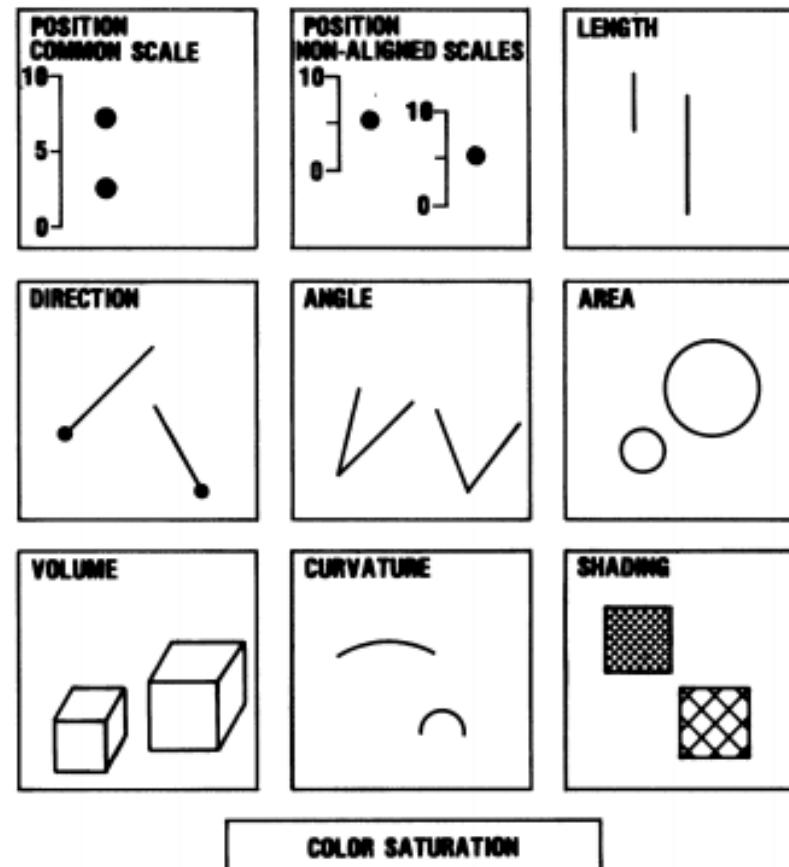
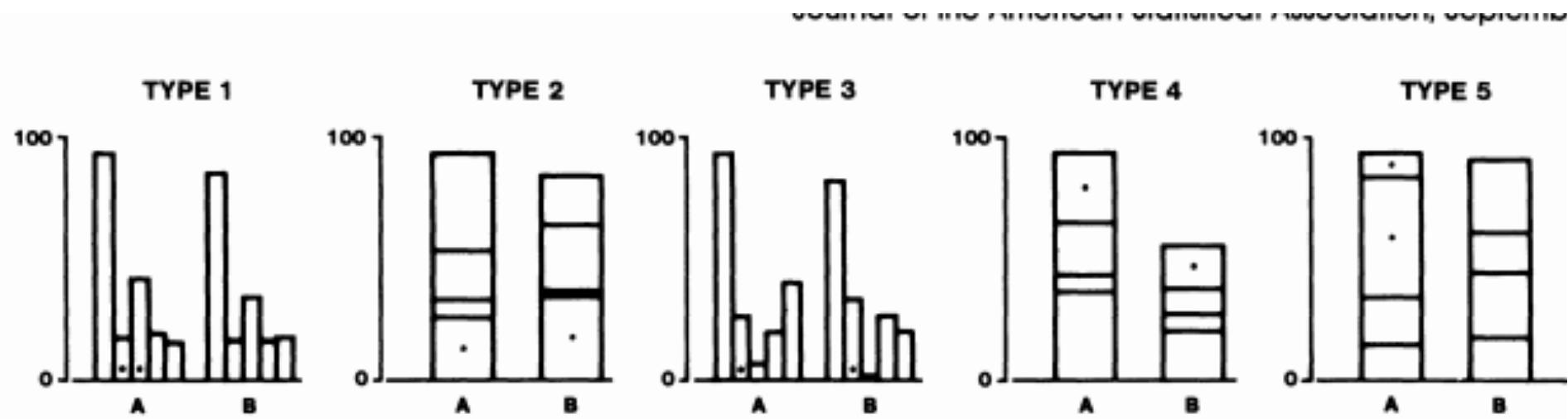


Figure 1. Elementary perceptual tasks.

Graphical perception: Theory, Experimentation, and Applications to the Development of Graphical Models

5/20

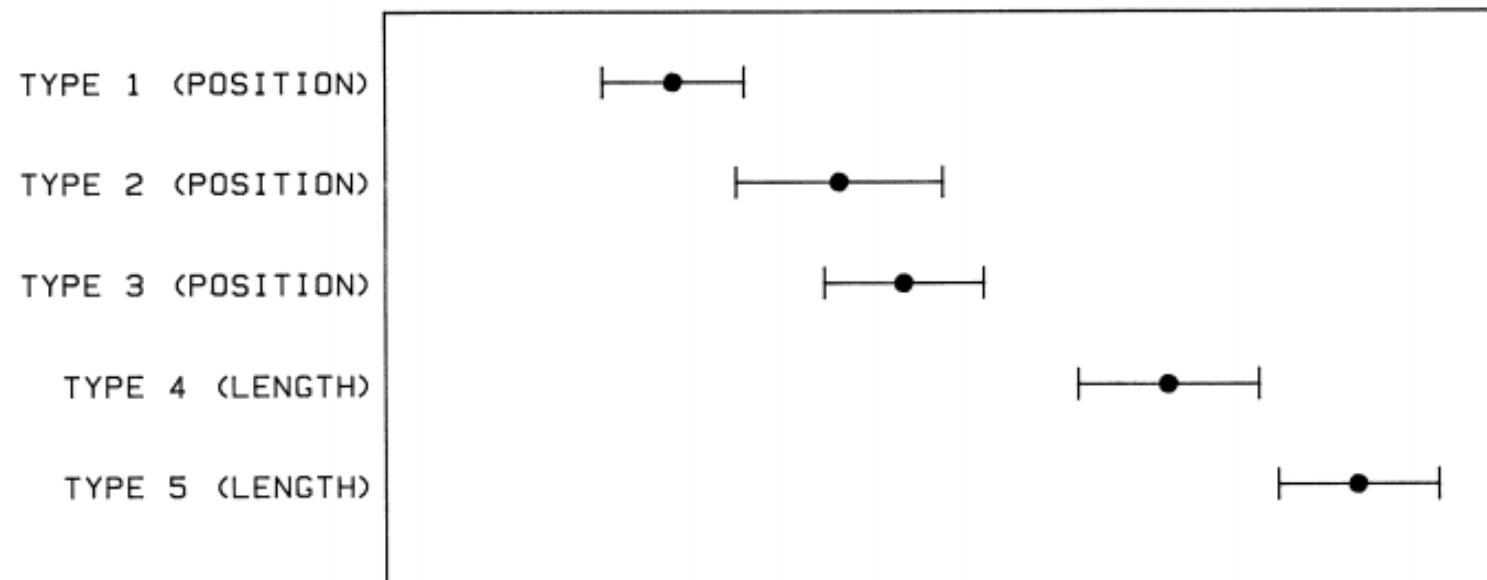
# Position versus length



*Figure 4. Graphs from position-length experiment.*

Graphical perception: Theory, Experimentation, and Applications to the Development of Graphical Models

# Position versus length - results



Graphical perception: Theory, Experimentation, and Applications to the Development of Graphical Models

# Position versus angle

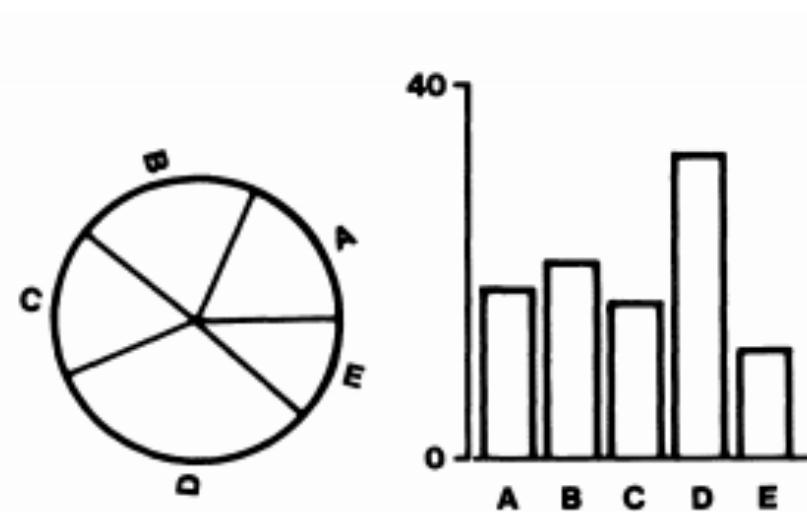
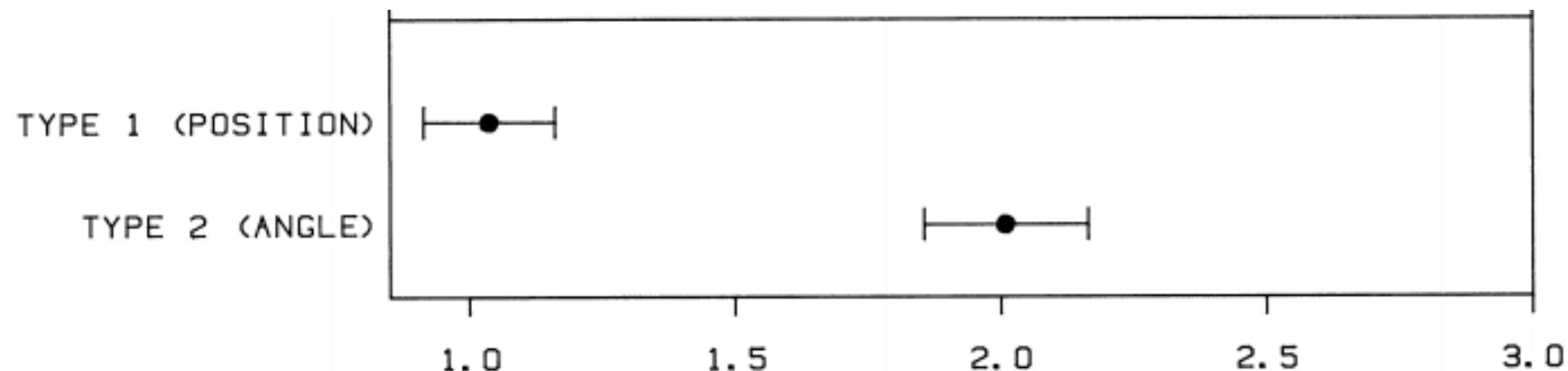


Figure 3. Graphs from position–angle experiment.

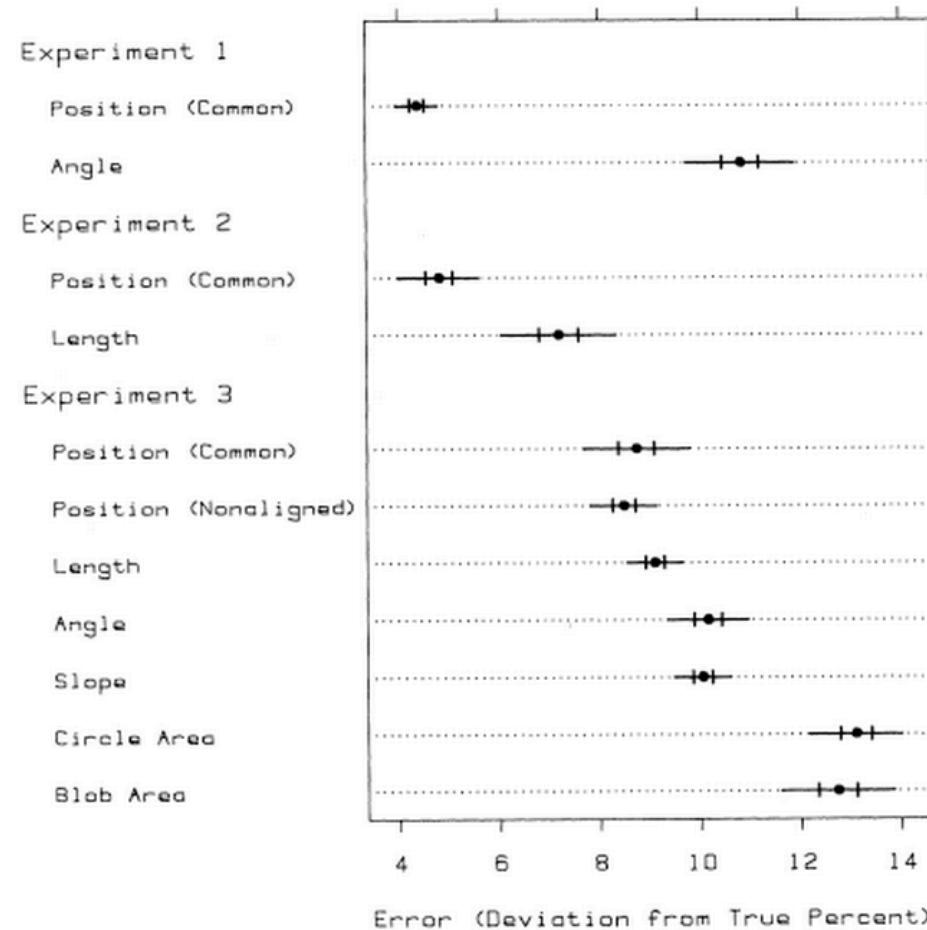
Graphical perception: Theory, Experimentation, and Applications to the Development of Graphical Models

# Position versus angle - results



Graphical perception: Theory, Experimentation, and Applications to the Development of Graphical Models

# More experimental results



## Graphical Perception and Graphical Methods for Analyzing Scientific Data

10/20

# Summary

- Use common scales when possible
- When possible use position comparisons
- Angle comparisons are frequently hard to interpret (no piecharts!)
- No 3-D barcharts

# Housing data

The screenshot shows a web browser displaying the 'American Community Survey' section of the Census Bureau's website. The URL in the address bar is [www.census.gov/acs/www/data\\_documentation/public\\_use\\_microdata\\_sample/](http://www.census.gov/acs/www/data_documentation/public_use_microdata_sample/). The page title is 'American Community Survey'. The main content area is titled 'Public Use Microdata Sample (PUMS)' and includes sections on 'About PUMS', 'PUMS Data', 'PUMS Documentation', 'PUMS on DataFerrett', 'PUMS FAQs', and 'Custom Tabulations'. On the left, there is a sidebar with links to 'Data Releases', 'Data Product Descriptions', 'Documentation', 'Geography', 'Downloadable data via FTP', and 'Summary File'. The 'Public Use Microdata Sample (PUMS)' link is highlighted in blue. At the top of the page, there is a navigation bar with links for 'People', 'Business', 'Geography', 'Data', 'Research', and 'Newsroom'. A search bar is also present at the top right.

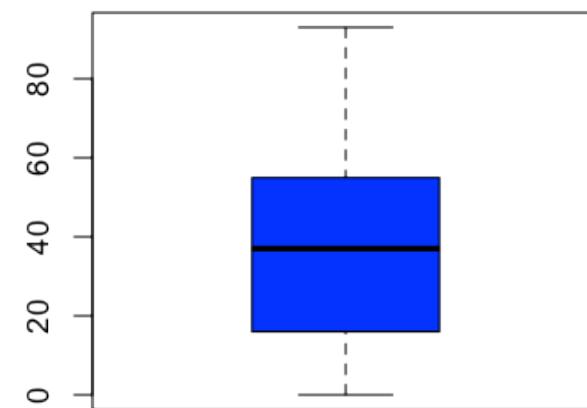
```
pData <- read.csv("./data/ss06pid.csv")
```

12/20

# Boxplots

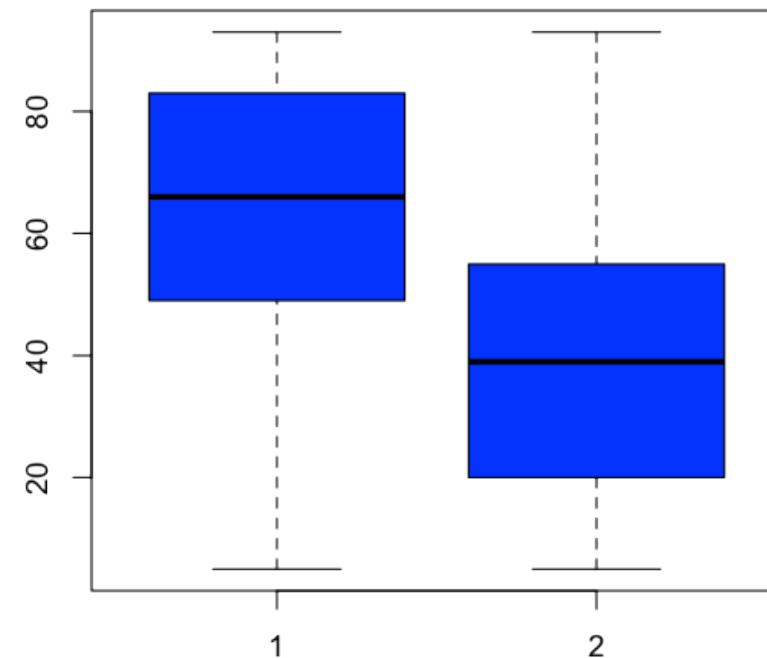
- Important parameters: *col, varwidth, names, horizontal*

```
boxplot(pData$AGEP,col="blue")
```



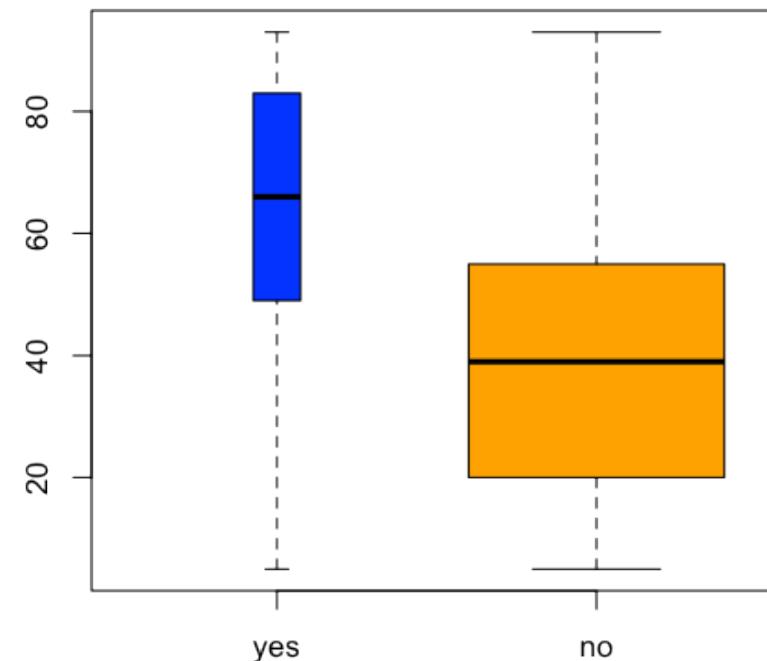
# Boxplots

```
boxplot(pData$AGEP ~ as.factor(pData$DDRS), col="blue")
```



# Boxplots

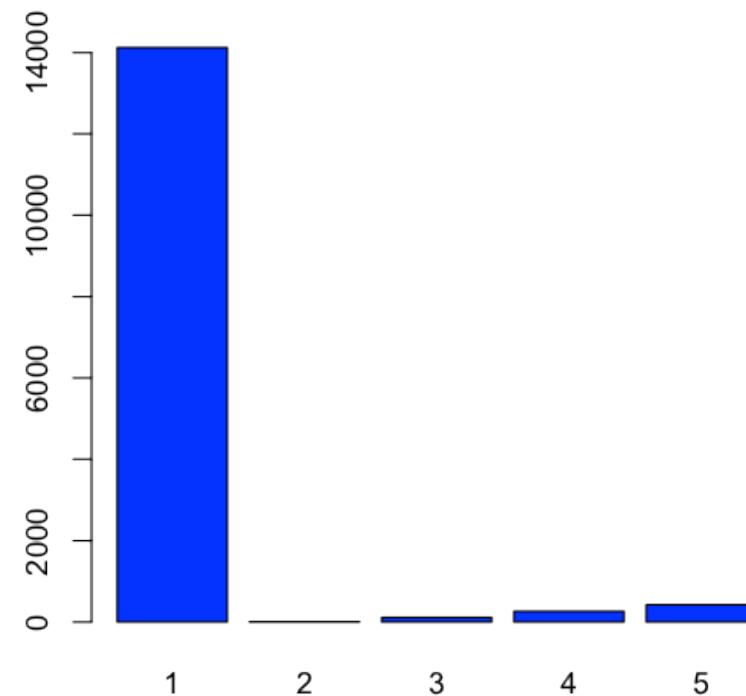
```
boxplot(pData$AGEP ~ as.factor(pData$DDRS), col=c("blue", "orange"), names=c("yes", "no"), varwidth=TRUE)
```



15/20

# Barplots

```
barplot(table(pData$CIT), col="blue")
```

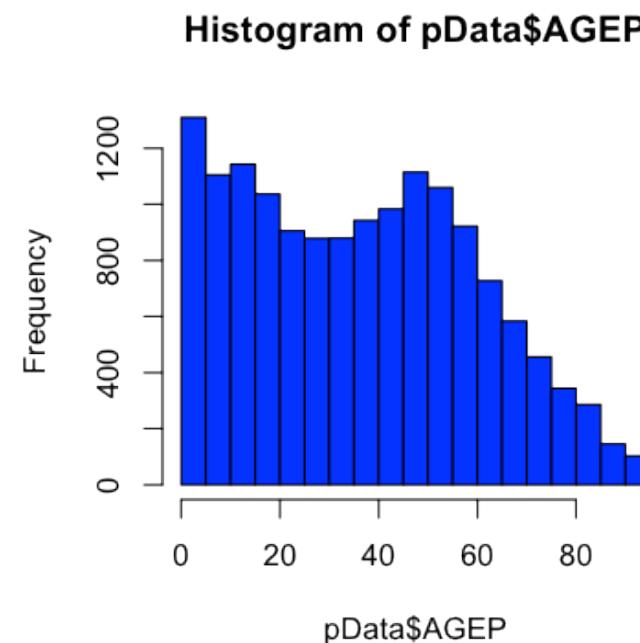


16/20

# Histograms

- Important parameters: *breaks, freq, col, xlab, ylab, xlim, ylim, main*

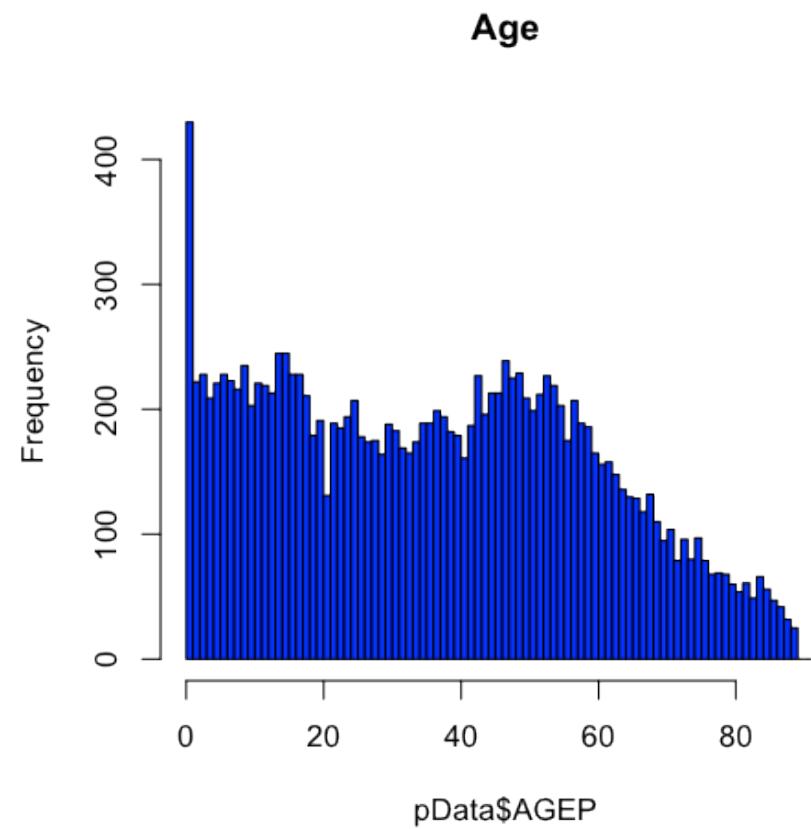
```
hist(pData$AGEP,col="blue")
```



17/20

# Histograms

```
hist(pData$AGEP,col="blue",breaks=100,main="Age")
```

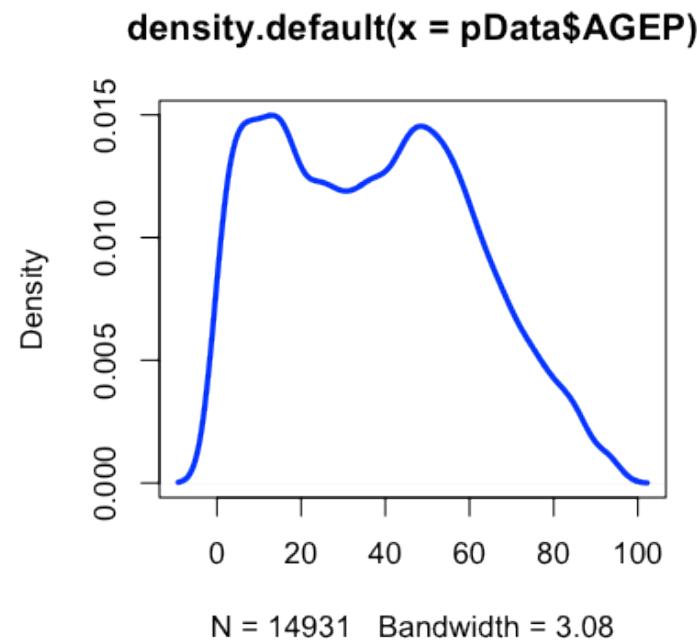


18/20

# Density plots

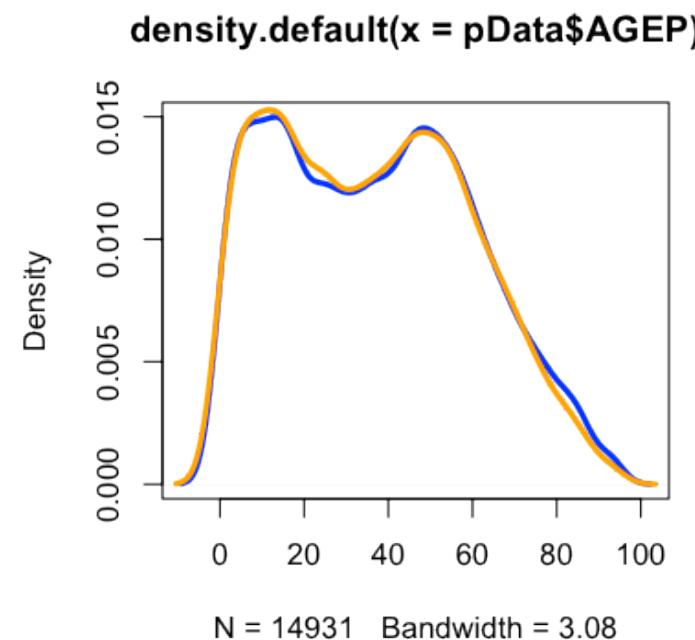
Important parameters (to plot): *col,lwd,xlab,ylab,xlim,ylim*

```
dens <- density(pData$AGEP)
plot(dens,lwd=3,col="blue")
```



# Density plots - multiple distributions

```
dens <- density(pData$AGEP)
densMales <- density(pData$AGEP[which(pData$SEX==1) ])
plot(dens,lwd=3,col="blue")
lines(densMales,lwd=3,col="orange")
```



20/20

# Exploratory graphs

## Part 2

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

# Why do we use graphs in data analysis?

- To understand data properties
- To find patterns in data
- To suggest modeling strategies
- To "debug" analyses
- To communicate results

# Exploratory graphs

- To understand data properties
- To find patterns in data
- To suggest modeling strategies
- To "debug" analyses
- To communicate results

# Characteristics of exploratory graphs

- They are made quickly
- A large number are made
- The goal is for personal understanding
- Axes/legends are generally not cleaned up
- Color/size are primarily used for information

# Housing data

The screenshot shows a web browser displaying the 'American Community Survey' section of the Census Bureau's website. The URL in the address bar is [www.census.gov/acs/www/data\\_documentation/public\\_use\\_microdata\\_sample/](http://www.census.gov/acs/www/data_documentation/public_use_microdata_sample/). The page title is 'American Community Survey'. The main content area is titled 'Public Use Microdata Sample (PUMS)' and includes sections on 'About PUMS', 'PUMS Data', 'PUMS Documentation', 'PUMS on DataFerrett', 'PUMS FAQs', and 'Custom Tabulations'. On the left, there is a sidebar with links to 'Data Releases', 'Data Product Descriptions', 'Documentation', 'Geography', 'Downloadable data via FTP', and 'Summary File'. A sub-menu for 'Public Use Microdata Sample (PUMS)' is expanded, showing the same six items. At the top of the page, there is a navigation bar with links for 'People', 'Business', 'Geography', 'Data', 'Research', and 'Newsroom', along with a search bar.

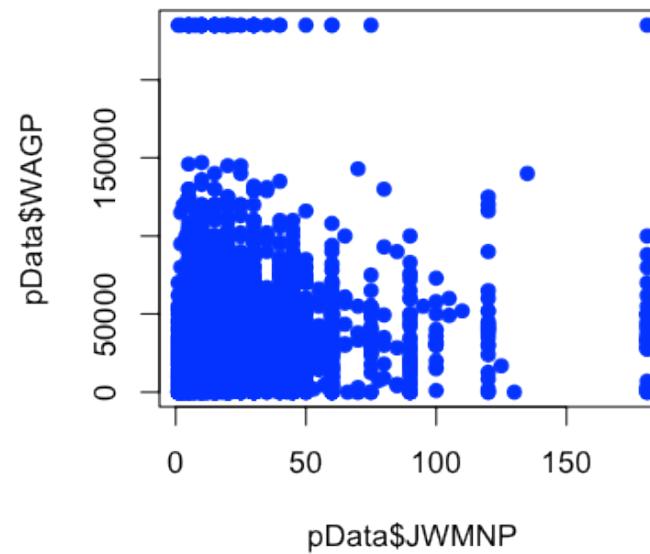
```
pData <- read.csv("./data/ss06pid.csv")
```

5/23

# Scatterplots

- Important parameters:  $x, y, type, xlab, ylab, xlim, ylim, cex, col, bg$
- See ?par for more

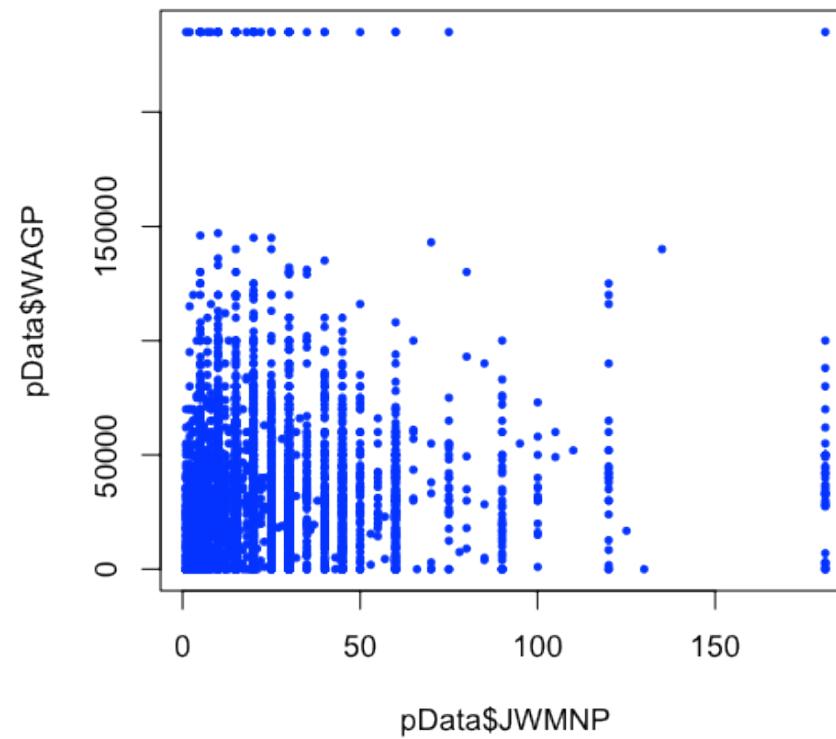
```
plot(pData$JWMNP, pData$WAGP, pch=19, col="blue")
```



6/23

# Scatterplots - size matters

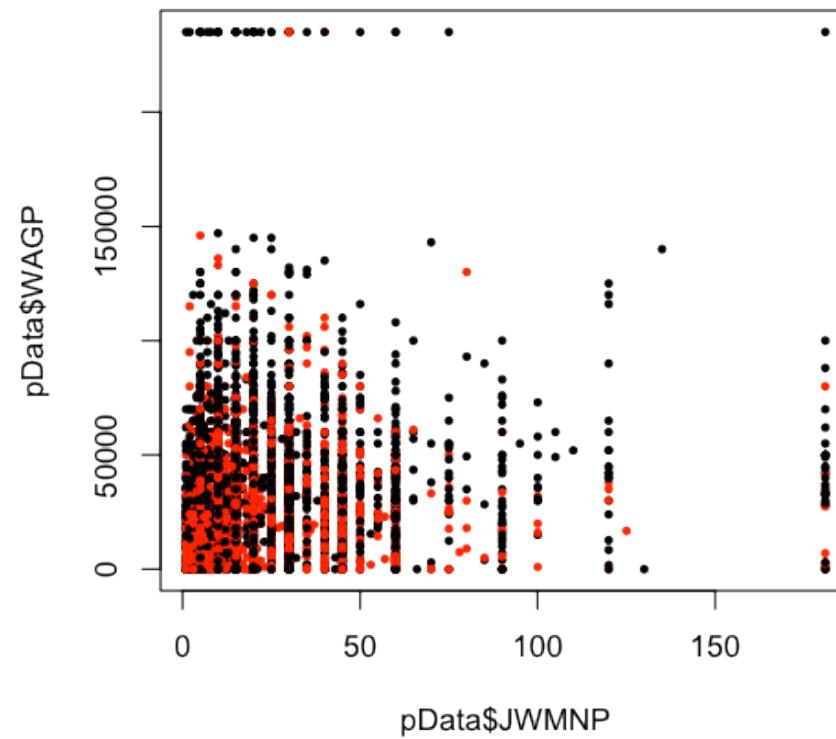
```
plot(pData$JWMNP,pData$WAGP,pch=19,col="blue",cex=0.5)
```



7/23

# Scatterplots - using color

```
plot(pData$JWMNP,pData$WAGP,pch=19,col=pData$SEX,cex=0.5)
```



8/23

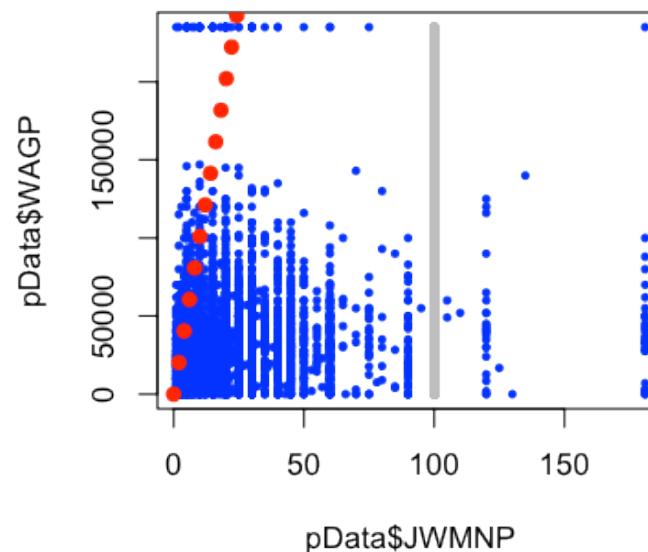
# Scatterplots - using size

```
percentMaxAge <- pData$AGEP/max(pData$AGEP)  
plot(pData$JWMNP,pData$WAGP,pch=19,col="blue",cex=percentMaxAge*0.5)
```

9/23

# Scatterplots - overlaying lines/points

```
plot(pData$JWMNP,pData$WAGP,pch=19,col="blue",cex=0.5)
lines(rep(100,dim(pData)[1]),pData$WAGP,col="grey",lwd=5)
points(seq(0,200,length=100),seq(0,20e5,length=100),col="red",pch=19)
```



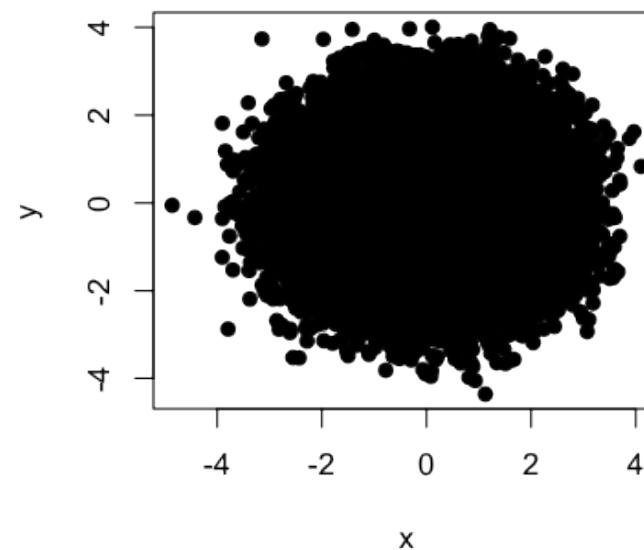
# Scatterplots - numeric variables as factors

```
library(Hmisc)
ageGroups <- cut2(pData$AGEP,g=5)
plot(pData$JWMNP,pData$WAGP,pch=19,col=ageGroups,cex=0.5)
```

11/23

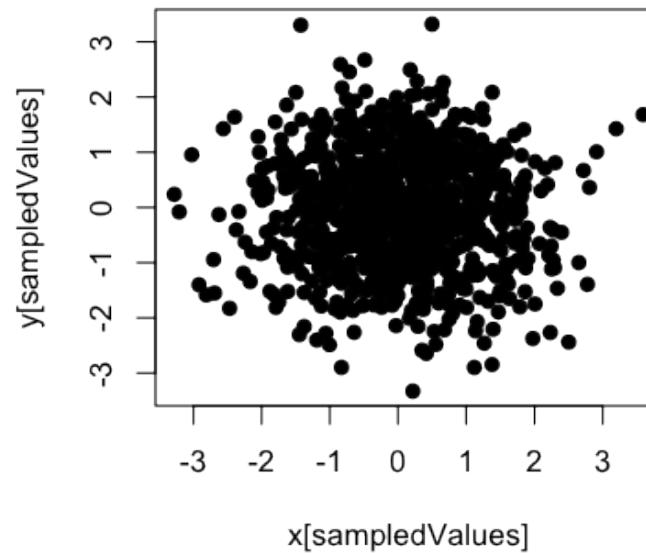
# If you have a lot of points

```
x <- rnorm(1e5)  
y <- rnorm(1e5)  
plot(x,y,pch=19)
```



# If you have a lot of points - sampling

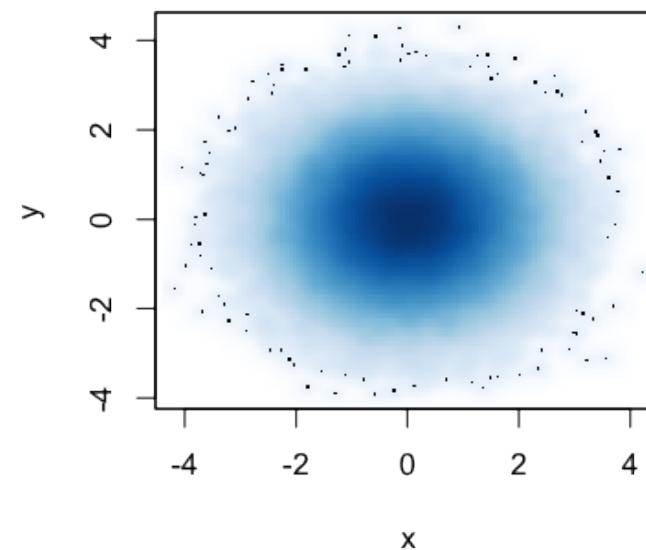
```
x <- rnorm(1e5)
y <- rnorm(1e5)
sampledValues <- sample(1:1e5, size=1000, replace=FALSE)
plot(x[sampledValues], y[sampledValues], pch=19)
```



13/23

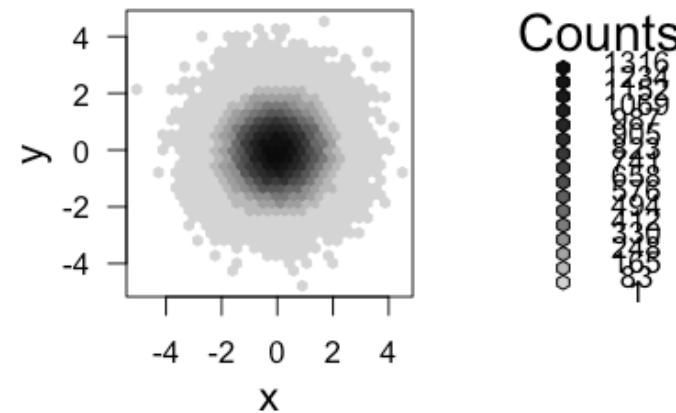
# If you have a lot of points - smoothScatter

```
x <- rnorm(1e5)  
y <- rnorm(1e5)  
smoothScatter(x,y)
```



# If you have a lot of points - hexbin {hexbin}

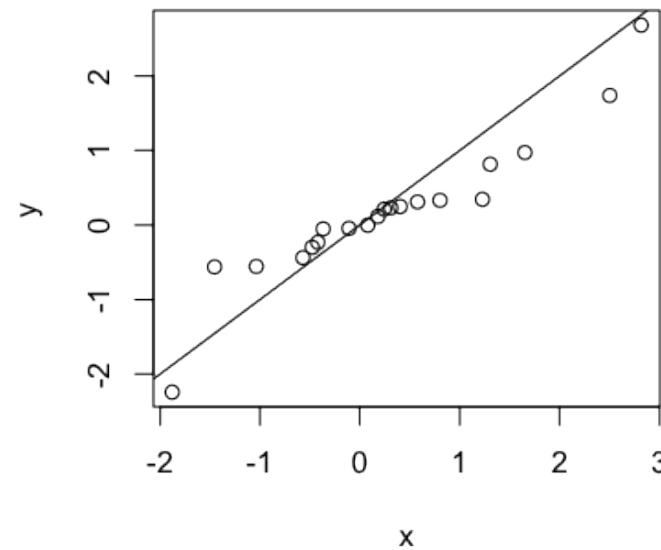
```
library(hexbin)
x <- rnorm(1e5)
y <- rnorm(1e5)
hbo <- hexbin(x,y)
plot(hbo)
```



# QQ-plots

- Important parameters:  $x, y$

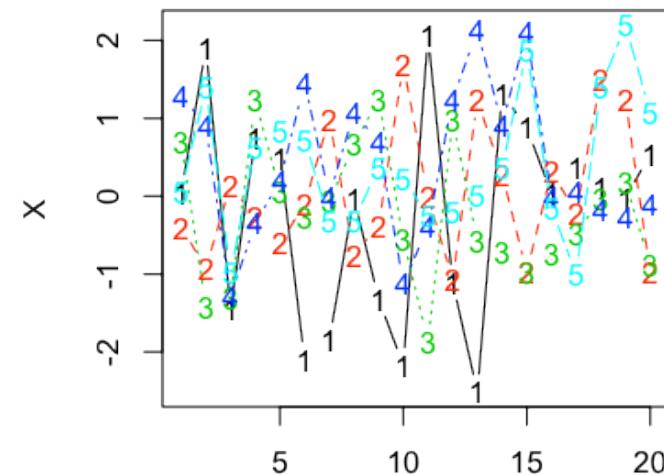
```
x <- rnorm(20); y <- rnorm(20)
qqplot(x,y)
abline(c(0,1))
```



# Matplot and spaghetti

- Important parameters:  $x, y, lty, lwd, pch, col$

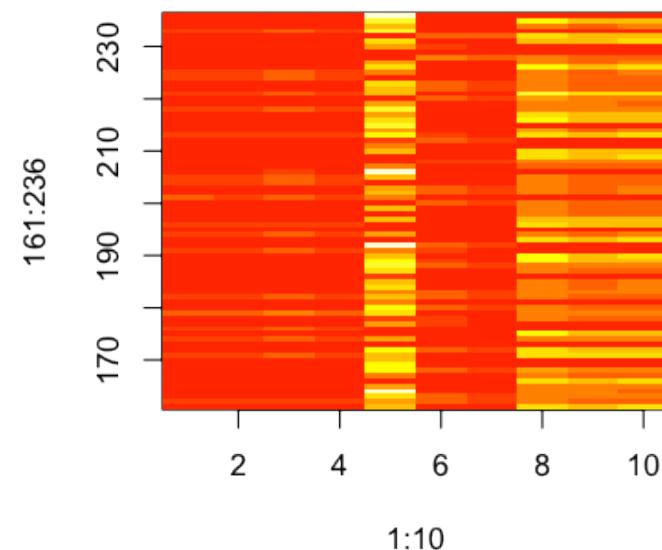
```
X <- matrix(rnorm(20*5), nrow=20)
matplot(X, type="b")
```



# Heatmaps

- Important parameters:  $x, y, z, col$

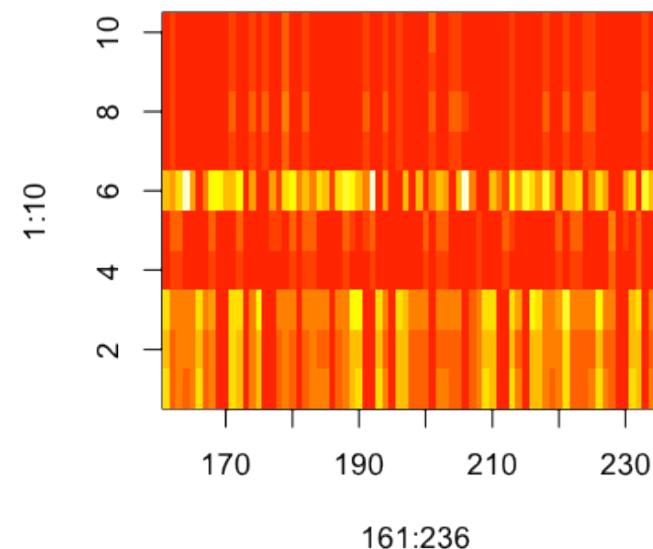
```
image(1:10, 161:236, as.matrix(pData[1:10, 161:236]))
```



18/23

# Heatmaps - matching intuition

```
newMatrix <- as.matrix(pData[1:10, 161:236])
newMatrix <- t(newMatrix)[, nrow(newMatrix):1]
image(161:236, 1:10, newMatrix)
```



19/23

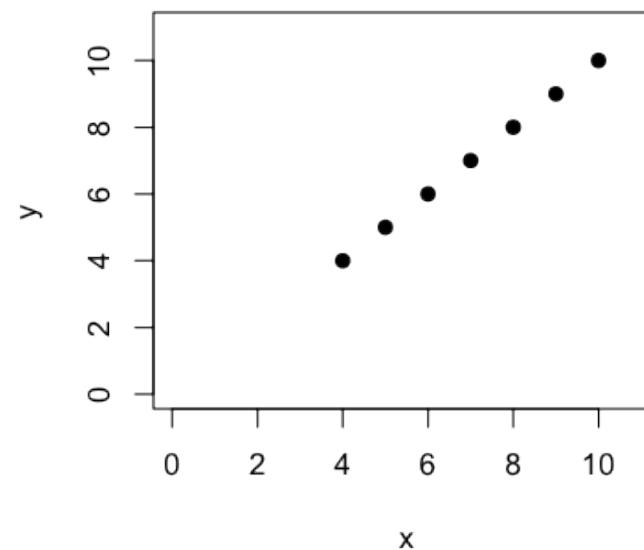
# Maps - very basics

```
library(maps)
map("world")
lat <- runif(40,-180,180); lon <- runif(40,-90,90)
points(lat,lon,col="blue",pch=19)
```

20/23

# Missing values and plots

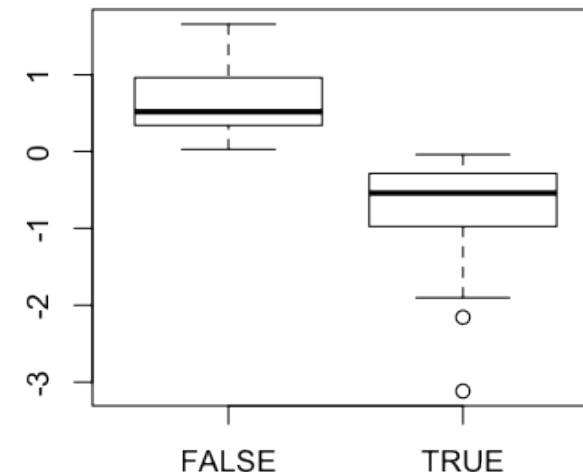
```
x <- c(NA, NA, NA, 4, 5, 6, 7, 8, 9, 10)
y <- 1:10
plot(x,y,pch=19,xlim=c(0,11),ylim=c(0,11))
```



21/23

# Missing values and plots

```
x <- rnorm(100)
y <- rnorm(100)
y[x < 0] <- NA
boxplot(x ~ is.na(y))
```



# Further resources

- [R Graph Gallery](#)
- [ggplot2, ggplot2 basic introduction](#)
- [lattice package, lattice introduction](#)
- [R bloggers](#)

# Expository graphs

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

# Why do we use graphs in data analysis?

- To understand data properties
- To find patterns in data
- To suggest modeling strategies
- To "debug" analyses
- To communicate results

# Expository graphs

- To understand data properties
- To find patterns in data
- To suggest modeling strategies
- To "debug" analyses
- **To communicate results**

3/21

# Characteristics of expository graphs

- The goal is to communicate information
- Information density is generally good
- Color/size are used both for aesthetics and communication
- Expository figures have understandable axes, titles, and legends

# Housing data

The screenshot shows a web browser displaying the 'American Community Survey' section of the Census Bureau's website. The URL in the address bar is [www.census.gov/acs/www/data\\_documentation/public\\_use\\_microdata\\_sample/](http://www.census.gov/acs/www/data_documentation/public_use_microdata_sample/). The page title is 'American Community Survey'. The main content area is titled 'Public Use Microdata Sample (PUMS)' and includes sections on 'About PUMS', 'PUMS Data', 'PUMS Documentation', 'PUMS on DataFerrett', 'PUMS FAQs', and 'Custom Tabulations'. On the left, there is a sidebar with links to 'Data Releases', 'Data Product Descriptions', 'Documentation', 'Geography', 'Downloadable data via FTP', and 'Summary File'. The 'Public Use Microdata Sample (PUMS)' link is highlighted in blue. At the top of the page, there is a navigation bar with links for 'People', 'Business', 'Geography', 'Data', 'Research', and 'Newsroom'. A search bar is also present at the top right.

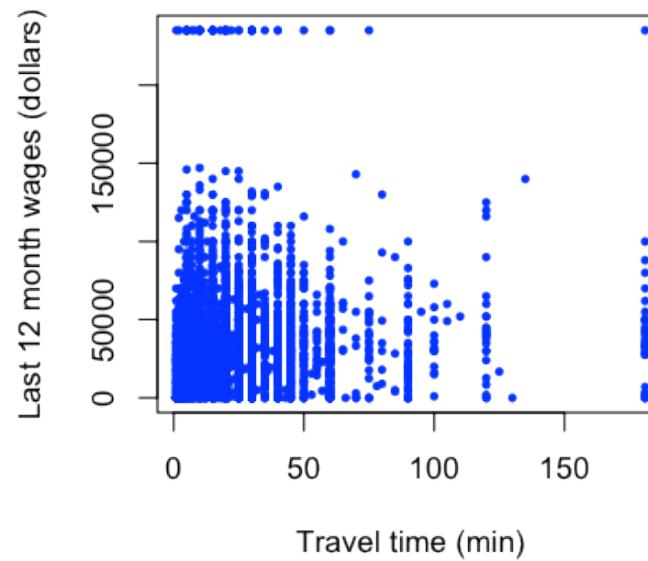
```
pData <- read.csv("./data/ss06pid.csv")
```

5/21

# Axes

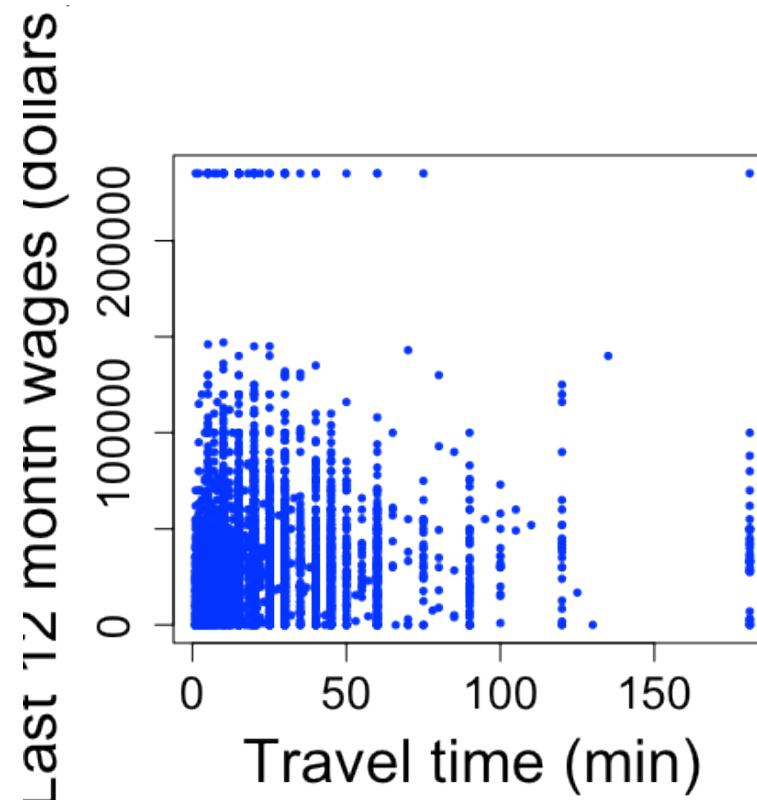
Important parameters: *xlab*, *ylab*, *cex.lab*, *cex.axis*

```
plot(pData$JWMNP,pData$WAGP,pch=19,col="blue",cex=0.5,  
     xlab="Travel time (min)",ylab="Last 12 month wages (dollars)")
```



# Axes

```
plot(pData$JWMNP,pData$WAGP,pch=19,col="blue",cex=0.5,  
xlab="Travel time (min)",ylab="Last 12 month wages (dollars)",cex.lab=2,cex.axis=1.5)
```

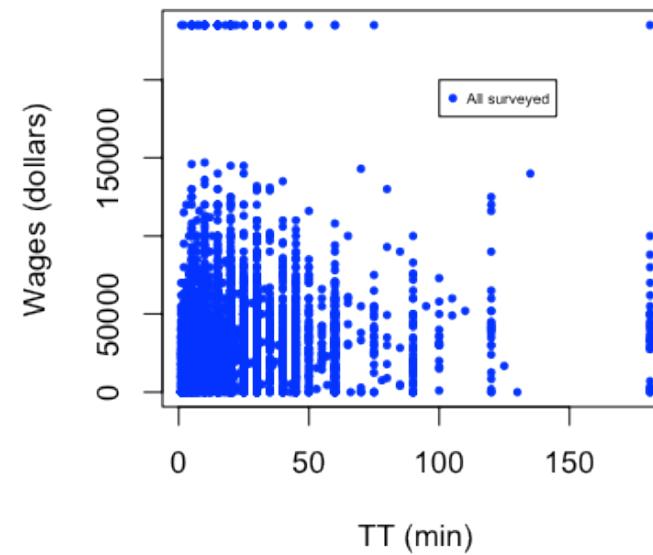


7/21

# Legends

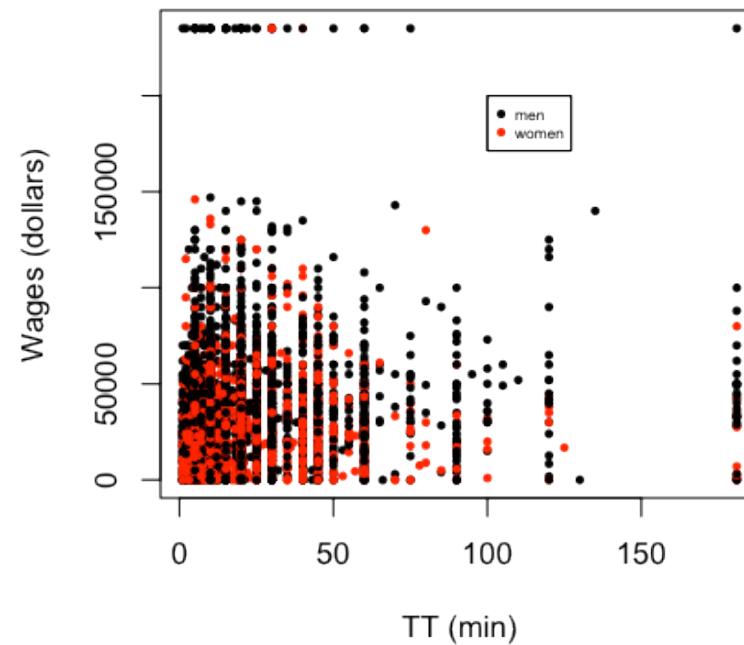
- Important parameters:  $x, y, legend, other\ plotting\ parameters$

```
plot(pData$JWMNP,pData$WAGP,pch=19,col="blue",cex=0.5,xlab="TT (min)",ylab="Wages (dollars)")  
legend(100,200000,legend="All surveyed",col="blue",pch=19,cex=0.5)
```



# Legends

```
plot(pData$JWMNP,pData$WAGP,pch=19,cex=0.5,xlab="TT (min)",ylab="Wages (dollars)",col=pData$SEX)
legend(100,200000,legend=c("men","women"),col=c("black","red"),pch=c(19,19),cex=c(0.5,0.5))
```

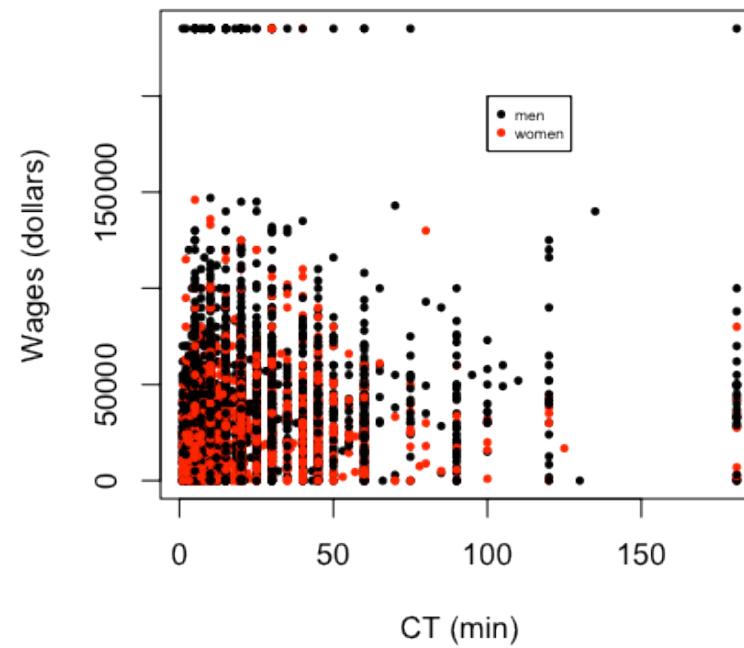


9/21

# Titles

```
plot(pData$JWMNP,pData$WAGP,pch=19,cex=0.5,xlab="CT (min)",  
      ylab="Wages (dollars)",col=pData$SEX,main="Wages earned versus commute time")  
legend(100,200000,legend=c("men", "women"),col=c("black", "red"),pch=c(19,19),cex=c(0.5,0.5))
```

**Wages earned versus commute time**

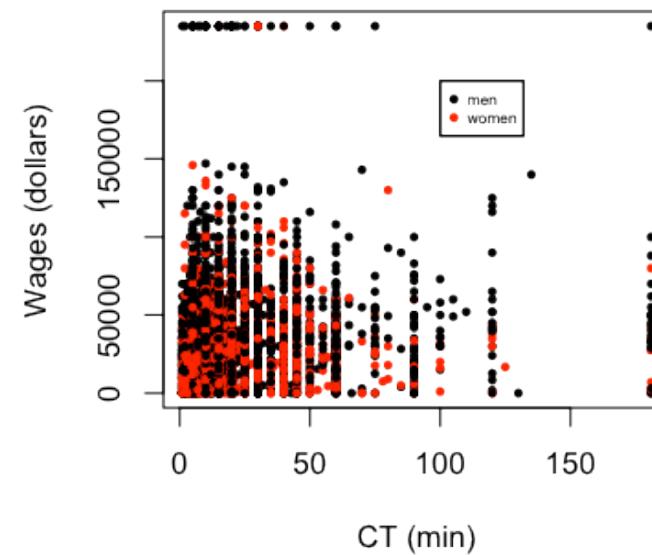
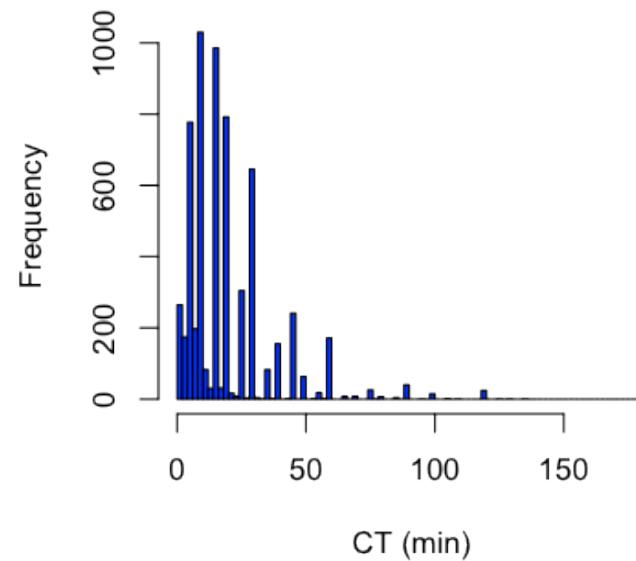


10/21

# Multiple panels

```
par(mfrow=c(1,2))

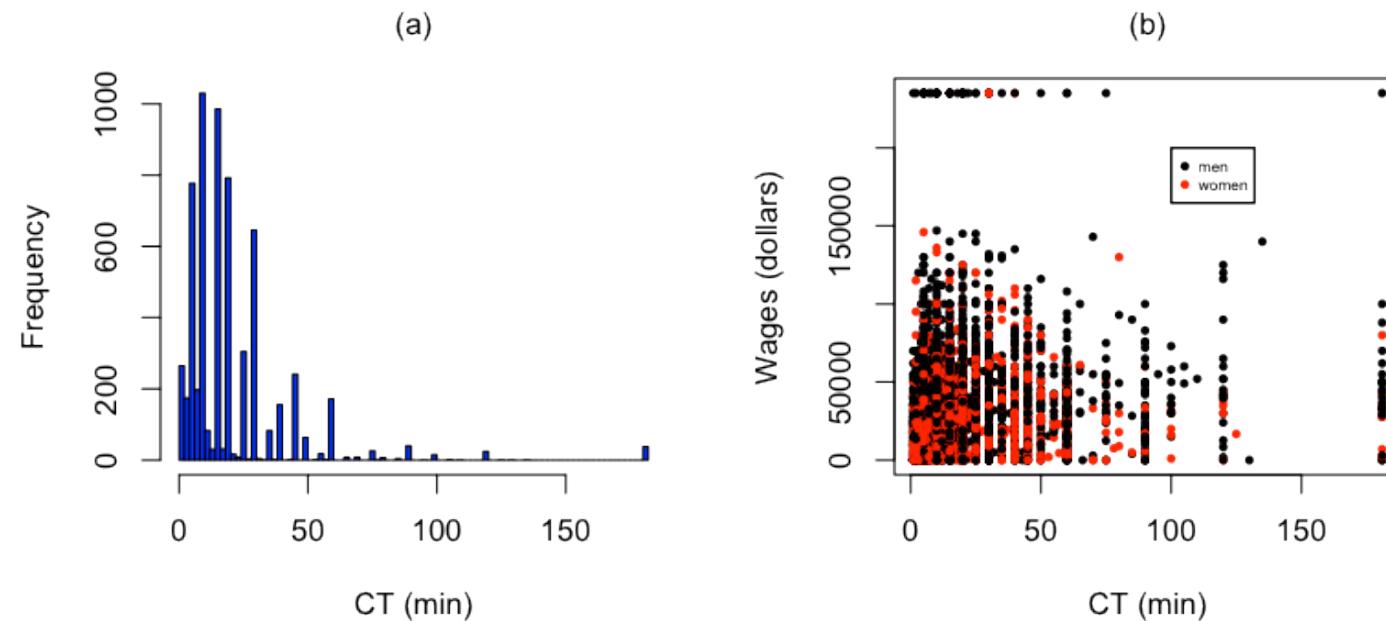
hist(pData$JWMNP,xlab="CT (min)",col="blue",breaks=100,main="")
plot(pData$JWMNP,pData$WAGP,pch=19,cex=0.5,xlab="CT (min)",ylab="Wages (dollars)",col=pData$SEX)
legend(100,200000,legend=c("men","women"),col=c("black","red"),pch=c(19,19),cex=c(0.5,0.5))
```



# Adding text

```
par(mfrow=c(1,2))
hist(pData$JWMNP,xlab="CT (min)",col="blue",breaks=100,main="")
mtext(text="(a)",side=3,line=1)
plot(pData$JWMNP,pData$WAGP,pch=19,cex=0.5,xlab="CT (min)",ylab="Wages (dollars)",col=pData$SEX)
legend(100,200000,legend=c("men","women"),col=c("black","red"),pch=c(19,19),cex=c(0.5,0.5))
mtext(text="(b)",side=3,line=1)
```

# Figure captions



**Figure 1. Distribution of commute time and relationship to wage earned by sex** (a) Commute times in the American Community Survey (ACS) are right skewed. (b) Commute times do not appear to be strongly correlated with wage for either sex.

# Colorblindness

The screenshot shows a web browser window with the title bar "Go to home page. Vischeck: VischeckImage". The address bar displays "www.vischeck.com/vischeck/vischeckImage.php". The main content area is titled "Vischeck" with three colored bars (red, green, blue) below it. On the left, there is a sidebar with links: Home, Vischeck (with sub-links Run Images and Run Webpages), Daltonize, Examples, Downloads, Info & Links, FAQ, and About Us. A "User quotes" box contains a testimonial from Brad C. At the bottom of the sidebar are links for Web, Vischeck (which is selected), and Google Search, along with a "SUPPORT WIKIPEDIA" button and the URL "www.vischeck.com". The main content area has a heading "Try Vischeck on Your Image Files" and a sub-heading "Select the type of color vision to simulate:". There are three radio buttons with corresponding images: Deuteranope (a form of red/green color deficit), Protanope (another form of red/green color deficit), and Tritanope (a blue/yellow deficit- very rare). Below these is a "Choose File" button with the file name "unnamed-chunk-6.png" and a "Run Vischeck!" button. A "Notes:" section lists several tips for using Vischeck.

**Try Vischeck on Your Image Files**

Select the type of color vision to simulate:

Deuteranope (a form of red/green color deficit)

Protanope (another form of red/green color deficit)

Tritanope (a blue/yellow deficit- very rare)

Image file:  unnamed-chunk-6.png

Notes:

- Vischeck accepts most common image formats. However, we recommend that you use PNG or JPEG format for uploading large images as these tend to transfer faster.
- For PowerPoint slides, you can save all your slides as PNG images with "Save As..." and run Vischeck on each slide.
- If you have many images to process, consider [downloading](#) Vischeck to run on your own computer.)
- Uploading a large file may take a while - please be patient!

Please read our [terms of use](#) before using Vischeck.

<http://www.vischeck.com/>

14/21

# Graphical workflow

- Start with a rough plot
- Tweak it to make it expository
- **Save the file**
- Include it in presentations

Saving files in R is done with graphics *devices*. Use the command `?Devices` to see a list. Here we will go over the most popular devices.

# pdf

- Important parameters: *file, height, width*

```
pdf(file="twoPanel.pdf",height=4,width=8)
par(mfrow=c(1,2))
hist(pData$JWMNP,xlab="CT (min)",col="blue",breaks=100,main="")
mtext(text="(a)",side=3,line=1)
plot(pData$JWMNP,pData$WAGP,pch=19,cex=0.5,xlab="CT (min)",ylab="Wages (dollars)",col=pData$SEX)
legend(100,200000,legend=c("men","women"),col=c("black","red"),pch=c(19,19),cex=c(0.5,0.5))
mtext(text="(b)",side=3,line=1)

dev.off()
```

```
pdf
```

```
2
```

16/21

# png

- Important parameters: *file, height, width*

```
png(file="twoPanel.png",height=480,width=(2*480))
par(mfrow=c(1,2))
hist(pData$JWMNP,xlab="CT (min)",col="blue",breaks=100,main="")
mtext(text="(a)",side=3,line=1)
plot(pData$JWMNP,pData$WAGP,pch=19,cex=0.5,xlab="CT (min)",ylab="Wages (dollars)",col=pData$SEX)
legend(100,200000,legend=c("men","women"),col=c("black","red"),pch=c(19,19),cex=c(0.5,0.5))
mtext(text="(b)",side=3,line=1)
dev.off()
```

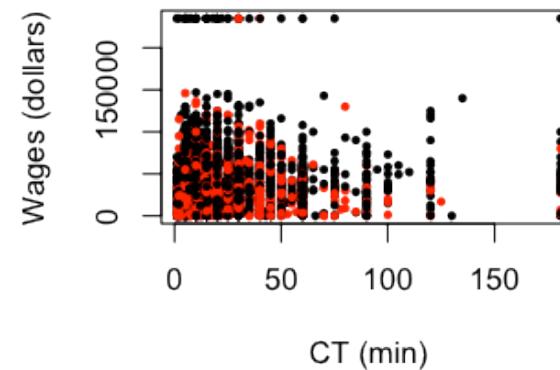
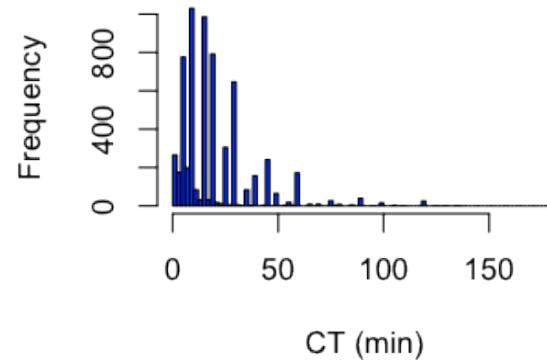
```
pdf
```

```
2
```

17/21

# dev.copy2pdf

```
par(mfrow=c(1,2))
hist(pData$JWMNP,xlab="CT (min)",col="blue",breaks=100,main="")
plot(pData$JWMNP,pData$WAGP,pch=19,cex=0.5,xlab="CT (min)",ylab="Wages (dollars)",col=pData$SEX)
```



```
dev.copy2pdf(file="twoPanelv2.pdf")
```

pdf

2

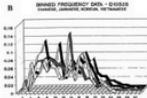
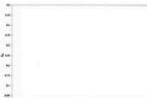
18/21

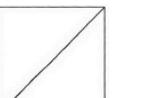
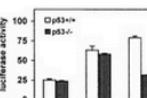
# Something to avoid

The top ten worst graphs

With apologies to the authors, we provide the following list of the top ten worst graphs in the scientific literature. As these examples indicate, good scientists can make mistakes.

---

1. Roeder K (1994) DNA fingerprinting: A review of the controversy (with discussion). *Statistical Science* 9:222-278, Figure 4  
[[The article](#) | [The figure](#) | [Discussion](#)]  

2. Wittke-Thompson JK, Pluzhnikov A, Cox NJ (2005) Rational inferences about departures from Hardy-Weinberg equilibrium. *American Journal of Human Genetics* 76:967-986, Figure 1  
[[The article](#) | [Fig 1AB](#) | [Fig 1CD](#) | [Discussion](#)]  

3. Epstein MP, Satten GA (2003) Inference on haplotype effects in case-control studies using unphased genotype data. *American Journal of Human Genetics* 73:1316-1329, Figure 1  
[[The article](#) | [The figure](#) | [Discussion](#)]  

4. Mykland P, Tierney L, Yu B (1995) Regeneration in Markov chain samplers. *Journal of the American Statistical Association* 90:233-241, Figure 1  
[[The article](#) | [The figure](#) | [Discussion](#)]  

5. Hummer BT, Li XL, Hassel BA (2001) Role for p53 in gene induction by double-stranded RNA. *J Virol* 75:7774-7777, Figure 4  
[[The article](#) | [The figure](#) | [Discussion](#)]  


| Condition | p53+/- | p53-/- |
|-----------|--------|--------|
| untrt     | ~20    | ~20    |
| IFN       | ~65    | ~60    |
| dsRNA     | ~75    | ~35    |
| SV        | ~85    | ~75    |

[http://www.biostat.wisc.edu/~kbroman/topten\\_worstgraphs/](http://www.biostat.wisc.edu/~kbroman/topten_worstgraphs/)

19/21

# Something to aspire to



<http://www.facebook.com/notes/facebook-engineering/visualizing-friendships/469716398919>

20/21

# Further resources

- [How to display data badly](#)
- [The visual display of quantitative information](#)
- [Creating more effective graphs](#)
- [R Graphics Cookbook](#)
- [ggplot2: Elegant Graphics for Data Analysis](#)
- [Flowing Data](#)

# Regression with factor variables

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

# Key ideas

- Outcome is still quantitative
- Covariate(s) are factor variables
- Fitting lines = fitting means
- Want to evaluate contribution of all factor levels at once

# Example: Movie ratings

The screenshot shows the homepage of Rotten Tomatoes (www.rottentomatoes.com). The header features the Rotten Tomatoes logo and a search bar. The main banner promotes the movie "DIANA VREELAND: THE EYE HAS TO TRAVEL" with a PG-13 rating and a "FRESH" critics' pick badge. Below the banner, the "TOP BOX OFFICE" section lists movies like "Warm Bodies" and "Hansel and Gretel: Witch Hunters". The "OPENING" section lists movies like "Identity Thief" and "Side Effects". A central feature is an interview with the cast of "Side Effects" featuring Rooney Mara and Channing Tatum. To the right, there's an advertisement for "INFOCHIMPS BIG DATA PLATFORM" with a call-to-action button. At the bottom, there's a "What's Hot on RT" section and a "Featured Movie Trailers" section.

<http://www.rottentomatoes.com/>

3/20

# Movie Data

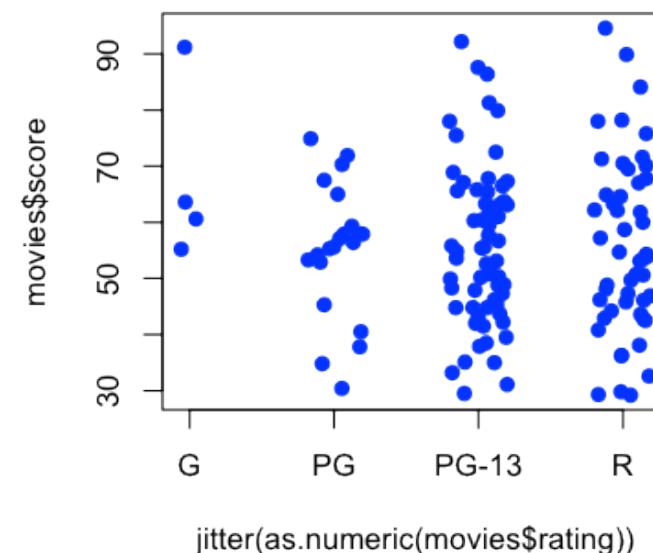
```
download.file("http://www.rossmanchance.com/iscam2/data/movies03RT.txt", destfile=".data/movies.txt")
movies <- read.table("./data/movies.txt", sep="\t", header=T, quote="")
head(movies)
```

|   | X | score            | rating | genre | box.office       | running.time |     |
|---|---|------------------|--------|-------|------------------|--------------|-----|
| 1 | 2 | Fast 2 Furious   | 48.9   | PG-13 | action/adventure | 127.15       | 107 |
| 2 |   | 28 Days Later    | 78.2   | R     | horror           | 45.06        | 113 |
| 3 |   | A Guy Thing      | 39.5   | PG-13 | rom comedy       | 15.54        | 101 |
| 4 |   | A Man Apart      | 42.9   | R     | action/adventure | 26.25        | 110 |
| 5 |   | A Mighty Wind    | 79.9   | PG-13 | comedy           | 17.78        | 91  |
| 6 |   | Agent Cody Banks | 57.9   | PG    | action/adventure | 47.81        | 102 |

<http://www.rossmanchance.com/>

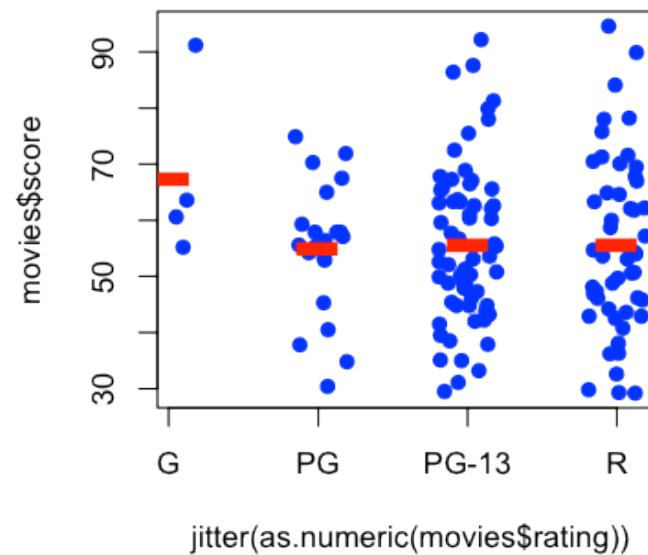
# Rotten tomatoes score vs. rating

```
plot(movies$score ~ jitter(as.numeric(movies$rating)), col="blue", xaxt="n", pch=19)
axis(side=1, at=unique(as.numeric(movies$rating)), labels=unique(movies$rating))
```



# Average score by rating

```
plot(movies$score ~ jitter(as.numeric(movies$rating)), col="blue", xaxt="n", pch=19)
axis(side=1, at=unique(as.numeric(movies$rating)), labels=unique(movies$rating))
meanRatings <- tapply(movies$score, movies$rating, mean)
points(1:4, meanRatings, col="red", pch="-", cex=5)
```



6/20

# Another way to write it down

$$S_i = b_0 + b_1 \mathbb{1}(Ra_i = "PG") + b_2 \mathbb{1}(Ra_i = "PG-13") + b_3 \mathbb{1}(Ra_i = "R") + e_i$$

The notation  $\mathbb{1}(Ra_i = "PG")$  is a logical value that is one if the movie rating is "PG" and zero otherwise.

## Average values

$b_0$  = average of the G movies

$b_0 + b_1$  = average of the PG movies

$b_0 + b_2$  = average of the PG-13 movies

$b_0 + b_3$  = average of the R movies

7/20

# Here is how you do it in R

```
lm1 <- lm(movies$score ~ as.factor(movies$rating))
summary(lm1)
```

Call:

```
lm(formula = movies$score ~ as.factor(movies$rating))
```

Residuals:

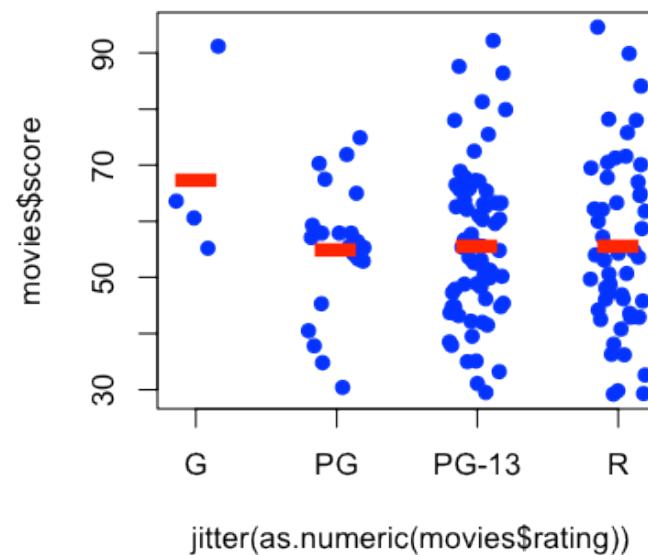
| Min    | 1Q    | Median | 3Q   | Max   |
|--------|-------|--------|------|-------|
| -26.43 | -9.98 | -0.98  | 9.34 | 38.97 |

Coefficients:

|                                | Estimate | Std. Error | t value  | Pr(> t ) |         |   |
|--------------------------------|----------|------------|----------|----------|---------|---|
| (Intercept)                    | 67.65    | 7.19       | 9.40     | <2e-16   | ***     |   |
| as.factor(movies\$rating)PG    | -12.59   | 7.85       | -1.60    | 0.11     |         |   |
| as.factor(movies\$rating)PG-13 | -11.81   | 7.41       | -1.59    | 0.11     |         |   |
| as.factor(movies\$rating)R     | -12.02   | 7.48       | -1.61    | 0.11     |         |   |
| ---                            |          |            |          |          |         |   |
| Signif. codes:                 | 0 '***'  | 0.001 '**' | 0.01 '*' | 0.05 '.' | 0.1 ' ' | 1 |

# Plot fitted values

```
plot(movies$score ~ jitter(as.numeric(movies$rating)), col="blue", xaxt="n", pch=19)
axis(side=1, at=unique(as.numeric(movies$rating)), labels=unique(movies$rating))
points(1:4, lm1$coeff[1] + c(0, lm1$coeff[2:4]), col="red", pch="-", cex=5)
```



# Question 1

## Average values

$b_0$  = average of the G movies

$b_0 + b_1$  = average of the PG movies

$b_0 + b_2$  = average of the PG-13 movies

$b_0 + b_3$  = average of the R movies

**What is the average difference in rating between G and R movies?**

$b_0 + b_3 - b_0 = b_3$

10/20

# Question 1 in R

```
lm1 <- lm(movies$score ~ as.factor(movies$rating))
summary(lm1)
```

Call:

```
lm(formula = movies$score ~ as.factor(movies$rating))
```

Residuals:

| Min    | 1Q    | Median | 3Q   | Max   |
|--------|-------|--------|------|-------|
| -26.43 | -9.98 | -0.98  | 9.34 | 38.97 |

Coefficients:

|                                | Estimate | Std. Error | t value  | Pr(> t ) |         |   |
|--------------------------------|----------|------------|----------|----------|---------|---|
| (Intercept)                    | 67.65    | 7.19       | 9.40     | <2e-16   | ***     |   |
| as.factor(movies\$rating)PG    | -12.59   | 7.85       | -1.60    | 0.11     |         |   |
| as.factor(movies\$rating)PG-13 | -11.81   | 7.41       | -1.59    | 0.11     |         |   |
| as.factor(movies\$rating)R     | -12.02   | 7.48       | -1.61    | 0.11     |         |   |
| ---                            |          |            |          |          |         |   |
| Signif. codes:                 | 0 '***'  | 0.001 '**' | 0.01 '*' | 0.05 '.' | 0.1 ' ' | 1 |

11/20

# Question 1 in R

```
lm1 <- lm(movies$score ~ as.factor(movies$rating))
confint(lm1)
```

|                                | 2.5 %  | 97.5 % |
|--------------------------------|--------|--------|
| (Intercept)                    | 53.42  | 81.875 |
| as.factor(movies\$rating)PG    | -28.11 | 2.928  |
| as.factor(movies\$rating)PG-13 | -26.47 | 2.842  |
| as.factor(movies\$rating)R     | -26.80 | 2.763  |

# Question 2

## Average values

$b_0$  = average of the G movies

$b_0 + b_1$  = average of the PG movies

$b_0 + b_2$  = average of the PG-13 movies

$b_0 + b_3$  = average of the R movies

**What is the average difference in rating between PG – 13 and R movies?**

$b_0 + b_2 - (b_0 + b_3) = b_2 - b_3$

# We could rewrite our model

$$S_i = b_0 + b_1 \mathbb{1}(Ra_i = "G") + b_2 \mathbb{1}(Ra_i = "PG") + b_3 \mathbb{1}(Ra_i = "PG-13") + e_i$$

## Average values

$b_0$  = average of the R movies

$b_0 + b_1$  = average of the G movies

$b_0 + b_2$  = average of the PG movies

$b_0 + b_3$  = average of the PG-13 movies

**What is the average difference in rating between PG-13 and R movies?**

$$b_0 + b_3 - b_0 = b_3$$

# Question 2 in R

```
lm2 <- lm(movies$score ~ relevel(movies$rating, ref = "R"))
summary(lm2)
```

Call:

```
lm(formula = movies$score ~ relevel(movies$rating, ref = "R"))
```

Residuals:

| Min    | 1Q    | Median | 3Q   | Max   |
|--------|-------|--------|------|-------|
| -26.43 | -9.98 | -0.98  | 9.34 | 38.97 |

Coefficients:

|   | Estimate                                  | Std. Error | t value | Pr(> t ) |     |
|---|---|------------|---------|----------|-----|
| (Intercept)                             | 55.630                                    | 2.035      | 27.34   | <2e-16   | *** |
| relevel(movies\$rating, ref = "R")G     | 12.020                                    | 7.476      | 1.61    | 0.11     |     |
| relevel(movies\$rating, ref = "R")PG    | -0.573                                    | 3.741      | -0.15   | 0.88     |     |
| relevel(movies\$rating, ref = "R")PG-13 | 0.205                                     | 2.706      | 0.08    | 0.94     |     |
| ---                                     |   |            |         |          |     |
| Signif. codes:                          | 0 **** 0.001 *** 0.01 ** 0.05 * . 0.1 . 1 |            |         |          |     |

# Question 2 in R

```
lm2 <- lm(movies$score ~ relevel(movies$rating, ref="R"))
confint(lm2)
```

|   | 2.5 %  | 97.5 % |
|---|--------|--------|
| (Intercept)                             | 51.606 | 59.654 |
| relevel(movies\$rating, ref = "R")G     | -2.763 | 26.803 |
| relevel(movies\$rating, ref = "R")PG    | -7.971 | 6.825  |
| relevel(movies\$rating, ref = "R")PG-13 | -5.146 | 5.557  |

# Question 3

$$S_i = b_0 + b_1 \mathbb{1}(Ra_i = "PG") + b_2 \mathbb{1}(Ra_i = "PG-13") + b_3 \mathbb{1}(Ra_i = "R") + e_i$$

## Average values

$b_0$  = average of the G movies

$b_0 + b_1$  = average of the PG movies

$b_0 + b_2$  = average of the PG-13 movies

$b_0 + b_3$  = average of the R movies

**Is there any difference in score between any of the movie ratings?**

17/20

# Question 3 in R

```
lm1 <- lm(movies$score ~ as.factor(movies$rating))
anova(lm1)
```

## Analysis of Variance Table

Response: movies\$score

|                           | Df  | Sum Sq | Mean Sq | F value | Pr(>F) |
|---------------------------|-----|--------|---------|---------|--------|
| as.factor(movies\$rating) | 3   | 570    | 190     | 0.92    | 0.43   |
| Residuals                 | 136 | 28149  | 207     |         |        |

# Sum of squares (G movies)

```
gMovies <- movies[movies$rating=="G",]; xVals <- seq(0.2,0.8,length=4)
plot(xVals,gMovies$score,ylab="Score",xaxt="n",xlim=c(0,1),pch=19)
abline(h=mean(gMovies$score),col="blue",lwd=3); abline(h=mean(movies$score),col="red",lwd=3)
segments(xVals+0.01,rep(mean(gMovies$score),length(xVals)),xVals+0.01,
         rep(mean(movies$score),length(xVals)),col="red",lwd=2)
segments(xVals-0.01,gMovies$score,xVals-0.01,rep(mean(gMovies$score),length(xVals)),col="blue",lwd=3)
```

# Tukey's (honestly significant difference test)

```
lm1 <- aov(movies$score ~ as.factor(movies$rating))  
TukeyHSD(lm1)
```

Tukey multiple comparisons of means

95% family-wise confidence level

Fit: aov(formula = movies\$score ~ as.factor(movies\$rating))

```
$`as.factor(movies$rating)`  
      diff     lwr     upr   p adj  
PG-G    -12.5929 -33.008  7.822 0.3795  
PG-13-G -11.8146 -31.092  7.463 0.3854  
R-G     -12.0200 -31.464  7.424 0.3776  
PG-13-PG  0.7782 -8.615 10.171 0.9964  
R-PG      0.5729 -9.158 10.304 0.9987  
R-PG-13  -0.2054 -7.245  6.834 0.9998
```

[http://en.wikipedia.org/wiki/Tukey's\\_range\\_test](http://en.wikipedia.org/wiki/Tukey's_range_test)

20/20

# Getting Data (Part 1)

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

# Get/set your working directory

Roger's lectures [windows](#), [mac](#) Andrew Jaffe's [lecture notes](#)

```
getwd()
```

```
[1] "/Users/jtleek/Dropbox/Jeff/teaching/2013/coursera/week2/004gettingData1"
```

```
setwd("/Users/jtleek/Dropbox/Jeff/teaching/2013/coursera/week2/004gettingData1/data")
getwd()
```

```
[1] "/Users/jtleek/Dropbox/Jeff/teaching/2013/coursera/week2/004gettingData1/data"
```

Important difference with Windows:

```
setwd("C:\\\\Users\\\\Andrew\\\\Downloads")
```

2/18

# Get/set your working directory (relative paths)

```
getwd()
```

```
[1] "/Users/jtleek/Dropbox/Jeff/teaching/2013/coursera/week2/004gettingData1"
```

```
setwd("./data")
getwd()
```

```
[1] "/Users/jtleek/Dropbox/Jeff/teaching/2013/coursera/week2/004gettingData1/data"
```

```
setwd("../")
getwd()
```

```
[1] "/Users/jtleek/Dropbox/Jeff/teaching/2013/coursera/week2/004gettingData1"
```

3/18

# Get/set your working directory (relative paths)

```
getwd()
```

```
[1] "/Users/jtleek/Dropbox/Jeff/teaching/2013/coursera/week2/004gettingData1"
```

```
setwd("/Users/jtleek/Dropbox/Jeff/teaching/2013/coursera/week2/004gettingData1/data")
getwd()
```

```
[1] "/Users/jtleek/Dropbox/Jeff/teaching/2013/coursera/week2/004gettingData1/data"
```

4/18

# Types of files data may come from

- Tab-delimited text
- Comma-separated text
- Excel file
- JSON File
- HTML/XML file
- Database

# Where you can get data

- From a colleague
- From the web
- From an application programming interface
- By scraping a web page

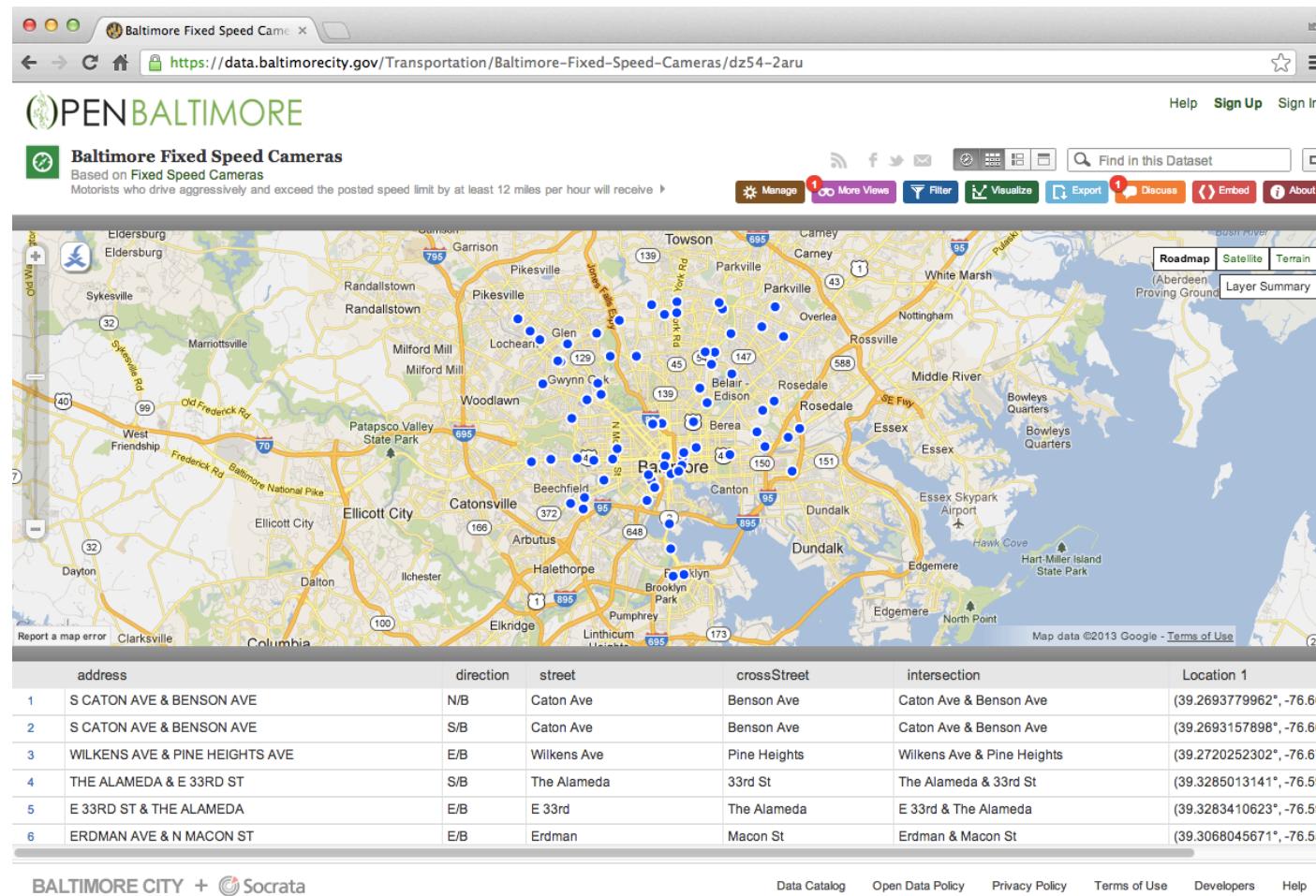
6/18

# Getting data from the internet - download.file()

- Downloads a file from the internet
- Even if you could do this by hand, helps with reproducibility
- Important parameters are *url*, *destfile*, *method*
- Useful for downloading tab-delimited, csv, etc.

7/18

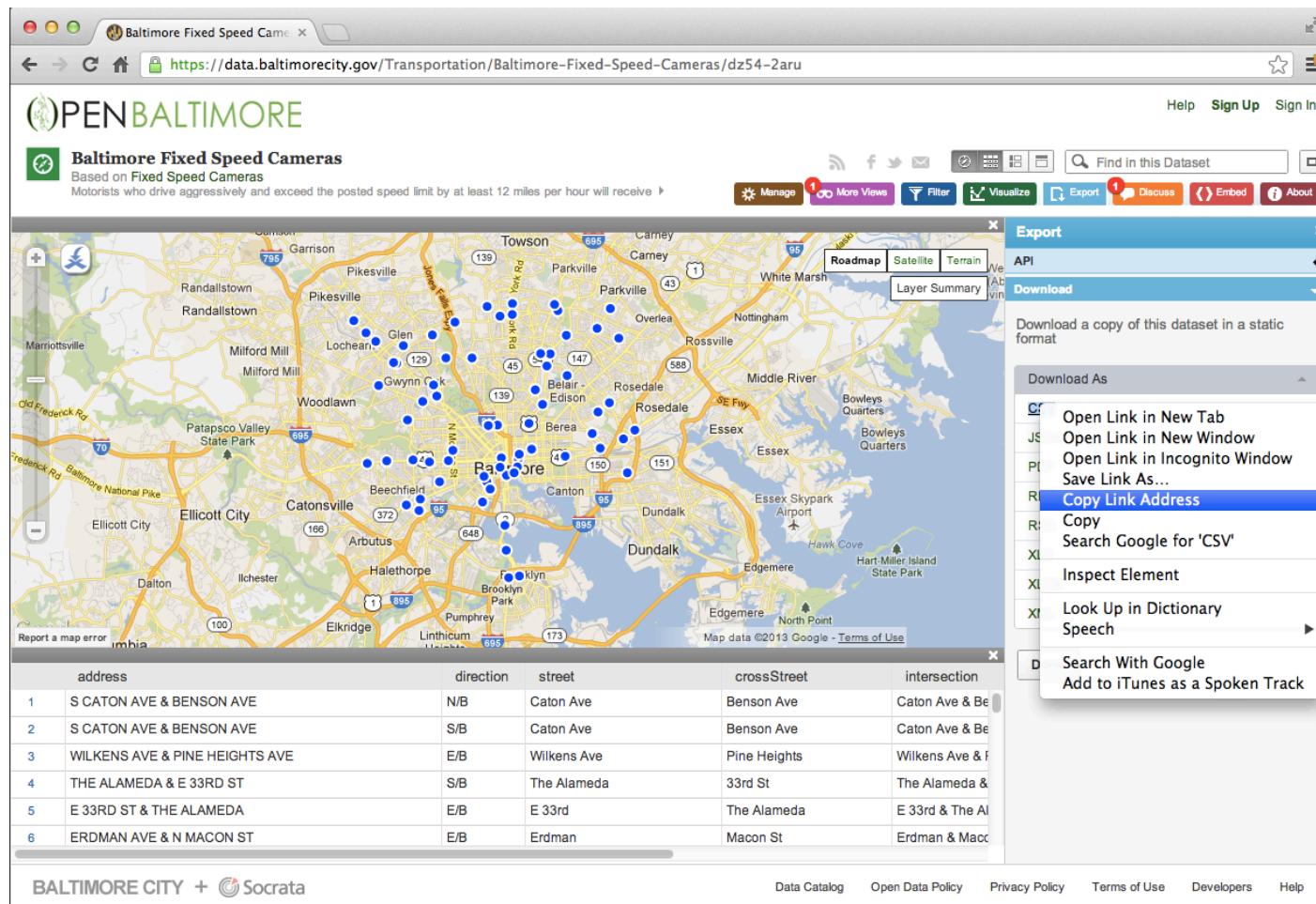
# Example - Baltimore camera data



<https://data.baltimorecity.gov/Transportation/Baltimore-Fixed-Speed-Cameras/dz54-2aru>

8/18

# Example - Baltimore camera data, csv



<https://data.baltimorecity.gov/Transportation/Baltimore-Fixed-Speed-Cameras/dz54-2aru>

9/18

# Download a file from the web

```
fileUrl <- "https://data.baltimorecity.gov/api/views/dz54-2aru/rows.csv?accessType=DOWNLOAD"
download.file(fileUrl, destfile = "./data/cameras.csv", method = "curl")
list.files("./data")
```

```
[1] "camera.json"           "camera.xlsx"          "cameras.csv"
[4] "cameras.rda"           "camerasModified.csv"
```

```
dateDownloaded <- date()
dateDownloaded
```

```
[1] "Sun Jan 27 12:21:15 2013"
```

10/18

# Some notes about download.file()

- If the url starts with *http* you can use download.file()
- If the url starts with *https* on Windows you may be ok
- If the url starts with *https* on Mac you may need to set *method="curl"*
- If the file is big, this might take a while
- Be sure to record when you downloaded.

# Loading data you have saved - `read.table()`

- This is the main function for reading data into R
- Flexible and robust but requires more parameters
- Reads the data into RAM - big data can cause problems
- Important parameters *file*, *header*, *sep*, *row.names*, *nrows*
- Related: *read.csv()*, *read.csv2()*

# Example: Baltimore camera data

```
getwd()
```

```
[1] "/Users/jtleek/Dropbox/Jeff/teaching/2013/coursera/week2/004gettingData1"
```

```
cameraData <- read.table("./data/cameras.csv")
```

```
Error: line 1 did not have 13 elements
```

```
head(cameraData)
```

```
Error: error in evaluating the argument 'x' in selecting a method for  
function 'head': Error: object 'cameraData' not found
```

13/18

# Example: Baltimore camera data

```
getwd()
```

```
[1] "/Users/jtleek/Dropbox/Jeff/teaching/2013/coursera/week2/004gettingData1"
```

```
cameraData <- read.table("./data/cameras.csv", sep=",", header=TRUE)
head(cameraData)
```

|   | address                        | direction                       | street      | crossStreet  |
|---|--------------------------------|---------------------------------|-------------|--------------|
| 1 | S CATON AVE & BENSON AVE       | N/B                             | Caton Ave   | Benson Ave   |
| 2 | S CATON AVE & BENSON AVE       | S/B                             | Caton Ave   | Benson Ave   |
| 3 | WILKENS AVE & PINE HEIGHTS AVE | E/B                             | Wilkens Ave | Pine Heights |
| 4 | THE ALAMEDA & E 33RD ST        | S/B                             | The Alameda | 33rd St      |
| 5 | E 33RD ST & THE ALAMEDA        | E/B                             | E 33rd      | The Alameda  |
| 6 |                                |                                 |             |              |
| 1 | Caton Ave & Benson Ave         | (39.2693779962, -76.6688185297) |             |              |
| 2 | Caton Ave & Benson Ave         | (39.2693157898, -76.6689698176) |             |              |
| 3 | Wilkens Ave & Pine Heights     | (39.2720252302, -76.676960806)  |             |              |
| 4 | The Alameda & 33rd St          | (39.3285013141, -76.5953545714) |             |              |

14/18

# Example: Baltimore camera data

read.csv sets *sep=","* and *header=TRUE*

```
cameraData <- read.csv("./data/cameras.csv")  
head(cameraData)
```

|   | address                        | direction                       | street      | crossStreet  |
|---|--------------------------------|---------------------------------|-------------|--------------|
| 1 | S CATON AVE & BENSON AVE       | N/B                             | Caton Ave   | Benson Ave   |
| 2 | S CATON AVE & BENSON AVE       | S/B                             | Caton Ave   | Benson Ave   |
| 3 | WILKENS AVE & PINE HEIGHTS AVE | E/B                             | Wilkens Ave | Pine Heights |
| 4 | THE ALAMEDA & E 33RD ST        | S/B                             | The Alameda | 33rd St      |
| 5 | E 33RD ST & THE ALAMEDA        | E/B                             | E 33rd      | The Alameda  |
| 6 |                                |                                 |             |              |
| 1 | Caton Ave & Benson Ave         | (39.2693779962, -76.6688185297) |             |              |
| 2 | Caton Ave & Benson Ave         | (39.2693157898, -76.6689698176) |             |              |
| 3 | Wilkens Ave & Pine Heights     | (39.2720252302, -76.676960806)  |             |              |
| 4 | The Alameda & 33rd St          | (39.3285013141, -76.5953545714) |             |              |
| 5 | E 33rd & The Alameda           | (39.3283410623, -76.5953594625) |             |              |
| 6 | Erdman & Macon St              | (39.3068045671, -76.5593167803) |             |              |

# read.xlsx(), read.xlsx2() {xlsx package}

- Reads .xlsx files, but slow
- Important parameters *file*, *sheetIndex*, *sheetIndex*, *rowIndex*, *colIndex*, *header*
- `read.xlsx2()` relies more on low level Java functions so may be a bit faster

# read.xlsx() - Baltimore camera data

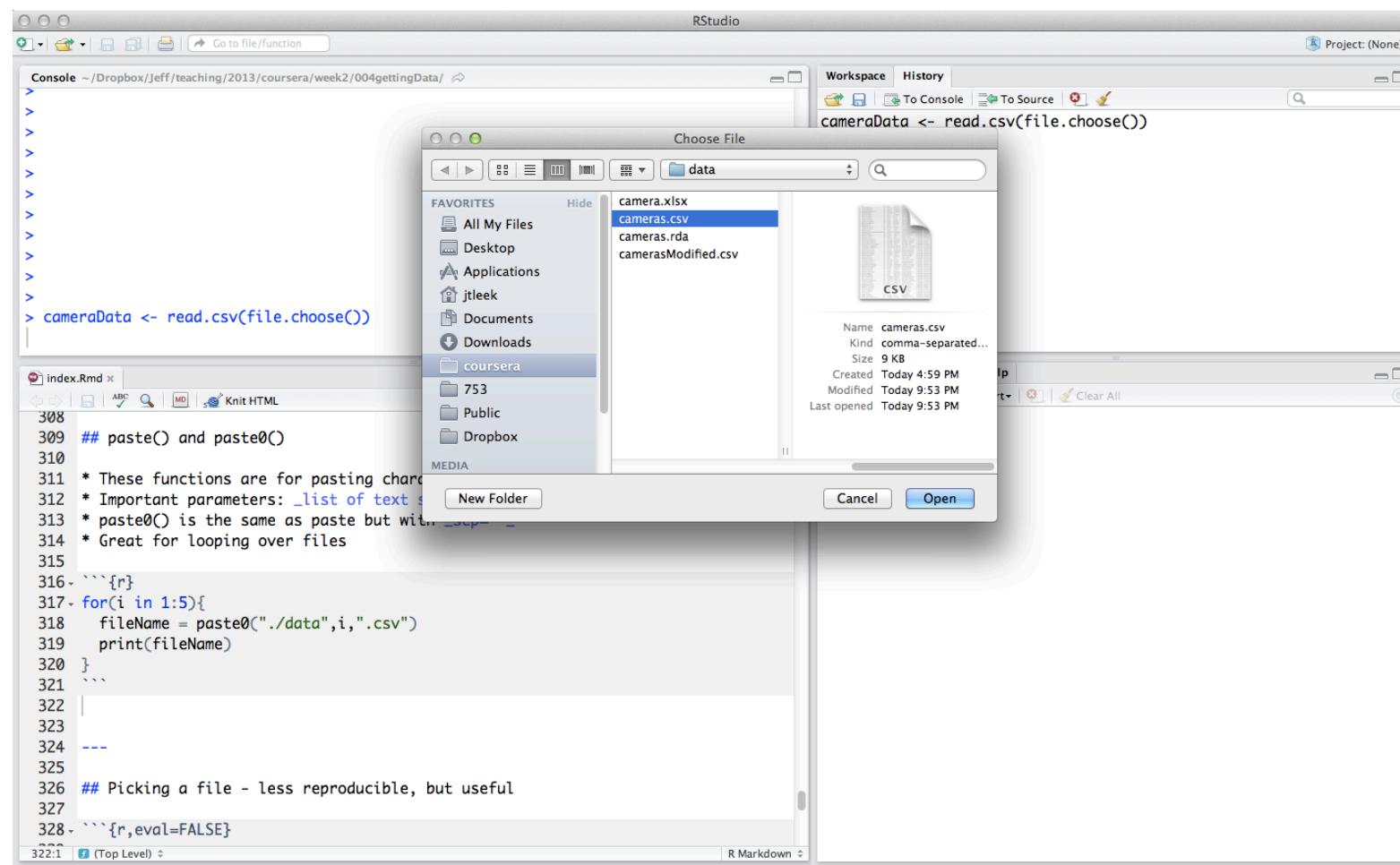
```
library(xlsx)
fileUrl <- "https://data.baltimorecity.gov/api/views/dz54-2aru/rows.xlsx?accessType=DOWNLOAD"
download.file(fileUrl, destfile="./data/camera.xlsx", method="curl")
cameraData <- read.xlsx2("./data/camera.xlsx", sheetIndex=1)
head(cameraData)
```

|   | address                        | direction                       | street      | crossStreet  |
|---|--------------------------------|---------------------------------|-------------|--------------|
| 1 | S CATON AVE & BENSON AVE       | N/B                             | Caton Ave   | Benson Ave   |
| 2 | S CATON AVE & BENSON AVE       | S/B                             | Caton Ave   | Benson Ave   |
| 3 | WILKENS AVE & PINE HEIGHTS AVE | E/B                             | Wilkens Ave | Pine Heights |
| 4 | THE ALAMEDA & E 33RD ST        | S/B                             | The Alameda | 33rd St      |
| 5 | E 33RD ST & THE ALAMEDA        | E/B                             | E 33rd      | The Alameda  |
| 6 |                                |                                 |             |              |
| 1 | Caton Ave & Benson Ave         | (39.2693779962, -76.6688185297) |             |              |
| 2 | Caton Ave & Benson Ave         | (39.2693157898, -76.6689698176) |             |              |
| 3 | Wilkens Ave & Pine Heights     | (39.2720252302, -76.676960806)  |             |              |
| 4 | The Alameda & 33rd St          | (39.3285013141, -76.5953545714) |             |              |
| 5 | E 33rd & The Alameda           | (39.3283410623, -76.5953594625) |             |              |
| 6 | Erdman & Macon St              | (39.3068045671, -76.5593167803) |             |              |

17/18

# Picking a file - less reproducible, but useful

```
cameraData <- read.csv(file.choose())
```



# Getting Data (Part 2)

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

# Interacting more directly with files

- file - open a connection to a text file
- url - open a connection to a url
- gzfile - open a connection to a .gz file
- bzfile - open a connection to a .bz2 file
- *?connections* for more information
- **Remember to close connections**

# readLines() - local file

- `readLines` - a function to read lines of text from a connection
- Important parameters: *con, n, encoding*

```
con <- file("./data/cameras.csv", "r")
cameraData <- read.csv(con)
close(con)
head(cameraData)
```

|   | address                        | direction                       | street      | crossStreet  |
|---|--------------------------------|---------------------------------|-------------|--------------|
| 1 | S CATON AVE & BENSON AVE       | N/B                             | Caton Ave   | Benson Ave   |
| 2 | S CATON AVE & BENSON AVE       | S/B                             | Caton Ave   | Benson Ave   |
| 3 | WILKENS AVE & PINE HEIGHTS AVE | E/B                             | Wilkens Ave | Pine Heights |
| 4 | THE ALAMEDA & E 33RD ST        | S/B                             | The Alameda | 33rd St      |
| 5 | E 33RD ST & THE ALAMEDA        | E/B                             | E 33rd      | The Alameda  |
| 6 |                                |                                 |             |              |
| 1 | Caton Ave & Benson Ave         | (39.2693779962, -76.6688185297) |             |              |
| 2 | Caton Ave & Benson Ave         | (39.2693157898, -76.6689698176) |             |              |
| 3 | Wilkens Ave & Pine Heights     | (39.2720252302, -76.676960806)  |             |              |
| 4 | The Alameda & 33rd St          | (39.3285013141, -76.5953545714) |             |              |

3/13

# readLines() - from the web

```
con <- url("http://simplystatistics.org", "r")
simplyStats <- readLines(con)
close(con)
head(simplyStats)

[1] "<!DOCTYPE html>"
[2] "<html lang=\"en-US\">"
[3] "<head>"
[4] "<meta charset=\"UTF-8\" />"
[5] "<title>Simply Statistics</title>"
[6] "<link rel=\"profile\" href=\"http://gmpg.org/xfn/11\" />"
```

# Reading JSON files {RJSONIO}

```
library(RJSONIO)
fileUrl <- "https://data.baltimorecity.gov/api/views/dz54-2aru/rows.json?accessType=DOWNLOAD"
download.file(fileUrl, destfile = "./data/camera.json", method = "curl")
con = file("./data/camera.json")
jsonCamera = fromJSON(con)
close(con)
head(jsonCamera)
```

```
$meta
$meta$view
$meta$view$id
[1] "dz54-2aru"
```

```
$meta$view$name
[1] "Baltimore Fixed Speed Cameras"
```

```
$meta$view$attribution
[1] "Department of Transportation"
```

```
$meta$view$attributionLink
```

5/13

# Writing data - write.table()

- The opposite of read.table
- Important parameters: *x, file, quote, sep, row.names, col.names*

```
cameraData <- read.csv("./data/cameras.csv")
tmpData <- cameraData[,-1]
write.table(tmpData,file="./data/camerasModified.csv",sep=",")
cameraData2 <- read.csv("./data/camerasModified.csv")
head(cameraData2)
```

|   | direction       | street          | crossStreet  | intersection               |
|---|-----------------|-----------------|--------------|----------------------------|
| 1 | N/B             | Caton Ave       | Benson Ave   | Caton Ave & Benson Ave     |
| 2 | S/B             | Caton Ave       | Benson Ave   | Caton Ave & Benson Ave     |
| 3 | E/B             | Wilkins Ave     | Pine Heights | Wilkins Ave & Pine Heights |
| 4 | S/B             | The Alameda     | 33rd St      | The Alameda & 33rd St      |
| 5 | E/B             | E 33rd          | The Alameda  | E 33rd & The Alameda       |
| 6 |                 |                 |              |                            |
| 1 | (39.2693779962, | -76.6688185297) |              |                            |
| 2 | (39.2693157898, | -76.6689698176) |              |                            |
| 3 | (39.2720252302, | -76.676960806)  |              |                            |

6/13

# Writing data - `save()`, `save.image()`

- `save` is used to save R objects
- Important parameters: *list of objects, file*
- `save.image` saves everything in your working directory

```
cameraData <- read.csv("./data/cameras.csv")
tmpData <- cameraData[,-1]
save(tmpData,cameraData,file="./data/cameras.rda")
```

# Reading saved data - load()

- Opposite of save()
- Important parameters: *file*

```
# Remove everything from the workspace
rm(list=ls())
ls()
```

```
character(0)
```

```
# Load data
load("./data/cameras.rda")
ls()
```

```
[1] "cameraData" "tmpData"
```

8/13

# paste() and paste0()

- These functions are for pasting character strings together.
- Important parameters: *list of text strings, sep*
- paste0() is the same as paste but with *sep=""*
- Great for looping over files
- See also [file.path](#)

```
for(i in 1:5){  
  fileName = paste0("./data", i, ".csv")  
  print(fileName)  
}
```

```
[1] "./data1.csv"  
[1] "./data2.csv"  
[1] "./data3.csv"  
[1] "./data4.csv"  
[1] "./data5.csv"
```

9/13

# Getting data off webpages

The screenshot shows a Google Scholar profile page for Jeff Leek. At the top, there's a photo of Jeff Leek wearing a straw hat and a blue shirt, with a link to "Change photo". Below the photo are his basic details: Assistant Professor of Biostatistics at Johns Hopkins Bloomberg School of Public Health, with links to edit them. He is associated with the fields of Statistics - Computing - Genomics - Personalized Medicine - Scientific Communication. His verified email is listed as jtleek@gmail.com. A link indicates his profile is public.

On the left, there's a "Citation indices" table:

|           | All  | Since 2008 |
|-----------|------|------------|
| Citations | 1285 | 1146       |
| h-index   | 10   | 10         |
| i10-index | 11   | 11         |

Next to it is a bar chart titled "Citations to my articles" showing citation counts from 2005 to 2013. The counts are approximately: 2005 (1), 2006 (2), 2007 (5), 2008 (10), 2009 (15), 2010 (30), 2011 (100), 2012 (200), 2013 (10).

The main content area lists his publications:

| Title / Author  | Cited by | Year |
|---|----------|------|
| Significance analysis of time course microarray experiments<br>JD Storey, W Xiao, JT Leek, RG Tompkins, RW Davis<br>Proceedings of the National Academy of Sciences of the United States of ...                     | 338      | 2005 |
| Capturing heterogeneity in gene expression studies by surrogate variable analysis<br>JT Leek, JD Storey<br>PLoS Genetics 3 (9), e161  | 171      | 2007 |
| EDGE: extraction and analysis of differential gene expression<br>JT Leek, E Monsen, AR Dabney, JD Storey<br>Bioinformatics 22 (4), 507-508  | 140      | 2006 |
| Tackling the widespread and critical impact of batch effects in high-throughput data<br>JT Leek, RB Scharpf, HC Bravo, D Simcha, B Langmead, WE Johnson, D Geman, K ...<br>Nature Reviews Genetics 11 (10), 733-739 | 133      | 2010 |
| The optimal discovery procedure for large-scale significance testing, with applications to comparative microarray experiments<br>JD Storey, JY Dai, JT Leek<br>UW Biostatistics Working Paper Series, 260           | 107      | 2005 |
| Systems-level dynamic analyses of fate change in murine embryonic stem  |          |      |

On the right side of the page, there are two sections: "Follow this author" (which has 5 followers) and "Add co-authors" (listing several names with "Add" buttons). There's also a "Co-authors" section which currently says "No co-authors".

<http://scholar.google.com/citations?user=HI-I6C0AAAAJ&hl=en>

10/13

# Getting data off webpages

```
library(XML)
con = url("http://scholar.google.com/citations?user=HI-I6C0AAAAJ&hl=en")
htmlCode = readLines(con)
close(con)
htmlCode
```

```
[1] "<!DOCTYPE html><html><head><title>Jeff Leek - Google Scholar Citations</title><meta name=\"rob"
```

11/13

# Getting data off webpages

```
html3 <- htmlTreeParse("http://scholar.google.com/citations?user=HI-I6C0AAAAJ&hl=en", useInternalNo  
xpathSApply(html3, "//title", xmlValue)
```

```
[1] "Jeff Leek - Google Scholar Citations"
```

```
xpathSApply(html3, "//td[@id='col-citedby']", xmlValue)
```

```
[1] "Cited by"  "338"      "171"      "140"      "133"      "107"  
[7] "95"        "78"       "78"       "53"       "16"       "10"  
[13] "9"         "9"        "8"        "8"        "6"        "6"  
[19] "6"         "5"        "3"
```

# Further resources

- Packages:
  - [httr](#) - for working with http connections
  - [RMySQL](#) - for interfacing with mySQL
  - [bigmemory](#) - for handling data larger than RAM
  - [RHadoop](#) - for interfacing R and Hadoop (by [Revolution Analytics](#))
  - [foreign](#) - for getting data into R from SAS, SPSS, Octave, etc.
- Reading/writing R videos [Part 1](#), [Part 2](#)

# Getting help

Jeffrey Leek, Assistant Professor of Biostatistics  
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"
- If you figure out the answer and see the same questions on the forum, post the solution you found

# 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
  - Post to [CrossValidated](#)

# 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)
## .Internal(rnorm(n, mean, sd))
## <bytecode: 0x7fc9fa7ce740>
## <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.

# A note on Googling data analysis questions

- The best place to start for general questions is our forum
- [Stackoverflow](#), [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
  - etc.

# Further resources

- Some R resources you might find useful
  - Roger's Computing for Data Analysis [Videos](#) on Youtube
  - A set of [two-minute R tutorials](#)
- Some Data Analysis Resources you might find useful
  - [The Elements of Statistical Learning](#)
  - [Advanced Data Analysis from an Elementary Point of View](#)

# Credits

- Roger's [Getting Help Video](#)
- Inspired by Eric Raymond's "How to ask questions the smart way"

# Hierarchical clustering

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

# Can we find things that are close together?

Clustering organizes things that are **close** into groups

- How do we define close?
- How do we group things?
- How do we visualize the grouping?
- How do we interpret the grouping?

# Hugely important/impactful

The screenshot shows a Google Scholar search results page for the query "cluster analysis". The search bar at the top contains "cluster analysis". The results are filtered by "Web" and show approximately 2,860,000 results found in 0.04 seconds. The results are listed in descending order of relevance. The first result is a book titled "Cluster analysis for applications" by MR Anderberg, published in 1973. The second result is a paper titled "Cluster analysis and display of genome-wide expression patterns" by MB Eisen, PT Spellman, PO Brown, et al., published in 1998. The third result is a paper titled "The application of cluster analysis in strategic management research: an analysis and critique" by DJ Ketchen, CL Shook, published in 1996. The fourth result is a paper titled "A cluster analysis method for grouping means in the analysis of variance" by AJ Scott, M Knott, published in 1974.

cluster analysis – Google Scholar

scholar.google.com/scholar?q=cluster+analysis&btnG=&hl=en&as\_sdt=0%2C21

Web Images More... jtleek@gmail.com

Google cluster analysis

Scholar About 2,860,000 results (0.04 sec) My Citations 0

Articles Legal documents

Any time Since 2013 Since 2012 Since 2009 Custom range...

Sort by relevance Sort by date

include patents  include citations

Create alert

**Cluster analysis for applications**  
MR Anderberg - 1973 - DTIC Document  
Abstract: Cluster analysis is a collective term covering a wide variety of techniques for delineating natural groups or clusters in data sets. This book integrates the necessary elements of data analysis, cluster analysis, and computer implementation to cover the ...  
Cited by 5438 Related articles All 12 versions Cite More▼

**Cluster analysis and display of genome-wide expression patterns**  
MB Eisen, PT Spellman, PO Brown... - Proceedings of the ..., 1998 - National Acad Sciences  
Abstract A system of cluster analysis for genome-wide expression data from DNA microarray hybridization is described that uses standard statistical algorithms to arrange genes according to similarity in pattern of gene expression. The output is displayed graphically, ...  
Cited by 12537 Related articles BL Direct All 259 versions Cite [HTML] from nih.gov

**The application of cluster analysis in strategic management research: an analysis and critique**  
DJ Ketchen, CL Shook - Strategic management journal, 1996 - Wiley Online Library  
Abstract Cluster analysis is a statistical technique that sorts observations into similar sets or groups. The use of cluster analysis presents a complex challenge because it requires several methodological choices that determine the quality of a cluster solution. This paper ...  
Cited by 754 Related articles BL Direct All 3 versions Cite

**A cluster analysis method for grouping means in the analysis of variance**  
AJ Scott, M Knott - Biometrics, 1974 - JSTOR  
It is sometimes useful in an analysis of variance to split the treatments into reasonably homogeneous groups. Multiple comparison procedures are often used for this purpose, but a more direct method is to use the techniques of cluster analysis. This approach is ...  
Cited by 1125 Related articles All 2 versions Cite

[http://scholar.google.com/scholar?hl=en&q=cluster+analysis&btnG=&as\\_sdt=1%2C21&as\\_sdtp=](http://scholar.google.com/scholar?hl=en&q=cluster+analysis&btnG=&as_sdt=1%2C21&as_sdtp=)

3/21

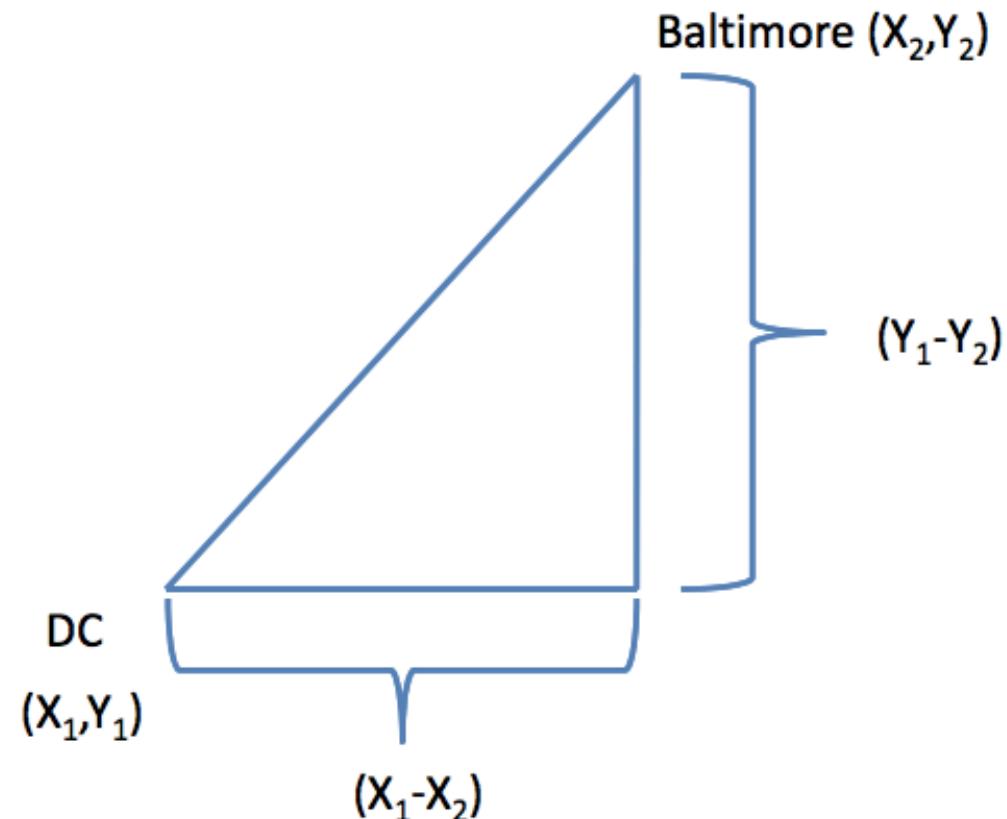
# Hierarchical clustering

- An agglomerative approach
  - Find closest two things
  - Put them together
  - Find next closest
- Requires
  - A defined distance
  - A merging approach
- Produces
  - A tree showing how close things are to each other

# How do we define close?

- Most important step
  - Garbage in -> garbage out
- Distance or similarity
  - Continuous - euclidean distance
  - Continous - correlation similarity
  - Binary - manhattan distance
- Pick a distance/similarity that makes sense for your problem

# Example distances - Euclidean

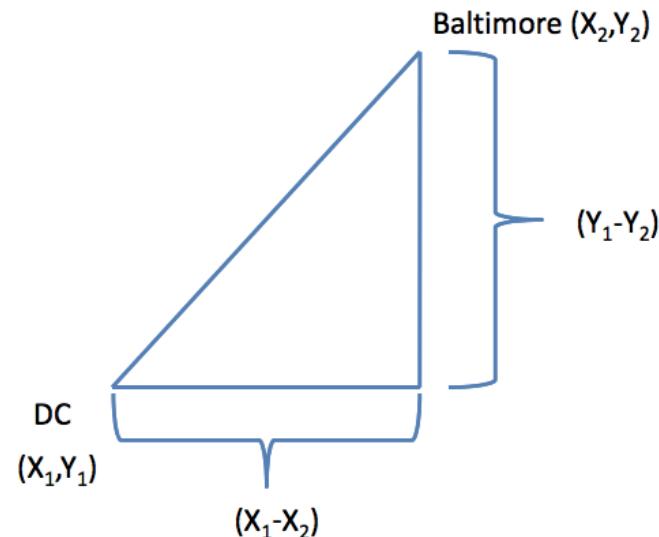


<http://rafalab.jhsph.edu/688/lec/lecture5-clustering.pdf>

6/21

# Example distances - Euclidean

$$\sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$$



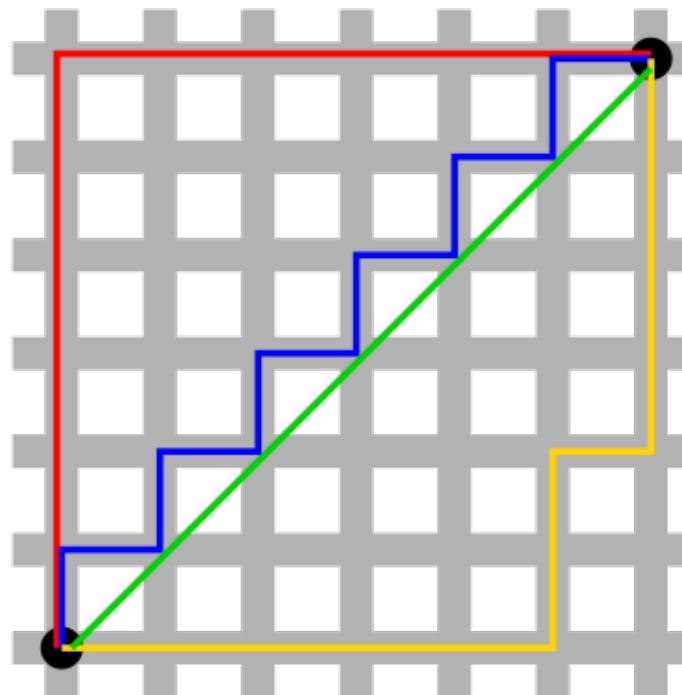
In general:

$$\sqrt{(A_1 - A_2)^2 + (B_1 - B_2)^2 + \dots + (Z_1 - Z_2)^2}$$

<http://rafalab.jhsph.edu/688/lec/lecture5-clustering.pdf>

7/21

# Example distances - Manhattan



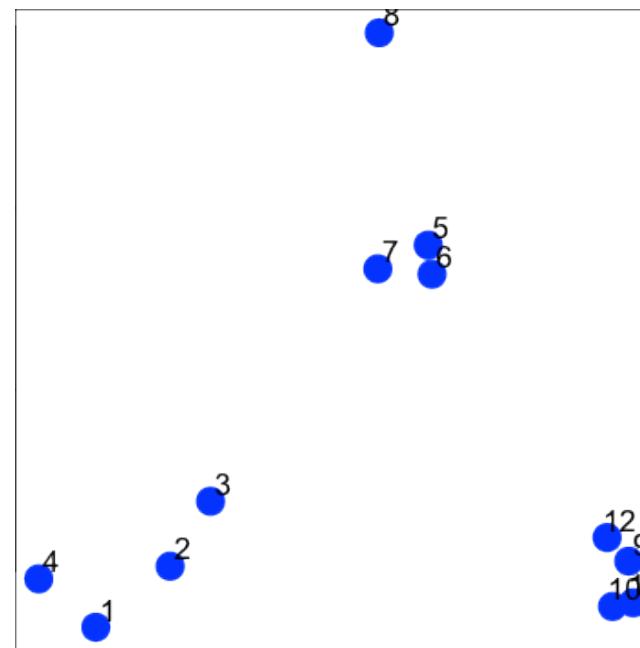
In general:

$$|A_1 - A_2| + |B_1 - B_2| + \dots + |Z_1 - Z_2|$$

[http://en.wikipedia.org/wiki/Taxicab\\_geometry](http://en.wikipedia.org/wiki/Taxicab_geometry)

# Hierarchical 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))
```



9/21

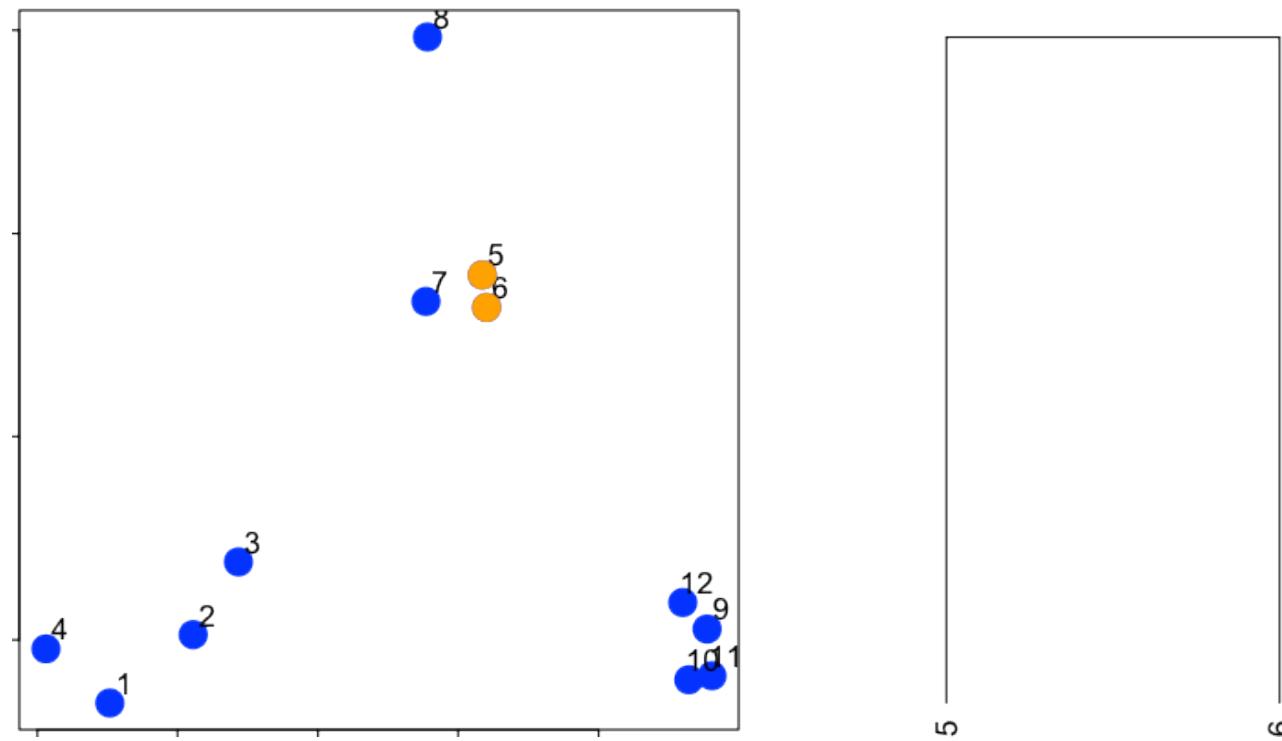
# Hierarchical clustering - dist

- Important parameters:  $x, method$

```
dataFrame <- data.frame(x=x,y=y)
dist(dataFrame)
```

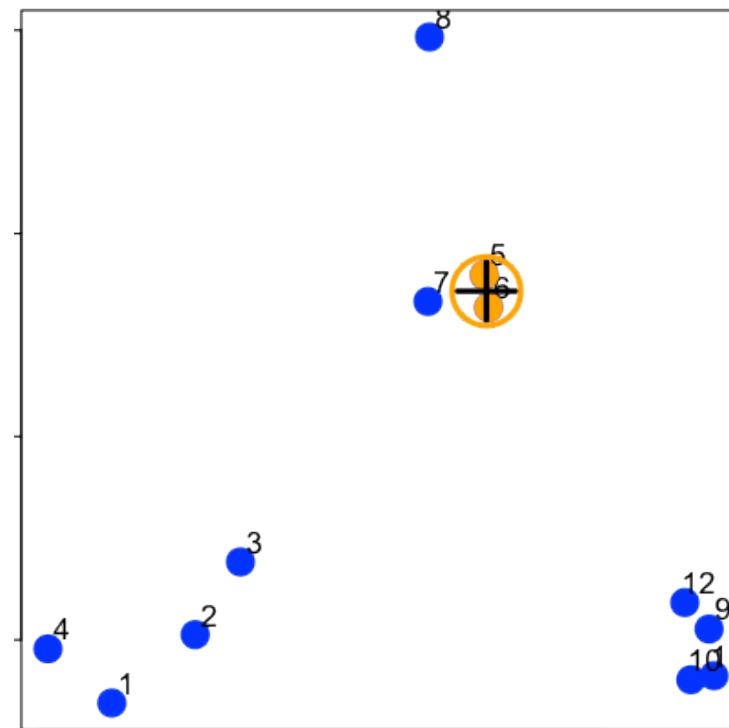
|    | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10      | 11      |
|----|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 2  | 0.34121 |         |         |         |         |         |         |         |         |         |         |
| 3  | 0.57494 | 0.24103 |         |         |         |         |         |         |         |         |         |
| 4  | 0.26382 | 0.52579 | 0.71862 |         |         |         |         |         |         |         |         |
| 5  | 1.69425 | 1.35818 | 1.11953 | 1.80667 |         |         |         |         |         |         |         |
| 6  | 1.65813 | 1.31960 | 1.08339 | 1.78081 | 0.08150 |         |         |         |         |         |         |
| 7  | 1.49823 | 1.16621 | 0.92569 | 1.60132 | 0.21110 | 0.21667 |         |         |         |         |         |
| 8  | 1.99149 | 1.69093 | 1.45649 | 2.02849 | 0.61704 | 0.69792 | 0.65063 |         |         |         |         |
| 9  | 2.13630 | 1.83168 | 1.67836 | 2.35676 | 1.18350 | 1.11500 | 1.28583 | 1.76461 |         |         |         |
| 10 | 2.06420 | 1.76999 | 1.63110 | 2.29239 | 1.23848 | 1.16550 | 1.32063 | 1.83518 | 0.14090 |         |         |
| 11 | 2.14702 | 1.85183 | 1.71074 | 2.37462 | 1.28154 | 1.21077 | 1.37370 | 1.86999 | 0.11624 | 0.08318 |         |
| 12 | 2.05664 | 1.74663 | 1.58659 | 2.27232 | 1.07701 | 1.00777 | 1.17740 | 1.66224 | 0.10849 | 0.19129 | 0.20803 |

# Hierarchical clustering - #1



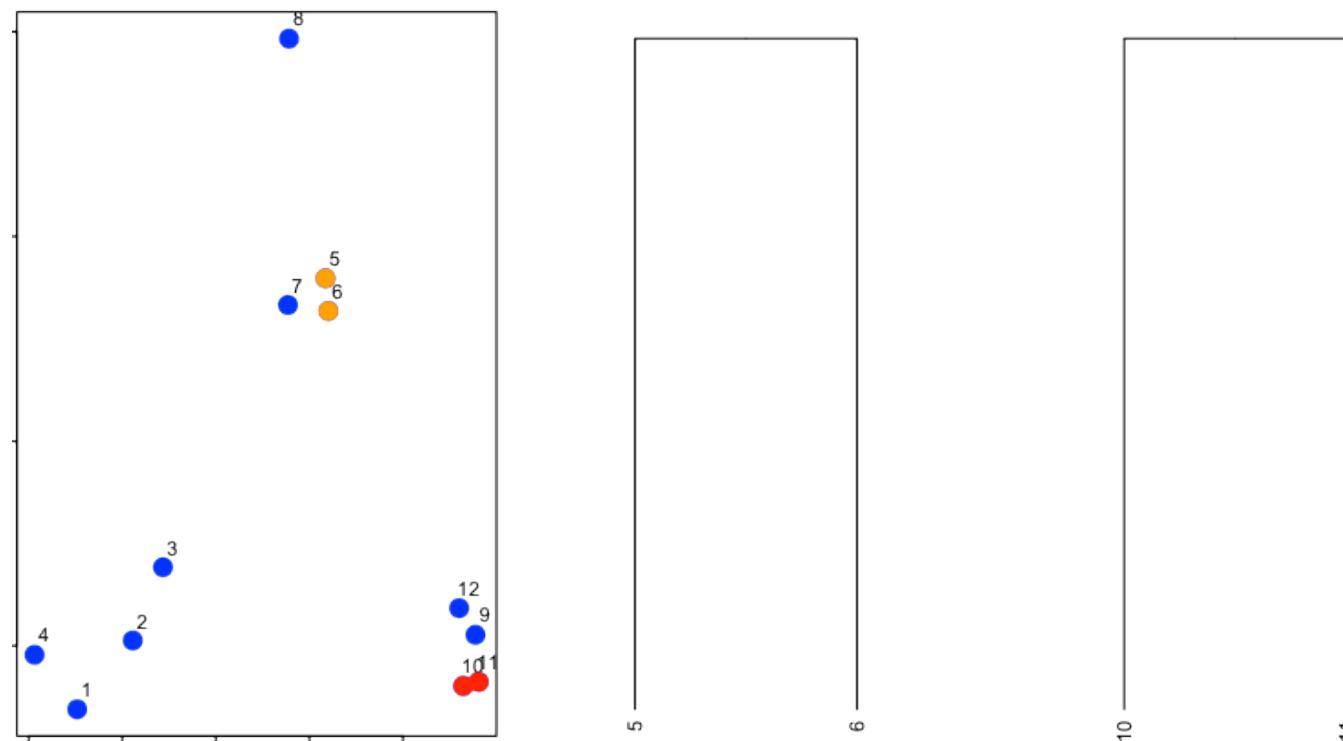
11/21

# Hierarchical clustering - #2



12/21

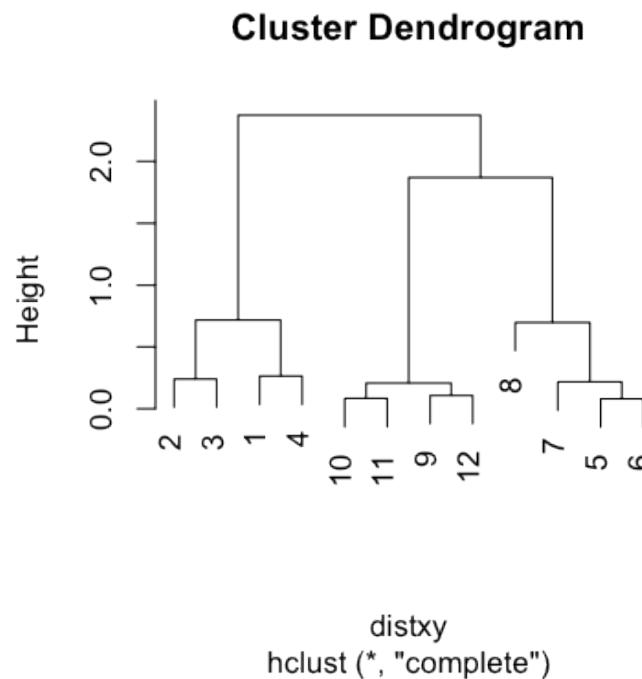
# Hierarchical clustering - #3



13/21

# Hierarchical clustering - hclust

```
dataFrame <- data.frame(x=x,y=y)
distxy <- dist(dataFrame)
hClustering <- hclust(distxy)
plot(hClustering)
```



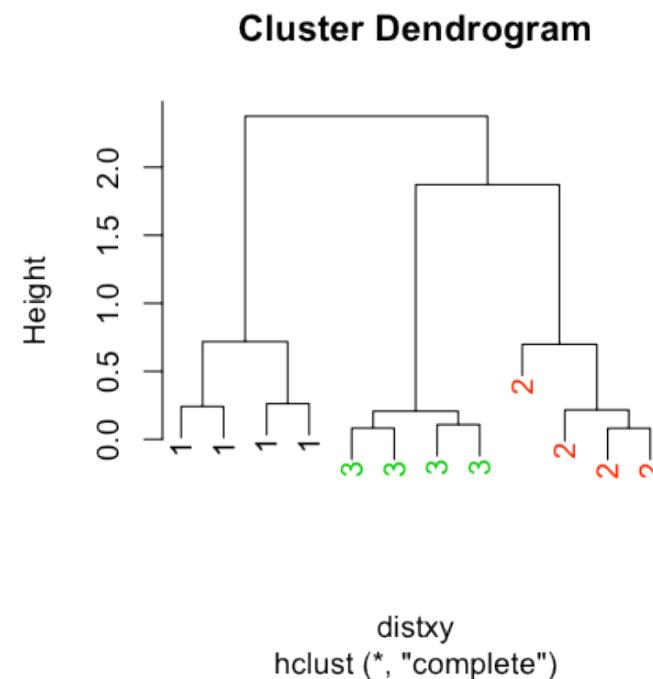
14/21

# Prettier dendograms

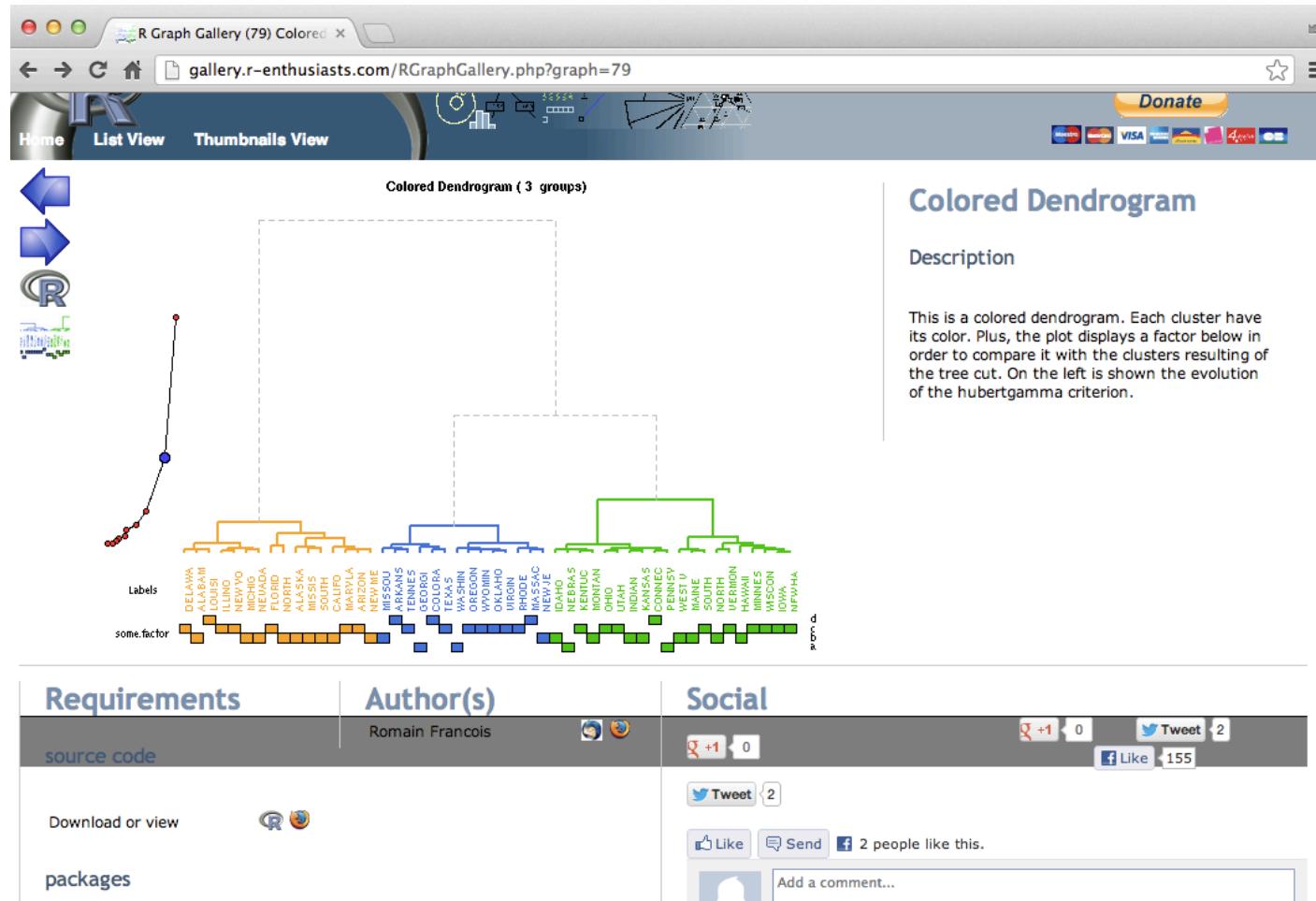
```
myplclust <- function( hclust, lab=hclust$labels, lab.col=rep(1,length(hclust$labels)), hang=0.1, ...
## modifiction of plclust for plotting hclust objects *in colour*!
## Copyright Eva KF Chan 2009
## Arguments:
##   hclust:      hclust object
##   lab:         a character vector of labels of the leaves of the tree
##   lab.col:     colour for the labels; NA=default device foreground colour
##   hang:        as in hclust & plclust
## Side effect:
##   A display of hierarchical cluster with coloured leaf labels.
y <- rep(hclust$height,2); x <- as.numeric(hclust$merge)
y <- y[which(x<0)]; x <- x[which(x<0)]; x <- abs(x)
y <- y[order(x)]; x <- x[order(x)]
plot( hclust, labels=FALSE, hang=hang, ... )
text( x=x, y=y[hclust$order]-(max(hclust$height)*hang),
      labels=lab[hclust$order], col=lab.col[hclust$order],
      srt=90, adj=c(1,0.5), xpd=NA, ... )
}
```

## Hierarchical clustering - hclust

```
dataFrame <- data.frame(x=x,y=y)
distxy <- dist(dataFrame)
hClustering <- hclust(distxy)
myplclust(hClustering,lab=rep(1:3,each=4),lab.col=rep(1:3,each=4))
```



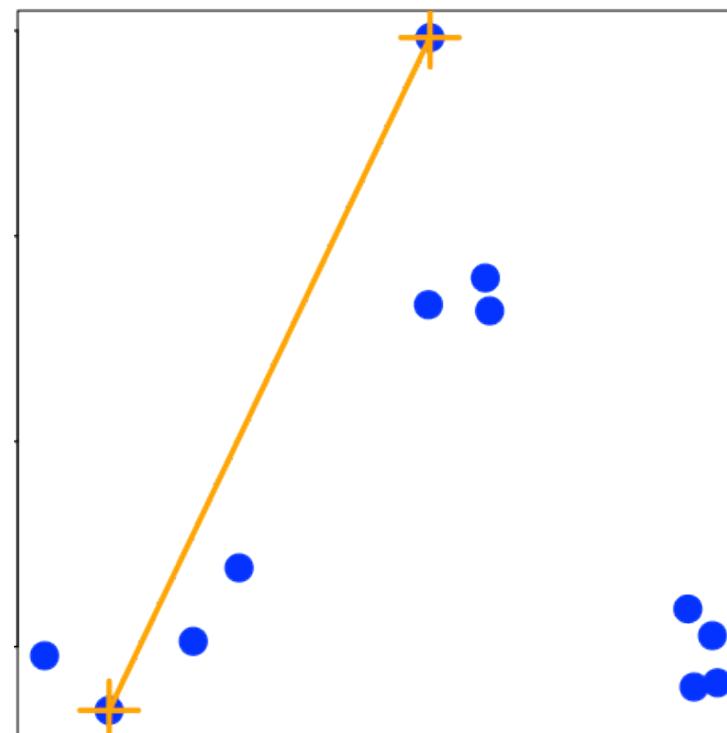
# Even Prettier dendograms



<http://gallery.r-enthusiasts.com/RGraphGallery.php?graph=79>

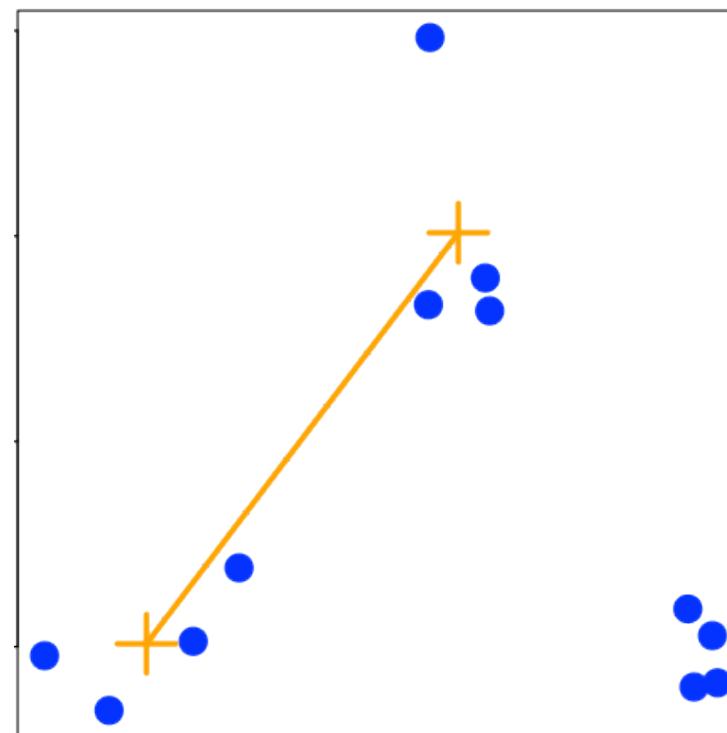
17/21

# Merging points - complete



18/21

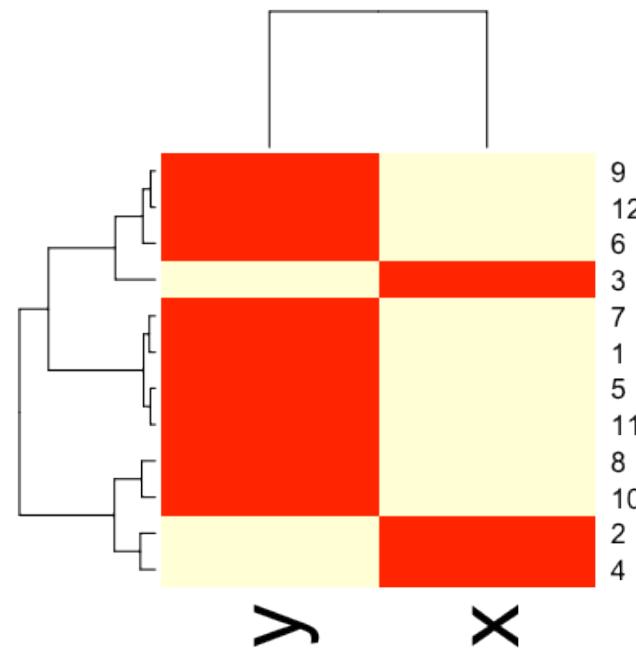
# Merging points - average



19/21

# heatmap()

```
dataFrame <- data.frame(x=x,y=y)
set.seed(143)
dataMatrix <- as.matrix(dataFrame)[sample(1:12),]
heatmap(dataMatrix)
```



20/21

# Notes and further resources

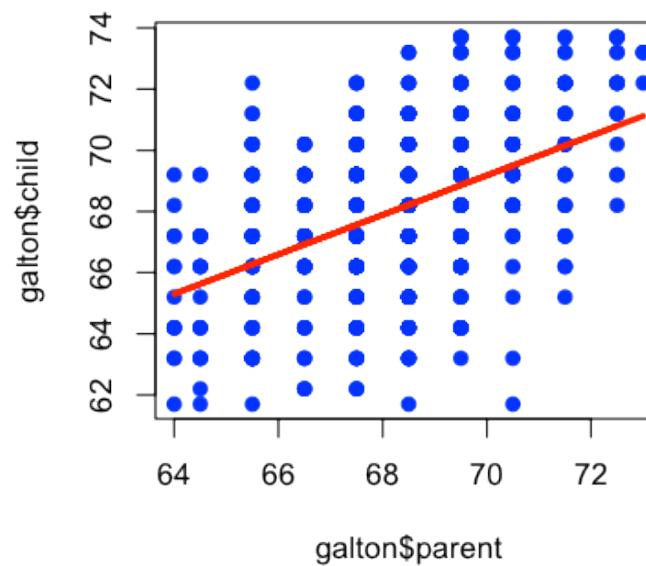
- Gives an idea of the relationships between variables/observations
- The picture may be unstable
  - Change a few points
  - Have different missing values
  - Pick a different distance
  - Change the merging strategy
  - Change the scale of points for one variable
- But it is deterministic
- Choosing where to cut isn't always obvious
- Should be primarily used for exploration
- [Rafa's Distances and Clustering Video](#)
- [Elements of statistical learning](#)

# Inference basics

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

# Fit a line to the Galton Data

```
library(UsingR); data(galton);
plot(galton$parent,galton$child,pch=19,col="blue")
lm1 <- lm(galton$child ~ galton$parent)
lines(galton$parent,lm1$fitted,col="red",lwd=3)
```



2/21

# Fit a line to the Galton Data

```
lm1
```

Call:

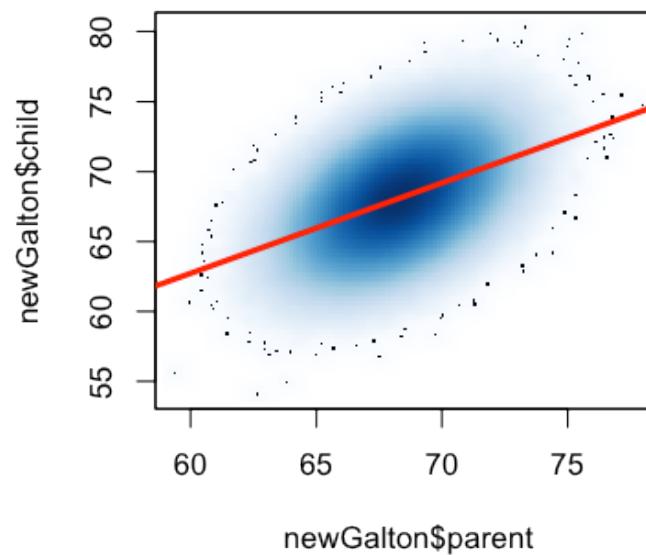
```
lm(formula = galton$child ~ galton$parent)
```

Coefficients:

| (Intercept) | galton\$parent |
|-------------|----------------|
| 23.942      | 0.646          |

# Create a "population" of 1 million families

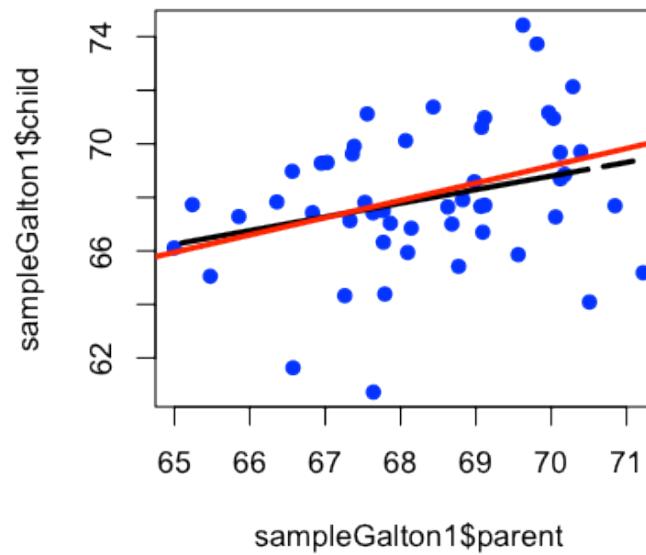
```
newGalton <- data.frame(parent=rep(NA, 1e6), child=rep(NA, 1e6))
newGalton$parent <- rnorm(1e6, mean=mean(galton$parent), sd=sd(galton$parent))
newGalton$child <- lm1$coeff[1] + lm1$coeff[2]*newGalton$parent + rnorm(1e6, sd=sd(lm1$residuals))
smoothScatter(newGalton$parent, newGalton$child)
abline(lm1, col="red", lwd=3)
```



4/21

# Let's take a sample

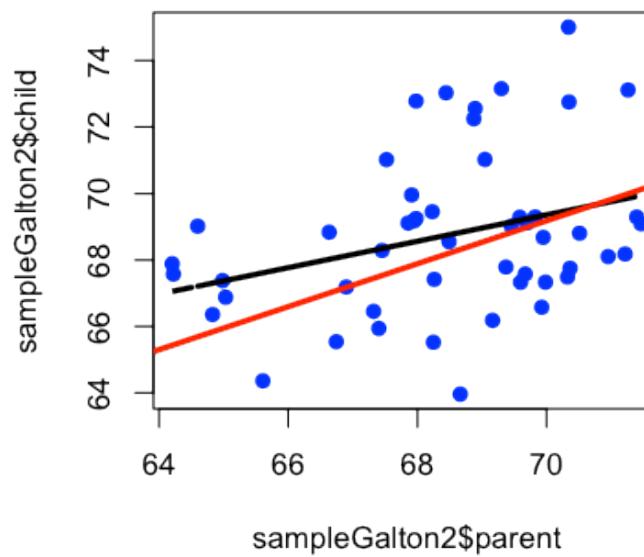
```
set.seed(134325); sampleGalton1 <- newGalton[sample(1:1e6, size=50, replace=F), ]  
sampleLm1 <- lm(sampleGalton1$child ~ sampleGalton1$parent)  
plot(sampleGalton1$parent, sampleGalton1$child, pch=19, col="blue")  
lines(sampleGalton1$parent, sampleLm1$fitted, lwd=3, lty=2)  
abline(lm1, col="red", lwd=3)
```



5/21

# Let's take another sample

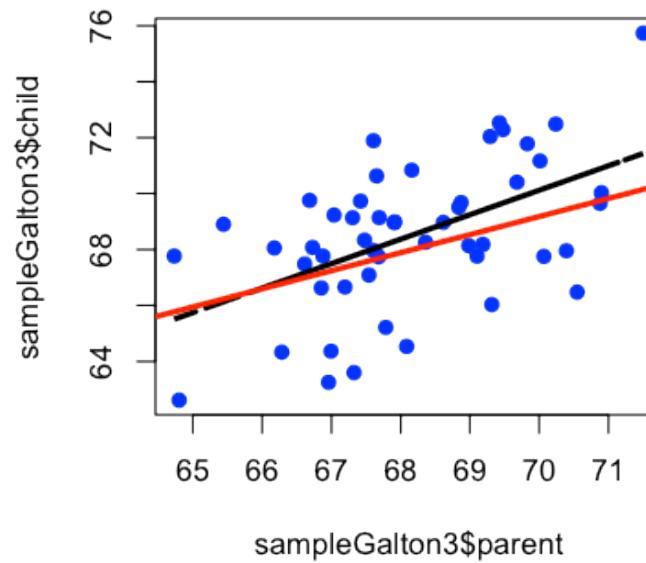
```
sampleGalton2 <- newGalton[sample(1:1e6, size=50, replace=F), ]  
sampleLm2 <- lm(sampleGalton2$child ~ sampleGalton2$parent)  
plot(sampleGalton2$parent, sampleGalton2$child, pch=19, col="blue")  
lines(sampleGalton2$parent, sampleLm2$fitted, lwd=3, lty=2)  
abline(lm1, col="red", lwd=3)
```



6/21

# Let's take another sample

```
sampleGalton3 <- newGalton[sample(1:1e6, size=50, replace=F), ]  
sampleLm3 <- lm(sampleGalton3$child ~ sampleGalton3$parent)  
plot(sampleGalton3$parent, sampleGalton3$child, pch=19, col="blue")  
lines(sampleGalton3$parent, sampleLm3$fitted, lwd=3, lty=2)  
abline(lm1, col="red", lwd=3)
```



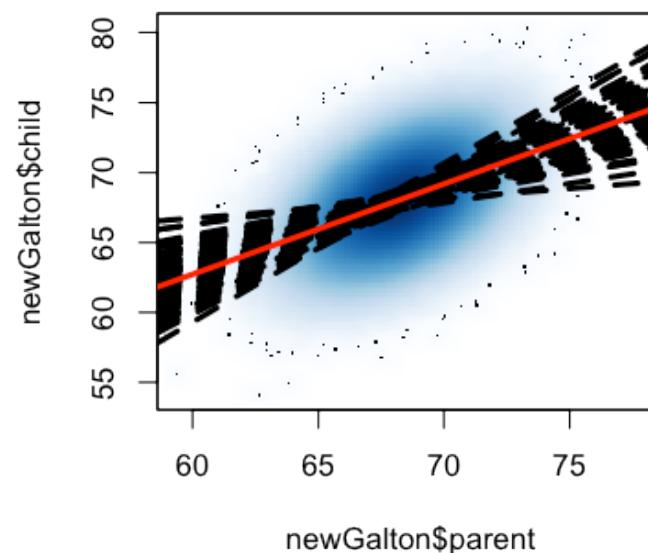
7/21

# Many samples

```
sampleLm <- vector(100, mode="list")
for(i in 1:100){
  sampleGalton <- newGalton[sample(1:1e6, size=50, replace=F), ]
  sampleLm[[i]] <- lm(sampleGalton$child ~ sampleGalton$parent)
}
```

# Many samples

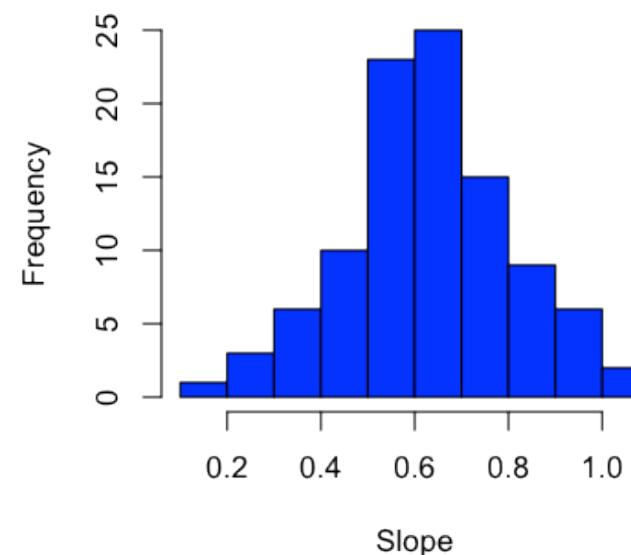
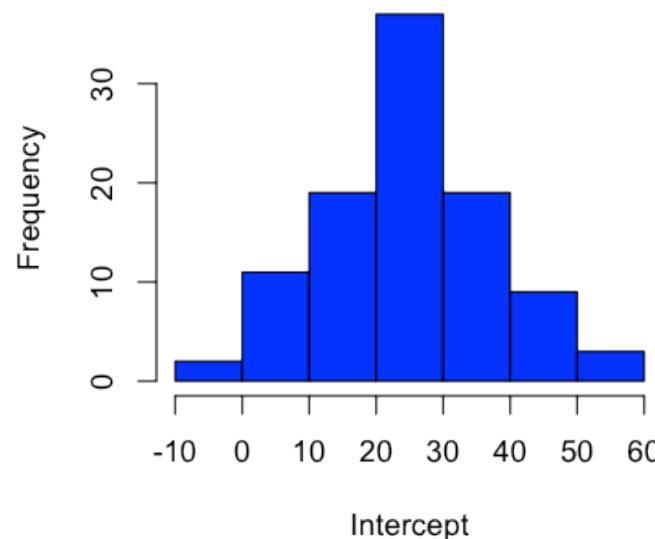
```
smoothScatter(newGalton$parent,newGalton$child)
for(i in 1:100){abline(sampleLm[[i]],lwd=3,lty=2)}
abline(lm1,col="red",lwd=3)
```



9/21

# Histogram of estimates

```
par(mfrow=c(1,2))
hist(sapply(sampleLm,function(x){coef(x)[1]}),col="blue",xlab="Intercept",main="")
hist(sapply(sampleLm,function(x){coef(x)[2]}),col="blue",xlab="Slope",main="")
```



# Distribution of coefficients

From the [central limit theorem](#) it turns out that in many cases:

$$\hat{b}_0 \sim N(b_0, Var(\hat{b}_0))$$

$$\hat{b}_1 \sim N(b_1, Var(\hat{b}_1))$$

which we can estimate with:

$$\hat{b}_0 \approx N(b_0, \hat{Var}(\hat{b}_0))$$

$$\hat{b}_1 \approx N(b_1, \hat{Var}(\hat{b}_1))$$

$\sqrt{\hat{Var}(\hat{b}_0)}$  is the "standard error" of the estimate  $\hat{b}_0$  and is abbreviated *S.E.*( $\hat{b}_0$ )

# Estimating the values in R

```
sampleGalton4 <- newGalton[sample(1:1e6, size=50, replace=F),]  
sampleLm4 <- lm(sampleGalton4$child ~ sampleGalton4$parent)  
summary(sampleLm4)
```

Call:

```
lm(formula = sampleGalton4$child ~ sampleGalton4$parent)
```

Residuals:

| Min    | 1Q     | Median | 3Q    | Max   |
|--------|--------|--------|-------|-------|
| -4.360 | -1.610 | -0.289 | 2.020 | 4.387 |

Coefficients:

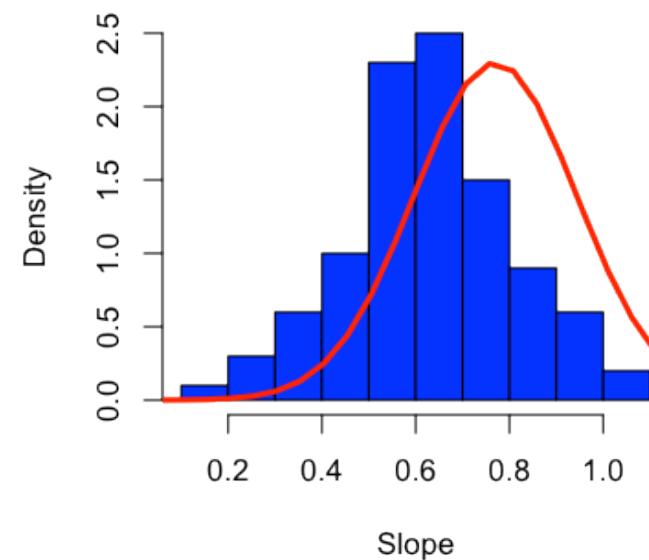
|                       | Estimate | Std. Error | t value | Pr(> t )    |
|-----------------------|----------|------------|---------|-------------|
| (Intercept)           | 15.863   | 11.773     | 1.35    | 0.18        |
| sampleGalton4\$parent | 0.770    | 0.174      | 4.43    | 5.4e-05 *** |
| ---                   |          |            |         |             |
| Signif. codes:        | 0 ***    | 0.001 **   | 0.01 *  | 0.05 .      |
|                       | '        | '          | '       | '           |
|                       | '        | '          | '       | '           |

Residual standard error: 2.29 on 48 degrees of freedom

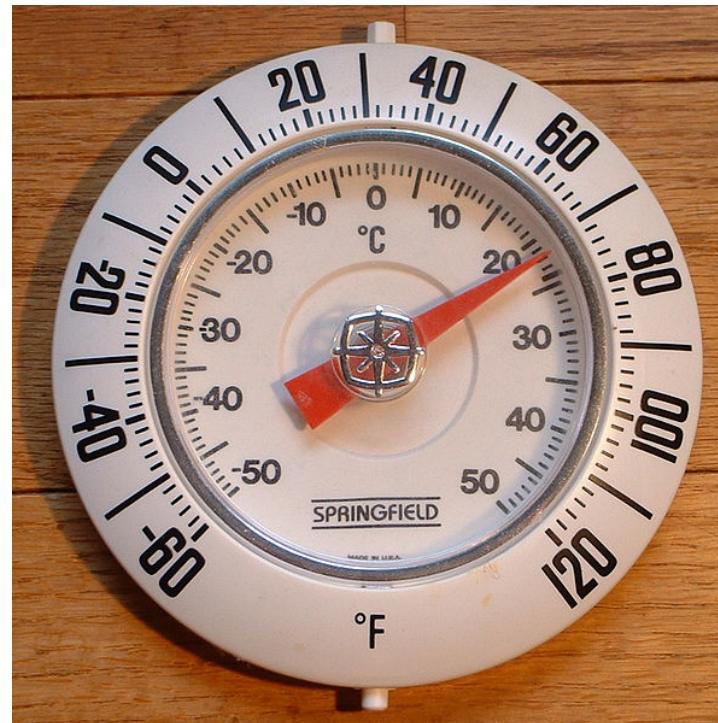
12/21

# Estimating the values in R

```
hist(sapply(sampleLm,function(x){coef(x)[2]}),col="blue",xlab="Slope",main="",freq=F)
lines(seq(0,5,length=100),dnorm(seq(0,5,length=100),mean=coef(summary(sampleLm4))[2],
sd=summary(sampleLm4)$coeff[2,2]),lwd=3,col="red")
```



# Why do we standardize?



$$K^\circ = C^\circ + 273.15$$

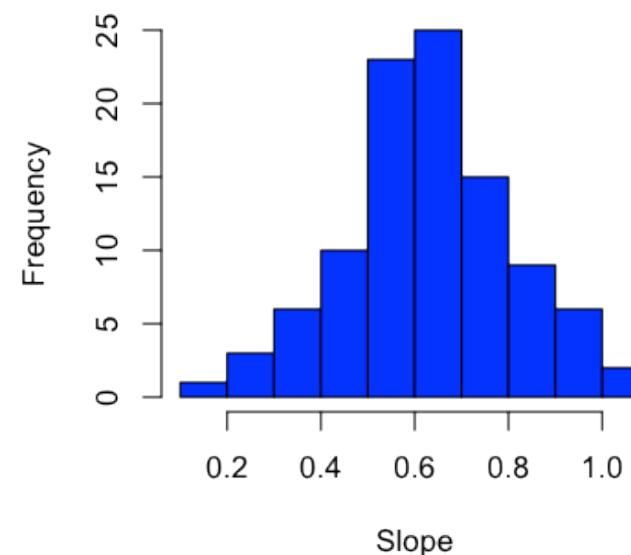
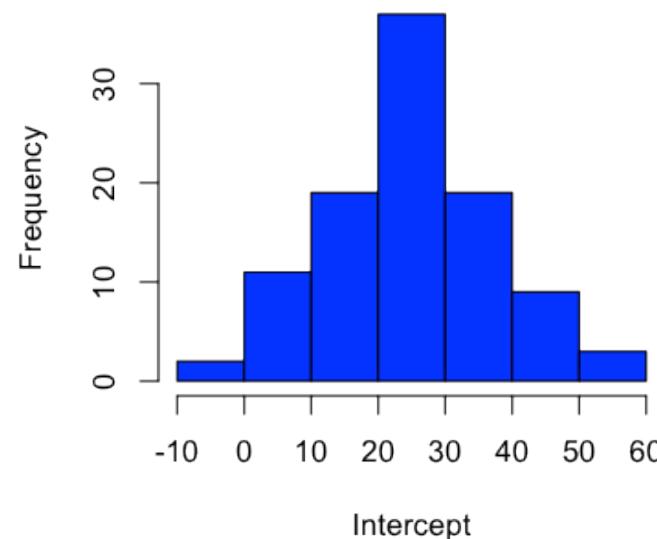
$$K^\circ = \frac{F^\circ + 459.67}{1.8}$$

<http://en.wikipedia.org/wiki/Kelvin>

14/21

# Why do we standardize?

```
par(mfrow=c(1,2))
hist(sapply(sampleLm,function(x){coef(x)[1]}),col="blue",xlab="Intercept",main="")
hist(sapply(sampleLm,function(x){coef(x)[2]}),col="blue",xlab="Slope",main="")
```



# Standardized coefficients

$$\hat{b}_0 \approx N(b_0, \hat{Var}(\hat{b}_0))$$

$$\hat{b}_1 \approx N(b_1, \hat{Var}(\hat{b}_1))$$

and

$$\frac{\hat{b}_0 - b_0}{S.E.(\hat{b}_0)} \sim t_{n-2}$$

$$\frac{\hat{b}_1 - b_1}{S.E.(\hat{b}_1)} \sim t_{n-2}$$

Degrees of Freedom  $\approx$  number of samples - number of things you estimated.

# $t_{n-2}$ versus $N(0, 1)$

```
x <- seq(-5,5,length=100)
plot(x,dnorm(x),type="l",lwd=3)
lines(x,dt(x,df=3),lwd=3,col="red")
lines(x,dt(x,df=10),lwd=3,col="blue")
```

17/21

# Confidence intervals

We have an estimate  $\hat{b}_1$  and we want to know something about how good our estimate is.

One way is to create a "level  $\alpha$  confidence interval".

A confidence interval will include the real parameter  $\alpha$  percent of the time in repeated studies.

# Confidence intervals

$$(\hat{b}_1 + T_{\alpha/2} \times S.E.(\hat{b}_1), \hat{b}_1 - T_{\alpha/2} \times S.E.(\hat{b}_1))$$

```
summary(sampleLm4)$coeff
```

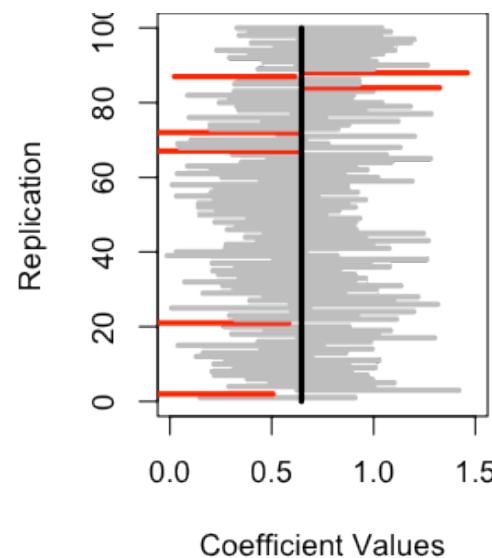
|                       | Estimate | Std. Error | t value | Pr(> t )  |
|-----------------------|----------|------------|---------|-----------|
| (Intercept)           | 15.8632  | 11.7726    | 1.347   | 1.842e-01 |
| sampleGalton4\$parent | 0.7698   | 0.1736     | 4.434   | 5.364e-05 |

```
confint(sampleLm4, level=0.95)
```

|                       | 2.5 %   | 97.5 % |
|-----------------------|---------|--------|
| (Intercept)           | -7.8072 | 39.534 |
| sampleGalton4\$parent | 0.4208  | 1.119  |

# Confidence intervals

```
par(mar=c(4,4,0,2));plot(1:10,type="n",xlim=c(0,1.5),ylim=c(0,100),
                           xlab="Coefficient Values",ylab="Replication")
for(i in 1:100){
  ci <- confint(sampleLm[[i]]); color="red";
  if((ci[2,1] < lm1$coeff[2]) & (lm1$coeff[2] < ci[2,2])){color = "grey"}
  segments(ci[2,1],i,ci[2,2],i,col=color,lwd=3)
}
lines(rep(lm1$coeff[2],100),seq(0,100,length=100),lwd=3)
```



20/21

# How you report the inference

```
sampleLm4$coeff
```

```
(Intercept) sampleGalton4$parent  
15.8632      0.7698
```

```
confint(sampleLm4, level=0.95)
```

|                       | 2.5 %   | 97.5 % |
|-----------------------|---------|--------|
| (Intercept)           | -7.8072 | 39.534 |
| sampleGalton4\$parent | 0.4208  | 1.119  |

A one inch increase in parental height is associated with a 0.77 inch increase in child's height (95% CI: 0.42-1.12 inches).

# K-means clustering

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

# Can we find things that are close together?

- How do we define close?
- How do we group things?
- How do we visualize the grouping?
- How do we interpret the grouping?

2/14

# How do we define close?

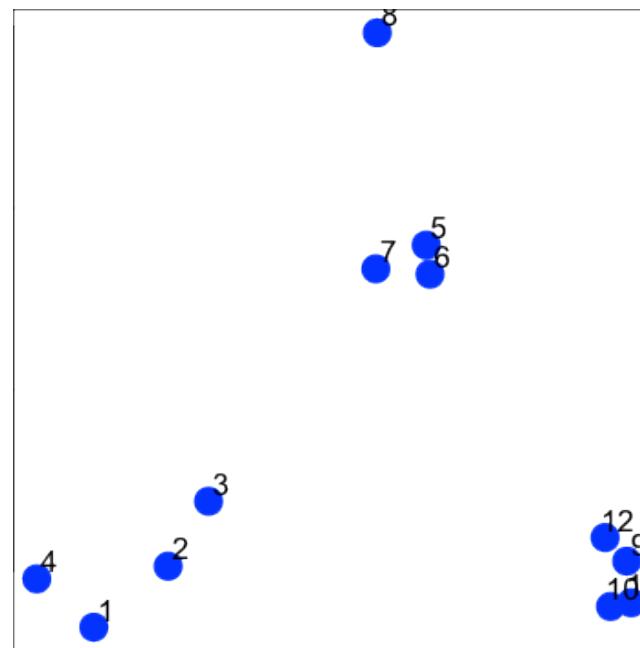
- Most important step
  - Garbage in -> garbage out
- Distance or similarity
  - Continuous - euclidean distance
  - Continous - correlation similarity
  - Binary - manhattan distance
- Pick a distance/similarity that makes sense for your problem

# K-means clustering

- A partitioning approach
  - Fix a number of clusters
  - Get "centroids" of each cluster
  - Assign things to closest centroid
  - Recalculate centroids
- Requires
  - A defined distance metric
  - A number of clusters
  - An initial guess as to cluster centroids
- Produces
  - Final estimate of cluster centroids
  - An assignment of each point to clusters

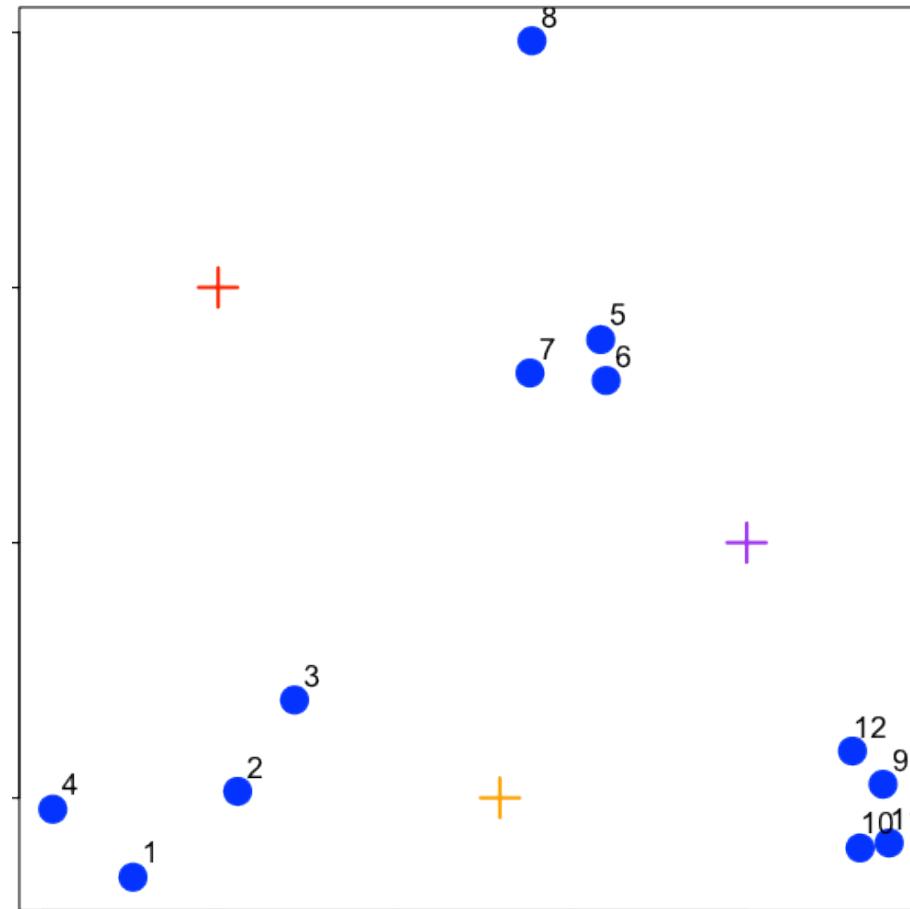
# K-means clustering - example

```
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))
```



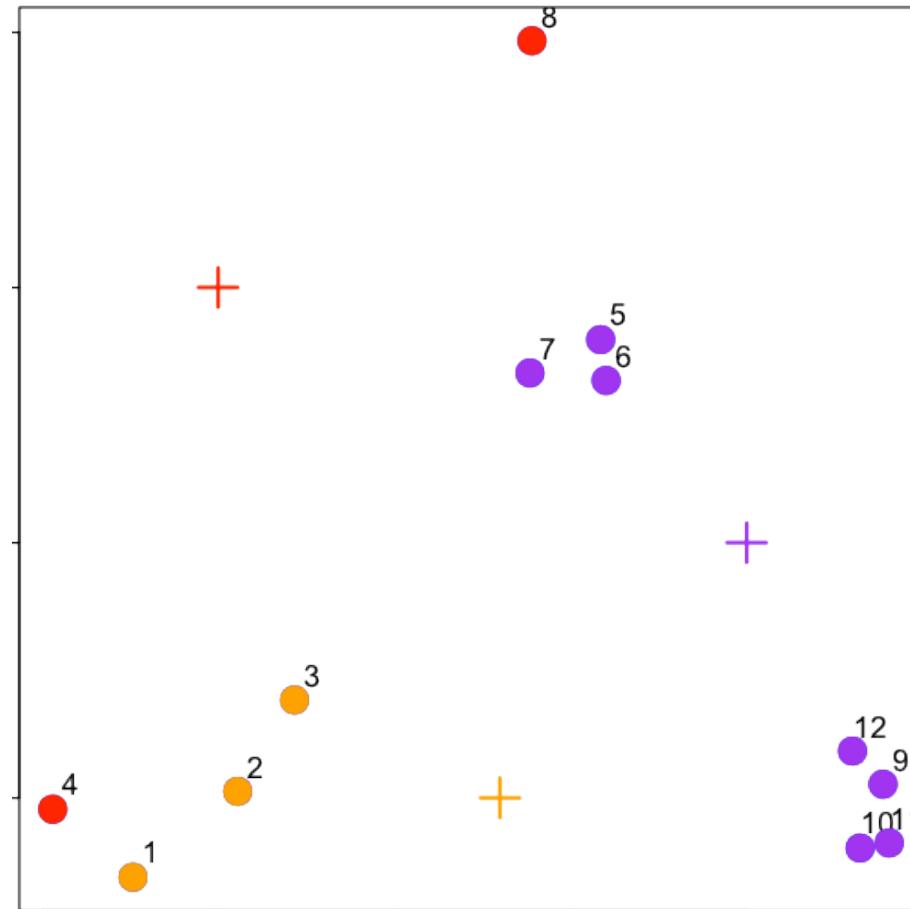
5/14

# K-means clustering - starting centroids



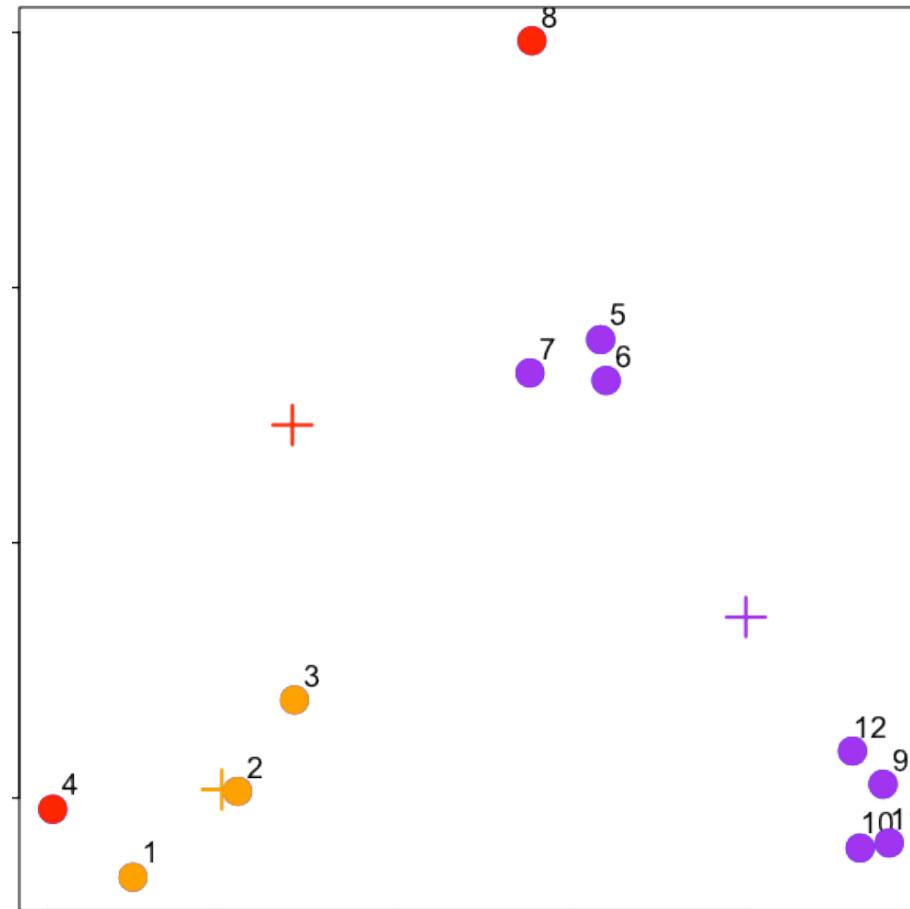
6/14

# K-means clustering - assign to closest centroid



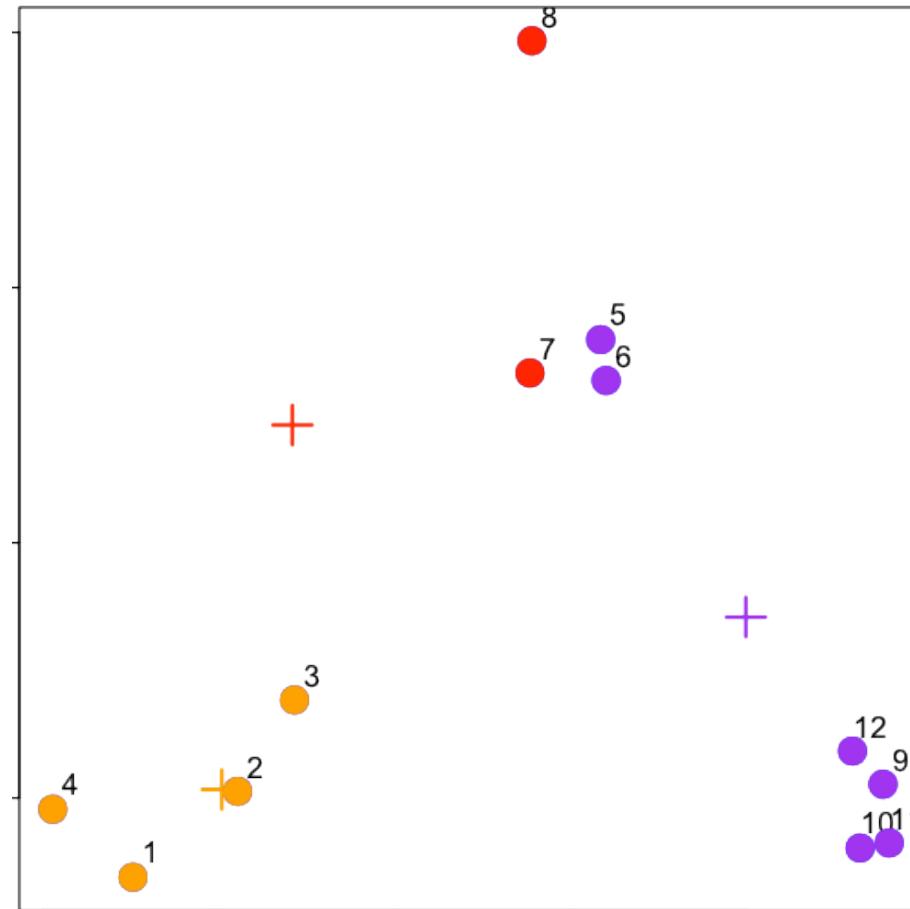
7/14

# K-means clustering - recalculate centroids



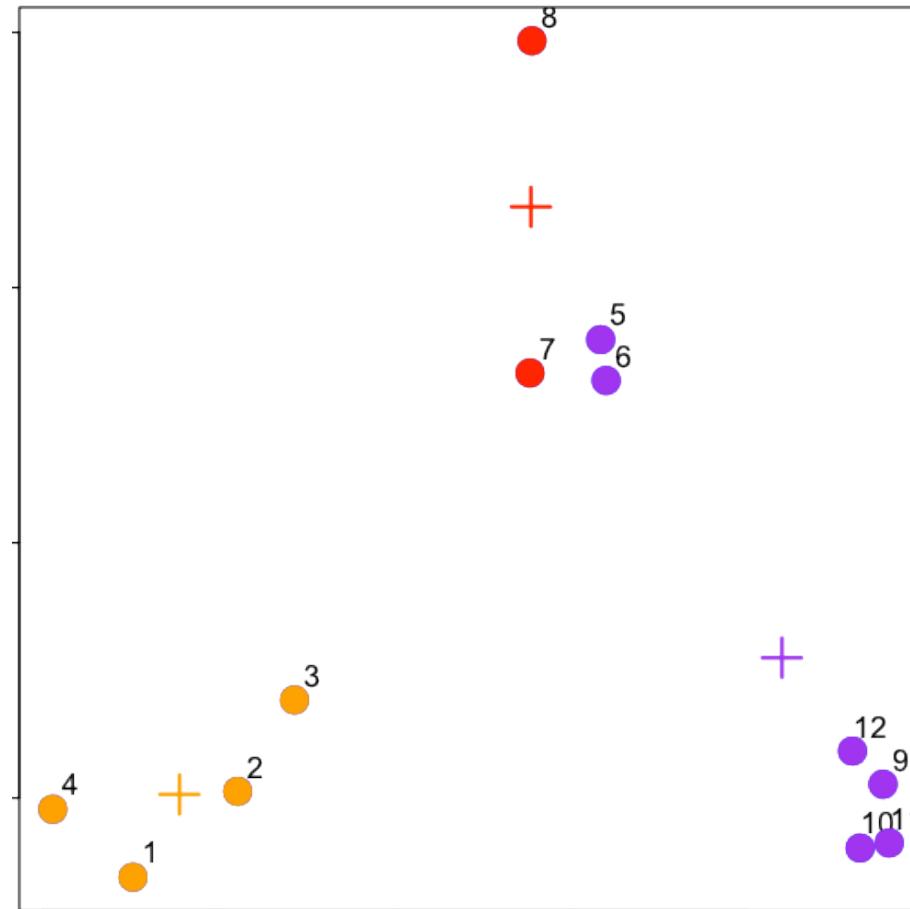
8/14

# K-means clustering - reassign values



9/14

# K-means clustering - update centroids



10/14

# kmeans()

- Important parameters: *x, centers, iter.max, nstart*

```
dataFrame <- data.frame(x,y)
kmeansObj <- kmeans(dataFrame,centers=3)
names(kmeansObj)
```

```
[1] "cluster"      "centers"       "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"
```

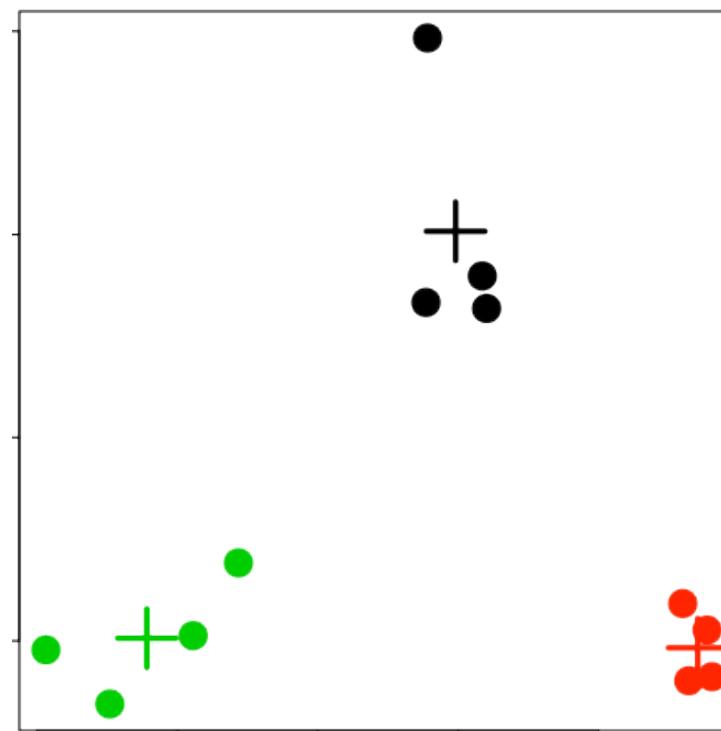
```
kmeansObj$cluster
```

```
[1] 3 3 3 3 1 1 1 1 2 2 2 2
```

11/14

# kmeans()

```
par(mar=rep(0.2,4))
plot(x,y,col=kmeansObj$cluster,pch=19,cex=2)
points(kmeansObj$centers,col=1:3,pch=3,cex=3,lwd=3)
```



12/14

# Heatmaps

```
set.seed(1234)
dataMatrix <- as.matrix(dataFrame)[sample(1:12), ]
kmeansObj2 <- kmeans(dataMatrix,centers=3)
par(mfrow=c(1,2),mar=rep(0.2,4))
image(t(dataMatrix)[,nrow(dataMatrix):1],yaxt="n")
image(t(dataMatrix)[,order(kmeansObj$cluster)],yaxt="n")
```



13/14

# Notes and further resources

- K-means requires a number of clusters
  - Pick by eye/intuition
  - Pick by cross validation/information theory, etc.
  - [Determining the number of clusters](#)
- K-means is not deterministic
  - Different # of clusters
  - Different number of iterations
- [Rafa's Distances and Clustering Video](#)
- [Elements of statistical learning](#)

# Model checking and model selection

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

# Model checking and model selection

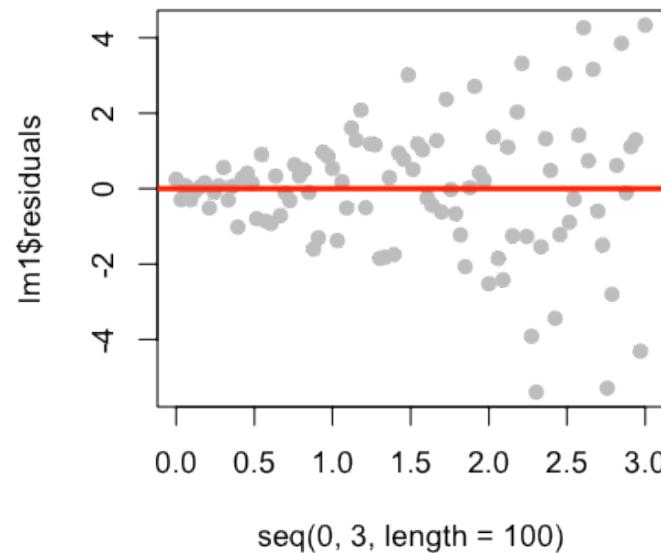
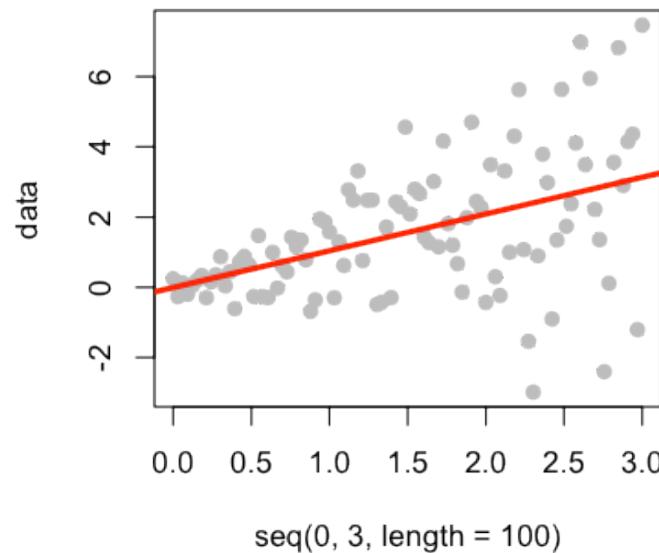
- Sometimes model checking/selection not allowed
- Often it can lead to problems
  - Overfitting
  - Overtesting
  - Biased inference
- *But* you don't want to miss something obvious

# Linear regression - basic assumptions

- Variance is constant
- You are summarizing a linear trend
- You have all the right terms in the model
- There are no big outliers

# Model checking - constant variance

```
set.seed(3433); par(mfrow=c(1,2))
data <- rnorm(100,mean=seq(0,3,length=100),sd=seq(0.1,3,length=100))
lm1 <- lm(data ~ seq(0,3,length=100))
plot(seq(0,3,length=100),data,pch=19,col="grey"); abline(lm1,col="red",lwd=3)
plot(seq(0,3,length=100),lm1$residuals,,pch=19,col="grey"); abline(c(0,0),col="red",lwd=3)
```



# What to do

- See if another variable explains the increased variance
- Use the *vcovHC* {sandwich} variance estimators (if n is big)

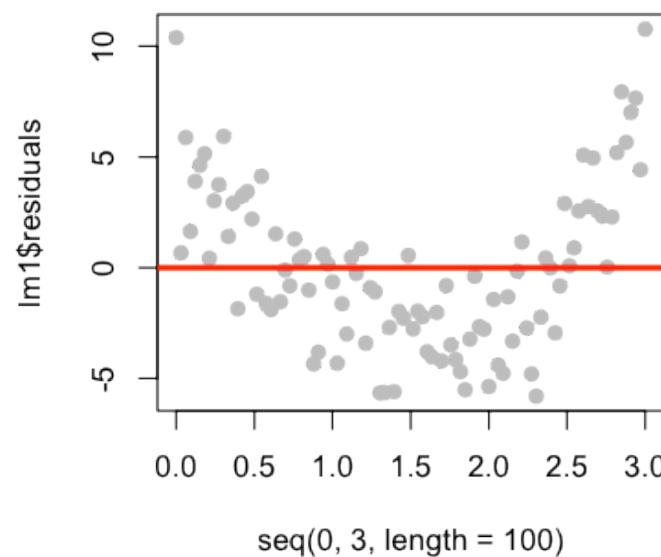
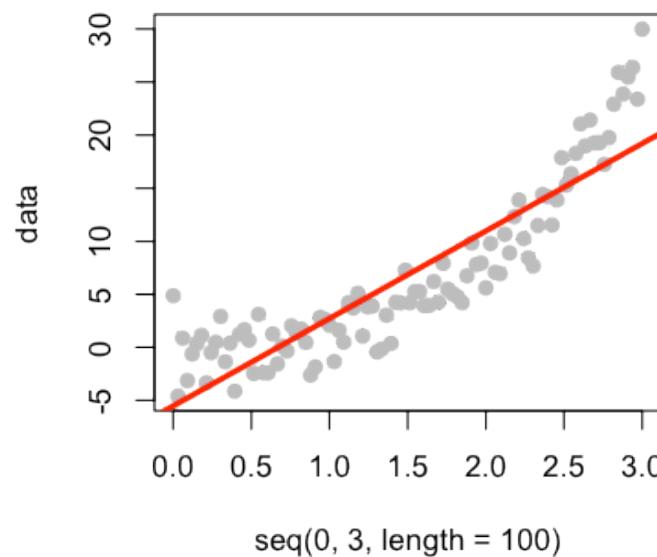
# Using the sandwich estimate

```
set.seed(3433); par(mfrow=c(1,2)); data <- rnorm(100,mean=seq(0,3,length=100),sd=seq(0.1,3,length=100))
lm1 <- lm(data ~ seq(0,3,length=100))
vcovHC(lm1)
summary(lm1)$cov.unscaled
```

|                         | (Intercept) seq(0, 3, length = 100) |          |
|-------------------------|-------------------------------------|----------|
| (Intercept)             | 0.03941                             | -0.01960 |
| seq(0, 3, length = 100) | -0.01960                            | 0.01307  |

# Model checking - linear trend

```
set.seed(3433); par(mfrow=c(1,2))
data <- rnorm(100,mean=seq(0,3,length=100)^3, sd=2)
lm1 <- lm(data ~ seq(0,3,length=100))
plot(seq(0,3,length=100),data,pch=19,col="grey"); abline(lm1,col="red",lwd=3)
plot(seq(0,3,length=100),lm1$residuals,,pch=19,col="grey"); abline(c(0,0),col="red",lwd=3)
```



# What to do

- Use Poisson regression (if it looks exponential/multiplicative)
- Use a data transformation (e.g. take the log)
- Smooth the data/fit a nonlinear trend (next week's lectures)
- Use linear regression anyway
  - Interpret as the linear trend between the variables
  - Use the *vcovHC* {sandwich} variance estimators (if n is big)

# Model checking - missing covariate

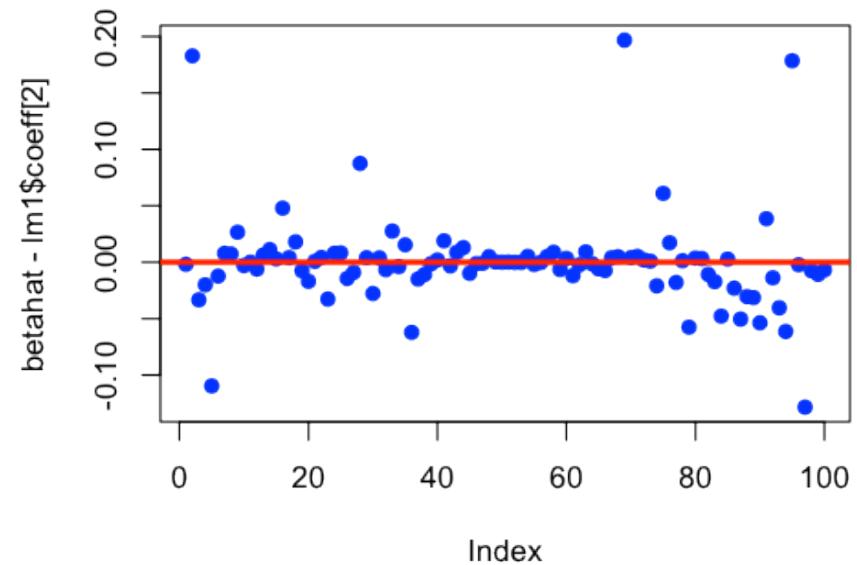
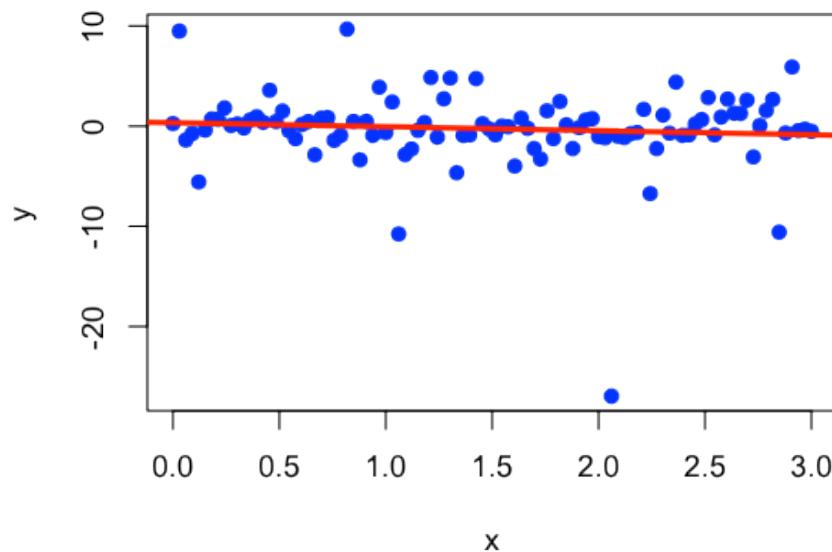
```
set.seed(3433); par(mfrow=c(1,3)); z <- rep(c(-0.5,0.5),50)
data <- rnorm(100,mean=(seq(0,3,length=100) + z),sd=seq(0.1,3,length=100))
lm1 <- lm(data ~ seq(0,3,length=100))
plot(seq(0,3,length=100),data,pch=19,col=((z>0)+3)); abline(lm1,col="red",lwd=3)
plot(seq(0,3,length=100),lm1$residuals,pch=19,col=((z>0)+3)); abline(c(0,0),col="red",lwd=3)
boxplot(lm1$residuals ~ z,col = ((z>0)+3) )
```

# What to do

- Use exploratory analysis to identify other variables to include
- Use the *vcovHC* {sandwich} variance estimators (if n is big)
- Report unexplained patterns in the data

# Model checking - outliers

```
set.seed(343); par(mfrow=c(1,2)); betahat <- rep(NA,100)
x <- seq(0,3,length=100); y <- rcauchy(100); lm1 <- lm(y ~ x)
plot(x,y,pch=19,col="blue"); abline(lm1,col="red",lwd=3)
for(i in 1:length(data)){betahat[i] <- lm(y[-i] ~ x[-i])$coeff[2]}
plot(betahat - lm1$coeff[2],col="blue",pch=19); abline(c(0,0),col="red",lwd=3)
```



# What to do

- If outliers are experimental mistakes -remove and document them
- If they are real - consider reporting how sensitive your estimate is to the outliers
- Consider using a robust linear model fit like *rlm* {MASS}

# Robust linear modeling

```
set.seed(343); x <- seq(0,3,length=100); y <- rcauchy(100);
lm1 <- lm(y ~ x); rlm1 <- rlm(y ~ x)
lm1$coeff
```

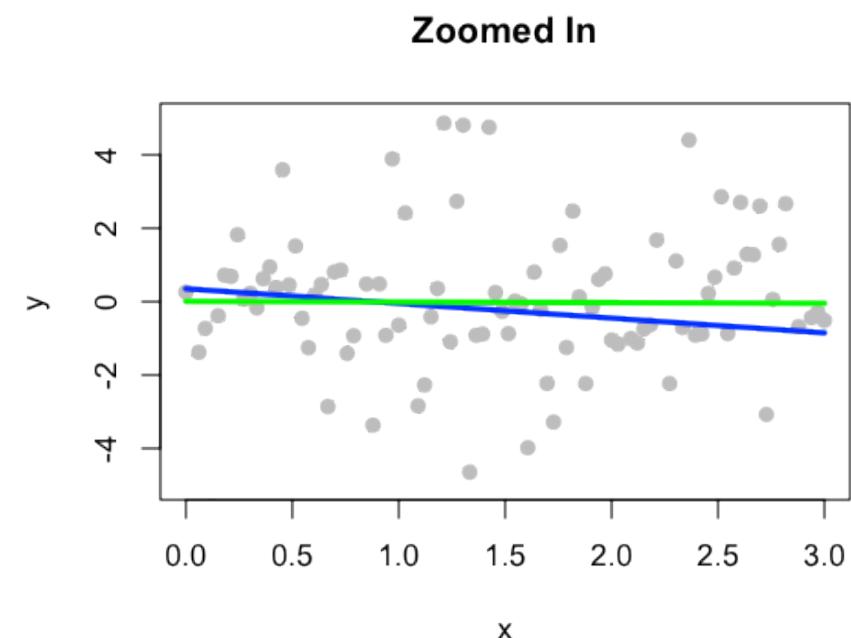
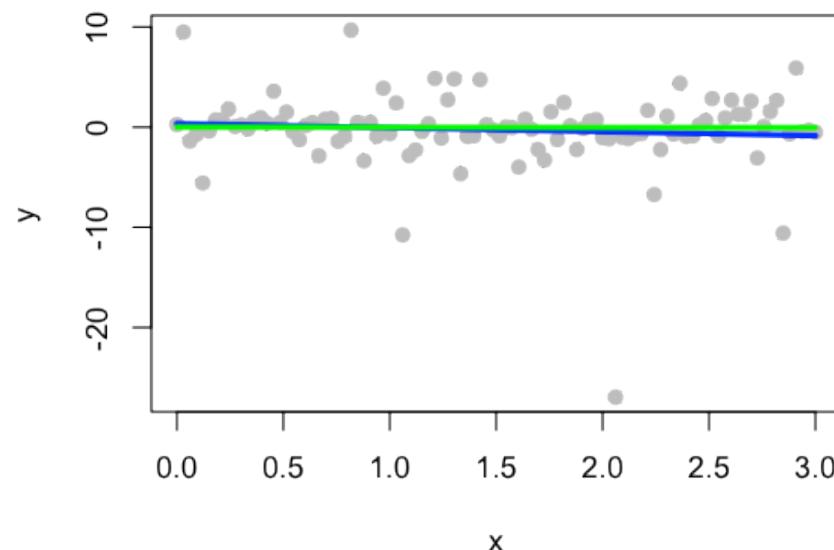
|             |         |
|-------------|---------|
| (Intercept) | x       |
| 0.3523      | -0.4011 |

```
rlm1$coeff
```

|             |           |
|-------------|-----------|
| (Intercept) | x         |
| 0.008527    | -0.017892 |

# Robust linear modeling

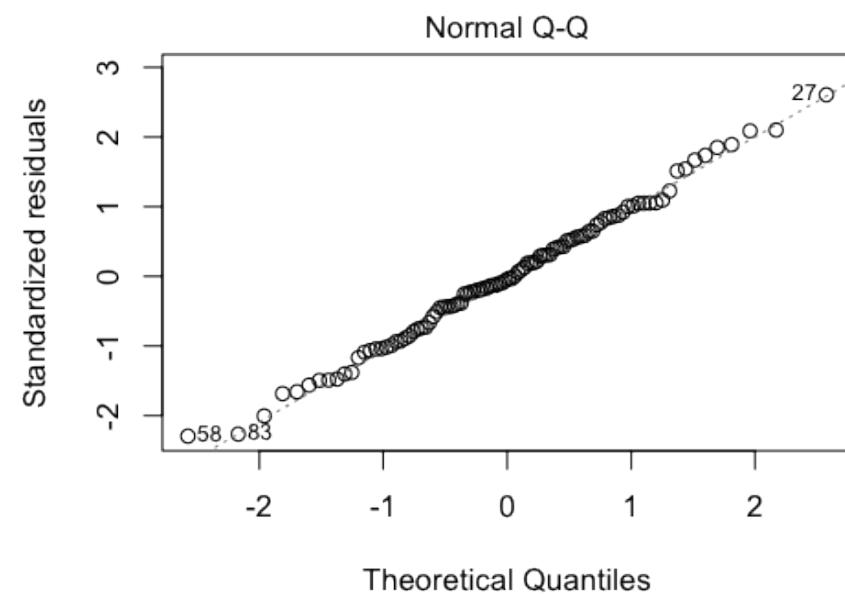
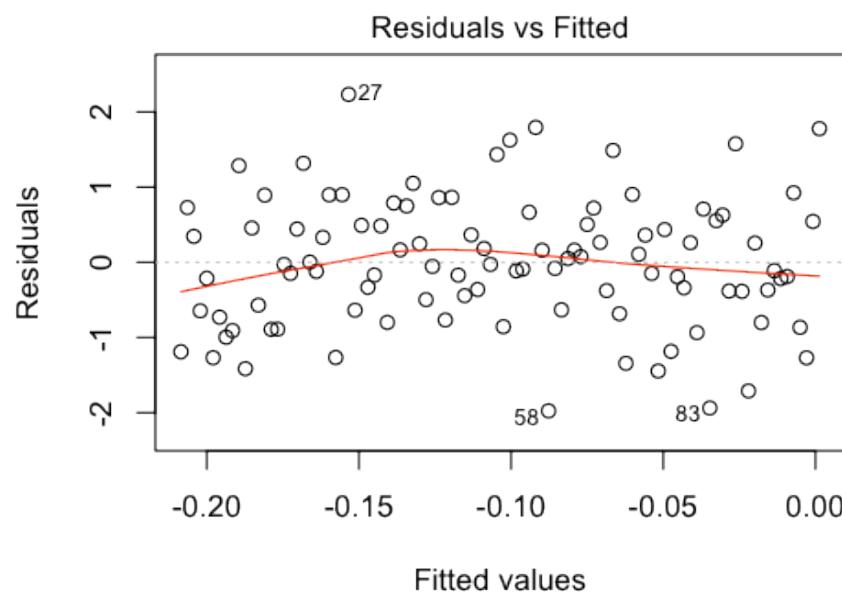
```
par(mfrow=c(1,2))
plot(x,y,pch=19,col="grey")
lines(x,lm1$fitted,col="blue",lwd=3); lines(x,rlm1$fitted,col="green",lwd=3)
plot(x,y,pch=19,col="grey",ylim=c(-5,5),main="Zoomed In")
lines(x,lm1$fitted,col="blue",lwd=3); lines(x,rlm1$fitted,col="green",lwd=3)
```



14/25

# Model checking - default plots

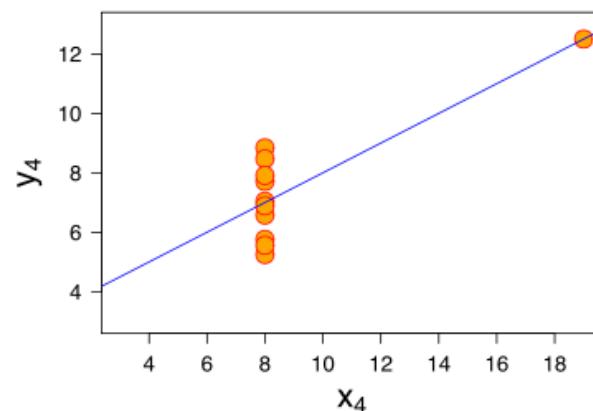
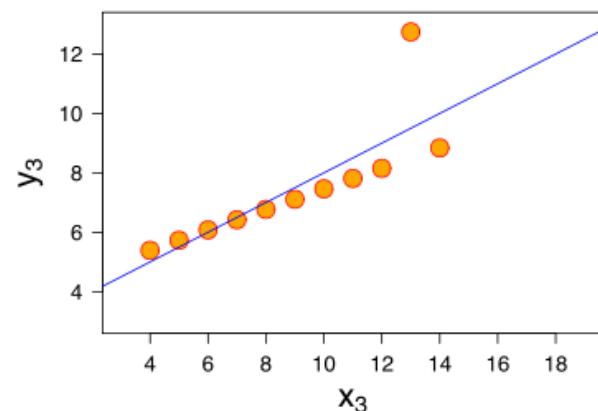
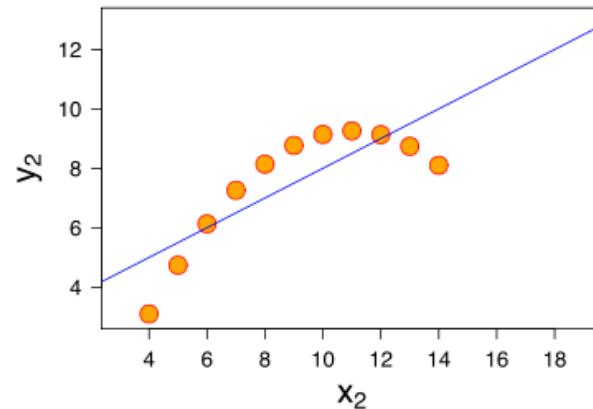
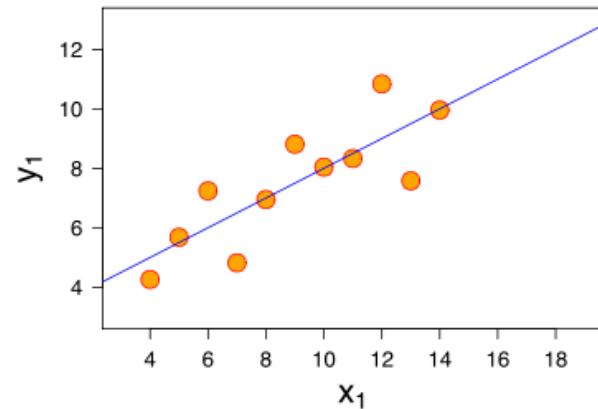
```
set.seed(343); par(mfrow=c(1,2))
x <- seq(0,3,length=100); y <- rnorm(100); lm1 <- lm(y ~ x)
plot(lm1)
```



# Model checking - deviance

- Commonly reported for GLM's
- Usually compares the model where every point gets its own parameter to the model you are using
- On its own it doesn't tell you what is wrong
- In large samples the deviance may be big even for "conservative" models
- You can not compare deviances for models with different sample sizes

# $R^2$ may be a bad summary



# Model selection

- Many times you have multiple variables to evaluate
- Options for choosing variables
  - Domain-specific knowledge
  - Exploratory analysis
  - Statistical selection
- There are many statistical selection options
  - Step-wise
  - AIC
  - BIC
  - Modern approaches: Lasso, Ridge-Regression, etc.
- Statistical selection may bias your inference
  - If possible, do selection on a held out sample

# Error measures

- $R^2$  alone isn't enough - more variables = bigger  $R^2$
- Adjusted  $R^2$  is  $R^2$  taking into account the number of estimated parameters
- AIC also penalizes models with more parameters
- BIC does the same, but with a bigger penalty

# Movie Data

```
download.file("http://www.rossmanchance.com/iscam2/data/movies03RT.txt", destfile="./data/movies.txt"
movies <- read.table("./data/movies.txt", sep="\t", header=T, quote="")
head(movies)
```

|   | X | score            | rating | genre | box.office       | running.time |     |
|---|---|------------------|--------|-------|------------------|--------------|-----|
| 1 | 2 | Fast 2 Furious   | 48.9   | PG-13 | action/adventure | 127.15       | 107 |
| 2 |   | 28 Days Later    | 78.2   | R     | horror           | 45.06        | 113 |
| 3 |   | A Guy Thing      | 39.5   | PG-13 | rom comedy       | 15.54        | 101 |
| 4 |   | A Man Apart      | 42.9   | R     | action/adventure | 26.25        | 110 |
| 5 |   | A Mighty Wind    | 79.9   | PG-13 | comedy           | 17.78        | 91  |
| 6 |   | Agent Cody Banks | 57.9   | PG    | action/adventure | 47.81        | 102 |

<http://www.rossmanchance.com/>

# Model selection - step

```
movies <- movies[,-1]
lm1 <- lm(score ~ ., data=movies)
aicFormula <- step(lm1)
```

Start: AIC=727.5

score ~ rating + genre + box.office + running.time

|                | Df | Sum of Sq | RSS   | AIC |
|----------------|----|-----------|-------|-----|
| - genre        | 12 | 2575      | 22132 | 721 |
| - rating       | 3  | 40        | 19596 | 722 |
| - running.time | 1  | 237       | 19793 | 727 |
| <none>         |    |           | 19556 | 728 |
| - box.office   | 1  | 3007      | 22563 | 746 |

Step: AIC=720.8

score ~ rating + box.office + running.time

|          | Df | Sum of Sq | RSS   | AIC |
|----------|----|-----------|-------|-----|
| - rating | 3  | 491       | 22623 | 718 |
| <none>   |    |           | 22132 | 721 |

# Model selection - step

```
aicFormula
```

Call:

```
lm(formula = score ~ box.office + running.time, data = movies)
```

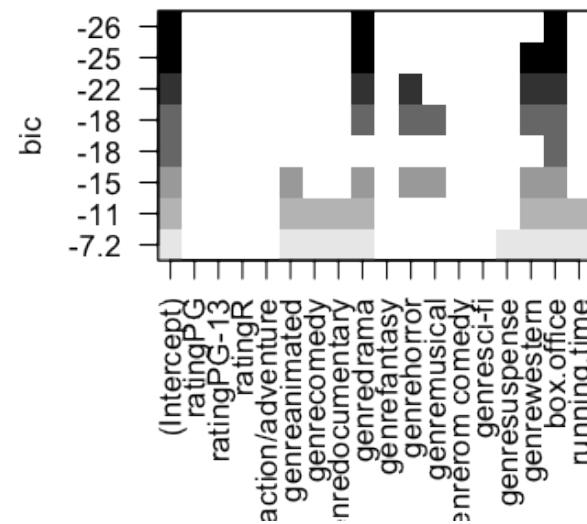
Coefficients:

| (Intercept) | box.office | running.time |
|-------------|------------|--------------|
| 37.2364     | 0.0824     | 0.1275       |

22/25

# Model selection - regsubsets

```
library(leaps);
regSub <- regsubsets(score ~ ., data=movies)
plot(regSub)
```



<http://cran.r-project.org/web/packages/leaps/leaps.pdf>

23/25

# Model selection - bic.glm

```
library(BMA)
bicglm1 <- bic.glm(score ~ ., data=movies, glm.family="gaussian")
print(bicglm1)
```

Call:

```
bic.glm.formula(f = score ~ ., data = movies, glm.family = "gaussian")
```

Posterior probabilities(%):

| rating | genre | box.office | running.time |
|--------|-------|------------|--------------|
| 0.0    | 100.0 | 100.0      | 18.2         |

Coefficient posterior expected values:

|                       | (Intercept) | ratingPG | ratingPG-13 | ratingR          |
|-----------------------|-------------|----------|-------------|------------------|
| genreaction/adventure | 45.263      | 0.000    | 0.000       | 0.000            |
| genredrama            | -0.120      | 7.628    | 2.077       | 8.642            |
| genrefantasy          | 13.041      | 1.504    | -3.458      | 24/25<br>-12.255 |
| genrehorror           |             |          |             |                  |
| genreanimated         |             |          |             |                  |
| genremusical          |             |          |             |                  |
| genredocumentary      |             |          |             |                  |

# Notes and further resources

- Exploratory/visual analysis is key
- Automatic selection produces an answer - but may bias inference
- You may think about separating the sample into two groups
- The goal is not to get the "causal" model
- [Lars package](#)
- [Elements of machine learning](#)

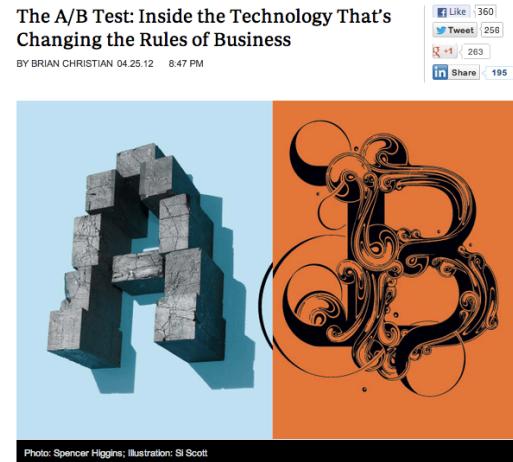
Jeffrey Leek, Assistant Professor of Biostatistics  
Johns Hopkins Bloomberg School of Public Health

# Key ideas

- Outcome is still quantitative
- You have multiple explanatory variables
- Goal is to identify contributions of different variables

2/15

# A successful example



"For the button, an A/B test of three new word choices—"Learn More," "Join Us Now," and "Sign Up Now"—revealed that "Learn More" garnered 18.6 percent more signups per visitor than the default of "Sign Up." Similarly, a black-and-white photo of the Obama family outperformed the default turquoise image by 13.1 percent. Using both the family image and "Learn More," signups increased by a thundering 40 percent."

[http://www.wired.com/business/2012/04/ff\\_abtesting/](http://www.wired.com/business/2012/04/ff_abtesting/)

# Movie Data

```
download.file("http://www.rossmanchance.com/iscam2/data/movies03RT.txt",
              destfile=".~/data/movies.txt")
movies <- read.table("./data/movies.txt", sep="\t", header=T, quote="")
head(movies)
```

|   | X                | score          | rating | genre            | box.office       | running.time |     |
|---|------------------|----------------|--------|------------------|------------------|--------------|-----|
| 1 | 2                | Fast 2 Furious | 48.9   | PG-13            | action/adventure | 127.15       | 107 |
| 2 | 28               | Days Later     | 78.2   | R                | horror           | 45.06        | 113 |
| 3 | A Guy Thing      | 39.5           | PG-13  | rom              | comedy           | 15.54        | 101 |
| 4 | A Man Apart      | 42.9           | R      | action/adventure | 26.25            | 110          |     |
| 5 | A Mighty Wind    | 79.9           | PG-13  |                  | comedy           | 17.78        | 91  |
| 6 | Agent Cody Banks | 57.9           | PG     | action/adventure | 47.81            | 102          |     |

<http://www.rossmanchance.com/>

# Relating score to rating

$$S_i = b_0 + b_1 \mathbb{1}(Ra_i = "PG") + b_2 \mathbb{1}(Ra_i = "PG-13") + b_3 \mathbb{1}(Ra_i = "R") + e_i$$

The notation  $\mathbb{1}(Ra_i = "PG")$  is a logical value that is one if the movie rating is "PG" and zero otherwise.

## Average values

$b_0$  = average of the G movies

$b_0 + b_1$  = average of the PG movies

$b_0 + b_2$  = average of the PG-13 movies

$b_0 + b_3$  = average of the R movies

5/15

# ANOVA in R

```
aovObject <- aov(movies$score ~ movies$rating)  
aovObject
```

Call:

```
aov(formula = movies$score ~ movies$rating)
```

Terms:

|                 | movies\$rating | Residuals |
|-----------------|----------------|-----------|
| Sum of Squares  | 570            | 28149     |
| Deg. of Freedom | 3              | 136       |

Residual standard error: 14.39

Estimated effects may be unbalanced

# ANOVA in R

```
aovObject$coeff
```

| (Intercept) | movies\$ratingPG | movies\$ratingPG-13 | movies\$ratingR |
|-------------|------------------|---------------------|-----------------|
| 67.65       | -12.59           | -11.81              | -12.02          |

# Adding a second factor

$$S_i = b_0 + b_1 \mathbb{1}(Ra_i = "PG") + b_2 \mathbb{1}(Ra_i = "PG - 13") + b_3 \mathbb{1}(Ra_i = "R") \\ + \gamma_1 \mathbb{1}(G_i = "action") + \gamma_2 \mathbb{1}(G_i = "animated") + \dots + e_i$$

The notation  $\mathbb{1}(Ra_i = "PG")$  is a logical value that is one if the movie rating is "PG" and zero otherwise.

# Adding a second factor

$$S_i = b_0 + \underbrace{b_1 \mathbb{1}(Ra_i = "PG") + b_2 \mathbb{1}(Ra_i = "PG - 13") + b_3 \mathbb{1}(Ra_i = "R")}_{rating} + \underbrace{\gamma_1 \mathbb{1}(G_i = "action") + \gamma_2 \mathbb{1}(G_i = "animated") + \dots}_{genre} + e_i$$

There are only 2 variables in this model. They have multiple levels.

# Second variable

```
aovObject2 <- aov(movies$score ~ movies$rating + movies$genre)  
aovObject2
```

Call:

```
aov(formula = movies$score ~ movies$rating + movies$genre)
```

Terms:

|                 | movies\$rating | movies\$genre | Residuals |
|-----------------|----------------|---------------|-----------|
| Sum of Squares  | 570            | 3935          | 24214     |
| Deg. of Freedom | 3              | 12            | 124       |

Residual standard error: 13.97

Estimated effects may be unbalanced

# ANOVA Summary

```
summary(aovObject2)
```

|                | Df  | Sum Sq | Mean Sq | F value | Pr(>F)  |     |      |      |     |     |   |
|----------------|-----|--------|---------|---------|---------|-----|------|------|-----|-----|---|
| movies\$rating | 3   | 570    | 190     | 0.97    | 0.408   |     |      |      |     |     |   |
| movies\$genre  | 12  | 3935   | 328     | 1.68    | 0.079 . |     |      |      |     |     |   |
| Residuals      | 124 | 24214  | 195     |         |         |     |      |      |     |     |   |
| ---            |     |        |         |         |         |     |      |      |     |     |   |
| Signif. codes: | 0   | '***'  | 0.001   | '**'    | 0.01    | '*' | 0.05 | '. ' | 0.1 | ' ' | 1 |

# Order matters

```
aovObject3 <- aov(movies$score ~ movies$genre + movies$rating)
summary(aovObject3)
```

|                | Df  | Sum Sq | Mean Sq | F value | Pr(>F)  |     |      |      |     |     |   |
|----------------|-----|--------|---------|---------|---------|-----|------|------|-----|-----|---|
| movies\$genre  | 12  | 4222   | 352     | 1.80    | 0.055 . |     |      |      |     |     |   |
| movies\$rating | 3   | 284    | 95      | 0.48    | 0.694   |     |      |      |     |     |   |
| Residuals      | 124 | 24214  | 195     |         |         |     |      |      |     |     |   |
| ---            |     |        |         |         |         |     |      |      |     |     |   |
| Signif. codes: | 0   | '***'  | 0.001   | '**'    | 0.01    | '*' | 0.05 | '. ' | 0.1 | ' ' | 1 |

```
summary(aovObject2)
```

|                | Df  | Sum Sq | Mean Sq | F value | Pr(>F)  |
|----------------|-----|--------|---------|---------|---------|
| movies\$rating | 3   | 570    | 190     | 0.97    | 0.408   |
| movies\$genre  | 12  | 3935   | 328     | 1.68    | 0.079 . |
| Residuals      | 124 | 24214  | 195     |         |         |
| ---            |     |        |         |         |         |

12/15

# Adding a quantitative variable

$$S_i = b_0 + \underbrace{b_1 \mathbb{1}(Ra_i = "PG") + b_2 \mathbb{1}(Ra_i = "PG-13") + b_3 \mathbb{1}(Ra_i = "R")}_{rating} \\ + \underbrace{\gamma_1 \mathbb{1}(G_i = "action") + \gamma_2 \mathbb{1}(G_i = "animated") + \dots}_{genre} + \eta_1 \underbrace{BO_i}_{box\ office} + e_i$$

There are three variables in this model - box office is quantitative so only has one term.

# ANOVA with quantitative variable in R

```
aovObject4 <- aov(movies$score ~ movies$genre + movies$rating + movies$box.office)
summary(aovObject4)
```

|                    | Df  | Sum Sq | Mean Sq | F value | Pr(>F)      |     |      |      |     |     |   |
|--------------------|-----|--------|---------|---------|-------------|-----|------|------|-----|-----|---|
| movies\$genre      | 12  | 4222   | 352     | 2.19    | 0.016 *     |     |      |      |     |     |   |
| movies\$rating     | 3   | 284    | 95      | 0.59    | 0.624       |     |      |      |     |     |   |
| movies\$box.office | 1   | 4421   | 4421    | 27.47   | 6.7e-07 *** |     |      |      |     |     |   |
| Residuals          | 123 | 19793  | 161     |         |             |     |      |      |     |     |   |
| ---                |     |        |         |         |             |     |      |      |     |     |   |
| Signif. codes:     | 0   | '***'  | 0.001   | '**'    | 0.01        | '*' | 0.05 | '. ' | 0.1 | ' ' | 1 |

# Language and further resources

- Units - one observation
- Treatments - applied to units
- Factors - controlled by experimenters
- Replicates - multiple (independent) units with the same factors/treatments
- [Wikipedia on Experimental Design](#)
- [Wikipedia on ANOVA](#)
- [Wikipedia on A/B Testing](#)

# Multiple testing

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

# Key ideas

- Hypothesis testing/significance analysis is commonly overused
- Correcting for multiple testing avoids false positives or discoveries
- Two key components
  - Error measure
  - Correction

# Three eras of statistics

**The age of Quetelet and his successors, in which huge census-level data sets were brought to bear on simple but important questions:** Are there more male than female births? Is the rate of insanity rising?

The classical period of Pearson, Fisher, Neyman, Hotelling, and their successors, intellectual giants who **developed a theory of optimal inference capable of wringing every drop of information out of a scientific experiment.** The questions dealt with still tended to be simple Is treatment A better than treatment B?

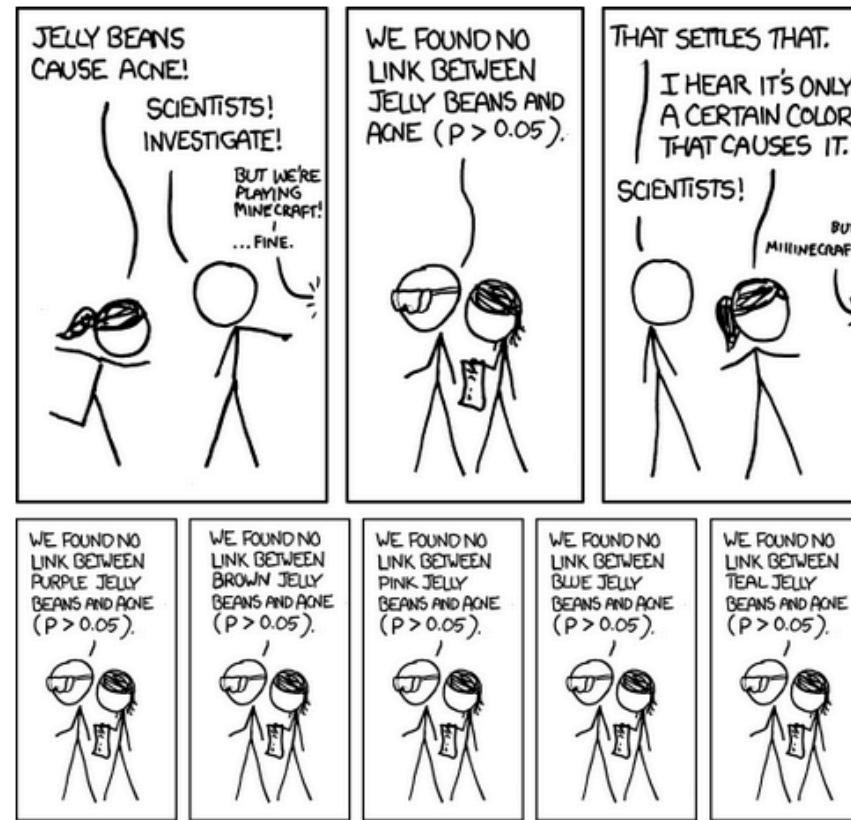
**The era of scientific mass production,** in which new technologies typified by the microarray allow a single team of scientists to produce data sets of a size Quetelet would envy. But now the flood of data is accompanied by a deluge of questions, perhaps thousands of estimates or hypothesis tests that the statistician is charged with answering together; not at all what the classical masters had in mind. Which variables matter among the thousands measured? How do you relate unrelated information?

<http://www-stat.stanford.edu/~ckirby/brad/papers/2010LSlexcerpt.pdf>

# Reasons for multiple testing

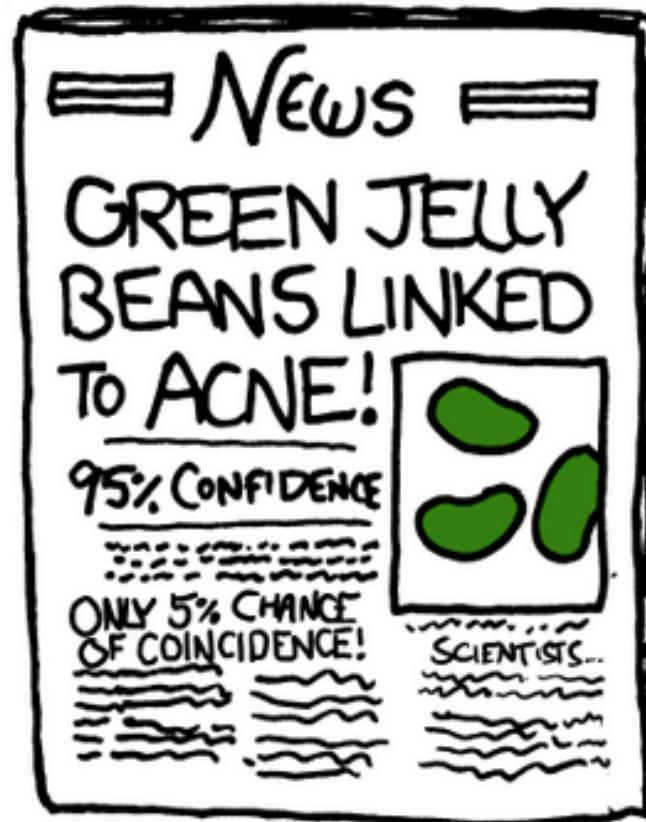


# Why correct for multiple tests?



<http://xkcd.com/882/>

# Why correct for multiple tests?



<http://xkcd.com/882/>

# Types of errors

Suppose you are testing a hypothesis that a parameter  $\beta$  equals zero versus the alternative that it does not equal zero. These are the possible outcomes.

|                      | $\beta = 0$ | $\beta \neq 0$ | HYPOTHESES |
|----------------------|-------------|----------------|------------|
| Claim $\beta = 0$    | $U$         | $T$            | $m - R$    |
| Claim $\beta \neq 0$ | $V$         | $S$            | $R$        |
| Claims               | $m_0$       | $m - m_0$      | $m$        |

**Type I error or false positive ( $V$ )** Say that the parameter does not equal zero when it does

**Type II error or false negative ( $T$ )** Say that the parameter equals zero when it doesn't

# Error rates

**False positive rate** - The rate at which false results ( $\beta = 0$ ) are called significant:  $E\left[\frac{V}{m_0}\right]^*$

**Family wise error rate (FWER)** - The probability of at least one false positive  $\Pr(V \geq 1)$

**False discovery rate (FDR)** - The rate at which claims of significance are false  $E\left[\frac{V}{R}\right]$

- The false positive rate is closely related to the type I error rate  
[http://en.wikipedia.org/wiki/False\\_positive\\_rate](http://en.wikipedia.org/wiki/False_positive_rate)

# Controlling the false positive rate

If P-values are correctly calculated calling all  $P < \alpha$  significant will control the false positive rate at level  $\alpha$  on average.

**Problem:** Suppose that you perform 10,000 tests and  $\beta = 0$  for all of them.

Suppose that you call all  $P < 0.05$  significant.

The expected number of false positives is:  $10,000 \times 0.05 = 500$  false positives.

**How do we avoid so many false positives?**

# Controlling family-wise error rate (FWER)

The [Bonferroni correction](#) is the oldest multiple testing correction.

**Basic idea:**

- Suppose you do  $m$  tests
- You want to control FWER at level  $\alpha$  so  $Pr(V \geq 1) < \alpha$
- Calculate P-values normally
- Set  $\alpha_{fwer} = \alpha/m$
- Call all  $P$ -values less than  $\alpha_{fwer}$  significant

**Pros:** Easy to calculate, conservative **Cons:** May be very conservative

# Controlling false discovery rate (FDR)

This is the most popular correction when performing *lots* of tests say in genomics, imaging, astronomy, or other signal-processing disciplines.

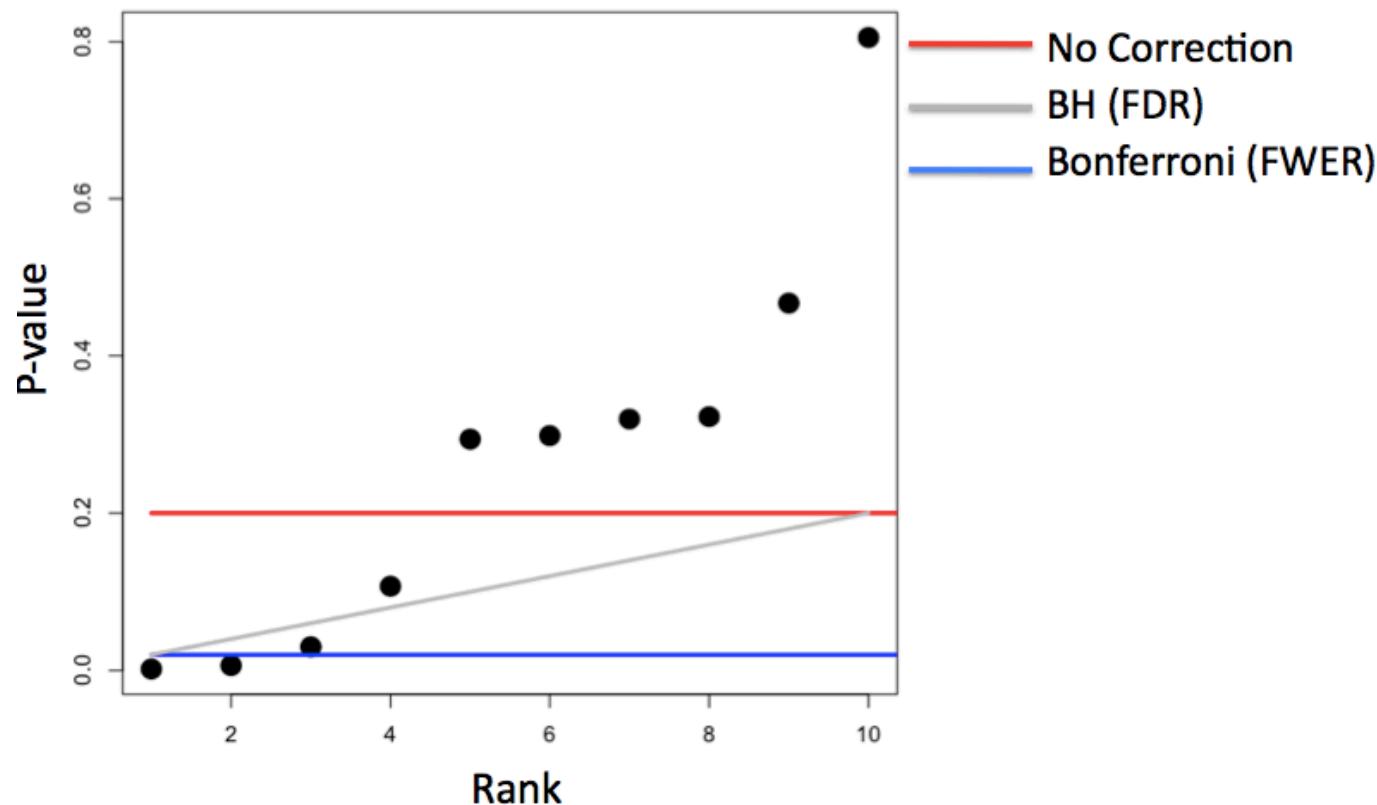
## Basic idea:

- Suppose you do  $m$  tests
- You want to control FDR at level  $\alpha$  so  $E\left[\frac{V}{R}\right]$
- Calculate P-values normally
- Order the P-values from smallest to largest  $P_{(1)}, \dots, P_{(m)}$
- Call any  $P_{(i)} \leq \alpha \times \frac{i}{m}$  significant

**Pros:** Still pretty easy to calculate, less conservative (maybe much less)

**Cons:** Allows for more false positives, may behave strangely under dependence

# Example with 10 P-values



Controlling all error rates at  $\alpha = 0.20$

12/19

# Adjusted P-values

- One approach is to adjust the threshold  $\alpha$
- A different approach is to calculate "adjusted p-values"
- They *are not p-values* anymore
- But they can be used directly without adjusting  $\alpha$

## Example:

- Suppose P-values are  $P_1, \dots, P_m$
- You could adjust them by taking  $P_i^{fwer} = \max(m \times P_i, 1)$  for each P-value.
- Then if you call all  $P_i^{fwer} < \alpha$  significant you will control the FWER.

# Case study I: no true positives

```
set.seed(1010093)
pValues <- rep(NA,1000)
for(i in 1:1000){
  y <- rnorm(20)
  x <- rnorm(20)
  pValues[i] <- summary(lm(y ~ x))$coeff[2,4]
}

# Controls false positive rate
sum(pValues < 0.05)
```

```
[1] 51
```

14/19

# Case study I: no true positives

```
# Controls FWER  
sum(p.adjust(pValues,method="bonferroni") < 0.05)
```

```
[1] 0
```

```
# Controls FDR  
sum(p.adjust(pValues,method="BH") < 0.05)
```

```
[1] 0
```

# Case study II: 50% true positives

```
set.seed(1010093)
pValues <- rep(NA,1000)
for(i in 1:1000){
  x <- rnorm(20)
  # First 500 beta=0, last 500 beta=2
  if(i <= 500){y <- rnorm(20)}else{ y <- rnorm(20,mean=2*x)}
  pValues[i] <- summary(lm(y ~ x))$coeff[2,4]
}
trueStatus <- rep(c("zero","not zero"),each=500)
table(pValues < 0.05, trueStatus)
```

| trueStatus | not zero | zero |
|------------|----------|------|
| FALSE      | 0        | 476  |
| TRUE       | 500      | 24   |

# Case study II: 50% true positives

```
# Controls FWER  
table(p.adjust(pValues,method="bonferroni") < 0.05,trueStatus)
```

```
trueStatus  
not zero zero  
FALSE      23  500  
TRUE       477    0
```

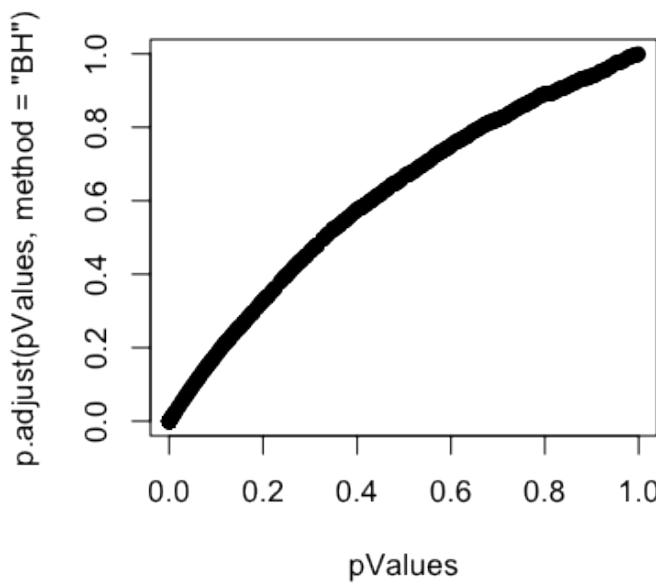
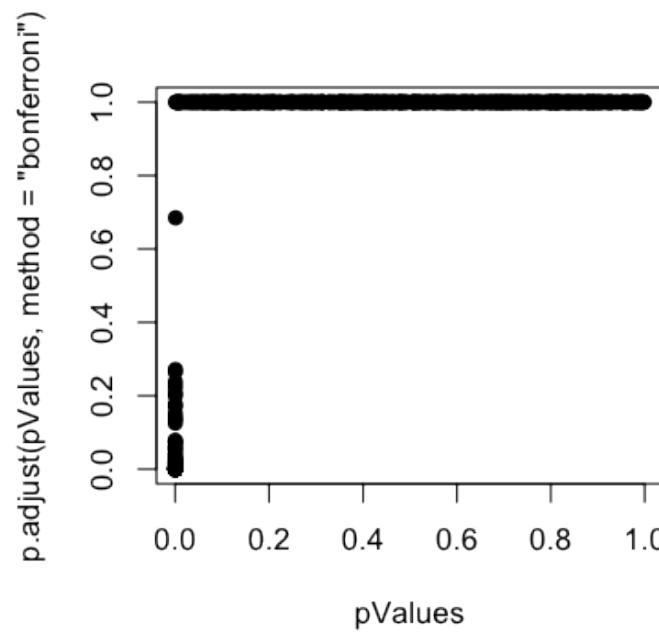
```
# Controls FDR  
table(p.adjust(pValues,method="BH") < 0.05,trueStatus)
```

```
trueStatus  
not zero zero  
FALSE      0  487  
TRUE      500   13
```

# Case study II: 50% true positives

## P-values versus adjusted P-values

```
par(mfrow=c(1,2))
plot(pValues,p.adjust(pValues,method="bonferroni"),pch=19)
plot(pValues,p.adjust(pValues,method="BH"),pch=19)
```



18/19

# Notes and resources

## Notes:

- Multiple testing is an entire subfield
- A basic Bonferroni/BH correction is usually enough
- If there is strong dependence between tests there may be problems
  - Consider method="BY"

## Further resources:

- [Multiple testing procedures with applications to genomics](#)
- [Statistical significance for genome-wide studies](#)
- [Introduction to multiple testing](#)

# Multiple regression

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

# Key ideas

- Regression with multiple covariates
- Still using least squares/central limit theorem
- Interpretation depends on all variables

2/16

# Example - Millenium Development Goal 1



## GOAL 1 Eradicate Extreme Poverty and Hunger

FACT SHEET

### TARGETS

1. Halve, between 1990 and 2015, the proportion of people whose income is less than \$1 a day
2. Achieve full and productive employment and decent work for all, including women and young people
3. Halve, between 1990 and 2015, the proportion of people who suffer from hunger

[http://www.un.org/millenniumgoals/pdf/MDG\\_FS\\_1\\_EN.pdf](http://www.un.org/millenniumgoals/pdf/MDG_FS_1_EN.pdf)

[http://apps.who.int/gho/athena/data/GHO/WHOSIS\\_000008.csv?  
profile=text&filter=COUNTRY:;SEX:](http://apps.who.int/gho/athena/data/GHO/WHOSIS_000008.csv?profile=text&filter=COUNTRY:;SEX:)

3/16

# WHO childhood hunger data

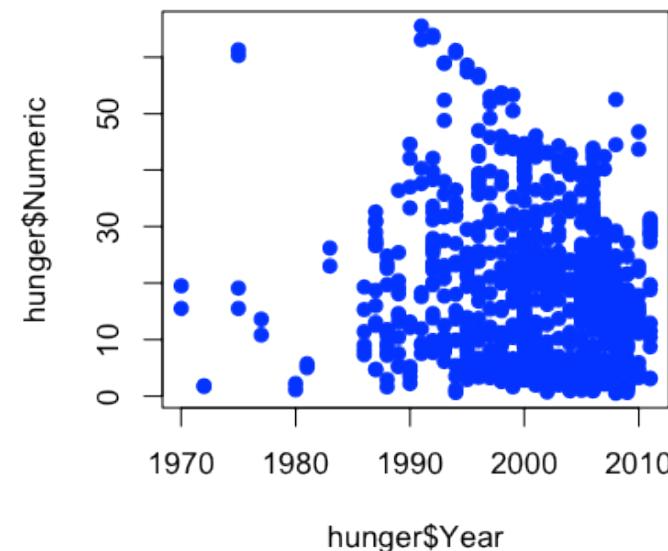
```
download.file("http://apps.who.int/gho/athena/data/GHO/WHOSIS_000008.csv?profile=text&filter=COUNTR
hunger <- read.csv("./data/hunger.csv")
hunger <- hunger[hunger$Sex!="Both sexes",]
head(hunger)
```

|    | Indicator                              | Data.Source | Country     | Sex    | Year     | WHO.region            |
|----|--|-------------|-------------|--------|----------|-----------------------|
| 2  | Children aged <5 years underweight (%) | NLIS_312819 | Afghanistan | Male   | 2004     | Eastern Mediterranean |
| 4  | Children aged <5 years underweight (%) | NLIS_312819 | Afghanistan | Female | 2004     | Eastern Mediterranean |
| 7  | Children aged <5 years underweight (%) | NLIS_312361 | Albania     | Male   | 2000     | Europe                |
| 8  | Children aged <5 years underweight (%) | NLIS_312361 | Albania     | Female | 2000     | Europe                |
| 9  | Children aged <5 years underweight (%) | NLIS_312879 | Albania     | Female | 2005     | Europe                |
| 10 | Children aged <5 years underweight (%) | NLIS_312879 | Albania     | Male   | 2005     | Europe                |
|    | Display.Value                          | Numeric     | Low         | High   | Comments |                       |
| 2  | 32.7                                   | 32.7        | NA          | NA     | NA       |                       |
| 4  | 33.0                                   | 33.0        | NA          | NA     | NA       |                       |
| 7  | 19.6                                   | 19.6        | NA          | NA     | NA       |                       |
| 8  | 14.2                                   | 14.2        | NA          | NA     | NA       |                       |
| 9  | 5.8                                    | 5.8         | NA          | NA     | NA       |                       |
| 10 | 7.3                                    | 7.3         | NA          | NA     | NA       |                       |

4/16

# Plot percent hungry versus time

```
lm1 <- lm(hunger$Numeric ~ hunger$Year)
plot(hunger$Year,hunger$Numeric,pch=19,col="blue")
```



# Remember the linear model

$$Hu_i = b_0 + b_1 Y_i + e_i$$

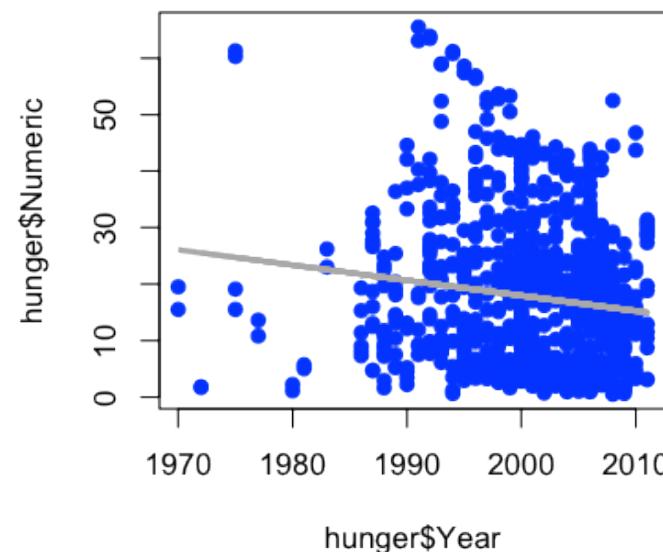
$b_0$  = percent hungry at Year 0

$b_1$  = decrease in percent hungry per year

$e_i$  = everything we didn't measure

# Add the linear model

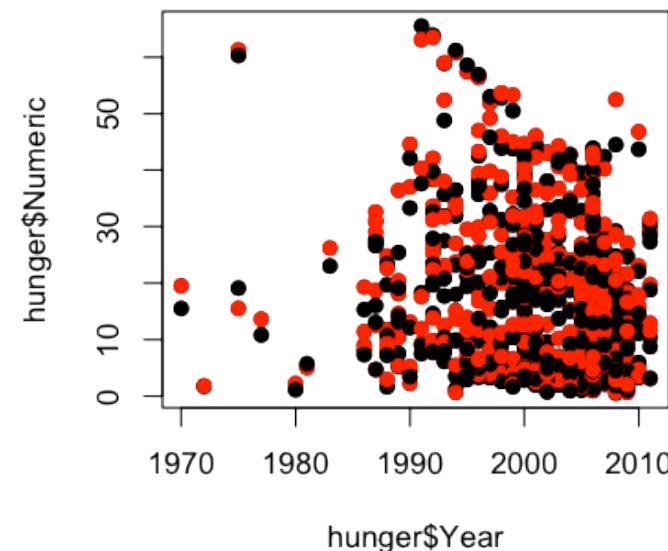
```
lm1 <- lm(hunger$Numeric ~ hunger$Year)
plot(hunger$Year,hunger$Numeric,pch=19,col="blue")
lines(hunger$Year,lm1$fitted,lwd=3,col="darkgrey")
```



7/16

# Color by male/female

```
plot(hunger$Year,hunger$Numeric,pch=19)
points(hunger$Year,hunger$Numeric,pch=19,col=((hunger$Sex=="Male")*1+1))
```



8/16

# Now two lines

$$HuF_i = bf_0 + bf_1 YF_i + ef_i$$

$bf_0$  = percent of girls hungry at Year 0

$bf_1$  = decrease in percent of girls hungry per year

$ef_i$  = everything we didn't measure

$$HuM_i = bm_0 + bm_1 YM_i + em_i$$

$bm_0$  = percent of boys hungry at Year 0

$bm_1$  = decrease in percent of boys hungry per year

$em_i$  = everything we didn't measure

# Color by male/female

```
lmM <- lm(hunger$Numeric[hunger$Sex=="Male"] ~ hunger$Year[hunger$Sex=="Male"])
lmF <- lm(hunger$Numeric[hunger$Sex=="Female"] ~ hunger$Year[hunger$Sex=="Female"])
plot(hunger$Year,hunger$Numeric,pch=19)
points(hunger$Year,hunger$Numeric,pch=19,col=(hunger$Sex=="Male")*1+1)
lines(hunger$Year[hunger$Sex=="Male"],lmM$fitted,col="black",lwd=3)
lines(hunger$Year[hunger$Sex=="Female"],lmF$fitted,col="red",lwd=3)
```

10/16

# Two lines, same slope

$$Hu_i = b_0 + b_1 \mathbb{1}(Sex_i = "Male") + b_2 Y_i + e_i^*$$

$b_0$  - percent hungry at year zero for females

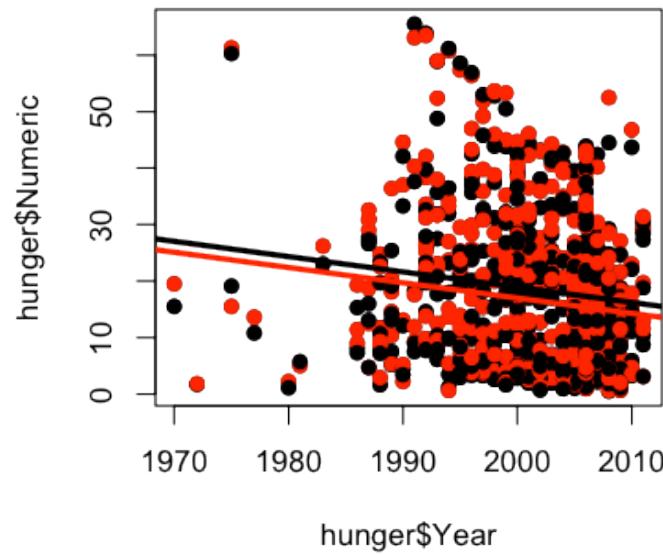
$b_0 + b_1$  - percent hungry at year zero for males

$b_2$  - change in percent hungry (for either males or females) in one year

$e_i^*$  - everything we didn't measure

# Two lines, same slope in R

```
lmBoth <- lm(hunger$Numeric ~ hunger$Year + hunger$Sex)
plot(hunger$Year,hunger$Numeric,pch=19)
points(hunger$Year,hunger$Numeric,pch=19,col=((hunger$Sex=="Male")*1+1))
abline(c(lmBoth$coeff[1],lmBoth$coeff[2]),col="red",lwd=3)
abline(c(lmBoth$coeff[1] + lmBoth$coeff[3],lmBoth$coeff[2] ),col="black",lwd=3)
```



12/16

# Two lines, different slopes (interactions)

$$Hu_i = b_0 + b_1 \mathbb{1}(Sex_i = "Male") + b_2 Y_i + b_3 \mathbb{1}(Sex_i = "Male") \times Y_i + e_i^+$$

$b_0$  - percent hungry at year zero for females

$b_0 + b_1$  - percent hungry at year zero for males

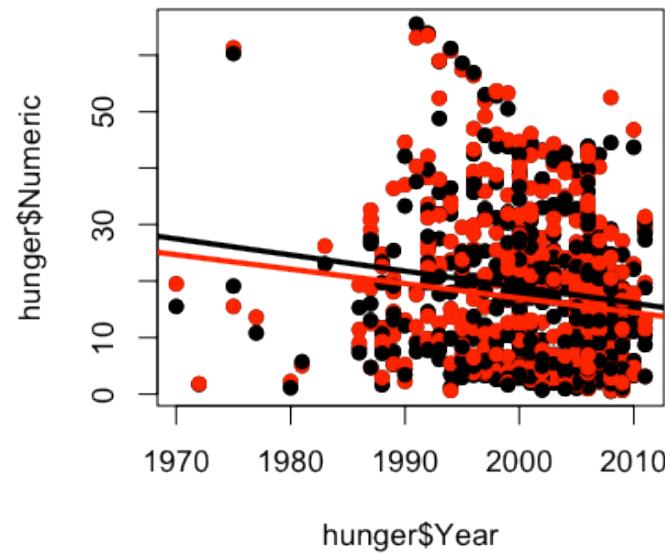
$b_2$  - change in percent hungry (females) in one year

$b_2 + b_3$  - change in percent hungry (males) in one year

$e_i^+$  - everything we didn't measure

# Two lines, different slopes in R

```
lmBoth <- lm(hunger$Numeric ~ hunger$Year + hunger$Sex + hunger$Sex*hunger$Year)
plot(hunger$Year,hunger$Numeric,pch=19)
points(hunger$Year,hunger$Numeric,pch=19,col=((hunger$Sex=="Male")*1+1))
abline(c(lmBoth$coeff[1],lmBoth$coeff[2]),col="red",lwd=3)
abline(c(lmBoth$coeff[1] + lmBoth$coeff[3],lmBoth$coeff[2] + lmBoth$coeff[4]),col="black",lwd=3)
```



14/16

# Two lines, different slopes in R

```
summary(lmBoth)
```

Call:

```
lm(formula = hunger$Numeric ~ hunger$Year + hunger$Sex + hunger$Sex *  
    hunger$Year)
```

Residuals:

| Min    | 1Q     | Median | 3Q   | Max   |
|--------|--------|--------|------|-------|
| -25.11 | -11.55 | -2.12  | 7.02 | 46.22 |

Coefficients:

|                              | Estimate | Std. Error | t value | Pr(> t  ) |
|------------------------------|----------|------------|---------|-----------|
| (Intercept)                  | 529.4033 | 190.8185   | 2.77    | 0.0057 ** |
| hunger\$Year                 | -0.2562  | 0.0954     | -2.69   | 0.0074 ** |
| hunger\$SexMale              | 59.5912  | 269.8581   | 0.22    | 0.8253    |
| hunger\$Year:hunger\$SexMale | -0.0288  | 0.1349     | -0.21   | 0.8309    |
| ---                          |          |            |         |           |
| Signif. codes:               | 0 ****   | 0.001 ***  | 0.01 ** | 0.05 *    |
|                              | .        | .          | .       | 1         |

# Interactions for continuous variables

$$Hu_i = b_0 + b_1 In_i + b_2 Y_i + b_3 In_i \times Y_i + e_i^+$$

$b_0$  - percent hungry at year zero for children with whose parents have no income

$b_1$  - change in percent hungry for each dollar of income in year zero

$b_2$  - change in percent hungry in one year for children whose parents have no income

$b_3$  - increased change in percent hungry by year for each dollar of income - e.g. if income is \$10,000, then change in percent hungry in one year will be

$$b_2 + 1e4 \times b_3$$

$e_i^+$  - everything we didn't measure

**Lot's of care/caution needed!**

# Organizing a data analysis

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

# 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

2/12

# Raw Data

| ALLERGIES                        |  | MEDICATION HISTORY               |  |
|----------------------------------|--|----------------------------------|--|
| Last Updated: 01 Dec 2011 @ 0851 |  | Last Updated: 11 Apr 2011 @ 1737 |  |
| Allergy Name:                    | TRIMETHOPRIM                                       | Medication:                      | AMLODIPINE BESYLATE 10MG TAB   |
| Location:                        | DAYT29   | Instructions:                    | TAKE ONE TABLET BY MOUTH TAKE ONE-HALF TABLET FOR GRAPEFRUIT JUICE-- |
| Date Entered:                    | 09 Mar 2011  | Status:                          | Active   |
| Action:                          |  | Refills Remaining:               | 3  |
| Allergy Type:                    | DRUG   | Last Filled On:                  | 28 Aug 2010  |
| A Drug Class:                    | ANTI-INFECTIVES, OTHER                             | Initially Ordered On:            | 13 Aug 2010  |
| Observed/Historical:             | HISTORICAL   | Quantity:                        | 45   |
| Comments:                        | The reaction to this allergy was MILD (NO SQUELAE) | Days Supply:                     | 90   |
| Allergy Name:                    | TRAMADOL   | Pharmacy:                        | DAYTON   |
| Location:                        | DAYT29   | Prescription Number:             | 2718953  |
| Date Entered:                    | 09 Mar 2011  | Medication:                      | IBUPROFEN 600MG TAB  |
| Action:                          | URINARY RETENTION                                  | Instructions:                    | TAKE ONE TABLET BY MOUTH FOUR TIMES A DAY WITH FOOD                  |
| Allergy Type:                    | DRUG   | Status:                          | Active   |
| A Drug Class:                    | NON-OPIOID ANALGESICS                              | Refills Remaining:               | 3  |
| Observed/Historical:             | HISTORICAL   | Last Filled On:                  | 28 Aug 2010  |
| Comments:                        | gradually worsening difficulty emptying bladder    | Initially Ordered On:            | 01 Jul 2010  |

- Should be stored in your analysis folder
- If accessed from the web, include url, description, and date accessed in README

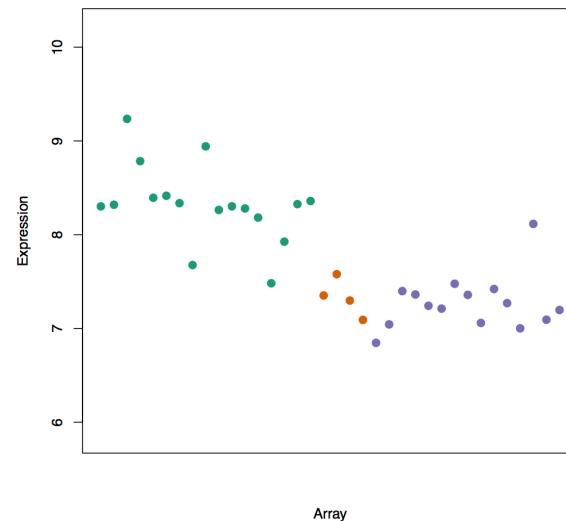
3/12

# Processed data

|    | A  | B          | C          | D          | E          | F         | G      | H | I | J | K | L | M | N | O | P |
|----|----|------------|------------|------------|------------|-----------|--------|---|---|---|---|---|---|---|---|---|
|    | id | problem_id | subject_id | start      | end        | 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 | 1307120013 | 2357      | A      |   |   |   |   |   |   |   |   |   |
| 7  | 6  | 101        | 19         | 1307119997 | 1307120021 | 2346      | 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         | 1307120021 | 1307120118 | 2310      | D      |   |   |   |   |   |   |   |   |   |
| 13 | 12 | 557        | 13         | 1307120021 | 1307120118 | 2357      | 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         | 1307120080 | 1307120249 | 2216      | B      |   |   |   |   |   |   |   |   |   |
| 19 | 18 | 472        | 12         | 1307120119 | 1307120170 | 2115      | 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         | 1307120153 | 1307120188 | 2197      | D      |   |   |   |   |   |   |   |   |   |
| 24 | 23 | 656        | 16         | 1307120154 | 1307120184 | 2080      | 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         | 1307120285 | 1307120343 | 2042      | E      |   |   |   |   |   |   |   |   |   |
| 29 | 28 | 22         | 18         | 1307120313 | 1307120353 | 2020      | C      |   |   |   |   |   |   |   |   |   |
| 30 | 29 | 54         | 19         | 1307120310 | 1307120386 | 2000      | B      |   |   |   |   |   |   |   |   |   |
| 31 | 30 | 502        | 16         | 1307120223 | 1307120336 | 2049      | B      |   |   |   |   |   |   |   |   |   |
| 32 | 31 | 44         | 16         | 1307120339 | 1307120352 | 2033      | A      |   |   |   |   |   |   |   |   |   |
| 33 | 32 | 315        | 14         | 1307120348 | 1307120362 | 2023      | B      |   |   |   |   |   |   |   |   |   |
| 34 | 33 | 385        | 15         | 1307120352 | 1307120383 | 1832      | E      |   |   |   |   |   |   |   |   |   |
| 35 | 34 | 550        | 13         | 1307120353 | 1307120444 | 1810      | 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 | 270        | 16         | 1307120437 | 1307120410 | 1875      | A      |   |   |   |   |   |   |   |   |   |

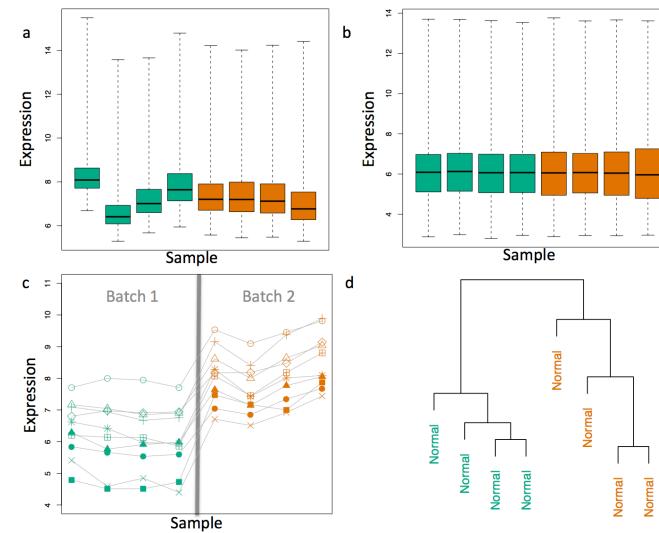
- Processed data should be named so it is easy to see which script generated the data.
- The processing script - processed data mapping should occur in the README
- Processed data should be tidy

# Exploratory figures



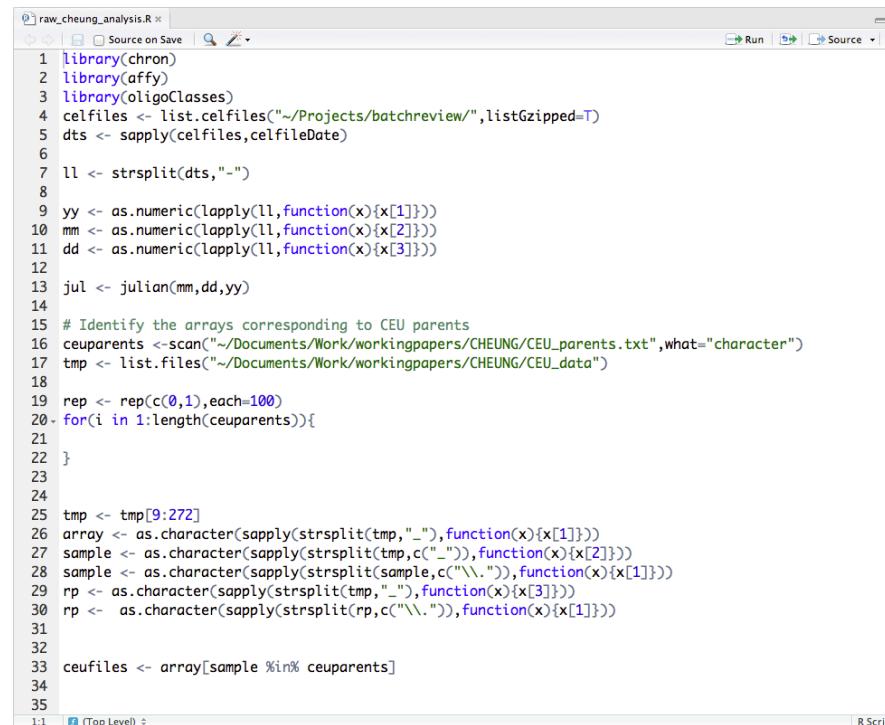
- Figures made during the course of your analysis, not necessarily part of your final report.
- They do not need to be "pretty"

# Final Figures



- Usually a small subset of the original figures
- Axes/colors set to make the figure clear
- Possibly multiple panels

# Raw scripts

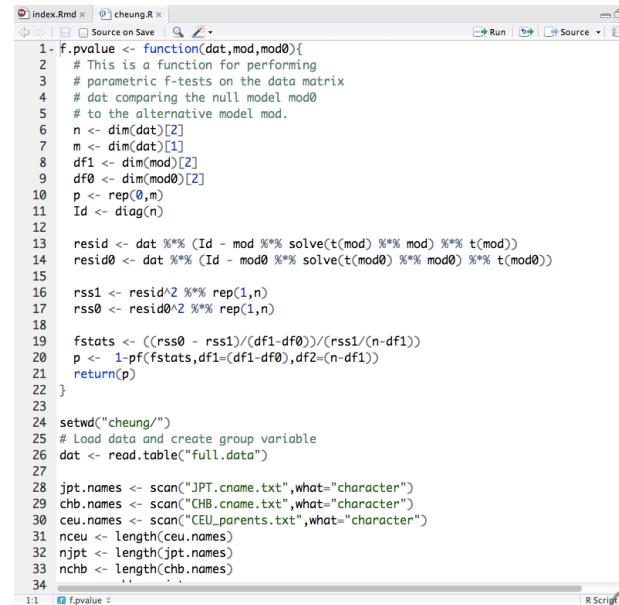


```
raw_cheung_analysis.R
library(chron)
library(affy)
library(oligoClasses)
celfiles <- list.celfiles("~/Projects/batchreview/",listGzipped=T)
dts <- sapply(celfiles,celfileDate)
ll <- strsplit(dts,"-")
yy <- as.numeric(lapply(ll,function(x){x[1]}))
mm <- as.numeric(lapply(ll,function(x){x[2]}))
dd <- as.numeric(lapply(ll,function(x){x[3]}))
jul <- julian(mm,dd,yy)
# Identify the arrays corresponding to CEU parents
ceuparents <- scan("~/Documents/Work/workingpapers/CHEUNG/CEU_parents.txt",what="character")
tmp <- list.files("~/Documents/Work/workingpapers/CHEUNG/CEU_data")
rep <- rep(c(0,1),each=100)
for(i in 1:length(ceuparents)){
array <- as.character(sapply(strsplit(tmp,"_"),function(x){x[1]}))
sample <- as.character(sapply(strsplit(tmp,c("_")),function(x){x[2]}))
sample <- as.character(sapply(strsplit(sample,c("\\.")),function(x){x[1]}))
rp <- as.character(sapply(strsplit(tmp,"_"),function(x){x[3]}))
rp <- as.character(sapply(strsplit(rp,c("\\.")),function(x){x[1]}))
ceufiles <- array[sample %in% ceuparents]
```

- May be less commented (but comments help you!)
- May be multiple versions
- May include analyses that are later discarded

7/12

# Final scripts



```
1 - f.pvalue <- function(dat,mod,mod0){  
2   # This is a function for performing  
3   # parametric f-tests on the data matrix  
4   # dat comparing the null model mod0  
5   # to the alternative model mod.  
6   n <- dim(dat)[2]  
7   m <- dim(dat)[1]  
8   df1 <- dim(mod)[2]  
9   df0 <- dim(mod0)[2]  
10  p <- rep(0,m)  
11  Id <- diag(n)  
12  
13  resid <- dat %*% (Id - mod %*% solve(t(mod) %*% mod) %*% t(mod))  
14  resid0 <- dat %*% (Id - mod0 %*% solve(t(mod0) %*% mod0) %*% t(mod0))  
15  
16  rss1 <- resid^2 %*% rep(1,n)  
17  rss0 <- resid0^2 %*% rep(1,n)  
18  
19  fstats <- ((rss0 - rss1)/(df1-df0))/(rss1/(n-df1))  
20  p <- 1-pf(fstats,df1=(df1-df0),df2=(n-df1))  
21  return(p)  
22 }  
23  
24 setwd("cheung")  
25 # Load data and create group variable  
26 dat <- read.table("full.data")  
27  
28 jpt.names <- scan("JPT cname.txt",what="character")  
29 chb.names <- scan("CHB cname.txt",what="character")  
30 ceu.names <- scan("CEU parents.txt",what="character")  
31 nceu <- length(ceu.names)  
32 njpt <- length(jpt.names)  
33 ncchb <- length(chb.names)  
34
```

- Clearly commented
  - Small comments liberally - what, when, why, how
  - Bigger commented blocks for whole sections
- Include processing details
- Only analyses that appear in the final write-up

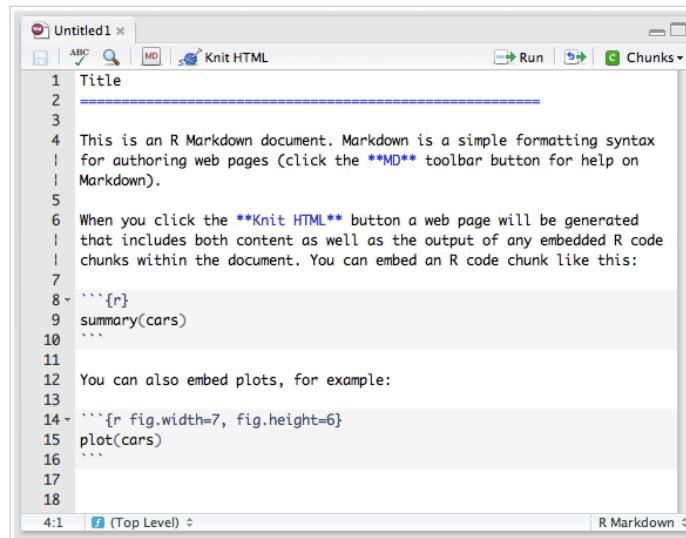
8/12

# R markdown files

## R Markdown Documents

To work with R Markdown (.Rmd) files in RStudio you first need to ensure that the [knitr](#) package (version 0.5 or later) is installed.

To create a new R Markdown file, go to **File | New | and select R Markdown**. A new file is created with a default template to get you oriented:



```
1 Title
2 =====
3
4 This is an R Markdown document. Markdown is a simple formatting syntax
| for authoring web pages (click the **MD** toolbar button for help on
| Markdown).
5
6 When you click the **Knit HTML** button a web page will be generated
| that includes both content as well as the output of any embedded R code
| chunks within the document. You can embed an R code chunk like this:
7
8 ```{r}
9 summary(cars)
10 ...
11
12 You can also embed plots, for example:
13
14 ```{r fig.width=7, fig.height=6}
15 plot(cars)
16 ...
17
18
```

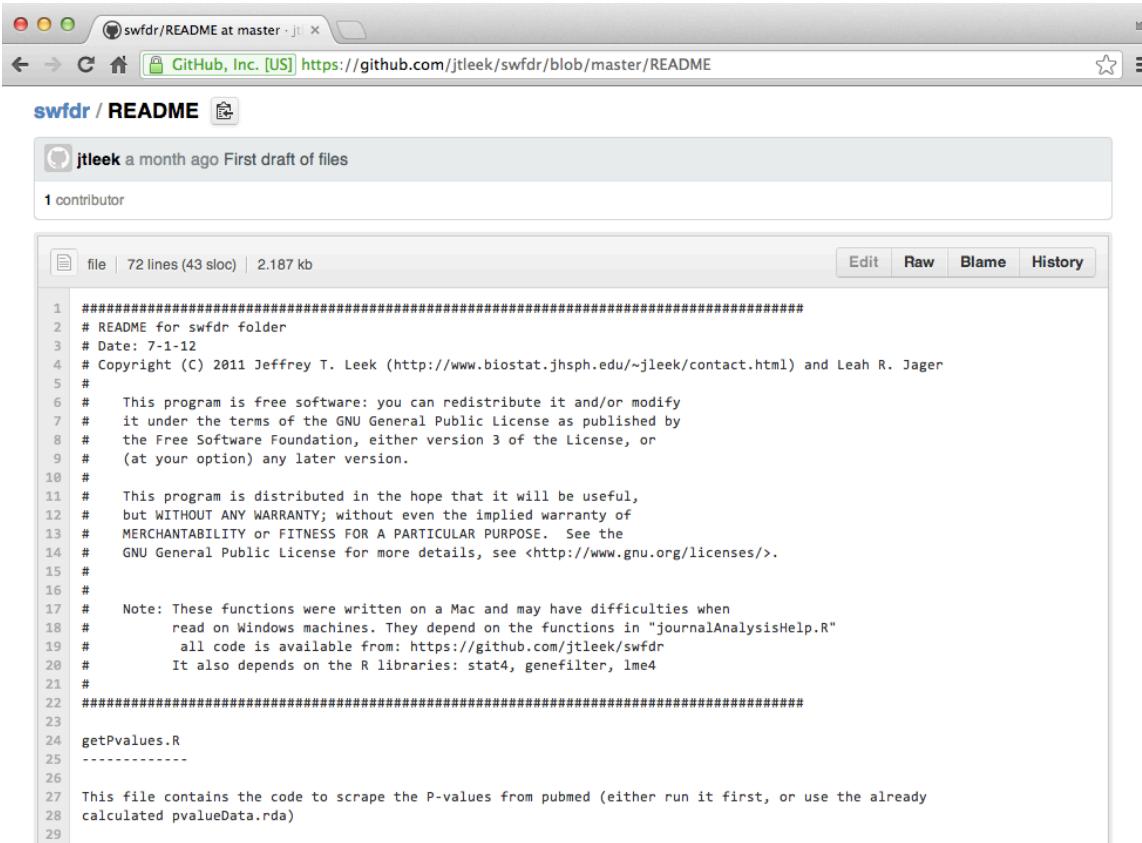
Note that the toolbar provides some useful tools for working with R Markdown:

- **Quick Reference** — Click the **MD** toolbar button to open a quick reference guide for Markdown.
- **Knit HTML** — Click to knit the current document to HTML, see the [Knitting to HTML](#) section below for more details.
- **Run** — Run the current line or selection of lines in the console. This allows running R code inside a code chunk similar to a normal R source file.
- **Chunks** — The chunks menu provides assistance with inserting, running, and chunk navigation. See the [Chunk Menu and Options](#) section below for more details.

- [R markdown](#) files can be used to generate reproducible reports
- Text and R code are integrated
- Very easy to create in [Rstudio](#)

9/12

# Readme files



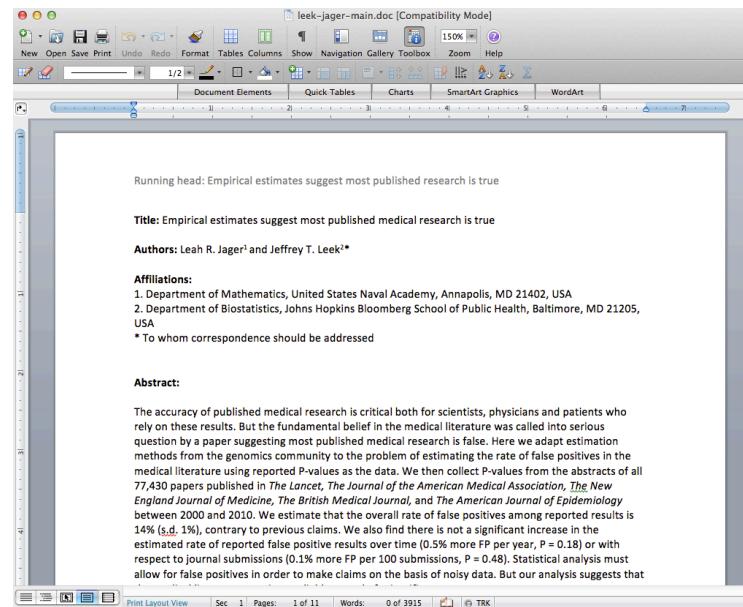
The screenshot shows a GitHub page for the 'swfdr' repository, specifically the 'README' file at the master branch. The page has a light gray header with the repository name 'swfdr/README at master - jt'. Below the header is a navigation bar with links for 'File', 'Raw', 'Blame', and 'History'. The main content area displays the README text, which is a standard GNU General Public License (GPL) notice. The text includes details about the program being free software, redistributable, and covered by the GPL version 3. It also notes dependencies on R libraries like stat4, genefilter, and lme4. At the bottom, it mentions a script named 'getPValues.R' and provides instructions for scraping P-values from PubMed. The text is presented in a monospaced font with line numbers on the left.

```
#####
# README for swfdr folder
# Date: 7-1-12
# Copyright (C) 2011 Jeffrey T. Leek (http://www.biostat.jhsph.edu/~jleek/contact.html) and Leah R. Jager
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details, see <http://www.gnu.org/licenses/>.
#
# Note: These functions were written on a Mac and may have difficulties when
#       read on Windows machines. They depend on the functions in "journalAnalysisHelp.R"
#       all code is available from: https://github.com/jtleek/swfdr
#       It also depends on the R libraries: stat4, genefilter, lme4
#
# -----
# getPValues.R
# -----
#
# This file contains the code to scrape the P-values from pubmed (either run it first, or use the already
# calculated pvalueData.rda)
```

- Not necessary if you use R markdown
- Should contain step-by-step instructions for analysis
- Here is an example <https://github.com/jtleek/swfdr/blob/master/README>

10/12

# Text of the document



- It should include a title, introduction (motivation), methods (statistics you used), results (including measures of uncertainty), and conclusions (including potential problems)
- It should tell a story
- *It should not include every analysis you performed*
- References should be included for statistical methods

# Further resources

- Information about a non-reproducible study that led to cancer patients being mistreated: [The Duke Saga Starter Set](#)
- [Reproducible research and Biostatistics](#)
- [Managing a statistical analysis project guidelines and best practices](#)
- [Project template](#) - a pre-organized set of files for data analysis

# P-values

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

# P-values

- Most common measure of "statistical significance"
- Commonly reported in papers
- Used for decision making (e.g. FDA)
- Controversial among statisticians
  - <http://warnercnr.colostate.edu/~anderson/thompson1.html>

# Not everyone thinks P-values are awful

## Simply Statistics

Home   About   Blog Roll   Courses   Editor's Picks   Interviews

← Why all #academics should have professional @twitter   Make us a part of your day – add Simply Statistics to your accounts   RSS feed →

### P-values and hypothesis testing get a bad rap – but we sometimes find them useful.

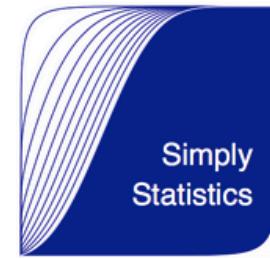
Posted on January 6, 2012 by admin

*This post written by Jeff Leek and Rafa Irizarry.*

The [p-value](#) is the most widely-known statistic. P-values are reported in a large majority of scientific publications that measure and report data. [R.A. Fisher](#) is widely credited with inventing the p-value. If he was cited every time a p-value was reported his paper would have, at the very least, **3 million** citations\* – making it the [most highly cited paper](#) of all time.

However, the p-value has a large number of very vocal critics. The criticisms of p-values, and hypothesis testing more generally, range from philosophical to practical. There are even [entire websites](#) dedicated to “debunking” p-values! One issue many statisticians raise with p-values is that they are easily misinterpreted, another is that p-values are not calibrated by sample size, another is that it ignores existing information or knowledge about the parameter in question, and yet another is that very significant (small) p-values may result even when the value of the parameter of interest is scientifically uninteresting.

We agree with all these criticisms. Yet, in practice, we find p-values useful and, if used



#### Recent Posts

- Issues with reproducibility at scale on Coursera
- Sunday data/statistics link roundup (2/3/2013)
- [pasteo](#) is statistical computing's most influential contribution of the 21st century
- Data supports claim that if Kobe stops ball hogging the Lakers will win more
- Sunday data/statistics link roundup (1/27/2013)

#### Recent Comments

<http://simplystatistics.org/2012/01/06/p-values-and-hypothesis-testing-get-a-bad-rap-but-we/>

3/19

# What is a P-value?

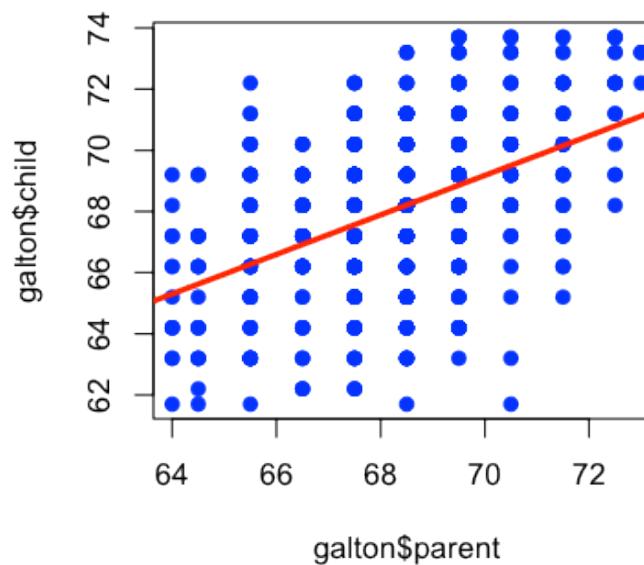
**Idea:** Suppose nothing is going on - how unusual is it to see the estimate we got?

**Approach:**

1. Define the hypothetical distribution of a data summary (statistic) when "nothing is going on" (*null hypothesis*)
2. Calculate the summary/statistic with the data we have (*test statistic*)
3. Compare what we calculated to our hypothetical distribution and see if the value is "extreme" (*p-value*)

# Galton data

```
library(UsingR); data(galton)
plot(galton$parent, galton$child, pch=19, col="blue")
lm1 <- lm(galton$child ~ galton$parent)
abline(lm1, col="red", lwd=3)
```



If there was no relation between mid-parent/child height would we be surprised to see a line that 5/19

# Null hypothesis/distribution

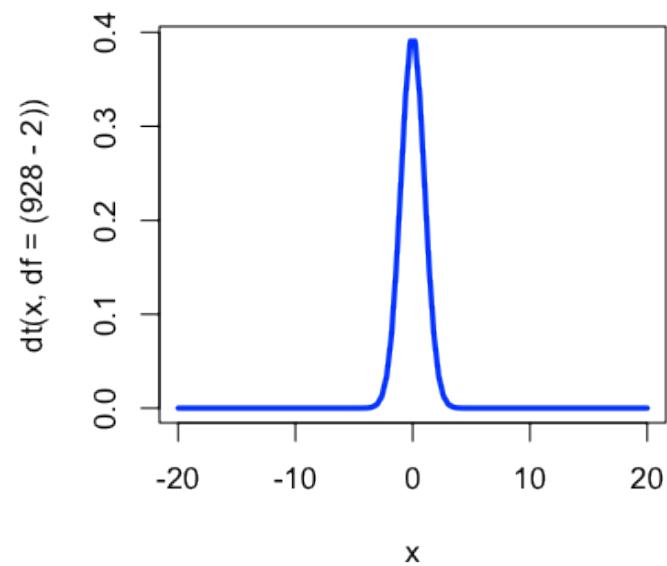
$$\frac{\hat{b}_1 - b_1}{S.E.(\hat{b}_1)} \sim t_{n-2}$$

$H_0$ : That there is no relationship between parent and child height ( $b_1 = 0$ ). Under the null hypothesis the distribution is:

$$\frac{\hat{b}_1}{S.E.(\hat{b}_1)} \sim t_{n-2}$$

# Null distribution

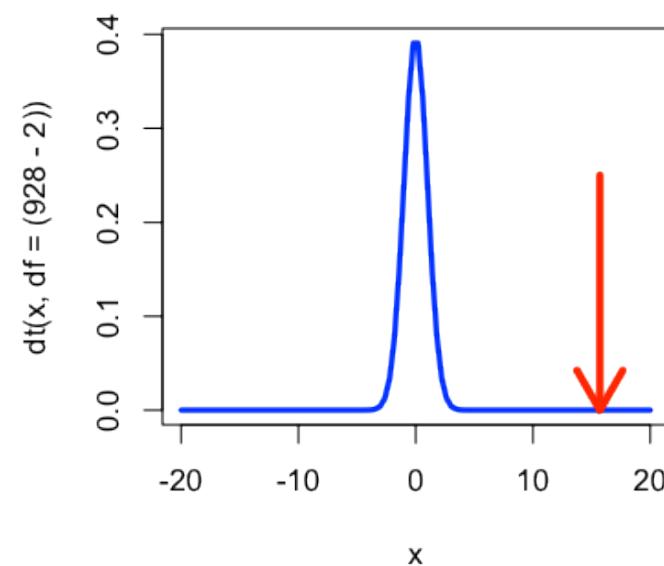
```
x <- seq(-20,20,length=100)
plot(x,dt(x,df=(928-2)),col="blue",lwd=3,type="l")
```



7/19

# Null distribution + observed statistic

```
x <- seq(-20,20,length=100)
plot(x,dt(x,df=(928-2)),col="blue",lwd=3,type="l")
arrows(summary(lm1)$coeff[2,3],0.25,summary(lm1)$coeff[2,3],0,col="red",lwd=4)
```



8/19

# Calculating p-values

```
summary(lm1)
```

Call:

```
lm(formula = galton$child ~ galton$parent)
```

Residuals:

| Min    | 1Q     | Median | 3Q    | Max   |
|--------|--------|--------|-------|-------|
| -7.805 | -1.366 | 0.049  | 1.634 | 5.926 |

Coefficients:

|                | Estimate | Std. Error | t value | Pr(> t )   |      |     |      |      |     |     |   |
|----------------|----------|------------|---------|------------|------|-----|------|------|-----|-----|---|
| (Intercept)    | 23.9415  | 2.8109     | 8.52    | <2e-16 *** |      |     |      |      |     |     |   |
| galton\$parent | 0.6463   | 0.0411     | 15.71   | <2e-16 *** |      |     |      |      |     |     |   |
| ---            |          |            |         |            |      |     |      |      |     |     |   |
| Signif. codes: | 0        | '***'      | 0.001   | '**'       | 0.01 | '*' | 0.05 | '. ' | 0.1 | ' ' | 1 |

Residual standard error: 2.24 on 926 degrees of freedom

Multiple R-squared: 0.21, Adjusted R-squared: 0.21

F-statistic: 247 on 1 and 926 DF, p-value: <2e-16

9/19

# A quick simulated example

```
set.seed(9898324)
yValues <- rnorm(10); xValues <- rnorm(10)
lm2 <- lm(yValues ~ xValues)
summary(lm2)
```

Call:

```
lm(formula = yValues ~ xValues)
```

Residuals:

| Min    | 1Q     | Median | 3Q    | Max   |
|--------|--------|--------|-------|-------|
| -1.546 | -0.570 | 0.136  | 0.771 | 1.052 |

Coefficients:

|             | Estimate | Std. Error | t value | Pr(> t ) |
|-------------|----------|------------|---------|----------|
| (Intercept) | 0.310    | 0.351      | 0.88    | 0.40     |
| xValues     | 0.289    | 0.389      | 0.74    | 0.48     |

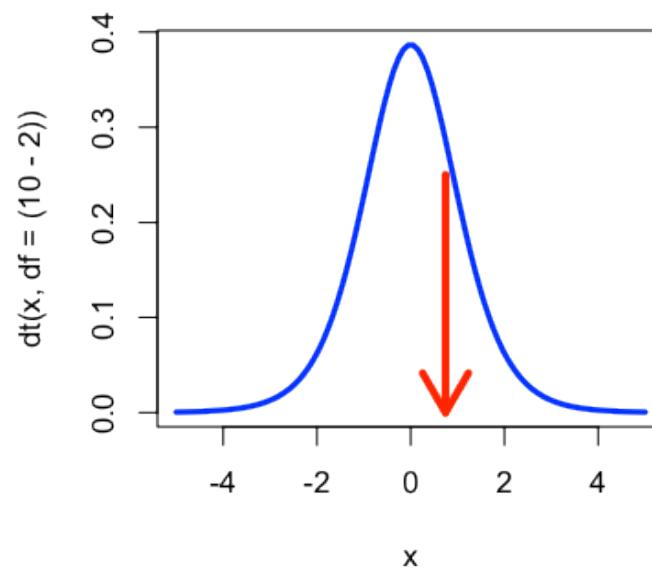
Residual standard error: 0.989 on 8 degrees of freedom

Multiple R-squared: 0.0644, Adjusted R-squared: -0.0525

10/19

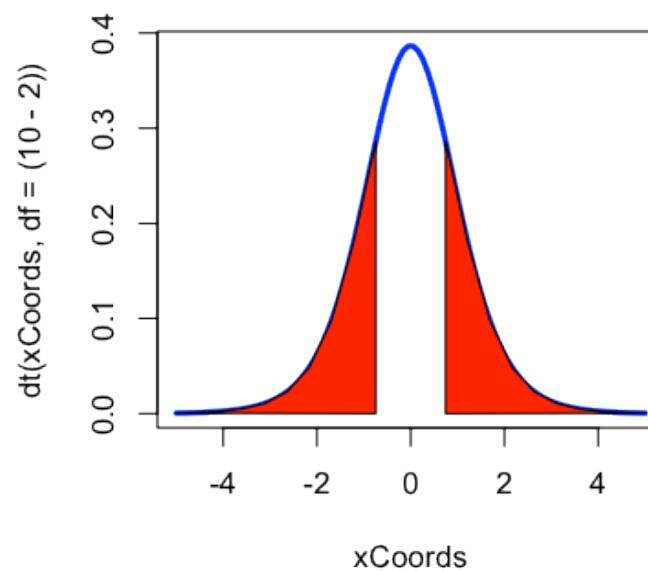
# A quick simulated example

```
x <- seq(-5,5,length=100)
plot(x,dt(x,df=(10-2)),col="blue",lwd=3,type="l")
arrows(summary(lm2)$coeff[2,3],0.25,sumMARY(lm2)$coeff[2,3],0,col="red",lwd=4)
```



# A quick simulated example

```
xCoords <- seq(-5,5,length=100)
plot(xCoords,dt(xCoords,df=(10-2)),col="blue",lwd=3,type="l")
xSequence <- c(seq(summary(lm2)$coeff[2,3],5,length=10),summary(lm2)$coeff[2,3])
ySequence <- c(dt(seq(summary(lm2)$coeff[2,3],5,length=10),df=8),0)
polygon(xSequence,ySequence,col="red"); polygon(-xSequence,ySequence,col="red")
```



12/19

# Simulate a ton of data sets with no signal

```
set.seed(8323); pValues <- rep(NA,100)
for(i in 1:100){
  xValues <- rnorm(20);yValues <- rnorm(20)
  pValues[i] <- summary(lm(yValues ~ xValues))$coeff[2,4]
}
hist(pValues,col="blue",main="",freq=F)
abline(h=1,col="red",lwd=3)
```

13/19

# Simulate a ton of data sets with signal

```
set.seed(8323); pValues <- rep(NA,100)
for(i in 1:100){
  xValues <- rnorm(20);yValues <- 0.2 * xValues + rnorm(20)
  pValues[i] <- summary(lm(yValues ~ xValues))$coeff[2,4]
}
hist(pValues,col="blue",main="",freq=F,xlim=c(0,1)); abline(h=1,col="red",lwd=3)
```

14/19

# Simulate a ton of data sets with signal

```
set.seed(8323); pValues <- rep(NA,100)
for(i in 1:100){
  xValues <- rnorm(100);yValues <- 0.2* xValues + rnorm(100)
  pValues[i] <- summary(lm(yValues ~ xValues))$coeff[2,4]
}
hist(pValues,col="blue",main="",freq=F,xlim=c(0,1)); abline(h=1,col="red",lwd=3)
```

# Some typical values (single test)

- $P < 0.05$  (significant)
- $P < 0.01$  (strongly significant)
- $P < 0.001$  (very significant)

In modern analyses, people generally report both the confidence interval and P-value. This is less true if many many hypotheses are tested.

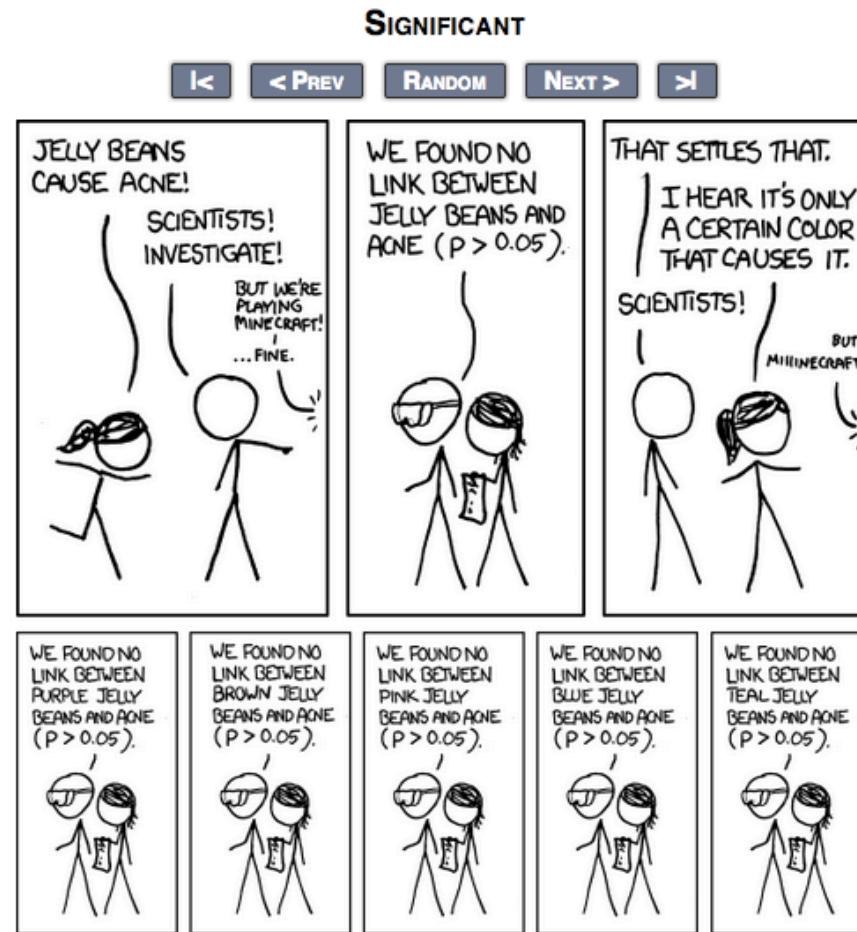
# How you interpret the results

```
summary(lm(galton$child ~ galton$parent))$coeff
```

|                | Estimate | Std. Error | t value | Pr(> t )  |
|----------------|----------|------------|---------|-----------|
| (Intercept)    | 23.9415  | 2.81088    | 8.517   | 6.537e-17 |
| galton\$parent | 0.6463   | 0.04114    | 15.711  | 1.733e-49 |

A one inch increase in parental height is associated with a 0.77 inch increase in child's height (95% CI: 0.42-1.12 inches). This difference was statistically significant ( $P < 0.001$ ).

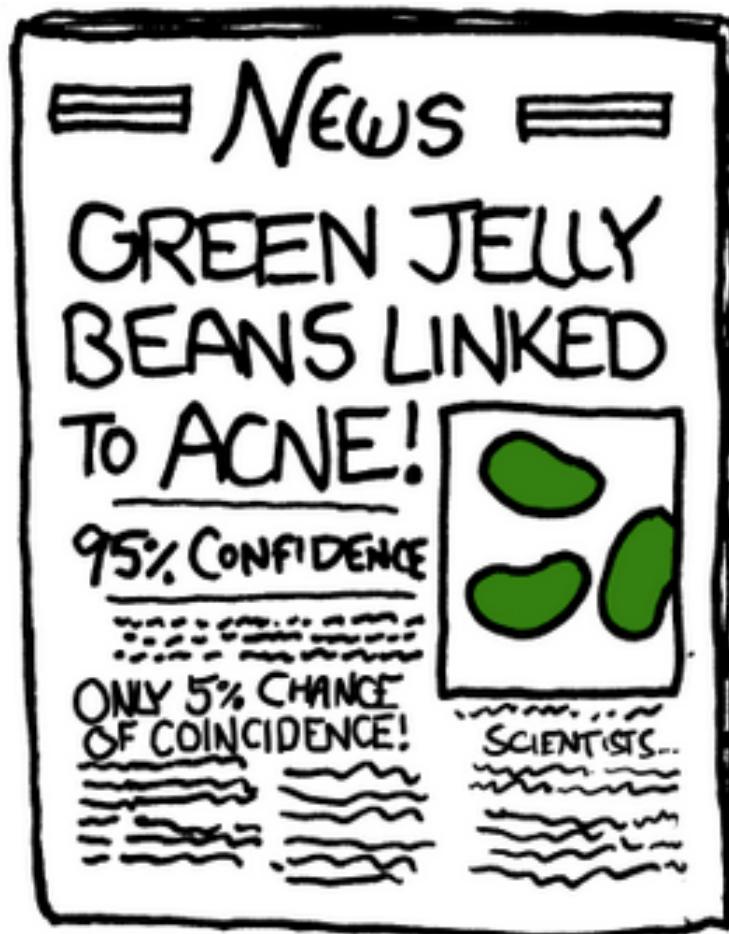
# Be careful!



<http://xkcd.com/882/>

18/19

# Be careful!



<http://xkcd.com/882/>

19/19

# Predicting with regression models

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

# Key ideas

- Use a standard regression model
  - lm
  - glm
- Predict new values with the coefficients
- Useful when the linear model is (nearly) correct

## Pros:

- Easy to implement
- Easy to interpret

## Cons:

- Often poor performance in nonlinear settings

# Example: Old faithful eruptions



(c) Wally Pacholka / AstroPics.com

Image Credit/Copyright Wally Pacholka <http://www.astropics.com/>

3/17

# Example: Old faithful eruptions

```
data(faithful)
dim(faithful)
```

```
[1] 272    2
```

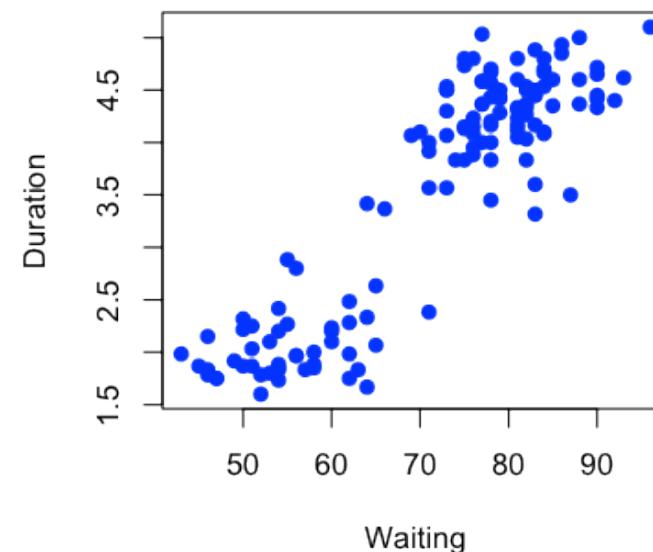
```
set.seed(333)
trainSamples <- sample(1:272, size=(272/2), replace=F)
trainFaith <- faithful[trainSamples,]
testFaith <- faithful[-trainSamples,]
head(trainFaith)
```

|     | eruptions | waiting |
|-----|-----------|---------|
| 128 | 4.500     | 82      |
| 23  | 3.450     | 78      |
| 263 | 1.850     | 58      |
| 154 | 4.600     | 81      |
| 6   | 2.883     | 55      |

4/17

# Eruption duration versus waiting time

```
plot(trainFaith$waiting,trainFaith$eruptions,pch=19,col="blue",xlab="Waiting",ylab="Duration")
```



# Fit a linear model

$$ED_i = b_0 + b_1 WT_i + e_i$$

```
lm1 <- lm(eruptions ~ waiting,data=trainFaith)
summary(lm1)
```

Call:

```
lm(formula = eruptions ~ waiting, data = trainFaith)
```

Residuals:

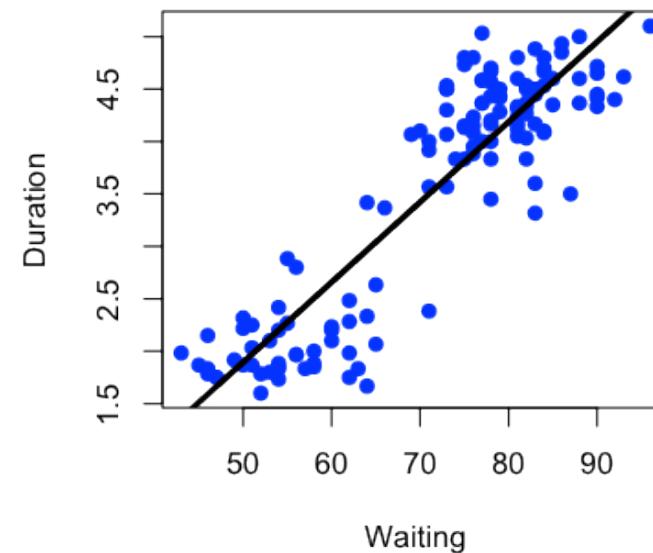
| Min     | 1Q      | Median | 3Q     | Max    |
|---------|---------|--------|--------|--------|
| -1.2969 | -0.3543 | 0.0487 | 0.3310 | 1.0760 |

Coefficients:

|                | Estimate | Std. Error | t value | Pr(> t ) |      |     |      |      |     |     |   |
|----------------|----------|------------|---------|----------|------|-----|------|------|-----|-----|---|
| (Intercept)    | -1.92491 | 0.22925    | -8.4    | 5.8e-14  | ***  |     |      |      |     |     |   |
| waiting        | 0.07639  | 0.00316    | 24.2    | < 2e-16  | ***  |     |      |      |     |     |   |
| ---            |          |            |         |          |      |     |      |      |     |     |   |
| Signif. codes: | 0        | '***'      | 0.001   | '**'     | 0.01 | '*' | 0.05 | '. ' | 0.1 | ' ' | 1 |

# Model fit

```
plot(trainFaith$waiting,trainFaith$eruptions,pch=19,col="blue",xlab="Waiting",ylab="Duration")  
lines(trainFaith$waiting,lm1$fitted,lwd=3)
```



# Predict a new value

$$\hat{ED} = \hat{b}_0 + \hat{b}_1 WT$$

```
coef(lm1)[1] + coef(lm1)[2]*80
```

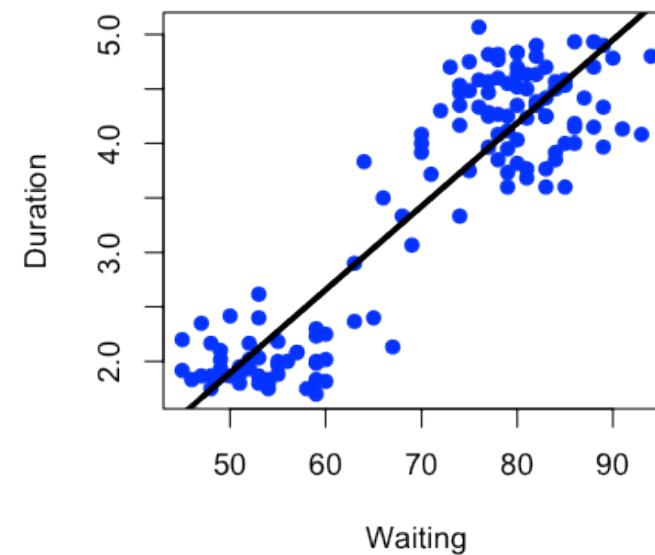
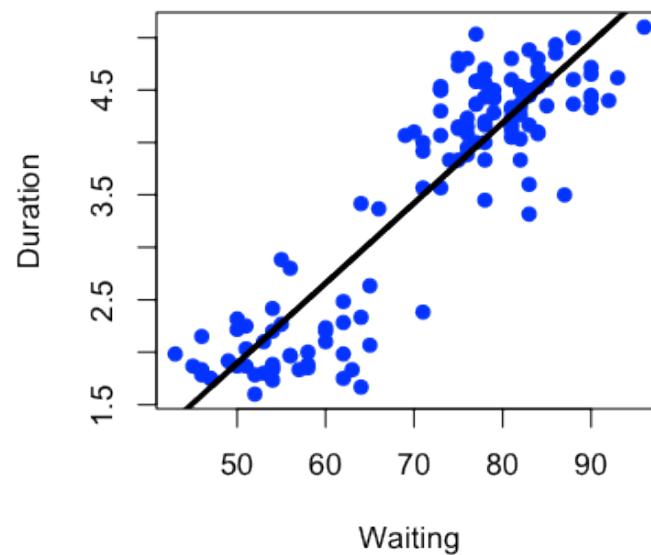
```
(Intercept)  
4.186
```

```
newdata <- data.frame(waiting=80)  
predict(lm1,newdata)
```

```
1  
4.186
```

# Plot predictions - training and test

```
par(mfrow=c(1,2))
plot(trainFaith$waiting,trainFaith$eruptions,pch=19,col="blue",xlab="Waiting",ylab="Duration")
lines(trainFaith$waiting,predict(lm1),lwd=3)
plot(testFaith$waiting,testFaith$eruptions,pch=19,col="blue",xlab="Waiting",ylab="Duration")
lines(testFaith$waiting,predict(lm1,newdata=testFaith),lwd=3)
```



9/17

# Get training set/test set errors

```
# Calculate RMSE on training  
sqrt(sum((lm1$fitted-trainFaith$eruptions)^2))
```

```
[1] 5.713
```

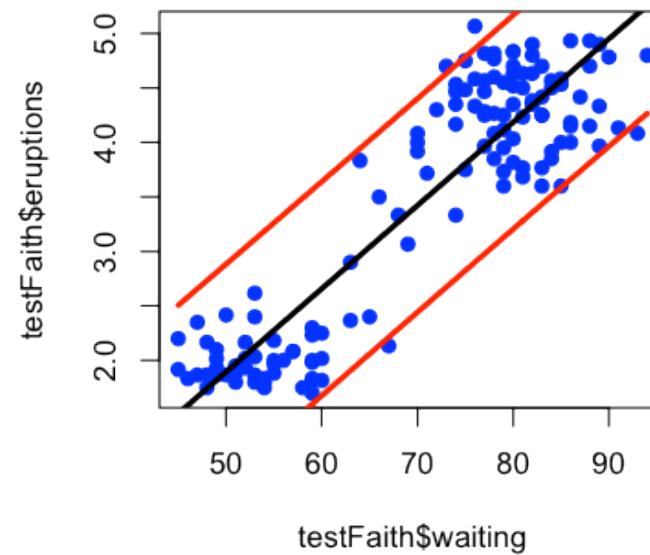
```
# Calculate RMSE on test  
sqrt(sum((predict(lm1,newdata=testFaith)-testFaith$eruptions)^2))
```

```
[1] 5.827
```

10/17

# Prediction intervals

```
pred1 <- predict(lm1,newdata=testFaith,interval="prediction")
ord <- order(testFaith$waiting)
plot(testFaith$waiting,testFaith$eruptions,pch=19,col="blue")
matlines(testFaith$waiting[ord],pred1[ord,],type="l",,col=c(1,2,2),lty = c(1,1,1), lwd=3)
```



11/17

# Example with binary data: Baltimore Ravens

**Baltimore Ravens** Sign in to personalize

AFC North 

[Clubhouse](#) [Stats](#) [Schedule](#) [Roster](#) [Splits](#) [Depth Chart](#) [Transactions](#) [Rankings](#) [Photos](#) [Stadium](#) [News](#) [Forum](#)

Sun Feb 3 Sun Feb 3 2012 Season

**Final Superbowl**

|  |   |  |
|--|---|--|
|  W<br>34-31 |  @<br>Baltimore (10-6) |  San Francisco (11-4-1) |
| Pass: Kaepernick 302 yds   | 1 2 3 4 T   | Overall: 10-6  |
| Rush: Gore 110 yds   | BAL 7 14 7 6 34   | vs AFC North: 4-2  |
| Rec: Crabtree 109 yds  | SF 3 3 17 8 31  | vs AFC: 8-4  |

Record:  
Overall: 10-6  
vs AFC North: 4-2  
vs AFC: 8-4

Team leaders:  
Pass: Flacco 3817 yds  
Rush: Rice 1143 yds  
Rec: Boldin 921 yds

[Tickets](#) [Shop](#)

**BALTIMORE TEAMS**



[http://espn.go.com/nfl/team/\\_/name/bal/baltimore-ravens](http://espn.go.com/nfl/team/_/name/bal/baltimore-ravens)

12/17

# Ravens Data

```
download.file("https://dl.dropbox.com/u/7710864/data/ravensData.rda",
              destfile=".~/data/ravensData.rda",method="curl")
load("./data/ravensData.rda")
head(ravensData)
```

|   | ravenWinNum | ravenWin | ravenScore | opponentScore |
|---|-------------|----------|------------|---------------|
| 1 | 1           | W        | 24         | 9             |
| 2 | 1           | W        | 38         | 35            |
| 3 | 1           | W        | 28         | 13            |
| 4 | 1           | W        | 34         | 31            |
| 5 | 1           | W        | 44         | 13            |
| 6 | 0           | L        | 23         | 24            |

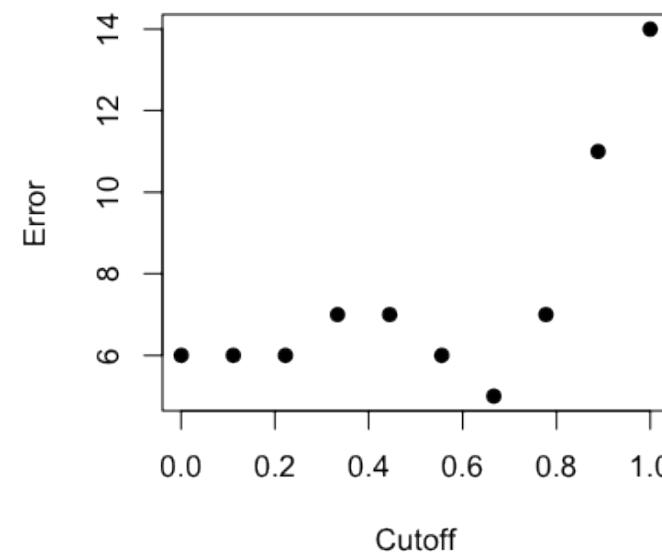
# Fit a logistic regression

$$\text{logit}(E[RW_i|RS_i]) = b_0 + b_1 RS_i$$

```
glm1 <- glm(ravenWinNum ~ ravenScore,family="binomial",data=ravensData)
par(mfrow=c(1,2))
boxplot(predict(glm1) ~ ravensData$ravenWinNum,col="blue")
boxplot(predict(glm1,type="response") ~ ravensData$ravenWinNum,col="blue")
```

# Choosing a cutoff (re-substitution)

```
xx <- seq(0,1,length=10); err <- rep(NA,10)
for(i in 1:length(xx)){
  err[i] <- sum((predict(glm1,type="response") > xx[i])) != ravensData$ravenWinNum
}
plot(xx,err,pch=19,xlab="Cutoff",ylab="Error")
```



# Comparing models with cross validation

```
library(boot)
cost <- function(win, pred = 0) mean(abs(win-pred) > 0.5)
glm1 <- glm(ravenWinNum ~ ravenScore,family="binomial",data=ravensData)
glm2 <- glm(ravenWinNum ~ ravenScore,family="gaussian",data=ravensData)
cv1 <- cv.glm(ravensData,glm1,cost,K=3)
cv2 <- cv.glm(ravensData,glm2,cost,K=3)
cv1$delta
```

```
[1] 0.350 0.365
```

```
cv2$delta
```

```
[1] 0.40 0.42
```

16/17

# Notes and further reading

- Regression models with multiple covariates can be included
- Often useful in combination with other models
- [Elements of statistical learning](#)
- [Modern applied statistics with S](#)

17/17

# Predicting with trees

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

# Key ideas

- Iteratively split variables into groups
- Split where maximally predictive
- Evaluate "homogeneity" within each branch
- Fitting multiple trees often works better (forests)

## Pros:

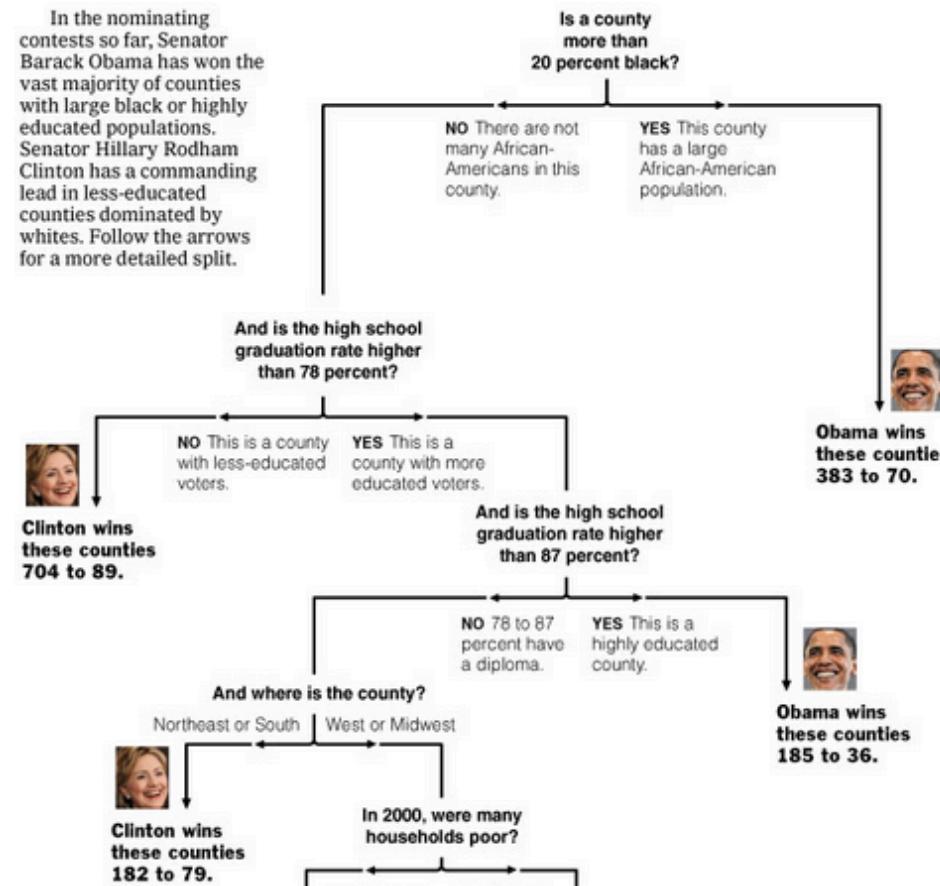
- Easy to implement
- Easy to interpret
- Better performance in nonlinear settings

## Cons:

- Without pruning/cross-validation can lead to overfitting
- Harder to estimate uncertainty
- Results may be variable

# Example Tree

## Decision Tree: The Obama-Clinton Divide



<http://graphics8.nytimes.com/images/2008/04/16/us/0416-nat-subOBAMA.jpg>

3/18

# Basic algorithm

1. Start with all variables in one group
2. Find the variable/split that best separates the outcomes
3. Divide the data into two groups ("leaves") on that split ("node")
4. Within each split, find the best variable/split that separates the outcomes
5. Continue until the groups are too small or sufficiently "pure"

# Measures of impurity

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \text{ in Leaf } m} \mathbb{1}(y_i = k)$$

**Misclassification Error:**

$$1 - \hat{p}_{mk(m)}$$

**Gini index:**

$$\sum_{k \neq k'} \hat{p}_{mk} \times \hat{p}_{mk'} = \sum_{k=1}^K \hat{p}_{mk} (1 - \hat{p}_{mk})$$

**Cross-entropy or deviance:**

$$-\sum_{k=1}^K \hat{p}_{mk} \ln \hat{p}_{mk}$$

# Example: Iris Data

```
data(iris)  
names(iris)
```

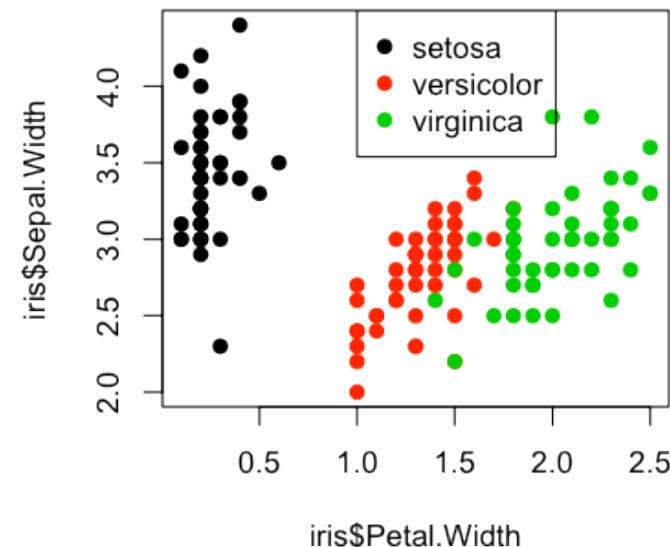
```
[1] "Sepal.Length" "Sepal.Width"   "Petal.Length" "Petal.Width"   "Species"
```

```
table(iris$Species)
```

|        |            |           |
|--------|------------|-----------|
| setosa | versicolor | virginica |
| 50     | 50         | 50        |

# Iris petal widths/sepal width

```
plot(iris$Petal.Width,iris$Sepal.Width,pch=19,col=as.numeric(iris$Species))  
legend(1,4.5,legend=unique(iris$Species),col=unique(as.numeric(iris$Species)),pch=19)
```



7/18

# Iris petal widths/sepal width

```
# An alternative is library(rpart)
library(tree)
tree1 <- tree(Species ~ Sepal.Width + Petal.Width,data=iris)
summary(tree1)
```

Classification tree:

```
tree(formula = Species ~ Sepal.Width + Petal.Width, data = iris)
```

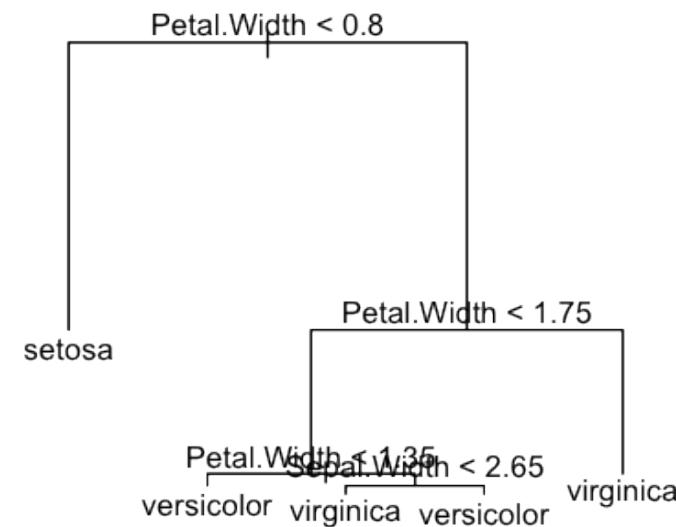
Number of terminal nodes: 5

Residual mean deviance: 0.204 = 29.6 / 145

Misclassification error rate: 0.0333 = 5 / 150

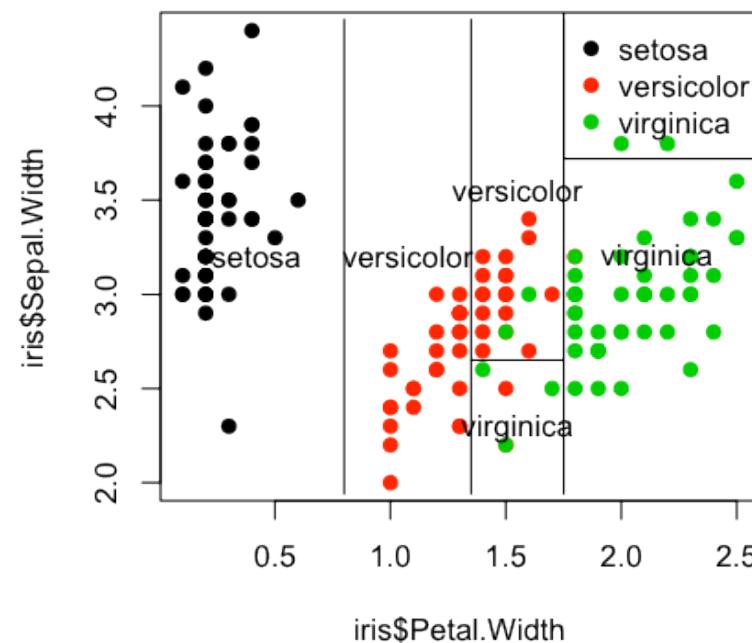
# Plot tree

```
plot(tree1)  
text(tree1)
```



# Another way of looking at a CART model

```
plot(iris$Petal.Width,iris$Sepal.Width,pch=19,col=as.numeric(iris$Species))
partition.tree(tree1,label="Species",add=TRUE)
legend(1.75,4.5,legend=unique(iris$Species),col=unique(as.numeric(iris$Species)),pch=19)
```



10/18

# Predicting new values

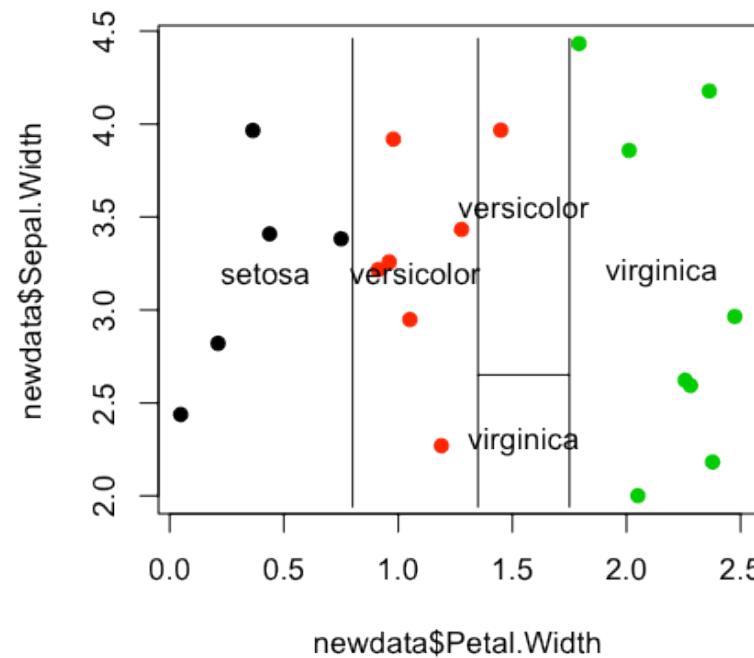
```
set.seed(32313)
newdata <- data.frame(Petal.Width = runif(20,0,2.5),Sepal.Width = runif(20,2,4.5))
pred1 <- predict(tree1,newdata)
pred1
```

|    | setosa | versicolor | virginica |
|----|--------|------------|-----------|
| 1  | 0      | 0.02174    | 0.97826   |
| 2  | 0      | 0.02174    | 0.97826   |
| 3  | 1      | 0.00000    | 0.00000   |
| 4  | 0      | 1.00000    | 0.00000   |
| 5  | 0      | 0.02174    | 0.97826   |
| 6  | 0      | 0.02174    | 0.97826   |
| 7  | 0      | 0.02174    | 0.97826   |
| 8  | 0      | 0.90476    | 0.09524   |
| 9  | 0      | 1.00000    | 0.00000   |
| 10 | 0      | 0.02174    | 0.97826   |
| 11 | 0      | 1.00000    | 0.00000   |
| 12 | 1      | 0.00000    | 0.00000   |
| 13 | 1      | 0.00000    | 0.00000   |
| 14 | 1      | 0.00000    | 0.00000   |

11/18

# Overlaying new values

```
pred1 <- predict(tree1,newdata,type="class")
plot(newdata$Petal.Width,newdata$Sepal.Width,col=as.numeric(pred1),pch=19)
partition.tree(tree1,"Species",add=TRUE)
```



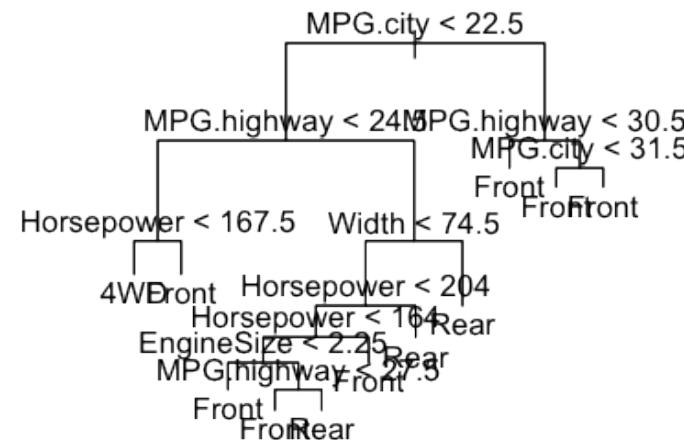
# Pruning trees example: Cars

```
data(Cars93, package="MASS")
head(Cars93)
```

|   | Manufacturer | Model     | Type       | Min.Price  | Price       | Max.Price      | MPG.city        | MPG.highway        | AirBags            |
|---|--------------|-----------|------------|------------|-------------|----------------|-----------------|--------------------|--------------------|
| 1 | Acura        | Integra   | Small      | 12.9       | 15.9        | 18.8           | 25              | 31                 | None               |
| 2 | Acura        | Legend    | Midsized   | 29.2       | 33.9        | 38.7           | 18              | 25                 | Driver & Passenger |
| 3 | Audi         | 90        | Compact    | 25.9       | 29.1        | 32.3           | 20              | 26                 | Driver only        |
| 4 | Audi         | 100       | Midsized   | 30.8       | 37.7        | 44.6           | 19              | 26                 | Driver & Passenger |
| 5 | BMW          | 535i      | Midsized   | 23.7       | 30.0        | 36.2           | 22              | 30                 | Driver only        |
| 6 | Buick        | Century   | Midsized   | 14.2       | 15.7        | 17.3           | 22              | 31                 | Driver only        |
|   | DriveTrain   | Cylinders | EngineSize | Horsepower | RPM         | Rev.per.mile   | Man.trans.avail | Fuel.tank.capacity |                    |
| 1 | Front        | 4         | 1.8        | 140        | 6300        | 2890           | Yes             |                    | 13.2               |
| 2 | Front        | 6         | 3.2        | 200        | 5500        | 2335           | Yes             |                    | 18.0               |
| 3 | Front        | 6         | 2.8        | 172        | 5500        | 2280           | Yes             |                    | 16.9               |
| 4 | Front        | 6         | 2.8        | 172        | 5500        | 2535           | Yes             |                    | 21.1               |
| 5 | Rear         | 4         | 3.5        | 208        | 5700        | 2545           | Yes             |                    | 21.1               |
| 6 | Front        | 4         | 2.2        | 110        | 5200        | 2565           | No              |                    | 16.4               |
|   | Passengers   | Length    | Wheelbase  | Width      | Turn.circle | Rear.seat.room | Luggage.room    | Weight             | Origin             |
| 1 | 5            | 177       | 102        | 68         | 37          | 26.5           | 11              | 2705               | non-USA            |
| 2 | 5            | 195       | 115        | 71         | 38          | 30.0           | 15              | 3560               | non-USA            |

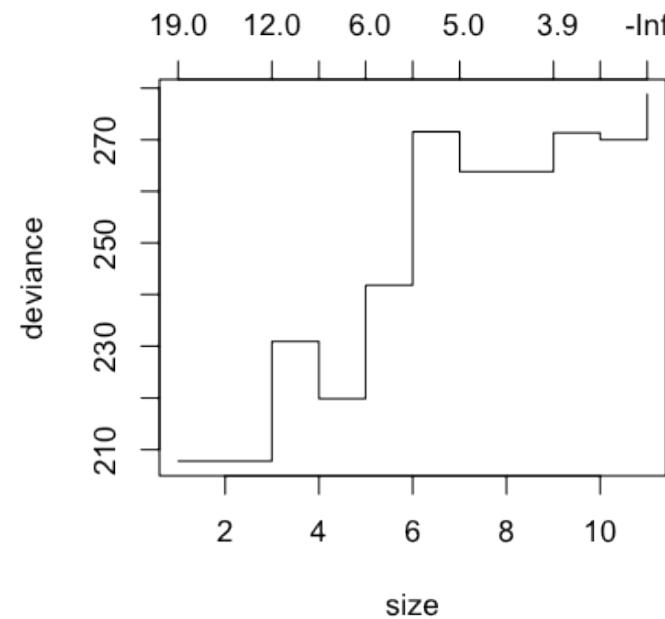
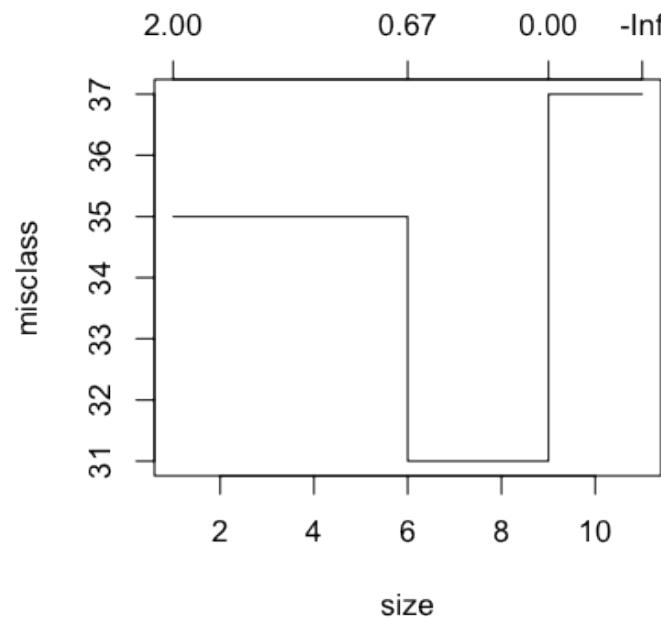
# Build a tree

```
treeCars <- tree(DriveTrain ~ MPG.city + MPG.highway + AirBags +  
    EngineSize + Width + Length + Weight + Price + Cylinders +  
    Horsepower + Wheelbase, data=Cars93)  
  
plot(treeCars)  
text(treeCars)
```



# Plot errors

```
par(mfrow=c(1,2))
plot(cv.tree(treeCars, FUN=prune.tree, method="misclass"))
plot(cv.tree(treeCars))
```

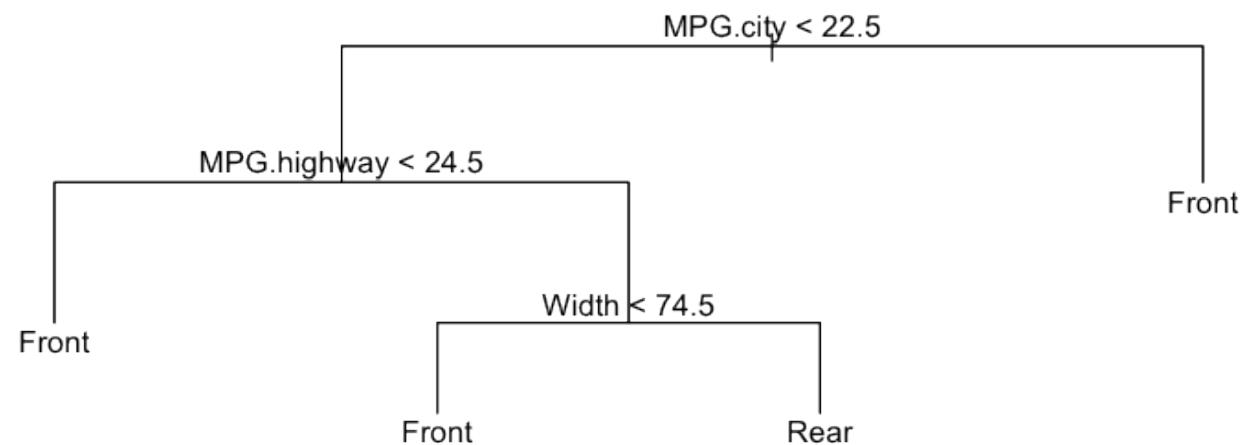


```
pruneTree <- prune.tree(treeCars, best=4)
```

15/18

# Prune the tree

```
pruneTree <- prune.tree(treeCars,best=4)  
plot(pruneTree)  
text(pruneTree)
```



# Show resubstitution error \*

```
table(Cars93$DriveTrain,predict(pruneTree,type="class"))
```

|       | 4WD | Front | Rear |
|-------|-----|-------|------|
| 4WD   | 5   | 5     | 0    |
| Front | 1   | 66    | 0    |
| Rear  | 1   | 10    | 5    |

```
table(Cars93$DriveTrain,predict(treeCars,type="class"))
```

|       | 4WD | Front | Rear |
|-------|-----|-------|------|
| 4WD   | 5   | 5     | 0    |
| Front | 2   | 61    | 4    |
| Rear  | 0   | 3     | 13   |

17/18

- Note that cross validation error is a better measure of test set accuracy

# Notes and further resources

- [Hector Corrada Bravo's Notes, code](#)
- [Cosma Shalizi's notes](#)
- [Elements of Statistical Learning](#)
- [Classification and regression trees](#)
- [Random forests](#)

# Prediction study design

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

# Key ideas

- Motivation
- Steps in predictive studies
- Choosing the right data
- Error measures
- Study design

2/15

# Why predict? Glory!



<http://www.zimbio.com/photos/Chris+Volinsky>

3/15

# Why predict? Riches!

Information Data Forum Leaderboard



**Improve Healthcare,  
Win \$3,000,000.**

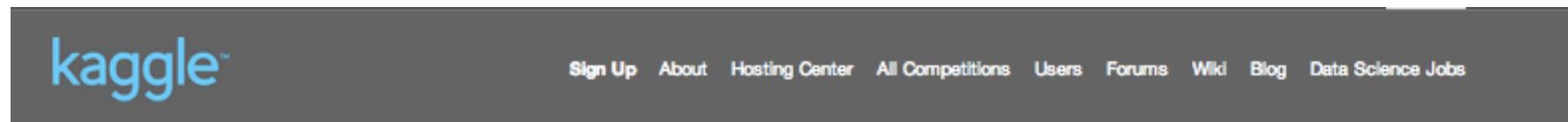
COMPETITION GOAL

Identify patients who will be admitted to a hospital within the next year, using historical claims data.

<http://www.heritagehealthprize.com/c/hhp>

4/15

# Why predict? For sport!

The image shows the top navigation bar of the Kaggle website. On the left is the "kaggle™" logo. To its right is a horizontal menu with the following items: Sign Up, About, Hosting Center, All Competitions, Users, Forums, Wiki, Blog, and Data Science Jobs.

## What's in your data?

### Participate in competitions

Kaggle is an arena where you can match your data science skills against a global cadre of experts in statistics, mathematics, and machine learning. Whether you're a world-class algorithm wizard competing for prize money or a novice looking to learn from the best, here's your chance to jump in and geek out, for fame, fortune, or fun.

[Join as a participant](#)

(Need convincing?)

### Create a competition

Kaggle is a platform for data prediction competitions that allows organizations to post their data and have it scrutinized by the world's best data scientists. In exchange for a prize, winning competitors provide the algorithms that beat all other methods of solving a data crunching problem. Most data problems can be framed as a competition.

[Learn more about hosting](#)

<http://www.kaggle.com/>

5/15

# Why predict? To save lives!

**Oncotype DX® reveals  
the underlying biology that  
changes treatment decisions  
37% of the time**

Uncover the Unexpected™



<http://www.oncotypedx.com/en-US/Home>

6/15

# Steps in building a prediction

1. Find the right data
2. Define your error rate
3. Split data into:
  - Training
  - Testing
  - Validation (optional)
4. On the training set pick features
5. On the training set pick prediction function
6. On the training set cross-validate
7. If no validation - apply 1x to test set
8. If validation - apply to test set and refine
9. If validation - apply 1x to validation

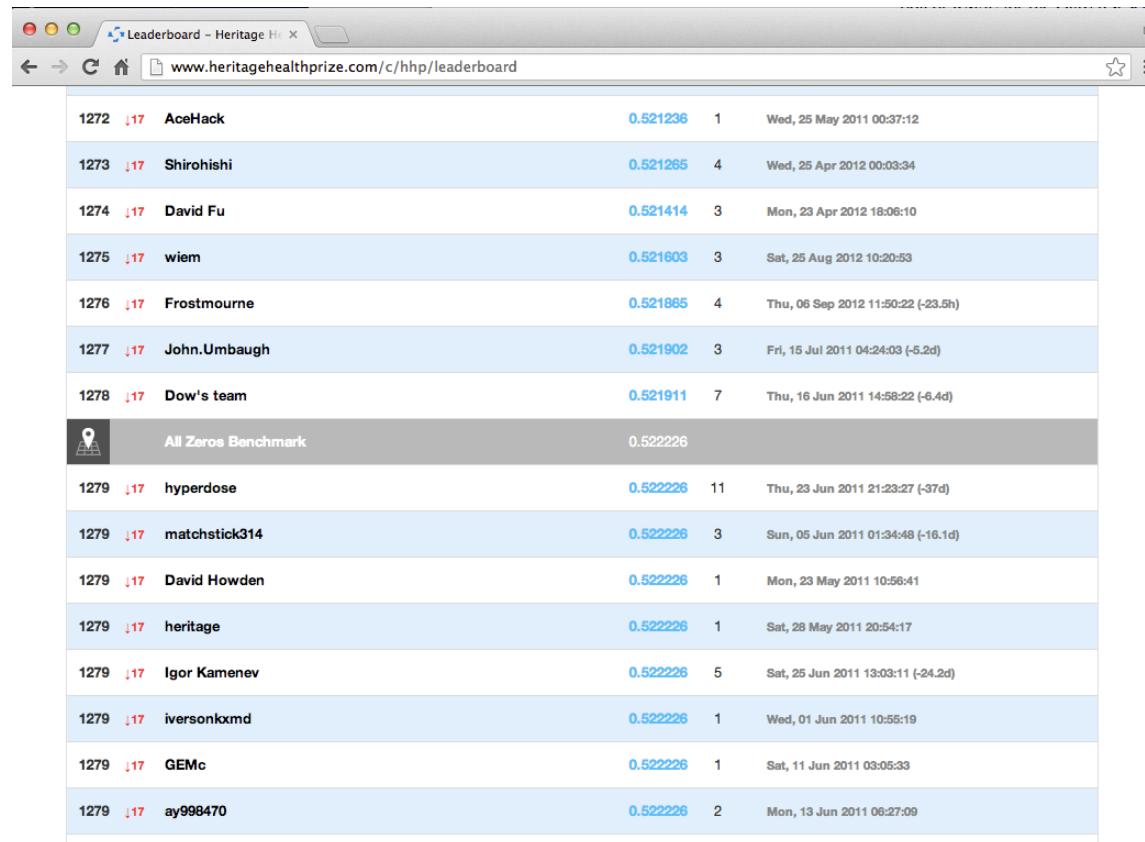
7/15

# Find the right data

1. In some cases it is easy (movie ratings -> new movie ratings)
2. It may be harder (gene expression data -> disease)
3. Depends strongly on the definition of "good prediction".
4. Often more data > better models
5. Know the bench mark
6. You need to start with raw data for predictions - processing is often cross-sample.

# Know the benchmarks

Probability of perfect classification is approximately  $\left(\frac{1}{2}\right)^{\text{test set sample size}}$



The screenshot shows a web browser window with the title "Leaderboard - Heritage Ho". The URL in the address bar is "www.heritagehealthprize.com/c/hhp/leaderboard". The page displays a table of 15 entries, each representing a participant's score and timestamp. The table includes a row for the "All Zeros Benchmark".

|   |     |               |          |    |                                    |
|---|-----|---------------|----------|----|------------------------------------|
| 1272  | ↓17 | AceHack       | 0.521236 | 1  | Wed, 25 May 2011 00:37:12          |
| 1273  | ↓17 | Shirohishi    | 0.521265 | 4  | Wed, 25 Apr 2012 00:03:34          |
| 1274  | ↓17 | David Fu      | 0.521414 | 3  | Mon, 23 Apr 2012 18:06:10          |
| 1275  | ↓17 | wiem          | 0.521603 | 3  | Sat, 25 Aug 2012 10:20:53          |
| 1276  | ↓17 | Frostmourne   | 0.521865 | 4  | Thu, 06 Sep 2012 11:50:22 (-23.5h) |
| 1277  | ↓17 | John.Umbaugh  | 0.521902 | 3  | Fri, 15 Jul 2011 04:24:03 (-5.2d)  |
| 1278  | ↓17 | Dow's team    | 0.521911 | 7  | Thu, 16 Jun 2011 14:58:22 (-6.4d)  |
|  All Zeros Benchmark |     |               | 0.522226 |    |                                    |
| 1279  | ↓17 | hyperdose     | 0.522226 | 11 | Thu, 23 Jun 2011 21:23:27 (-37d)   |
| 1279  | ↓17 | matchstick314 | 0.522226 | 3  | Sun, 05 Jun 2011 01:34:48 (-16.1d) |
| 1279  | ↓17 | David Howden  | 0.522226 | 1  | Mon, 23 May 2011 10:56:41          |
| 1279  | ↓17 | heritage      | 0.522226 | 1  | Sat, 28 May 2011 20:54:17          |
| 1279  | ↓17 | Igor Kamenev  | 0.522226 | 5  | Sat, 25 Jun 2011 13:03:11 (-24.2d) |
| 1279  | ↓17 | iversonlxmd   | 0.522226 | 1  | Wed, 01 Jun 2011 10:55:19          |
| 1279  | ↓17 | GEMc          | 0.522226 | 1  | Sat, 11 Jun 2011 03:05:33          |
| 1279  | ↓17 | ay990470      | 0.522226 | 2  | Mon, 13 Jun 2011 06:27:09          |

<http://www.heritagehealthprize.com/c/hhp/leaderboard>

9/15

# Defining true/false positives

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](http://en.wikipedia.org/wiki/Sensitivity_and_specificity)

10/15

# Define your error rate

|                 |                             | Condition<br>(as determined by "Gold standard")                                     |   |  |
|-----------------|-----------------------------|---|---|--|
|                 |                             | Condition Positive  | Condition Negative  |  |
| Test<br>Outcome | Test<br>Outcome<br>Positive | True Positive   | False Positive<br>(Type I error)  | Positive predictive value =<br>$\frac{\sum \text{True Positive}}{\sum \text{Test Outcome Positive}}$ |
|                 | Test<br>Outcome<br>Negative | False Negative<br>(Type II error)   | True Negative   | Negative predictive value =<br>$\frac{\sum \text{True Negative}}{\sum \text{Test Outcome Negative}}$ |
|                 |                             | Sensitivity =<br>$\frac{\sum \text{True Positive}}{\sum \text{Condition Positive}}$ | Specificity =<br>$\frac{\sum \text{True Negative}}{\sum \text{Condition Negative}}$ |  |

[http://en.wikipedia.org/wiki/Sensitivity\\_and\\_specificity](http://en.wikipedia.org/wiki/Sensitivity_and_specificity)

11/15

# Why your choice matters

|  |                       | Patients with bowel cancer<br>(as confirmed on endoscopy)                        |   |   |
|--|-----------------------|--|---|---|
|  |                       | Condition Positive   | Condition Negative  |   |
| Fecal Occult Blood Screen Test Outcome | Test Outcome Positive | True Positive (TP) = 20  | False Positive (FP) = 180   | Positive predictive value<br>$= TP / (TP + FP)$<br>$= 20 / (20 + 180)$<br>$= 10\%$            |
|  | Test Outcome Negative | False Negative (FN) = 10   | True Negative (TN) = 1820   | Negative predictive value<br>$= TN / (FN + TN)$<br>$= 1820 / (10 + 1820)$<br>$\approx 99.5\%$ |
|  |                       | <b>Sensitivity</b><br>$= TP / (TP + FN)$<br>$= 20 / (20 + 10)$<br>$\approx 67\%$ | <b>Specificity</b><br>$= TN / (FP + TN)$<br>$= 1820 / (180 + 1820)$<br>$= 91\%$ |   |

[http://en.wikipedia.org/wiki/Sensitivity\\_and\\_specificity](http://en.wikipedia.org/wiki/Sensitivity_and_specificity)

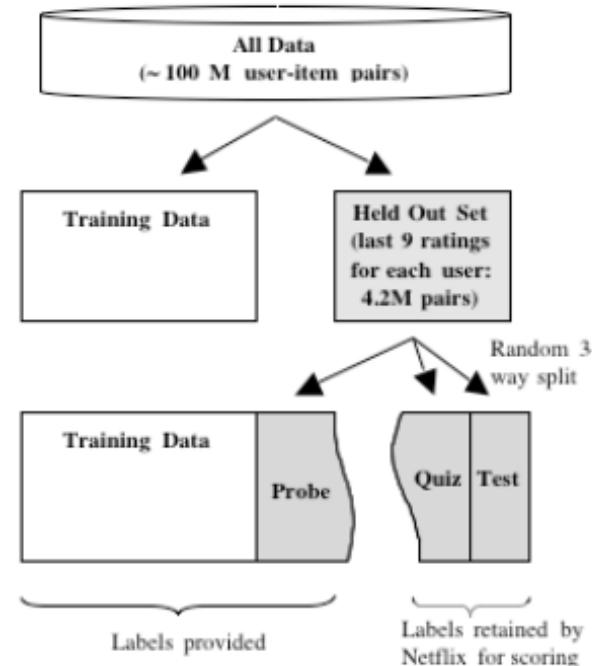
12/15

# Common error measures

1. Mean squared error (or root mean squared error)
  - Continuous data, sensitive to outliers
2. Median absolute deviation
  - Continuous data, often more robust
3. Sensitivity (recall)
  - If you want few missed positives
4. Specificity
  - If you want few negatives called positives
5. Accuracy
  - Weights false positives/negatives equally
6. Concordance
  - One example is kappa

13/15

# Study design



<http://www2.research.att.com/~volinsky/papers/ASASStatComp.pdf>

# Key issues and further resources

*Issues:*

1. Accuracy
2. Overfitting
3. Interpretability
4. Computational speed

*Resources:*

1. [Practical machine learning](#)
2. [Elements of statistical learning](#)
3. [Coursera machine learning](#)
4. [Machine learning for hackers](#)

15/15

# Regression in the real world

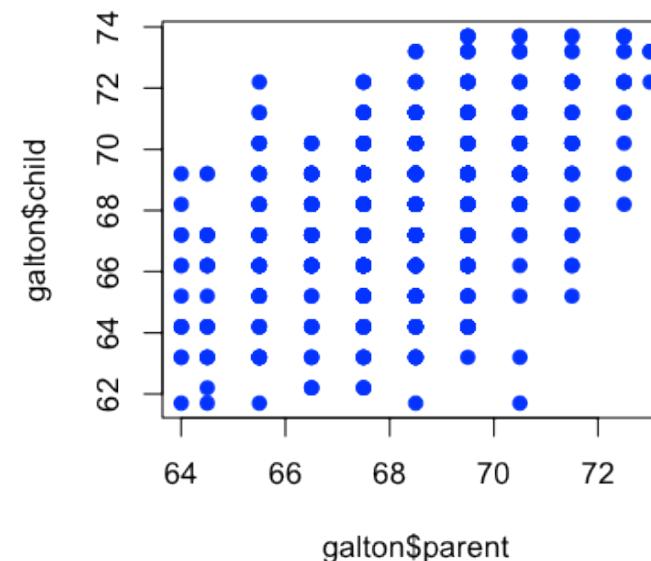
Jeffrey Leek, Assistant Professor of Biostatistics  
Johns Hopkins Bloomberg School of Public Health

# Things to pay attention to

- Confounders
- Complicated interactions
- Skewness
- Outliers
- Non-linear patterns
- Variance changes
- Units/scale issues
- Overloading regression
- Correlation and causation

# The ideal data for regression

```
library(UsingR); data(galton)
plot(galton$parent, galton$child, col="blue", pch=19)
```



3/30

# Confounders

**Confounder:** A variable that is correlated with both the outcome and the covariates

- Confounders can change the regression line
- They can even change the sign of the line
- They can sometimes be detected by careful exploration

# Example - Millenium Development Goal 1



## GOAL 1 Eradicate Extreme Poverty and Hunger

FACT SHEET

### TARGETS

1. Halve, between 1990 and 2015, the proportion of people whose income is less than \$1 a day
2. Achieve full and productive employment and decent work for all, including women and young people
3. Halve, between 1990 and 2015, the proportion of people who suffer from hunger

[http://www.un.org/millenniumgoals/pdf/MDG\\_FS\\_1\\_EN.pdf](http://www.un.org/millenniumgoals/pdf/MDG_FS_1_EN.pdf)

5/30

# WHO childhood hunger data

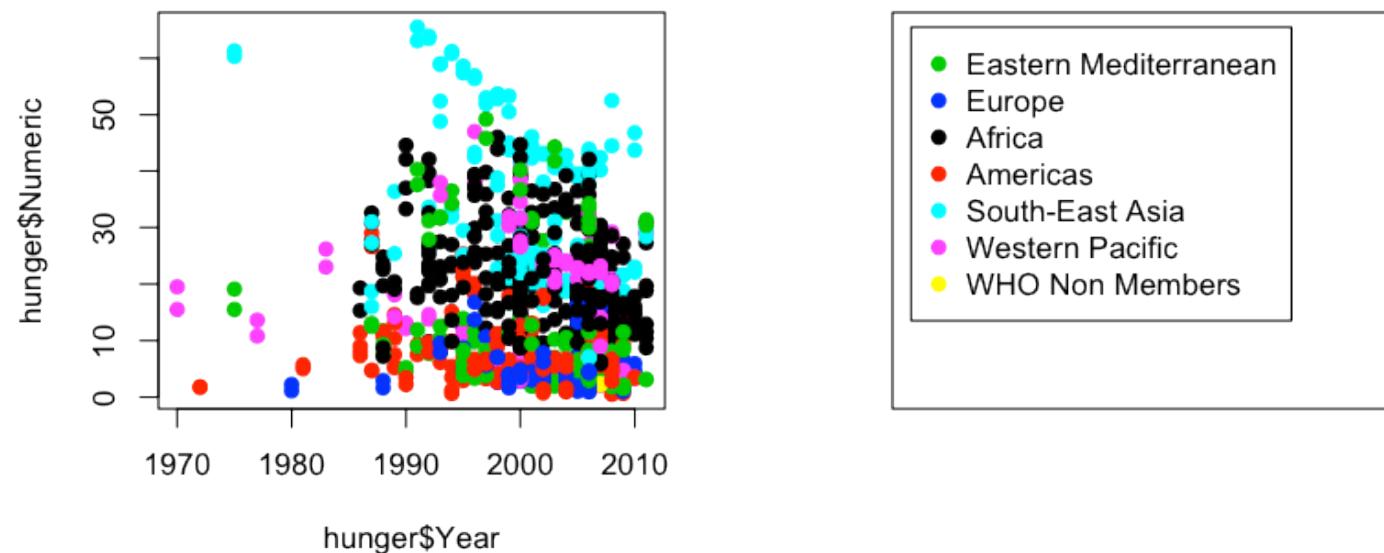
```
download.file("http://apps.who.int/gho/athena/data/GHO/WHOSIS_000008.csv?profile=text&filter=COUNTRY:*;SEX:*, ./data/hunger.csv", method="curl")
hunger <- read.csv("./data/hunger.csv")
hunger <- hunger[hunger$Sex!="Both sexes",]
head(hunger)
```

|    | Indicator                              | Data.Source | Country     | Sex    | Year     | WHO.region            |
|----|--|-------------|-------------|--------|----------|-----------------------|
|    | Display.Value                          | Numeric     | Low         | High   | Comments |                       |
| 2  | Children aged <5 years underweight (%) | NLIS_312819 | Afghanistan | Male   | 2004     | Eastern Mediterranean |
| 4  | Children aged <5 years underweight (%) | NLIS_312819 | Afghanistan | Female | 2004     | Eastern Mediterranean |
| 7  | Children aged <5 years underweight (%) | NLIS_312361 | Albania     | Male   | 2000     | Europe                |
| 8  | Children aged <5 years underweight (%) | NLIS_312361 | Albania     | Female | 2000     | Europe                |
| 9  | Children aged <5 years underweight (%) | NLIS_312879 | Albania     | Female | 2005     | Europe                |
| 10 | Children aged <5 years underweight (%) | NLIS_312879 | Albania     | Male   | 2005     | Europe                |
|    |  |             |             |        |          |                       |
| 2  | 32.7                                   | 32.7        | NA          | NA     | NA       |                       |
| 4  | 33.0                                   | 33.0        | NA          | NA     | NA       |                       |
| 7  | 19.6                                   | 19.6        | NA          | NA     | NA       |                       |
| 8  | 14.2                                   | 14.2        | NA          | NA     | NA       |                       |
| 9  | 5.8                                    | 5.8         | NA          | NA     | NA       |                       |
| 10 | 7.3                                    | 7.3         | NA          | NA     | NA       |                       |

6/30

# Hunger over time by region

```
par(mfrow=c(1,2))
plot(hunger$Year,hunger$Numeric,col=as.numeric(hunger$WHO.region),pch=19)
plot(1:10,type="n",xaxt="n",yaxt="n",xlab="",ylab="")
legend(1,10,col=unique(as.numeric(hunger$WHO.region))),legend=unique(hunger$WHO.region),pch=19)
```



7/30

# Region correlated with year

```
anova(lm(hunger$Year ~ hunger$WHO.region))
```

Analysis of Variance Table

Response: hunger\$Year

|                    | Df  | Sum Sq | Mean Sq | F value | Pr(>F)  |
|--------------------|-----|--------|---------|---------|---------|
| hunger\$WHO.region | 6   | 623    | 103.8   | 2.25    | 0.037 * |
| Residuals          | 851 | 39315  | 46.2    |         |         |

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Region correlated with hunger

```
anova(lm(hunger$Numeric ~ hunger$WHO.region))
```

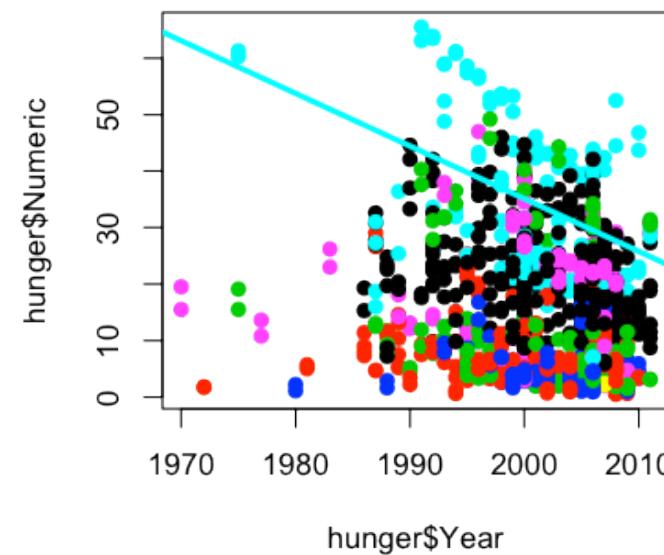
Analysis of Variance Table

Response: hunger\$Numeric

|                    | Df  | Sum Sq | Mean Sq | F value    | Pr(>F) |     |      |     |     |     |   |
|--------------------|-----|--------|---------|------------|--------|-----|------|-----|-----|-----|---|
| hunger\$WHO.region | 6   | 75042  | 12507   | 127 <2e-16 | ***    |     |      |     |     |     |   |
| Residuals          | 851 | 83853  | 99      |            |        |     |      |     |     |     |   |
| ---                |     |        |         |            |        |     |      |     |     |     |   |
| Signif. codes:     | 0   | '***'  | 0.001   | '**'       | 0.01   | '*' | 0.05 | '.' | 0.1 | ' ' | 1 |

# Including region - a complicated interaction

```
plot(hunger$Year,hunger$Numeric,pch=19,col=as.numeric(hunger$WHO.region))
lmRegion <- lm(hunger$Numeric ~ hunger$Year + hunger$WHO.region + hunger$Year*hunger$WHO.region )
abline(c(lmRegion$coeff[1] + lmRegion$coeff[6],lmRegion$coeff[2]+ lmRegion$coeff[12]),col=5,lwd=3)
```



10/30

# Income data

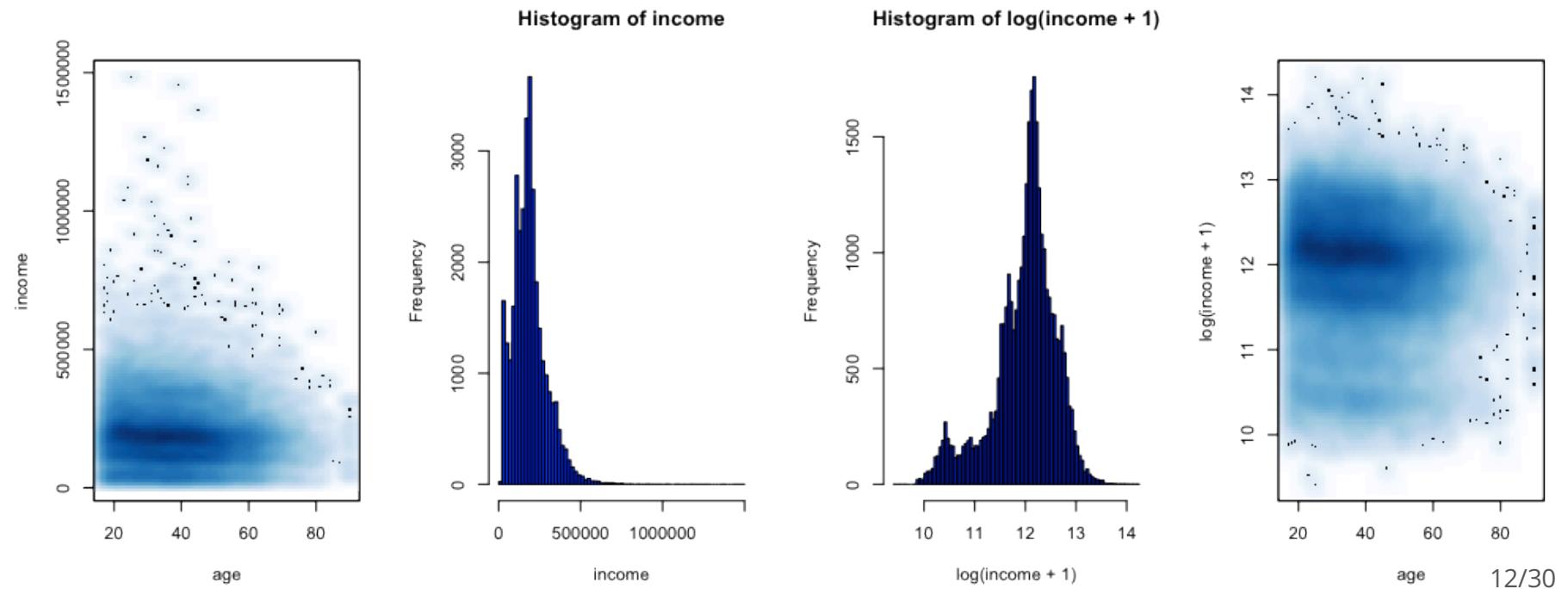
```
download.file("http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data", "./data/incomeData <- read.csv("./data/income.csv", header=FALSE)  
income <- incomeData[,3]  
age <- incomeData[,1]
```

<http://archive.ics.uci.edu/ml/datasets/Census+Income>

11/30

# Logs to address right-skew

```
par(mfrow=c(1,4))
smoothScatter(age,income)
hist(income,col="blue",breaks=100)
hist(log(income+1),col="blue",breaks=100)
smoothScatter(age,log(income+1))
```



# Outliers

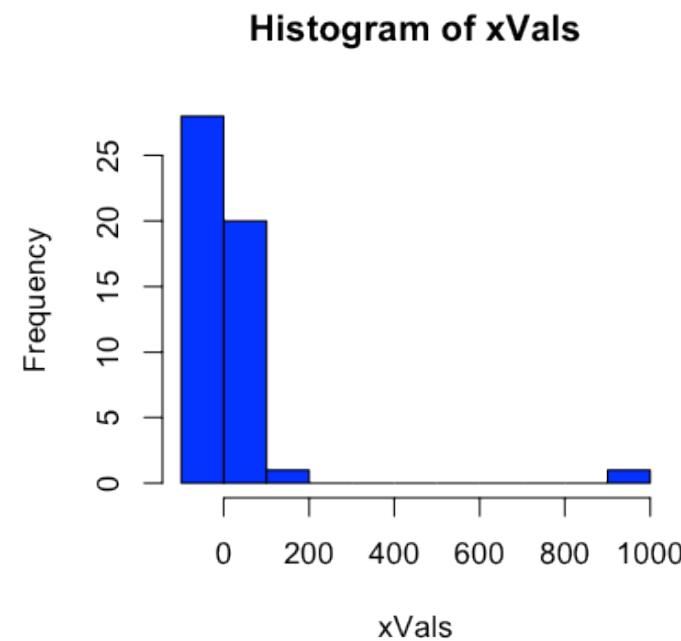
"outliers" ... are data points that do not appear to follow the pattern of the other data points.

A dataset that is 44% outliers

13/30

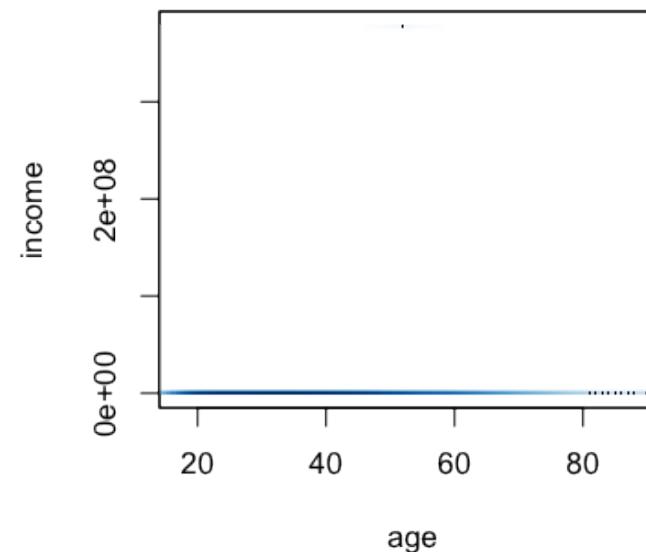
# Example - extreme points

```
set.seed(1235)
xVals <- rcauchy(50)
hist(xVals,col="blue")
```



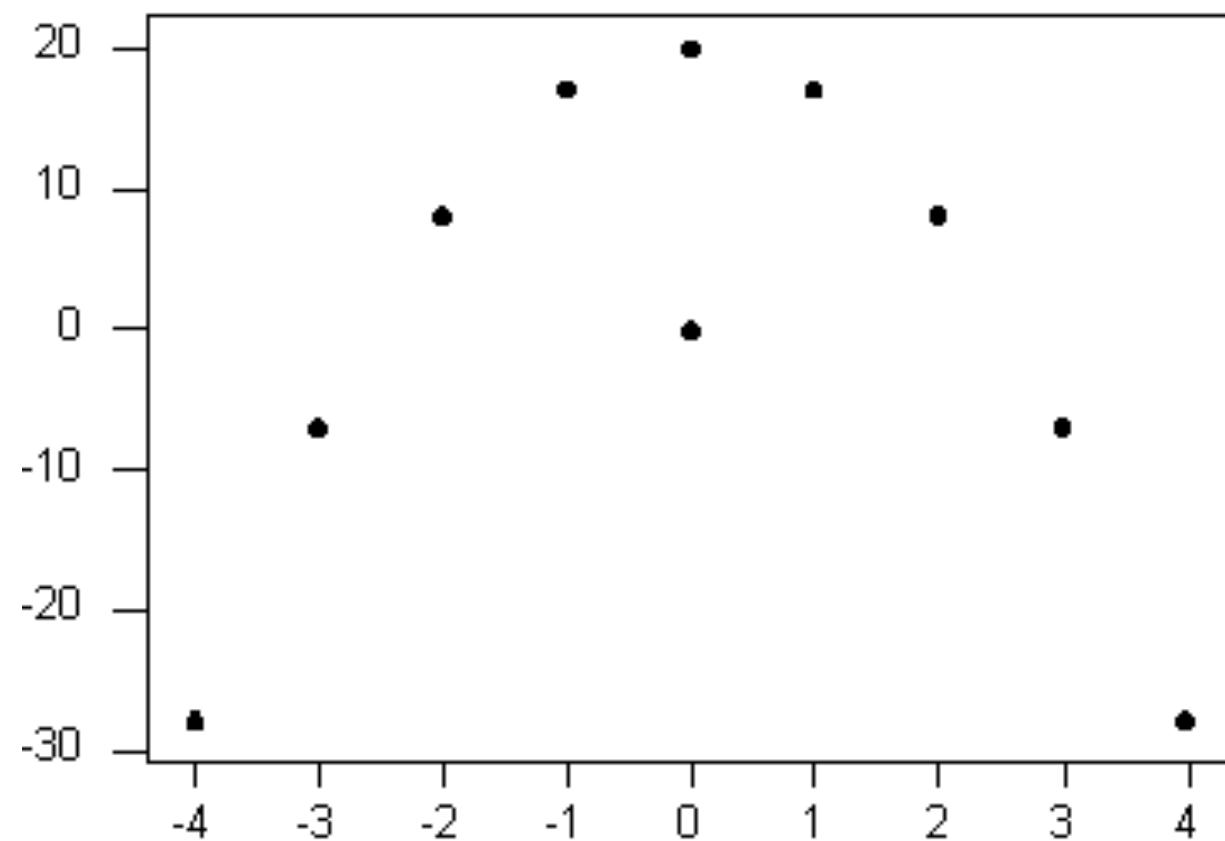
# Example - Outliers may be real

```
# Add Tim Cook, CEO of Apple 2011 income  
age <- c(age,52)  
income <- c(income,378e6)  
smoothScatter(age,income)
```



<http://www.macworld.com/article/2023491/apple-gives-tim-cook-51-percent-salary-increase.html> 15/30

# Example - Does not fit the trend



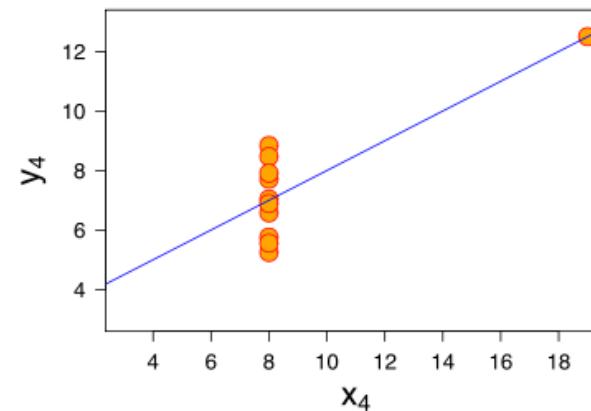
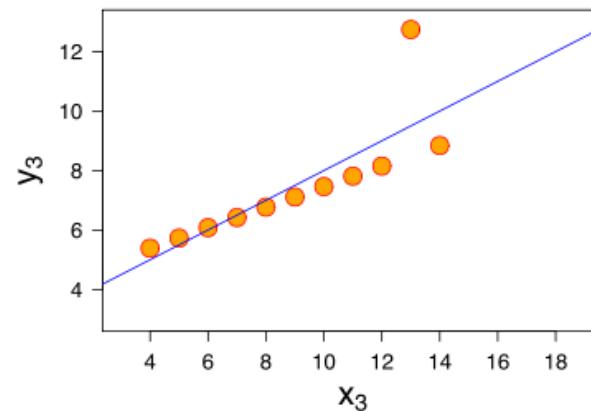
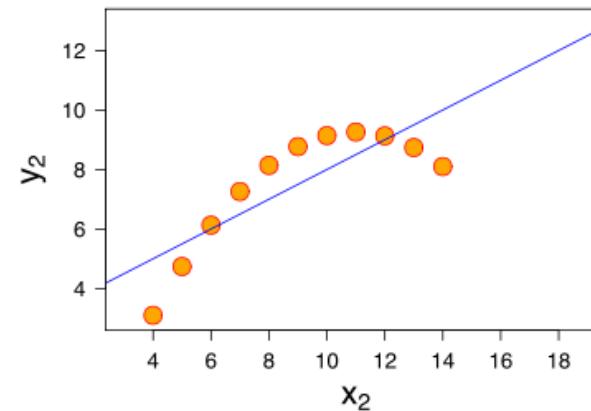
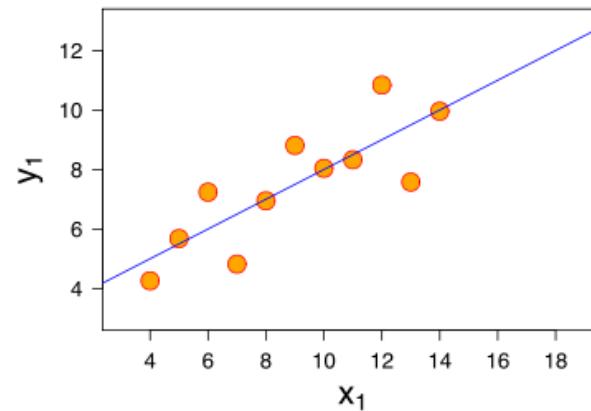
A dataset that is 44% outliers

16/30

# Outliers - what you can do

- If you know they aren't real/of interest, remove them (but changes question!)
- Alternatively
  - Sensitivity analysis - is it a big difference if you leave it in/take it out?
  - Logs - if the data are right skewed (lots of outliers)
  - Robust methods - we've been doing averages, but there are more robust approaches  
([Robust, rlm](#))

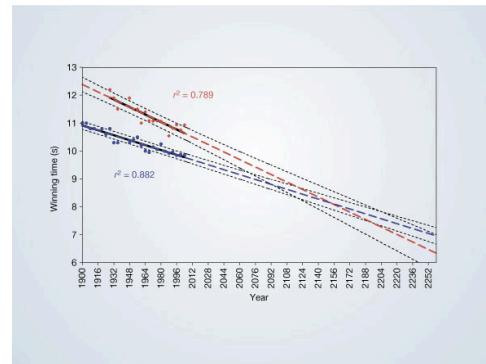
# A line isn't always the best summary



[http://en.wikipedia.org/wiki/Linear\\_regression](http://en.wikipedia.org/wiki/Linear_regression)

18/30

# You can end up saying some pretty silly stuff



[http://www.nature.com/nature/journal/v431/n7008/fig\\_tab/431525a\\_F1.html](http://www.nature.com/nature/journal/v431/n7008/fig_tab/431525a_F1.html)

"We are students aged 16–18 in a Texas high school. Our biology teacher Vidya Rajan asked us to comment on the paper by A. J. Tatem and colleagues (Nature 431, 525; 2004); we believe the projection on which it is based is riddled with flaws..."

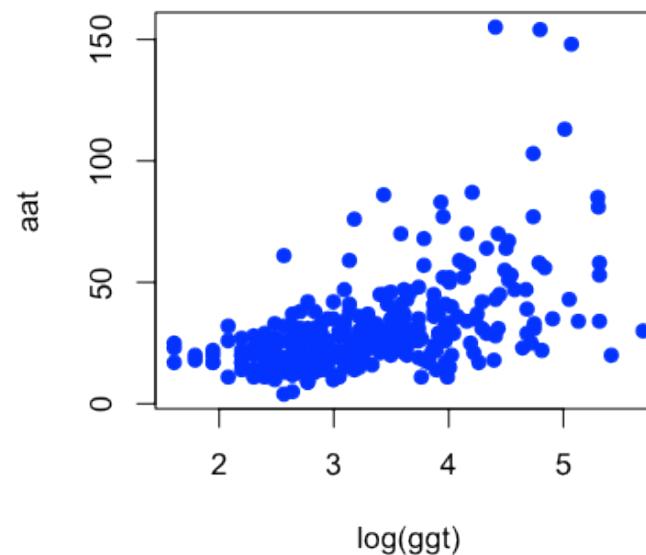
<http://www.nature.com/nature/journal/v432/n7014/full/432147c.html>

"They omit to mention, however, that (according to their analysis) a far more interesting race should occur in about 2636, when times of less than zero seconds will be recorded"

<http://www.nature.com/nature/journal/v432/n7014/full/432147b.html>

# Variance changes

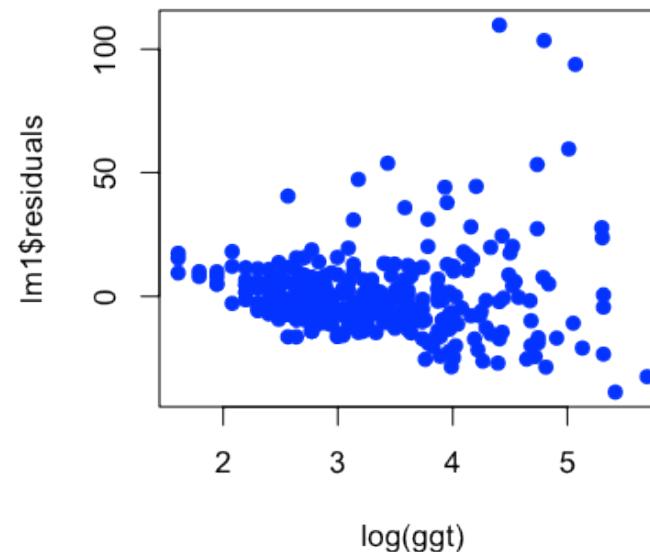
```
bupaData <- read.csv("ftp://ftp.ics.uci.edu/pub/machine-learning-databases/liver-disorders/bupa.dat"
ggt <- bupaData[,5]; aat <- bupaData[,3]
plot(log(ggt),aat,col="blue",pch=19)
```



20/30

# Plot the residuals

```
lm1 <- lm(aat ~ log(ggt))
plot(log(ggt), lm1$residuals, col="blue", pch=19)
```



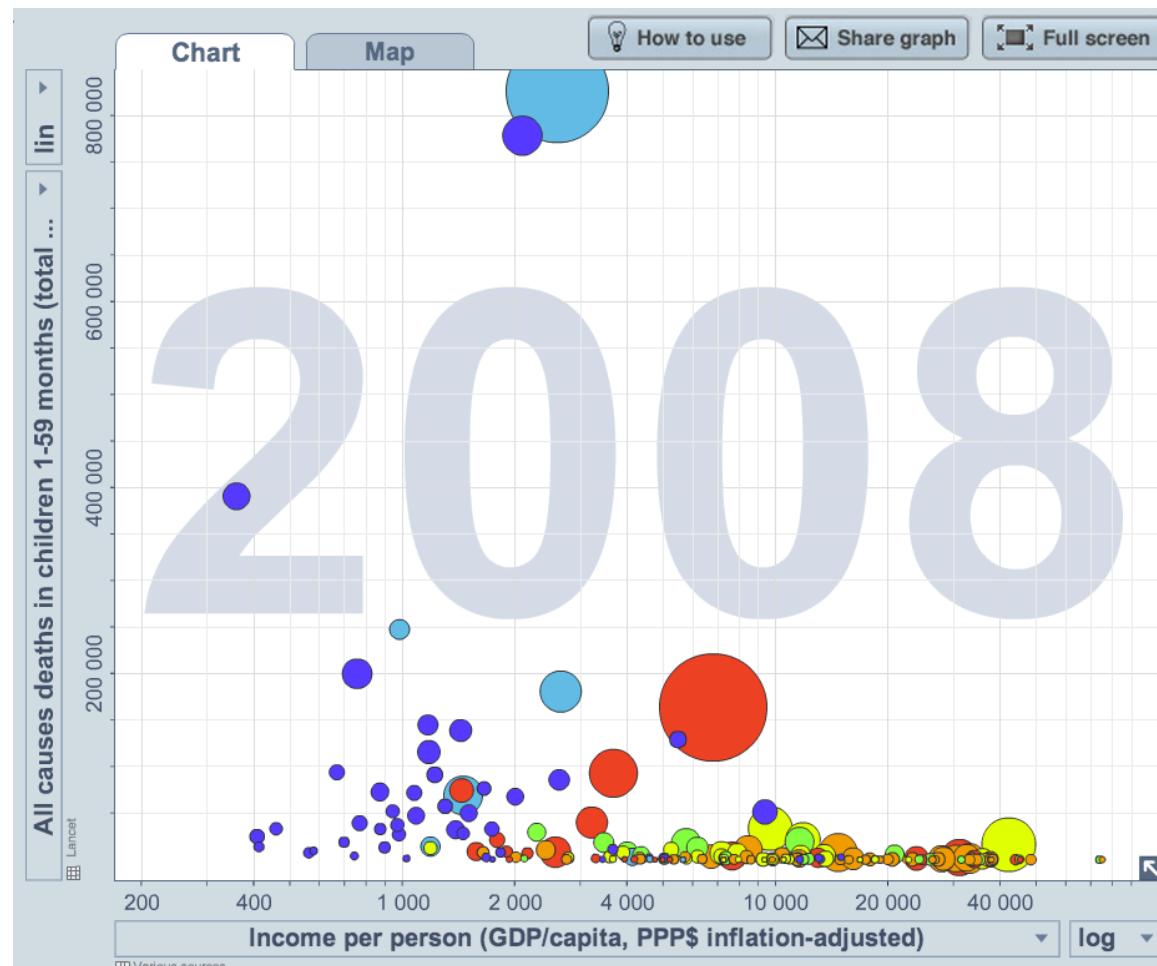
## Power (a.k.a. Box-Cox) transform

21/30

# Changing variance - what you can do

- There is a long literature on this problem (heteroskedasticity)
- A few examples
  - [Box-Cox Transform](#)
  - [Variance stabilizing transform](#)
  - [Weighted least squares](#)
  - [Huber-white standard errors](#)

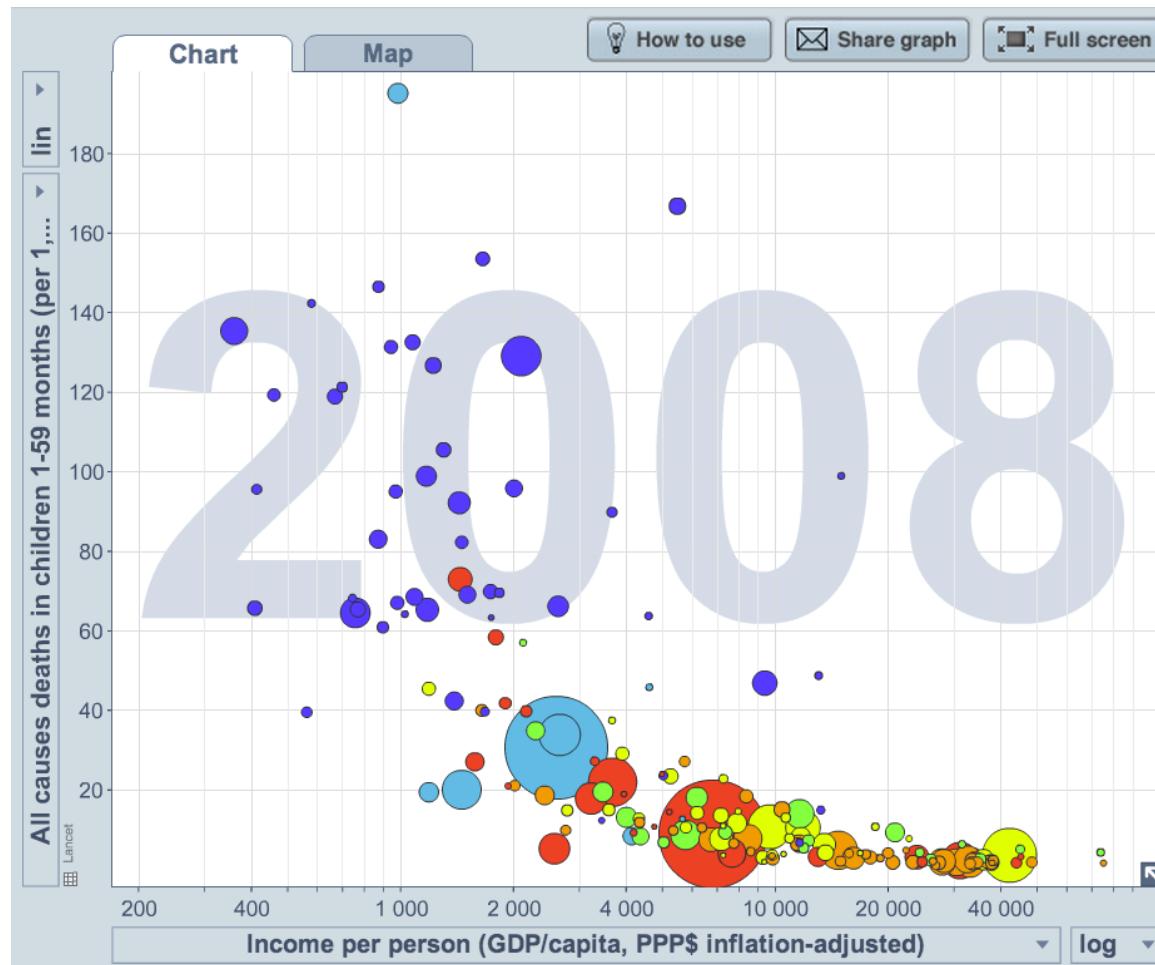
# Variation in units



## All Deaths

23/30

# Relative units



Per 1000 Deaths

24/30

# When there is variation in units

- Standardize, but keep track
  - Affects model fits
  - Affects interpretation
  - Affects inference

25/30

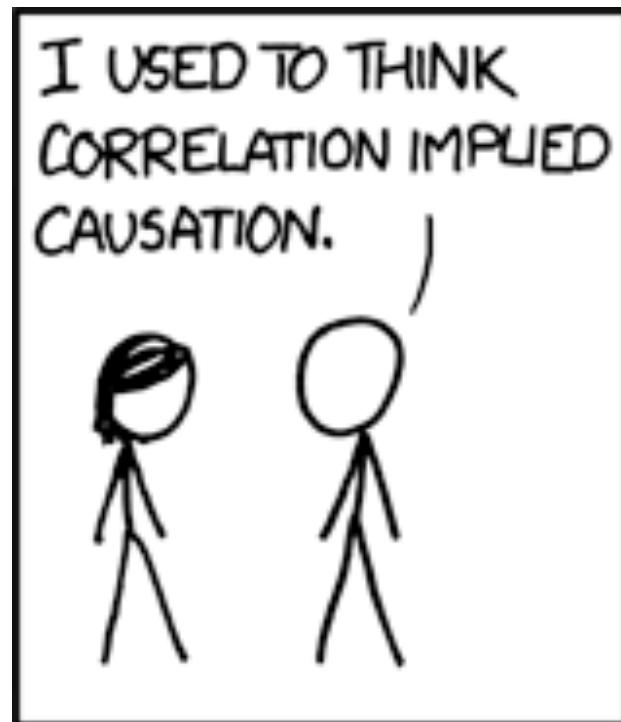
# Overloading regression

$$AMR_{it} = \alpha + \beta_1 PRIV_{it} + \beta_2 GDP_{it} + \beta_3 LIB_{it} + \beta_4 TRADE_{it} + \\ \beta_5 DEM_{it} + \beta_6 WAR_{it} + \beta_7 DEP_{it} + \beta_8 URBAN_{it} + \beta_9 EDUC_{it} + \mu_i + \varepsilon_{it}$$

<http://bit.ly/YiB5Um> <http://wmbriggs.com/blog/?p=7026>

26/30

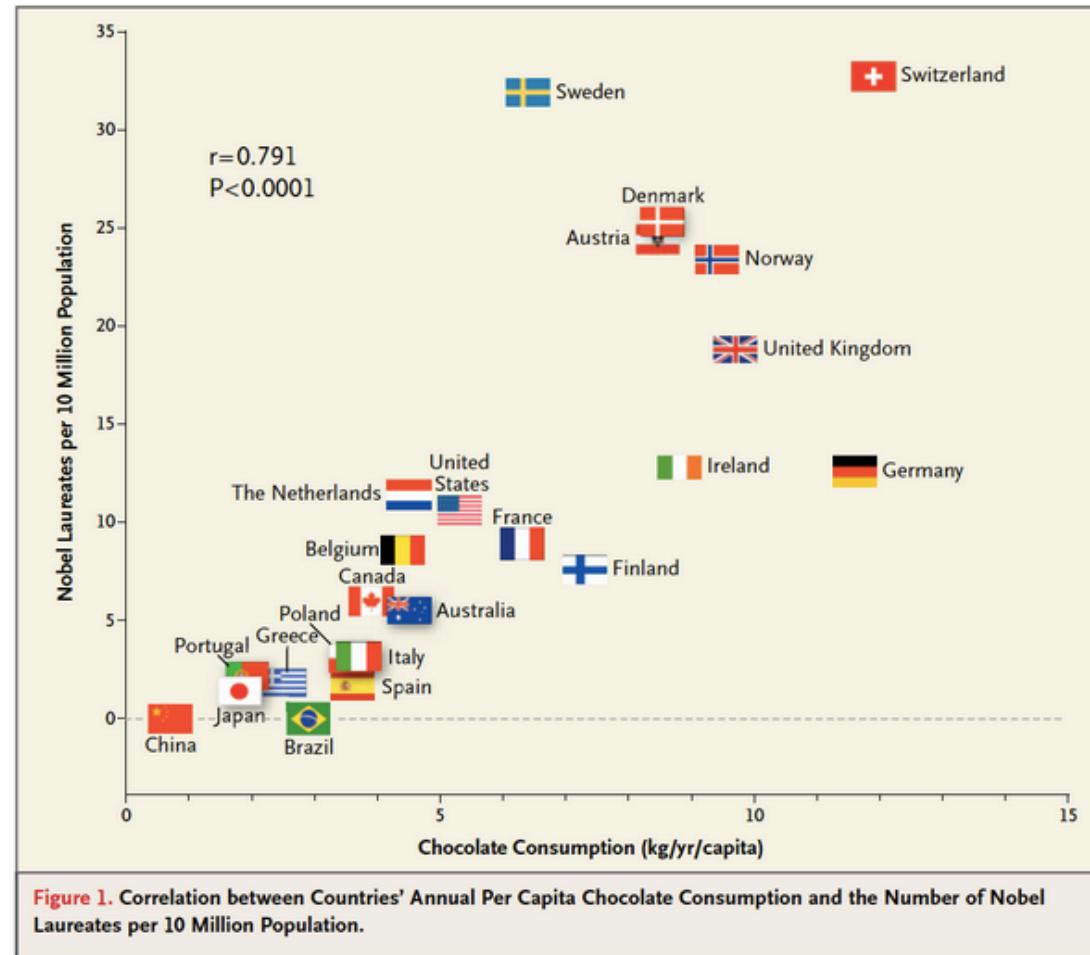
# Correlation and Causation



<http://xkcd.com/552/>

27/30

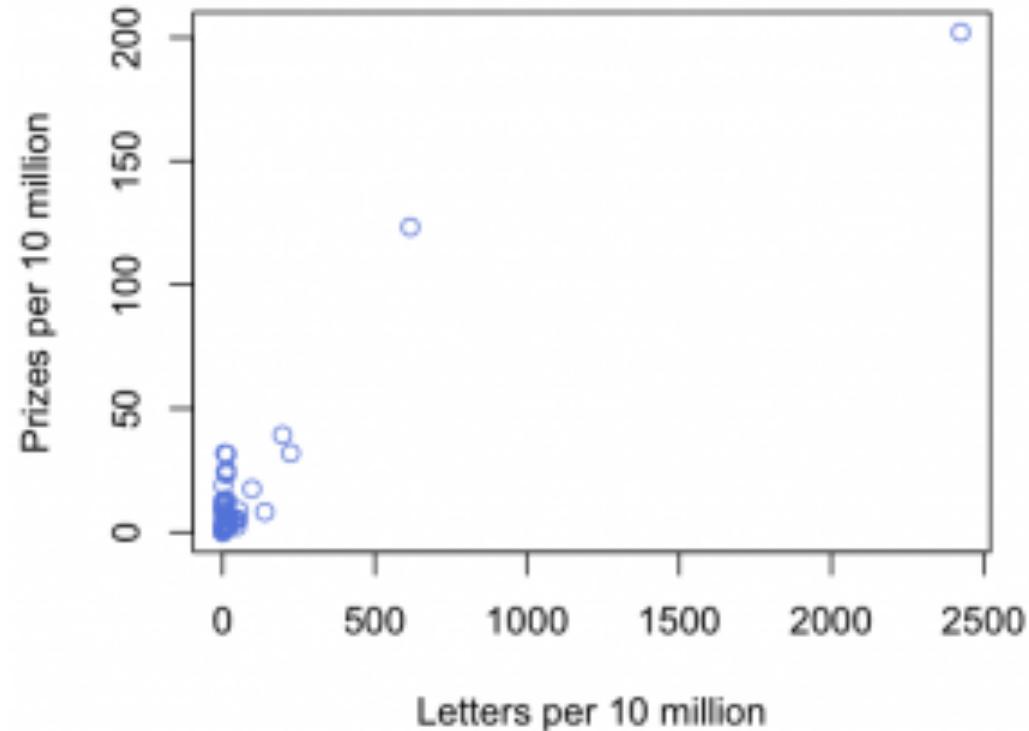
# Even when looking for associations



<http://www.nejm.org/doi/full/10.1056/NEJMoa1211064>

28/30

# Again, it can get silly



<http://www.statschat.org.nz/2012/10/12/even-better-than-chocolate/>

29/30

# Correlation vs. Causation

- Use caution when interpreting regression results
- Be critical of surprising associations
- Consider alternative explanations

30/30

# How do we represent data?

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

# How do we write about data?

- Each data point is usually represented by a capital letter.
  - $H$  for height,  $W$  for weight.
- If there are more than one data point of the same type we use subscripts.
  - $H_1, H_2, H_3$  for three different people's heights.
- Sometimes it is more compact to write  $X_1$  for height and  $X_2$  for weight.
- Then we need another subscript for the individual data point
  - $X_{11}$  for the height of the first person.
- $Y$  represents general outcomes and  $X$  general covariates.
- In this course we will try to use informative letters when possible.

# Randomness

- Variables like  $X$  and  $Y$  are called *random variables* because we expect them to be *random* in some way.
- In general, randomness is a hard thing to define
- In this class a variable may be random because
  - It represents an incompletely measured variable
  - It represents a sample drawn from a population using a random mechanism.
- Once we are talking about a specific value of a variable we have observed it isn't random anymore, we write these values with lower case letters  $x, y$ , etc.
- We write  $X = x$  or  $X = 1$  to indicate we have observed a specific value  $x$  or 1.

# Randomness and measurement

- A coin flip is commonly considered random
- But it can be modeled by deterministic equations
  - Dynamical bias in the coin toss ([Diaconis, Holmes and Montgomery SIAM Review 2007](#))
  - Modeled the tossing as a dynamical system
  - Showed that a coin is more likely to land on the side it started
  - Did experiments that demonstrated it was a 51% chance
- Some have taken it a bit farther making [predictable coin flipping machines](#) based on [physical properties](#).

# Distributions

- In statistical modeling, random variables like  $X$  are assumed to be samples from a *distribution*
- A distribution tells us the possible values of  $X$  and the probabilities for each value.
- Probability is the chance something will happen and is abbreviated  $Pr$
- The probabilities must all be between 0 and 1.
- The probabilities must add up to 1.
- An example:
  - Let's flip a coin and allow  $X$  to represent whether it is heads or tails
  - $X = 1$  if it is heads and  $X = 0$  if it is tails
  - We expect that about 50% of the time it will be heads.
  - The distribution can then be written  $\Pr(X = 1)=0.5$  and  $\Pr(X = 0)=0.5$

# Continuous versus discrete distributions

- *discrete* distributions specify probabilities for discrete values
  - Qualitative variables are discrete
  - So are variables that take on all values 0,1,2,3...
- *continuous* distributions specify probabilities for ranges of values
  - Quantitative variables are often assumed to be continuous
  - But we might only see specific values

# Parameters

- Distributions are defined by a set of fixed values called *parameters*.
- *parameters* are sometimes represented by Greek letters like  $\mu, \sigma, \tau$ .
- Distributions are written as letters with the parameters in parentheses like  $N(\mu, \sigma)$  or  $Poisson(\lambda)$ .
- $X \sim N(\mu, \sigma)$  means that  $X$  has the  $N(\mu, \sigma)$  distribution.

# The three most important parameters

- If  $X$  is a random variable, the mean of that random variable is written  $E[X]$ 
  - Stands for expected value
  - Measures the "center" of a distribution
- The variance of that random variable is written  $Var[X]$ 
  - Measures how "spread out" a distribution is
  - Measurement is in  $(\text{units of } X)^2$
- The standard deviation is written  $SD[X] = \sqrt{Var[X]}$ 
  - Also measures how "spread out" a distribution is
  - Measurement is in units of  $X$

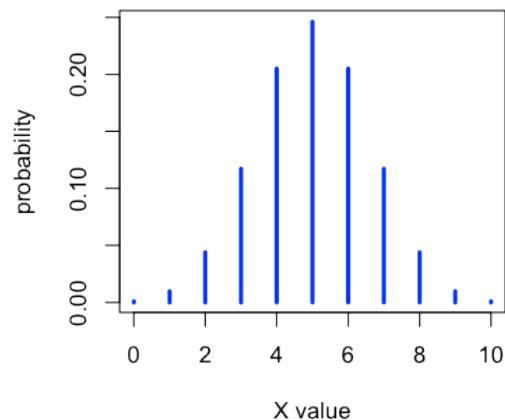
# Conditioning

- The variables  $X$  are considered to be random
- The parameters are considered to be fixed values
- Sometimes we want to talk about a case where one of the random variables is fixed
- To indicate what is fixed, we *condition* using the symbol "l"
  - $X|\mu$  means that  $X$  is a random variable with fixed parameter  $\mu$
  - $Y|X = 2$  means  $Y$  is the random variable  $Y$  when  $X$  is fixed at 2.

# Example distribution: Binomial

**Binomial distribution:**  $Bin(n, p)$

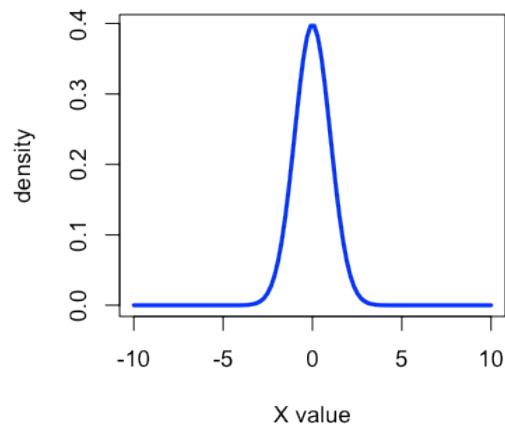
- $X \sim Bin(10, 0.5)$



# Example distribution: Normal

**Normal Distribution:**  $N(\mu, \sigma)$

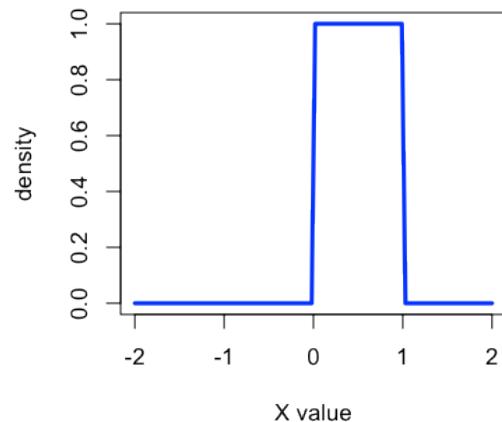
- $X \sim N(0, 1)$



# Example distribution: Uniform

Uniform distribution:  $U(\alpha, \beta)$

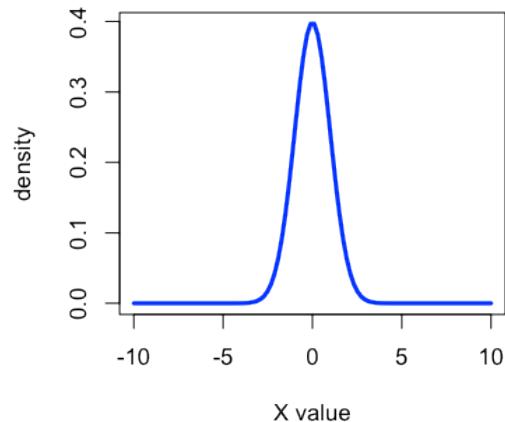
- $X \sim U(0, 1)$



# Changing parameters

**Normal Distribution:**  $N(\mu, \sigma)$

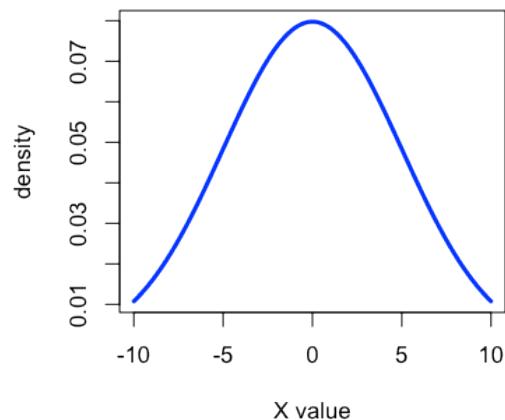
- $X \sim N(0, 1)$ ,  $E[X] = \mu = 0$ ,  $Var[X] = \sigma^2 = 1$



# Changing parameters: the variance

**Normal Distribution:**  $N(\mu, \sigma)$

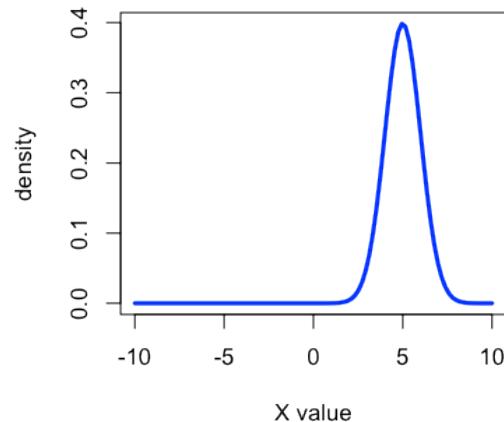
- $X \sim N(0, 5)$ ,  $E[X] = \mu = 0$ ,  $Var[X] = \sigma^2 = 25$



# Changing parameters: the mean

**Normal Distribution:**  $N(\mu, \sigma)$

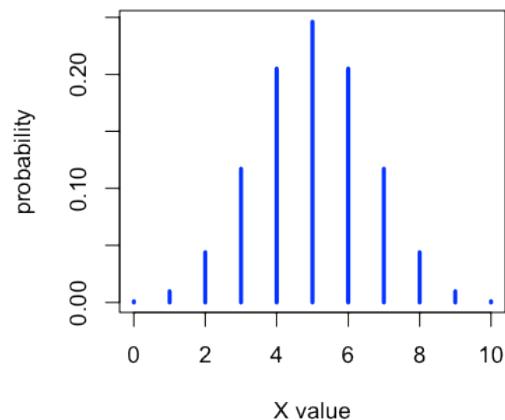
- $X \sim N(5, 1)$ ,  $E[X] = \mu = 5$ ,  $Var[X] = \sigma^2 = 1$



# Example distribution: Binomial

**Binomial distribution:**  $Bin(n, p)$

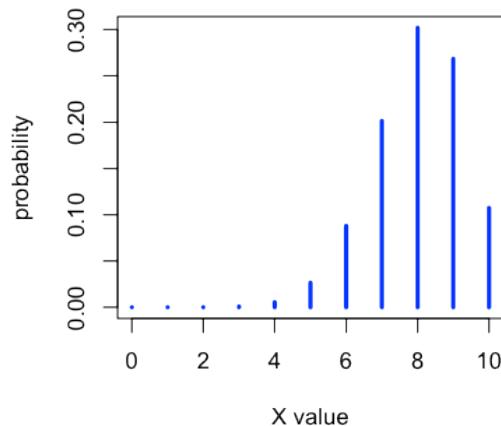
- $X \sim Bin(10, 0.5)$ ,  $E[X] = n \times p = 5$ ,  $Var[X] = n \times p \times (1 - p) = 2.5$



# Changing parameters: both mean and variance

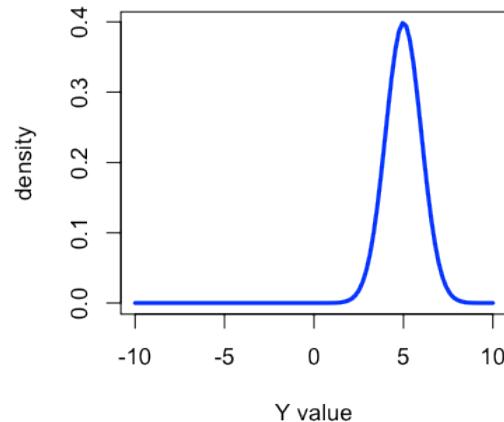
**Binomial distribution:**  $Bin(n, p)$

- $X \sim Bin(10, 0.8)$ ,  $E[X] = n \times p = 8$ ,  $Var[X] = n \times p \times (1 - p) = 1.6$



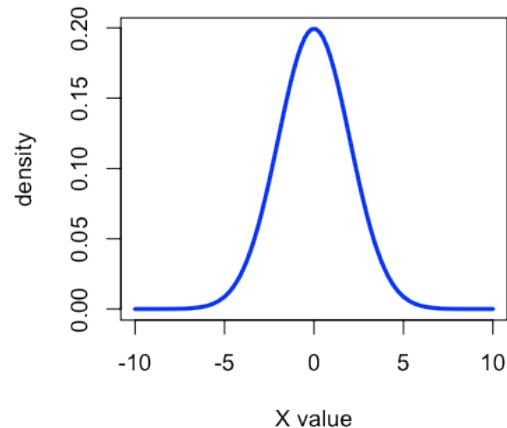
# Conditioning

- Suppose  $Y \sim N(X, 1)$  and  $X \sim N(0, 1)$ , then the distribution of  $Y|X = 5$  is



# Conditioning

- Suppose  $Y \sim N(X, 1)$  and  $X \sim N(0, 1)$ , then the distribution of  $Y$  is



[http://en.wikipedia.org/wiki/Law\\_of\\_total\\_variance](http://en.wikipedia.org/wiki/Law_of_total_variance)

[http://en.wikipedia.org/wiki/Law\\_of\\_total\\_expectation](http://en.wikipedia.org/wiki/Law_of_total_expectation)

# Learning more about a specific distribution

Poisson distribution - Wikipedia

en.wikipedia.org/wiki/Poisson\_distribution

**WIKIPEDIA**  
The Free Encyclopedia

Main page  
Contents  
Featured content  
Current events  
Random article  
Donate to Wikipedia  
Wikimedia Shop

Interaction  
Help  
About Wikipedia  
Community portal  
Recent changes  
Contact Wikipedia

Toolbox  
Print/export

Languages  
العربية  
Български  
Català  
Česky  
Deutsch  
Ελληνικά  
Español  
Euskara  
فارسی  
Français  
한국어  
Bahasa Indonesia  
Italiano  
עברית  
Lietuvių

This week we are launc  
Join us in creating a free travel g

## Poisson distribution

From Wikipedia, the free encyclopedia

In probability theory and statistics, the **Poisson distribution** (pronounced [pwasɔ̃]) is a discrete probability distribution that gives the probability of a given number of events occurring in a fixed interval of time and/or space if these events occur with a known constant rate and independently of the time since the last event.<sup>[1]</sup> The Poisson distribution can also be used for the number of events in other such as distance, area or volume.

For instance, suppose someone typically gets 4 pieces of mail per day on average. There will be, however, a certain spread: more, sometimes a little less, once in a while nothing at all.<sup>[2]</sup> Given only the average rate, for a certain period of observation (day, phonecalls per hour, etc.), and assuming that the process, or mix of processes, that produce the event flow are essentially Poisson distribution specifies how likely it is that the count will be 3, or 5, or 11, or any other number, during one period of observation. It predicts the degree of spread around a known average rate of occurrence.<sup>[2]</sup>

The section: *Derivation of the Poisson distribution* shows the relation with the formal definition.

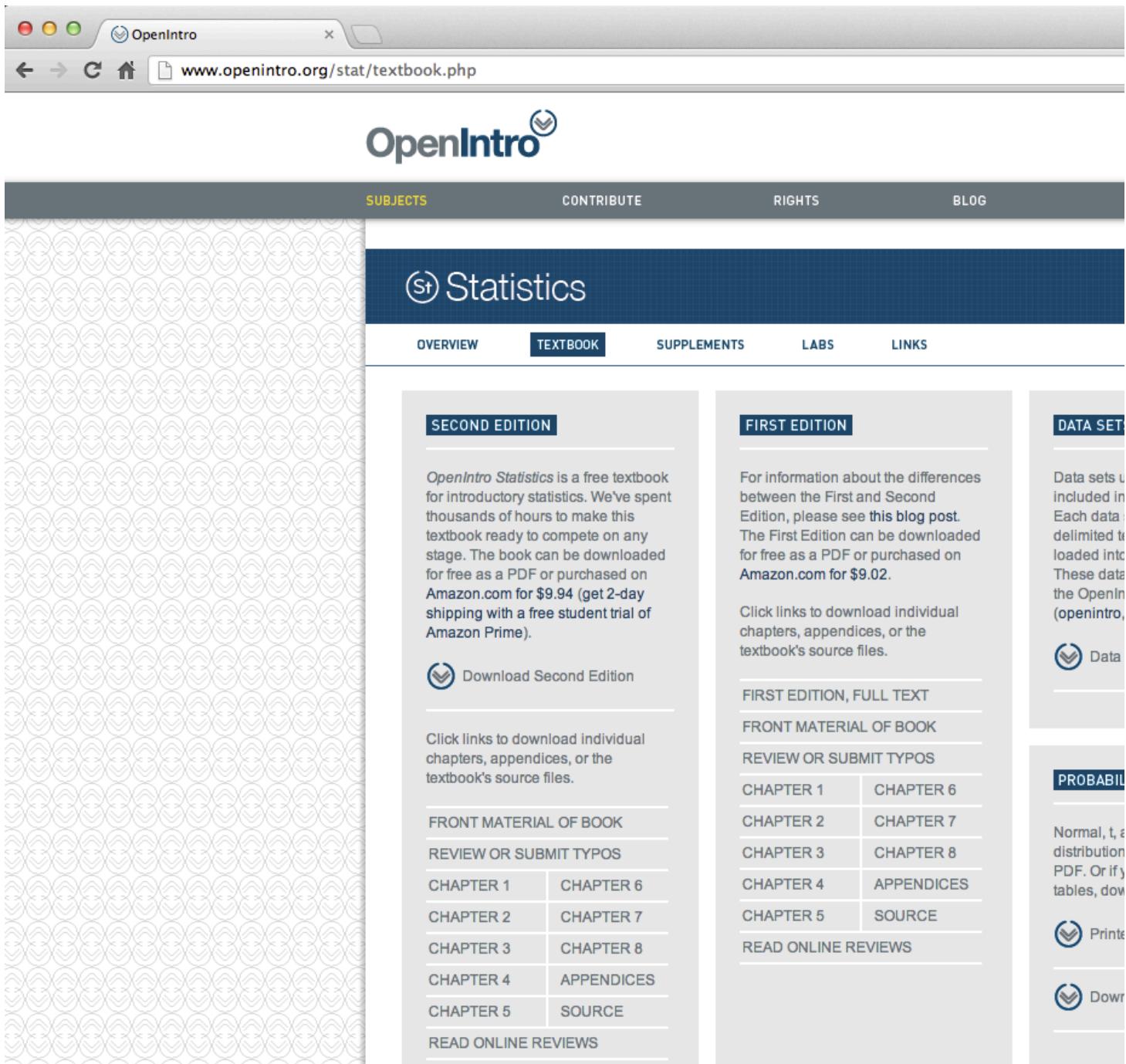
Historical background of the Poisson distribution has been described by Gullberg (1997).<sup>[3]</sup>

|   |
|---|
| <b>Contents</b> [hide] <ul style="list-style-type: none"> <li><a href="#">1 History</a></li> <li><a href="#">2 Definition</a></li> <li><a href="#">3 Properties</a> <ul style="list-style-type: none"> <li><a href="#">3.1 Mean</a></li> <li><a href="#">3.2 Median</a></li> <li><a href="#">3.3 Higher moments</a></li> <li><a href="#">3.4 Other properties</a></li> </ul> </li> <li><a href="#">4 Related distributions</a></li> <li><a href="#">5 Occurrence</a> <ul style="list-style-type: none"> <li><a href="#">5.1 Derivation of Poisson distribution — The law of rare events</a></li> <li><a href="#">5.2 Multi-dimensional Poisson process</a></li> <li><a href="#">5.3 Other applications in science</a></li> </ul> </li> <li><a href="#">6 Generating Poisson-distributed random variables</a></li> <li><a href="#">7 Parameter estimation</a> <ul style="list-style-type: none"> <li><a href="#">7.1 Maximum likelihood</a></li> </ul> </li> </ul> |
|---|

[http://en.wikipedia.org/wiki/Poisson\\_distribution](http://en.wikipedia.org/wiki/Poisson_distribution)

20/21

# Learning more about representing data



The screenshot shows a web browser window for 'OpenIntro' at the URL [www.openintro.org/stat/textbook.php](http://www.openintro.org/stat/textbook.php). The page features a dark header with the 'OpenIntro' logo and navigation links for 'SUBJECTS', 'CONTRIBUTE', 'RIGHTS', and 'BLOG'. Below the header, a large section title 'Statistics' is displayed, followed by tabs for 'OVERVIEW', 'TEXTBOOK' (which is selected), 'SUPPLEMENTS', 'LABS', and 'LINKS'. On the left, there's a decorative background pattern of repeating shapes. The main content area is divided into two main sections: 'SECOND EDITION' and 'FIRST EDITION'. The 'SECOND EDITION' section contains text about the book and a 'Download Second Edition' button. The 'FIRST EDITION' section contains text about differences between editions and download links. To the right, there's a sidebar titled 'DATA SET' with a link to 'Data sets used in the book'. At the bottom of the page, there are links for 'PROBABIL' (partially visible) and 'Normal, t, and F distributions'.

<http://www.openintro.org/stat/textbook.php>

# Representing data in R

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

# Important data types in R

## Classes

- Character, Numeric, Integer, Logical
- 

## Objects

- Vectors, Matrices, Data frames, Lists, Factors, Missing values
- 

## Operations

- Subsetting, Logical subsetting
- 

*For more information:*

- [Data Types](#)

# Character

```
firstName = "jeff"  
class(firstName)
```

```
## [1] "character"
```

```
firstName
```

```
## [1] "jeff"
```

# Numeric

```
heightCM = 188.2  
class(heightCM)
```

```
## [1] "numeric"
```

```
heightCM
```

```
## [1] 188.2
```

# Integer

```
numberSons = 1L  
class(numberSons)
```

```
## [1] "integer"
```

```
numberSons
```

```
## [1] 1
```

# Logical

```
teachingCoursera = TRUE  
class(teachingCoursera)
```

```
## [1] "logical"
```

```
teachingCoursera
```

```
## [1] TRUE
```

# Vectors

A set of values with the same class

```
heights = c(188.2, 181.3, 193.4)
```

```
heights
```

```
## [1] 188.2 181.3 193.4
```

```
firstNames = c("jeff", "roger", "andrew", "brian")
```

```
firstNames
```

```
## [1] "jeff"    "roger"   "andrew"  "brian"
```

# Lists

A vector of values of possibly different classes

```
vector1 = c(188.2, 181.3, 193.4)
vector2 = c("jeff", "roger", "andrew", "brian")
myList = list(heights = vector1, firstNames = vector2)
myList
```

```
## $heights
## [1] 188.2 181.3 193.4
##
## $firstNames
## [1] "jeff"    "roger"   "andrew"  "brian"
```

# Matrices

Vectors with multiple dimensions

```
myMatrix = matrix(c(1, 2, 3, 4), byrow = T, nrow = 2)
myMatrix
```

```
##      [,1] [,2]
## [1,]     1     2
## [2,]     3     4
```

# Data frames

Multiple vectors of possibly different classes, of the same length

```
vector1 = c(188.2, 181.3, 193.4)
vector2 = c("jeff", "roger", "andrew", "brian")
myDataFrame = data.frame(heights = vector1, firstNames = vector2)
```

```
## Error: arguments imply differing number of rows: 3, 4
```

```
myDataFrame
```

```
## Error: object 'myDataFrame' not found
```

# Data frames

```
vector1 = c(188.2, 181.3, 193.4, 192.3)
vector2 = c("jeff", "roger", "andrew", "brian")
myDataFrame = data.frame(heights = vector1, firstNames = vector2)
myDataFrame

##   heights firstNames
## 1    188.2      jeff
## 2    181.3     roger
## 3    193.4    andrew
## 4    192.3    brian
```

# Factors

Qualitative variables that can be included in models

```
smoker = c("yes", "no", "yes", "yes")
smokerFactor = as.factor(smoker)
smokerFactor
```

```
## [1] yes no yes yes
## Levels: no yes
```

# Missing values

In R they are usually coded NA

```
vector1 = c(188.2, 181.3, 193.4, NA)  
vector1
```

```
## [1] 188.2 181.3 193.4     NA
```

```
is.na(vector1)
```

```
## [1] FALSE FALSE FALSE  TRUE
```

# Subsetting

```
vector1 = c(188.2, 181.3, 193.4, 192.3)
vector2 = c("jeff", "roger", "andrew", "brian")
myDataFrame = data.frame(heights = vector1, firstNames = vector2)

vector1[1]

## [1] 188.2

vector1[c(1, 2, 4)]

## [1] 188.2 181.3 192.3
```

# Subsetting

```
myDataFrame[1, 1:2]  
  
##   heights firstNames  
## 1    188.2      jeff  
  
myDataFrame$firstNames  
  
## [1] jeff  roger andrew brian  
## Levels: andrew brian jeff roger
```

# Logical subsetting

```
myDataFrame[myDataFrame$firstNames == "jeff", ]
```

```
##   heights firstNames
## 1    188.2      jeff
```

```
myDataFrame[heights < 190, ]
```

```
##   heights firstNames
## 1    188.2      jeff
## 2    181.3     roger
## 4    192.3     brian
```

# Variable naming conventions

Variable names should be short, but descriptive. Here are some common styles

## Camel caps

```
myHeightCM = 188
```

## Underscore

```
my_height_cm = 188
```

## Dot separated

```
my.height.cm = 188
```

# Style guides

- <http://4dpiecharts.com/r-code-style-guide/>
- <http://google-styleguide.googlecode.com/svn/trunk/google-r-style.html>
- [http://wiki.fhcrc.org/bioc/Coding\\_Standards](http://wiki.fhcrc.org/bioc/Coding_Standards)

# Simulation basics

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

# Important simulation functions

## Distributions

- `rbeta`, `rbinom`, `rcauchy`, `rchisq`, `rexp`, `rf`, `rgamma`, `rgeom`, `rhyper`, `rlogis`, `rlnorm`, `rnbinom`, `rnorm`, `rpois`, `rt`, `runif`, `rweibull`

## Densities

- `dbeta`, `dbinom`, `dcauchy`, `dchisq`, `dexp`, `df`, `dgamma`, `dgeom`, `dhyper`, `dlogis`, `dlnorm`, `dnbinom`, `dnorm`, `dpois`, `dt`, `dunif`, `dweibull`

## Sampling

- `sample`(`replace=TRUE`), `sample`(`replace=FALSE`)

# rfoo functions generate data

## Normal

```
args(rnorm)
```

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

```
heights = rnorm(10,mean=188,sd=3)
heights
```

```
[1] 186.0 191.2 187.6 187.9 186.6 187.2 187.2 189.5 190.8 186.4
```

# rfoo functions generate data

## Binomial

```
args(rbinom)
```

```
function (n, size, prob)
NULL
```

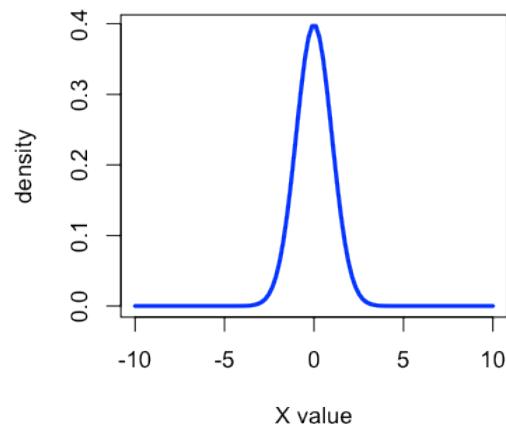
```
coinFlips = rbinom(10, size=10, prob=0.5)
coinFlips
```

```
[1] 3 4 6 5 7 6 5 8 5 6
```

# Example distribution: Normal

**Normal Distribution:**  $N(\mu, \sigma)$

- $X \sim N(0, 1)$



# dfoo functions calculate the density

## Normal

```
args(dnorm)
```

```
function (x, mean = 0, sd = 1, log = FALSE)  
NULL
```

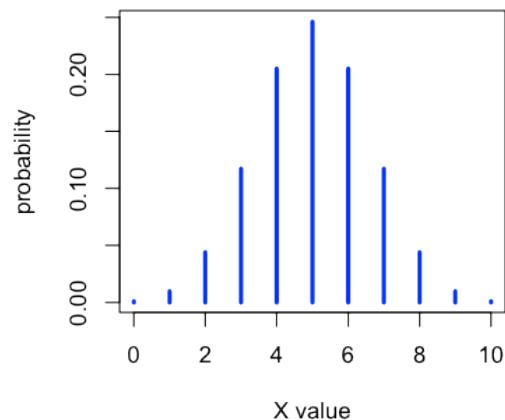
```
x = seq(from=-5,to=5,length=10)  
normalDensity = dnorm(x,mean=0,sd=1)  
round(normalDensity,2)
```

```
[1] 0.00 0.00 0.01 0.10 0.34 0.34 0.10 0.01 0.00 0.00
```

# Example distribution: Binomial

Binomial distribution:  $Bin(n, p)$

- $X \sim Bin(10, 0.5)$



# dfoo functions calculate the density

## Binomial

```
args(dbinom)
```

```
function (x, size, prob, log = FALSE)  
NULL
```

```
x = seq(0,10,by=1)  
binomialDensity = dbinom(x,size=10,prob=0.5)  
round(binomialDensity,2)
```

```
[1] 0.00 0.01 0.04 0.12 0.21 0.25 0.21 0.12 0.04 0.01 0.00
```

# Sample draws a random sample

```
args(sample)

function (x, size, replace = FALSE, prob = NULL)
NULL

heights = rnorm(10,mean=188,sd=3)
heights

[1] 187.2 185.4 187.9 187.3 184.8 190.3 185.0 188.2 190.0 188.1

sample(heights,size=10,replace=TRUE)

[1] 188.2 188.2 184.8 185.0 187.2 188.2 187.9 185.4 184.8 185.4
```

# Sample draws a random sample

heights

```
[1] 187.2 185.4 187.9 187.3 184.8 190.3 185.0 188.2 190.0 188.1
```

```
sample(heights, size=10, replace=FALSE)
```

```
[1] 185.0 188.2 188.1 184.8 190.3 187.9 187.2 185.4 190.0 187.3
```

# Sample can draw according to a set of probabilities

heights

```
[1] 187.2 185.4 187.9 187.3 184.8 190.3 185.0 188.2 190.0 188.1
```

```
probs = c(0.4,0.3,0.2,0.1,0,0,0,0,0,0)  
sum(probs)
```

```
[1] 1
```

```
sample(heights,size=10,replace=TRUE,prob=probs)
```

```
[1] 187.2 185.4 187.9 187.3 185.4 187.2 185.4 187.2 187.2 185.4
```

# Setting a seed

Setting a seed ensures reproducible results from random processes in R

```
set.seed(12345)
rnorm(5,mean=0,sd=1)

[1] 0.5855 0.7095 -0.1093 -0.4535 0.6059
```

```
set.seed(12345)
rnorm(5,mean=0,sd=1)

[1] 0.5855 0.7095 -0.1093 -0.4535 0.6059
```

# For more information

**More on distributions in R**

<http://cran.r-project.org/web/views/Distributions.html>

**Computing for Data Analysis**

[Simulation in R](#)

# Simulation for model checking

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

# Basic ideas

- Way back in the first week we talked about simulating data from distributions in R using the *rfoo* functions.
- In general simulations are way more flexible/useful
  - For bootstrapping as we saw in week 7
  - For evaluating models
  - For testing different hypotheses
  - For sensitivity analysis
- At minimum it is useful to simulate
  - A best case scenario
  - A few examples where you know your approach won't work
  - The importance of simulating the extremes

# Simulating data from a model

Suppose that you have a regression model

$$Y_i = b_0 + b_1 X_i + e_i$$

Here is an example of generating data from this model where  $X_i$  and  $e_i$  are normal:

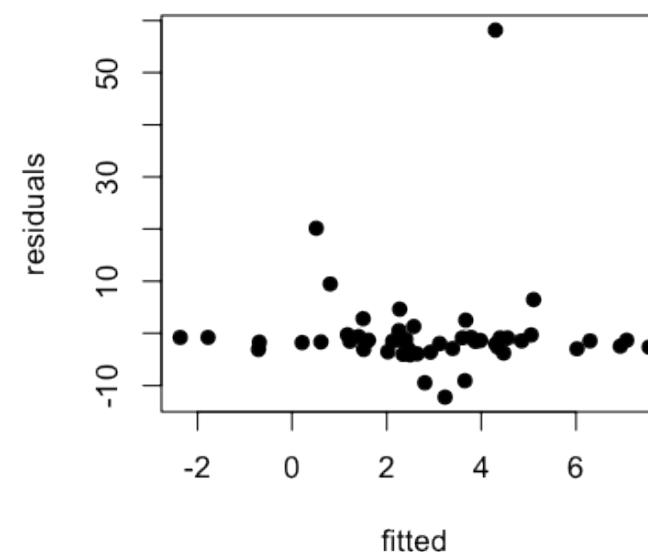
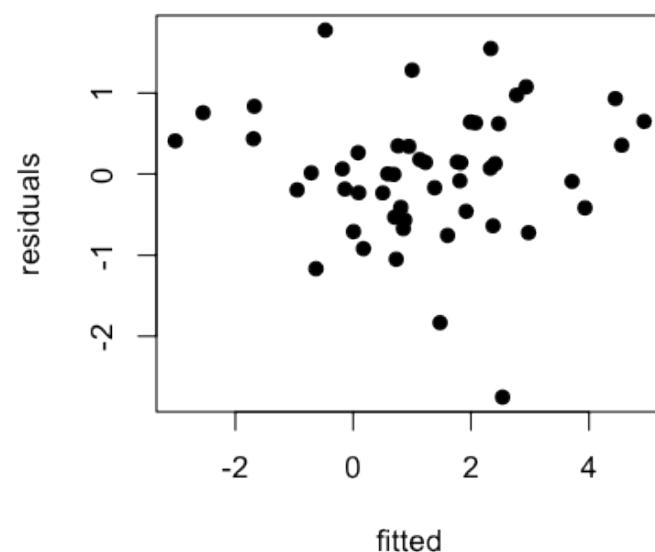
```
set.seed(44333)
x <- rnorm(50)
e <- rnorm(50)
b0 <- 1; b1 <- 2
y <- b0 + b1*x + e
```

# Violating assumptions

```
set.seed(44333)
x <- rnorm(50)
e <- rnorm(50); e2 <- rcauchy(50)
b0 <- 1; b1 <- 2
y <- b0 + b1*x + e; y2 <- b0 + b1*x + e2
```

# Violating assumptions

```
par(mfrow=c(1,2))
plot(lm(y ~ x)$fitted, lm(y~x)$residuals, pch=19, xlab="fitted", ylab="residuals")
plot(lm(y2 ~ x)$fitted, lm(y2~x)$residuals, pch=19, xlab="fitted", ylab="residuals")
```



# Repeated simulations

```
set.seed(44333)
betaNorm <- betaCauch <- rep(NA,1000)
for(i in 1:1000){
  x <- rnorm(50); e <- rnorm(50); e2 <- rcauchy(50); b0 <- 1; b1 <- 2
  y <- b0 + b1*x + e; y2 <- b0 + b1*x + e2
  betaNorm[i] <- lm(y ~ x)$coeff[2]; betaCauch[i] <- lm(y2 ~ x)$coeff[2]
}
quantile(betaNorm)
```

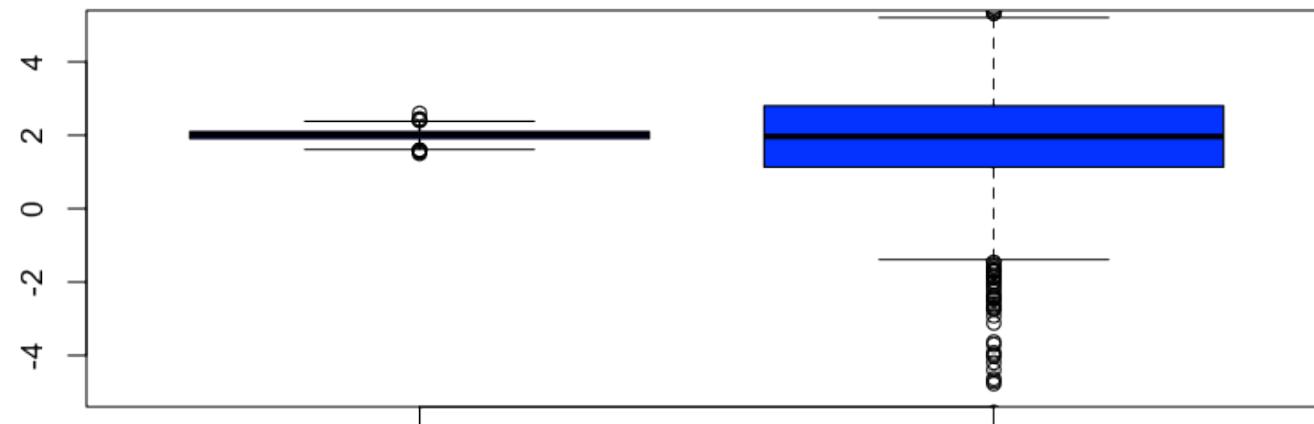
0% 25% 50% 75% 100%  
1.500 1.906 2.013 2.100 2.596

```
quantile(betaCauch)
```

0% 25% 50% 75% 100%  
-278.352 1.130 1.965 2.804 272.391

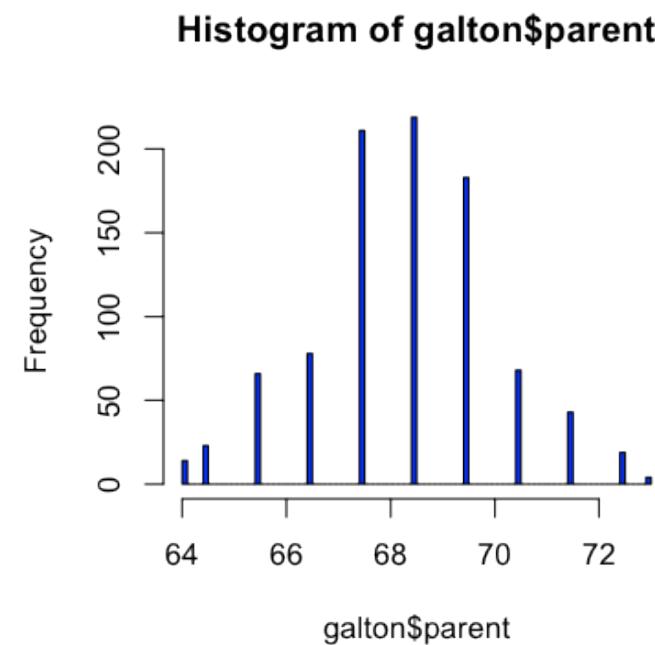
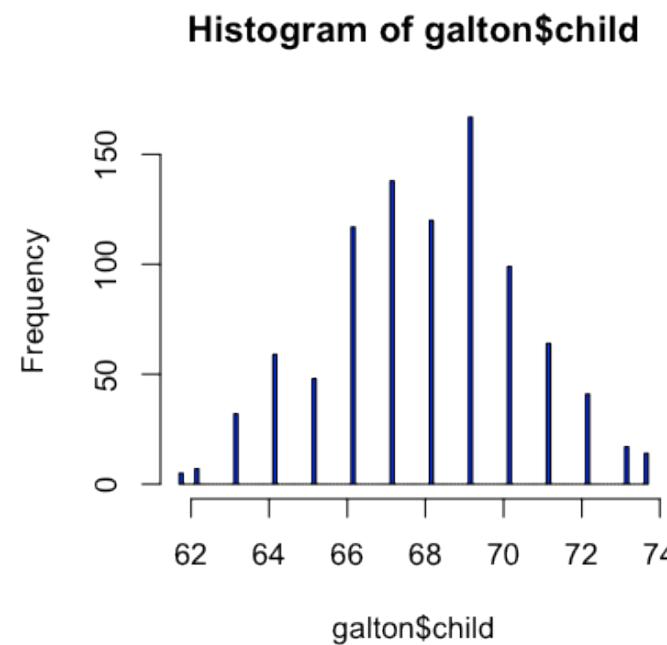
# Monte Carlo Error

```
boxplot(betaNorm,betaCauch,col="blue",ylim=c(-5,5))
```



# Simulation based on a data set

```
library(UsingR); data(galton); nobs <- dim(galton)[1]
par(mfrow=c(1,2))
hist(galton$child,col="blue",breaks=100)
hist(galton$parent,col="blue",breaks=100)
```

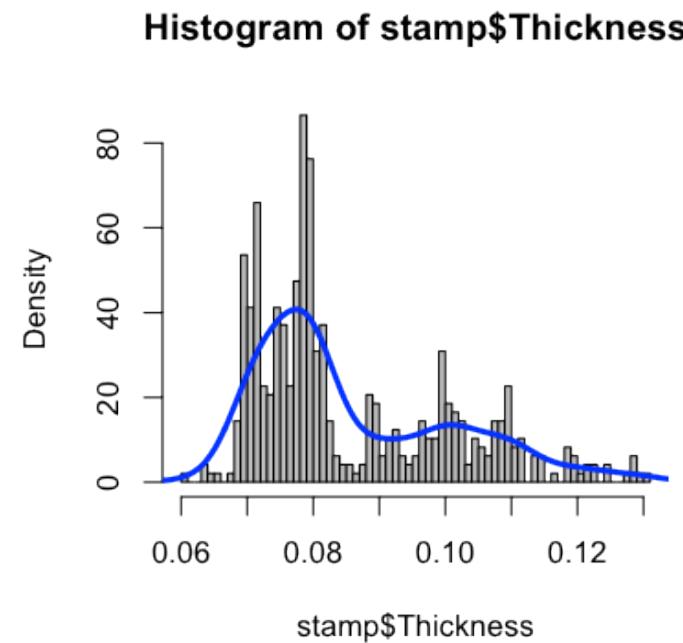


# Calculating means, variances

```
lm1 <- lm(galton$child ~ galton$parent)
parent0 <- rnorm(nobs, sd=sd(galton$parent), mean=mean(galton$parent))
child0 <- lm1$coeff[1] + lm1$coeff[2]*parent0 + rnorm(nobs, sd=summary(lm1)$sigma)
par(mfrow=c(1,2))
plot(galton$parent, galton$child, pch=19)
plot(parent0, child0, pch=19, col="blue")
```

# Simulating more complicated scenarios

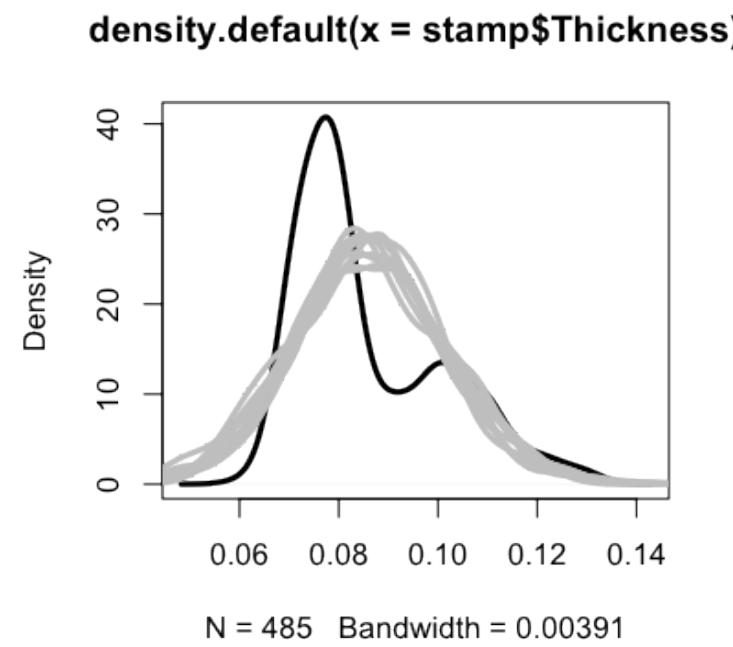
```
library(bootstrap); data(stamp); nobs <- dim(stamp)[1]
hist(stamp$Thickness,col="grey",breaks=100,freq=F)
dens <- density(stamp$Thickness)
lines(dens,col="blue",lwd=3)
```



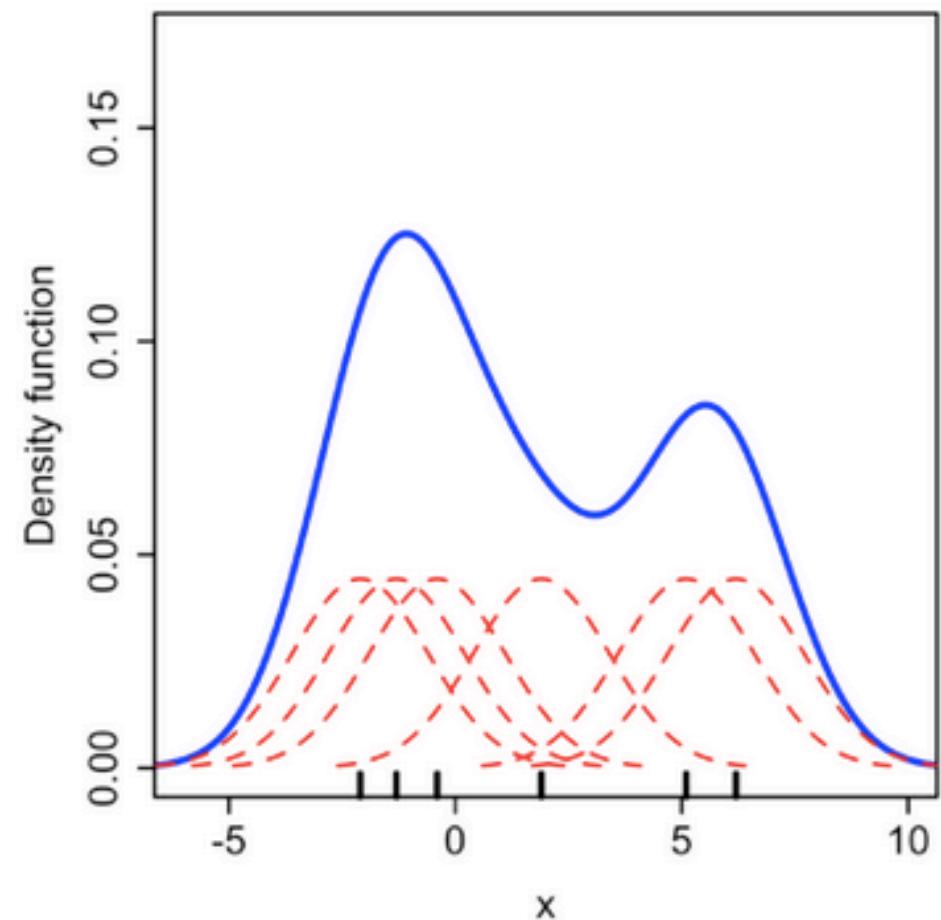
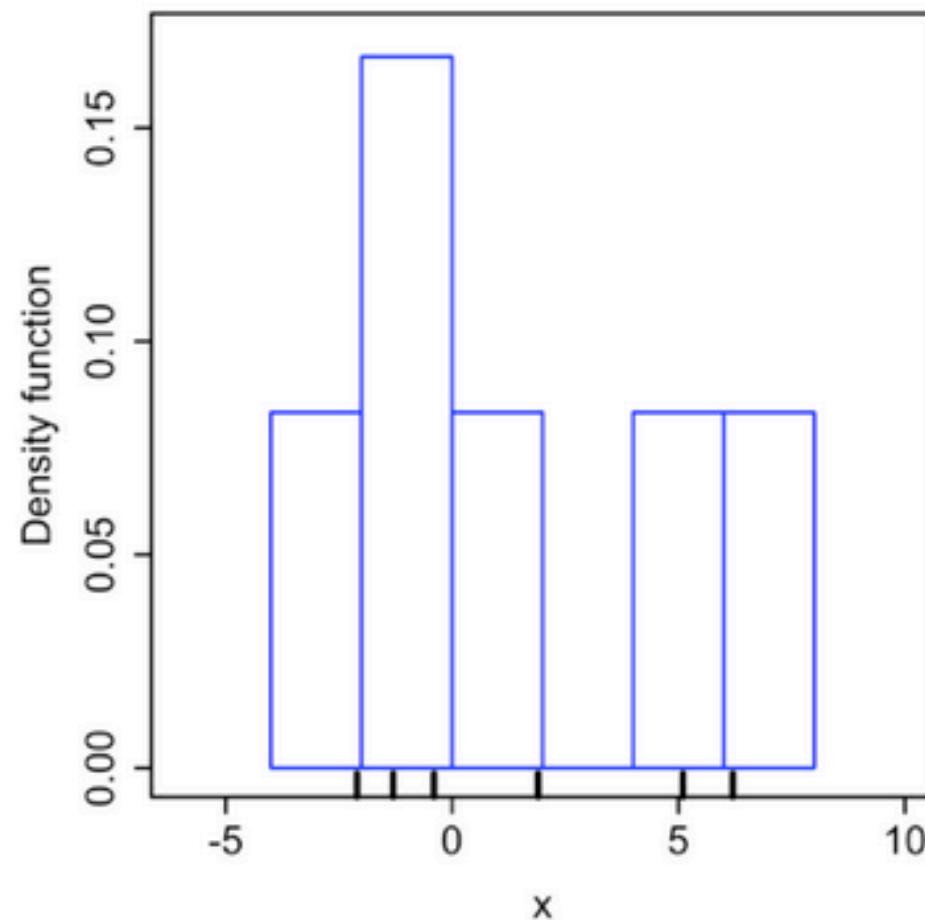
10/15

# A simulation that is too simple

```
plot(density(stamp$Thickness), col="black", lwd=3)
for(i in 1:10){
  newThick <- rnorm(nobs, mean=mean(stamp$Thickness), sd=sd(stamp$Thickness))
  lines(density(newThick), col="grey", lwd=3)
}
```



# How density estimation works

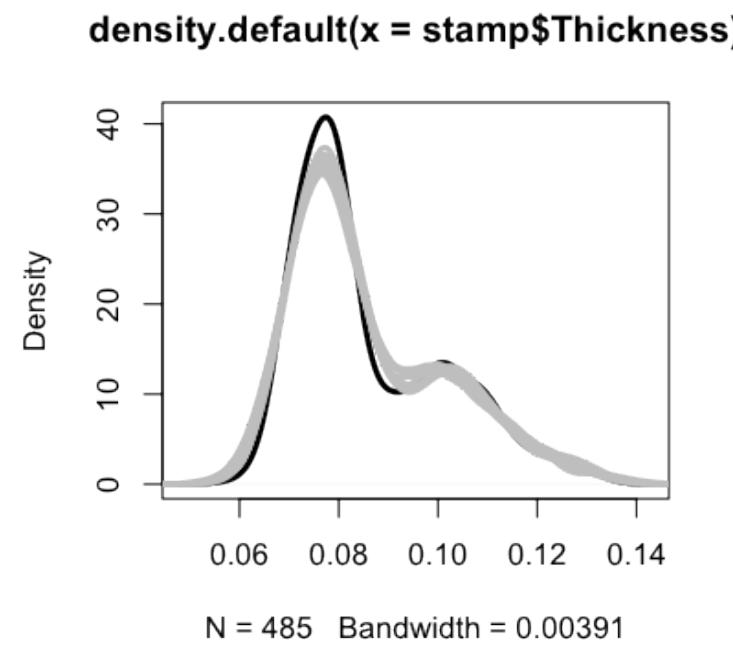


[http://en.wikipedia.org/wiki/File:Comparison\\_of\\_1D\\_histogram\\_and\\_KDE.png](http://en.wikipedia.org/wiki/File:Comparison_of_1D_histogram_and_KDE.png)

12/15

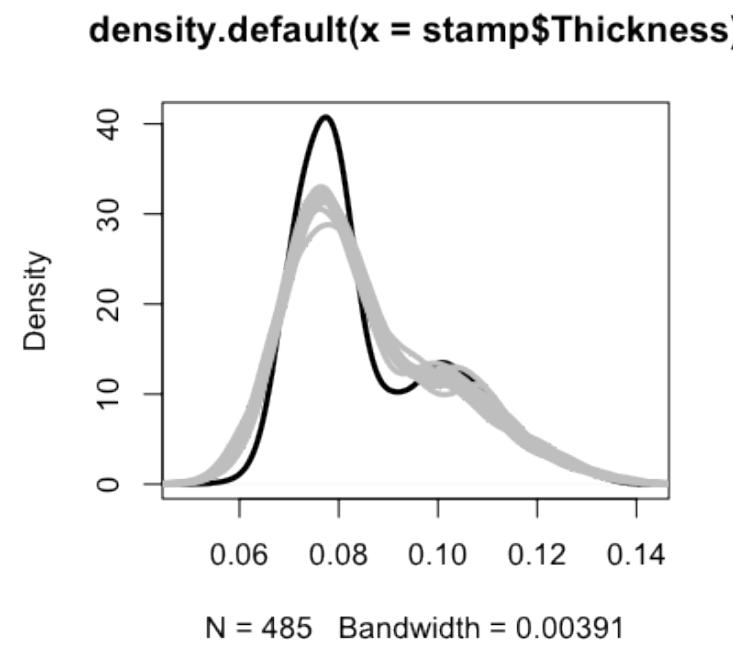
# Simulating from the density estimate

```
plot(density(stamp$Thickness), col="black", lwd=3)
for(i in 1:10){
  newThick <- rnorm(nobs, mean=stamp$Thickness, sd=dens$bw)
  lines(density(newThick), col="grey", lwd=3)
}
```



# Increasing variability

```
plot(density(stamp$Thickness), col="black", lwd=3)
for(i in 1:10){
  newThick <- rnorm(nobs, mean=stamp$Thickness, sd=dens$bw*1.5)
  lines(density(newThick, bw=dens$bw), col="grey", lwd=3)
}
```



14/15

# Notes and further resources

## Notes

- Simulation can be applied to missing data problems - simulate what missing data might be
- Simulation values are often drawn from standard distributions, but this may not be appropriate
- Sensitivity analysis means trying different simulations with different assumptions and seeing how estimates change

## Further resources

- [Advanced Data Analysis From An Elementary Point of View](#)
- [The design of simulation studies in medical statistics](#)
- [Simulation studies in statistics](#)

# Smoothing

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

# Key ideas

- Sometimes there are non-linear trends in data
- We can use "smoothing" to try to capture these
- Still a risk of overfitting
- Often hard to interpret

# CD4 Data

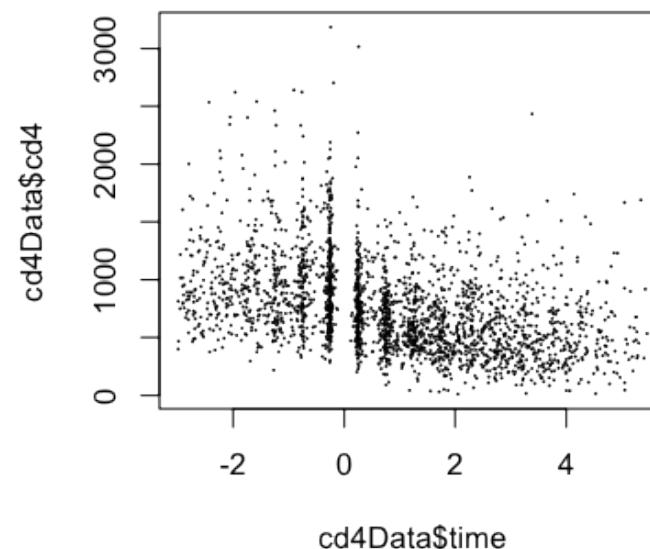
```
download.file("https://spark-public.s3.amazonaws.com/dataanalysis/cd4.data",
              destfile="./data/cd4.data", method="curl")
cd4Data <- read.table("./data/cd4.data",
                      col.names=c("time", "cd4", "age", "packs", "drugs", "sex",
                                 "cesd", "id"))
cd4Data <- cd4Data[order(cd4Data$time),]
head(cd4Data)
```

|      | time   | cd4  | age   | packs | drugs | sex | cesd | id    |
|------|--------|------|-------|-------|-------|-----|------|-------|
| 1279 | -2.990 | 814  | 6.17  | 3     | 1     | 5   | -3   | 30183 |
| 2190 | -2.990 | 400  | -6.02 | 0     | 0     | 3   | -4   | 41406 |
| 1167 | -2.984 | 467  | 13.94 | 0     | 1     | 1   | 0    | 30046 |
| 1427 | -2.957 | 749  | -4.54 | 0     | 1     | -1  | -7   | 30498 |
| 2032 | -2.951 | 1218 | 5.57  | 3     | 1     | 5   | 3    | 41032 |
| 1813 | -2.949 | 1015 | -9.15 | 2     | 1     | 0   | -7   | 40375 |

<http://www.cbcn.umd.edu/~hcorrada/PracticalML/>

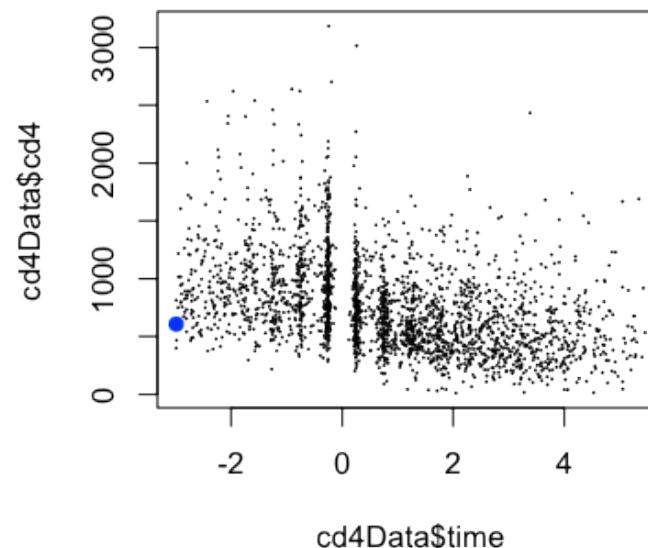
# CD4 over time

```
plot(cd4Data$time,cd4Data$cd4,pch=19,cex=0.1)
```



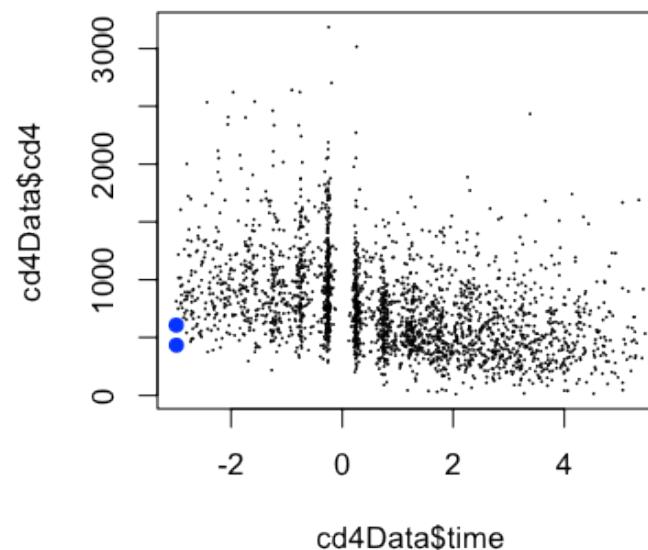
# Average first 2 points

```
plot(cd4Data$time,cd4Data$cd4,pch=19,cex=0.1)
points(mean(cd4Data$time[1:2]),mean(cd4Data$cd4[1:2]),col="blue",pch=19)
```



# Average second and third points

```
plot(cd4Data$time,cd4Data$cd4,pch=19,cex=0.1)
points(mean(cd4Data$time[1:2]),mean(cd4Data$cd4[1:2]),col="blue",pch=19)
points(mean(cd4Data$time[2:3]),mean(cd4Data$cd4[2:3]),col="blue",pch=19)
```



6/21

# A moving average

```
plot(cd4Data$time,cd4Data$cd4,pch=19,cex=0.1)
aveTime <- aveCd4 <- rep(NA,length(3:(dim(cd4Data)[1]-2)))
for(i in 3:(dim(cd4Data)[1]-2)){
  aveTime[i] <- mean(cd4Data$time[(i-2):(i+2)])
  aveCd4[i] <- mean(cd4Data$cd4[(i-2):(i+2)])
}
lines(aveTime,aveCd4,col="blue",lwd=3)
```

# Average more points

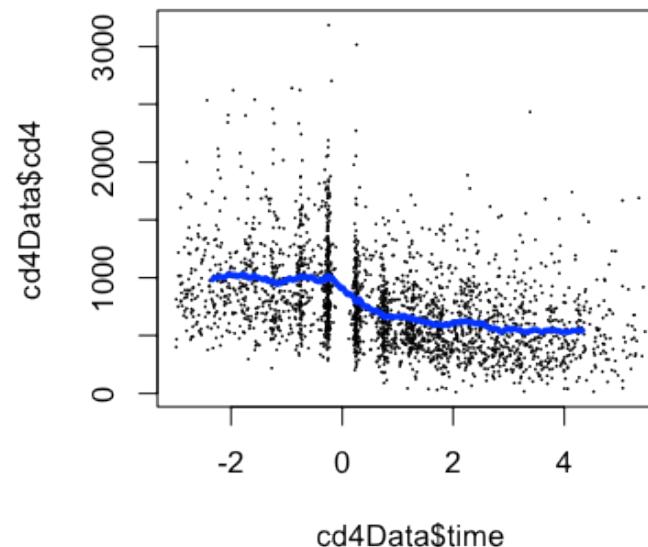
```
plot(cd4Data$time,cd4Data$cd4,pch=19,cex=0.1)
aveTime <- aveCd4 <- rep(NA,length(11:(dim(cd4Data)[1]-10)))
for(i in 11:(dim(cd4Data)[1]-2)){
  aveTime[i] <- mean(cd4Data$time[(i-10):(i+10)])
  aveCd4[i] <- mean(cd4Data$cd4[(i-10):(i+10)])
}
lines(aveTime,aveCd4,col="blue",lwd=3)
```

# Average many more

```
plot(cd4Data$time,cd4Data$cd4,pch=19,cex=0.1)
aveTime <- aveCd4 <- rep(NA,length(201:(dim(cd4Data)[1]-200)))
for(i in 201:(dim(cd4Data)[1]-2)){
  aveTime[i] <- mean(cd4Data$time[(i-200):(i+200)])
  aveCd4[i] <- mean(cd4Data$cd4[(i-200):(i+200)])
}
lines(aveTime,aveCd4,col="blue",lwd=3)
```

# A faster way

```
filtTime <- as.vector(filter(cd4Data$time,filter=rep(1,200))/200)
filtCd4 <- as.vector(filter(cd4Data$cd4,filter=rep(1,200))/200)
plot(cd4Data$time,cd4Data$cd4,pch=19,cex=0.1); lines(filtTime,filtCd4,col="blue",lwd=3)
```



10/21

# Averaging = weighted sums

```
filtCd4 <- as.vector(filter(cd4Data$cd4,filter=rep(1,4))/4)
filtCd4[2]
```

```
[1] 607.5
```

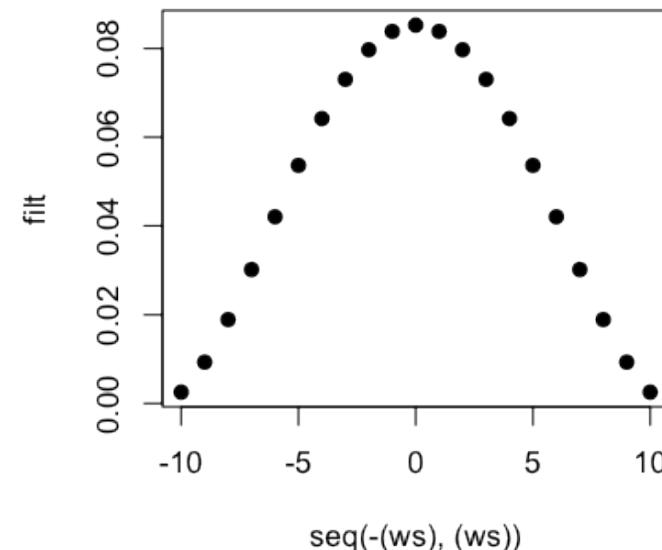
```
sum(cd4Data$cd4[1:4] * rep(1/4,4))
```

```
[1] 607.5
```

11/21

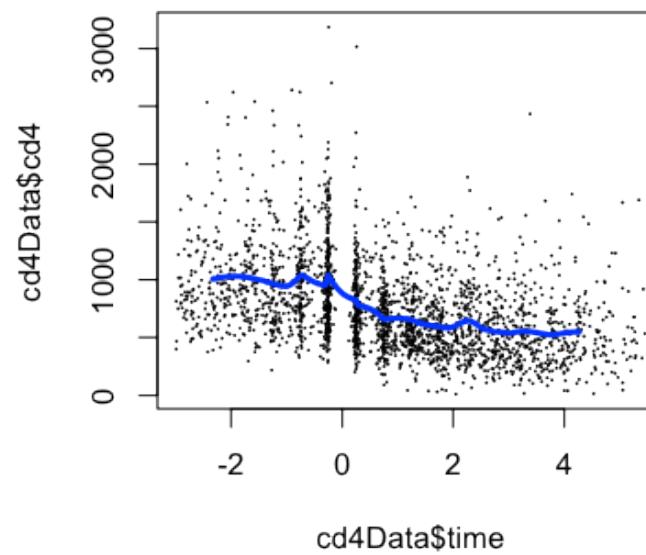
# Other weights -> should sum to one

```
ws = 10; tukey = function(x) pmax(1 - x^2,0)^2
filt= tukey(seq(-ws,ws)/(ws+1));filt=filt/sum(filt)
plot(seq(-(ws),(ws)),filt,pch=19)
```



# Other weights -> should sum to one

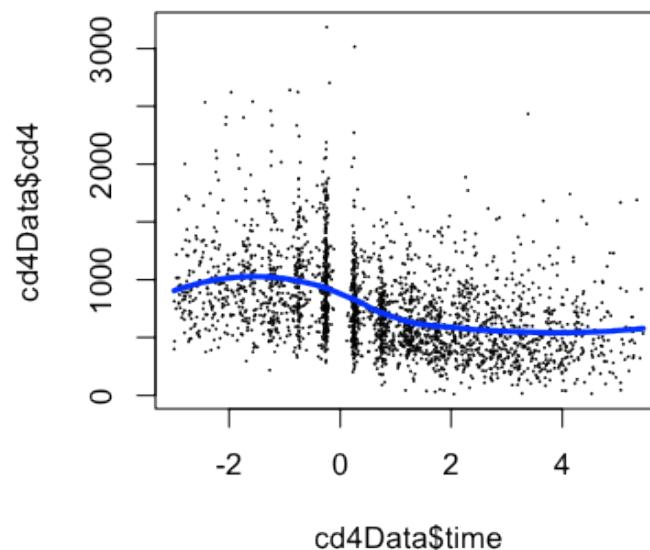
```
ws = 100; tukey = function(x) pmax(1 - x^2,0)^2
filt= tukey(seq(-ws,ws)/(ws+1));filt=filt/sum(filt)
filtTime <- as.vector(filter(cd4Data$time,filter=filt))
filtCd4 <- as.vector(filter(cd4Data$cd4,filter=filt))
plot(cd4Data$time,cd4Data$cd4,pch=19,cex=0.1); lines(filtTime,filtCd4,col="blue",lwd=3)
```



13/21

# Lowess (loess)

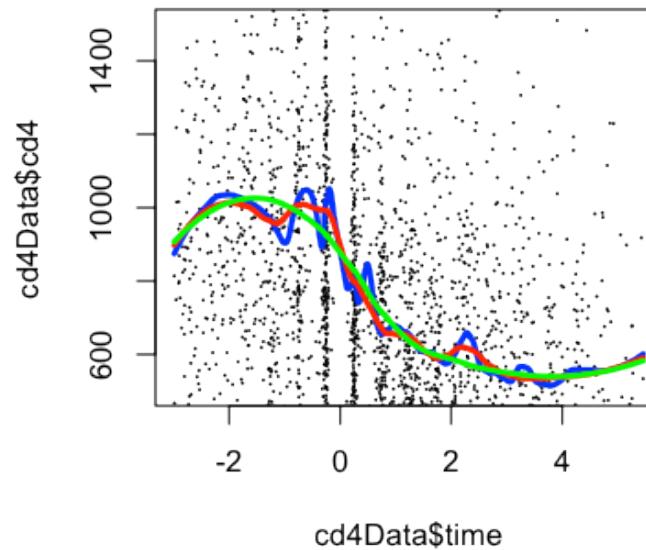
```
lw1 <- loess(cd4 ~ time,data=cd4Data)
plot(cd4Data$time,cd4Data$cd4,pch=19,cex=0.1)
lines(cd4Data$time,lw1$fitted,col="blue",lwd=3)
```



14/21

# Span

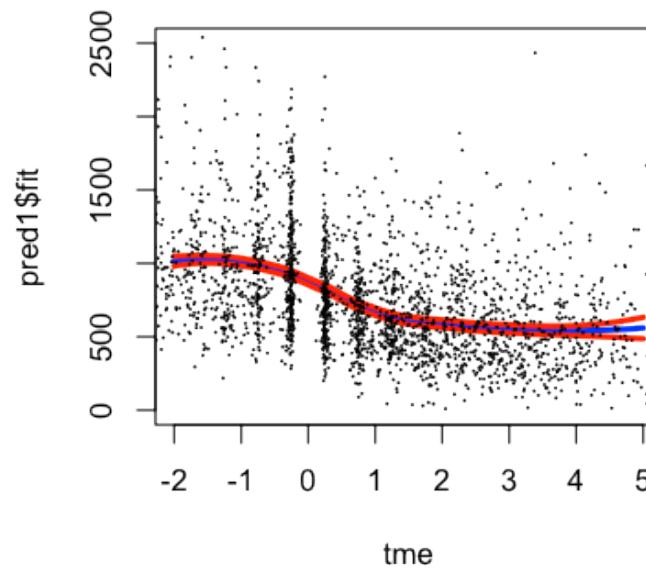
```
plot(cd4Data$time,cd4Data$cd4,pch=19,cex=0.1,ylim=c(500,1500))
lines(cd4Data$time,loess(cd4 ~ time,data=cd4Data,span=0.1)$fitted,col="blue",lwd=3)
lines(cd4Data$time,loess(cd4 ~ time,data=cd4Data,span=0.25)$fitted,col="red",lwd=3)
lines(cd4Data$time,loess(cd4 ~ time,data=cd4Data,span=0.76)$fitted,col="green",lwd=3)
```



15/21

# Predicting with loess

```
tme <- seq(-2,5,length=100); pred1 = predict(lw1,newdata=data.frame(time=tme),se=TRUE)
plot(tme,pred1$fit,col="blue",lwd=3,type="l",ylim=c(0,2500))
lines(tme,pred1$fit + 1.96*pred1$se.fit,col="red",lwd=3)
lines(tme,pred1$fit - 1.96*pred1$se.fit,col="red",lwd=3)
points(cd4Data$time,cd4Data$cd4,pch=19,cex=0.1)
```



16/21

# Splines

$$Y_i = b_0 + \sum_{k=1}^K b_k s_k(x_i) + e_i$$

$Y_i$  - outcome for  $i$ th observation

$b_0$  - Intercept term

$b_k$  - Coefficient for  $k$ th spline function

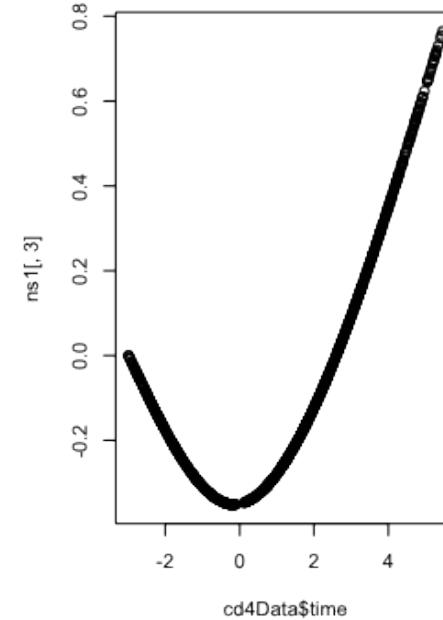
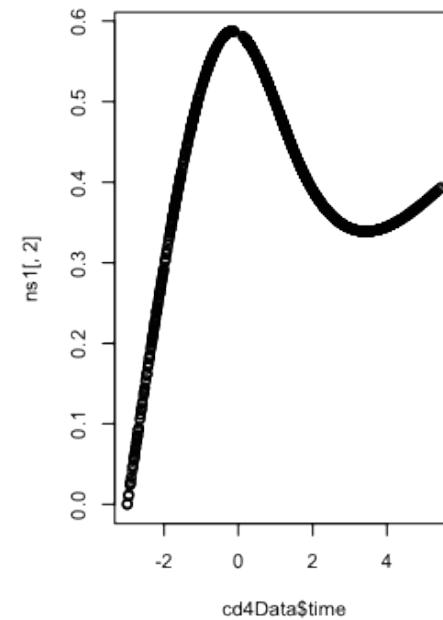
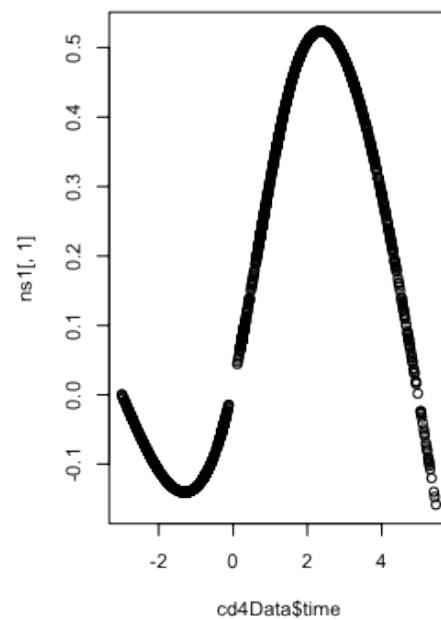
$s_k$  -  $k$ th spline function

$x_i$  - covariate for  $i$ th observation

$e_i$  - everything we didn't measure/model

# Splines in R

```
library(splines)
ns1 <- ns(cd4Data$time, df=3)
par(mfrow=c(1,3))
plot(cd4Data$time, ns1[,1]); plot(cd4Data$time, ns1[,2]); plot(cd4Data$time, ns1[,3])
```



18/21

# Regression with splines

```
lm1 <- lm(cd4Data$cd4 ~ ns1)
summary(lm1)
```

Call:

```
lm(formula = cd4Data$cd4 ~ ns1)
```

Residuals:

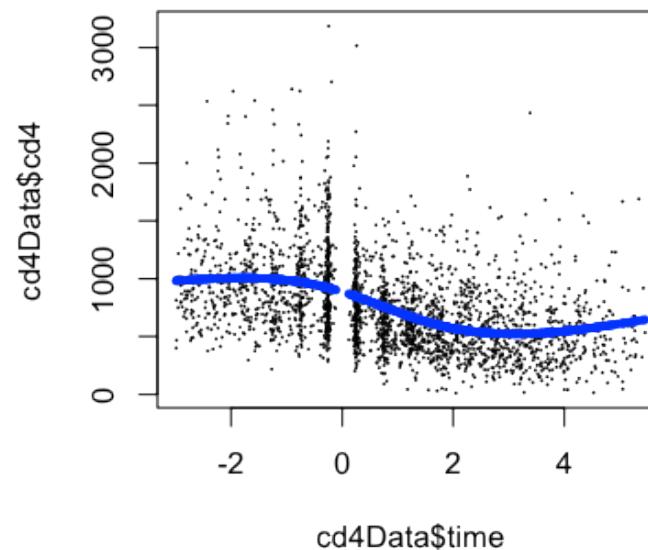
| Min    | 1Q     | Median | 3Q    | Max    |
|--------|--------|--------|-------|--------|
| -780.0 | -242.4 | -61.3  | 169.5 | 2263.7 |

Coefficients:

|                | Estimate | Std. Error | t value | Pr(> t )    |
|----------------|----------|------------|---------|-------------|
| (Intercept)    | 982.0    | 33.9       | 29.01   | < 2e-16 *** |
| ns11           | -611.3   | 32.6       | -18.78  | < 2e-16 *** |
| ns12           | -373.7   | 79.4       | -4.71   | 2.6e-06 *** |
| ns13           | -374.8   | 41.2       | -9.09   | < 2e-16 *** |
| ---            |          |            |         |             |
| Signif. codes: | 0        | '***'      | 0.001   | '**'        |
|                |          |            | 0.01    | '*'         |
|                |          |            | 0.05    | '. '        |
|                |          |            | 0.1     | ' '         |
|                |          |            | 1       |             |

# Fitted values

```
plot(cd4Data$time,cd4Data$cd4,pch=19,cex=0.1)
points(cd4Data$time,lm1$fitted,col="blue",pch=19,cex=0.5)
```



20/21

# Notes and further resources

## Notes:

- Cross-validation with splines/smoothing is a good idea
- Do not predict outside the range of observed data

## Further resources:

- [Hector Corrada Bravo's Lecture Notes](#)
- [Rafa Irizarry's Lecture Notes on smoothing, On splines](#)
- [Elements of Statistical Learning](#)
- [Advanced Data Analysis from An Elementary Point of View](#)

# Sources of data sets

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

# Data are defined by how they are collected

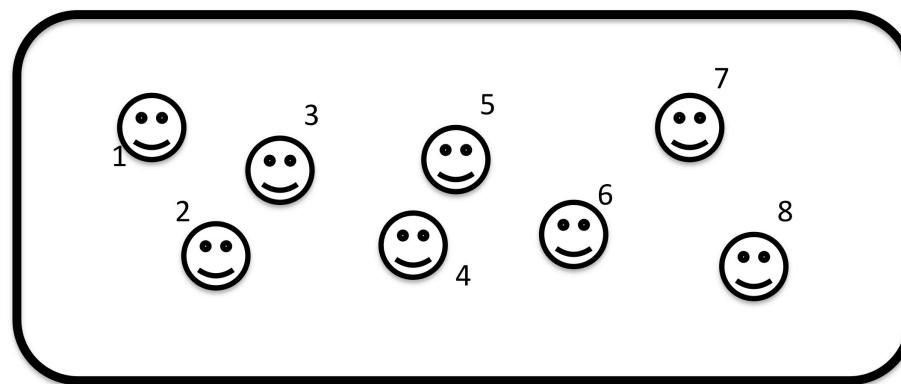
## Main types

- Census (descriptive)
- Observational study (inferential)
- Convenience sample (all types - may be biased)
- Randomized trial (causal)

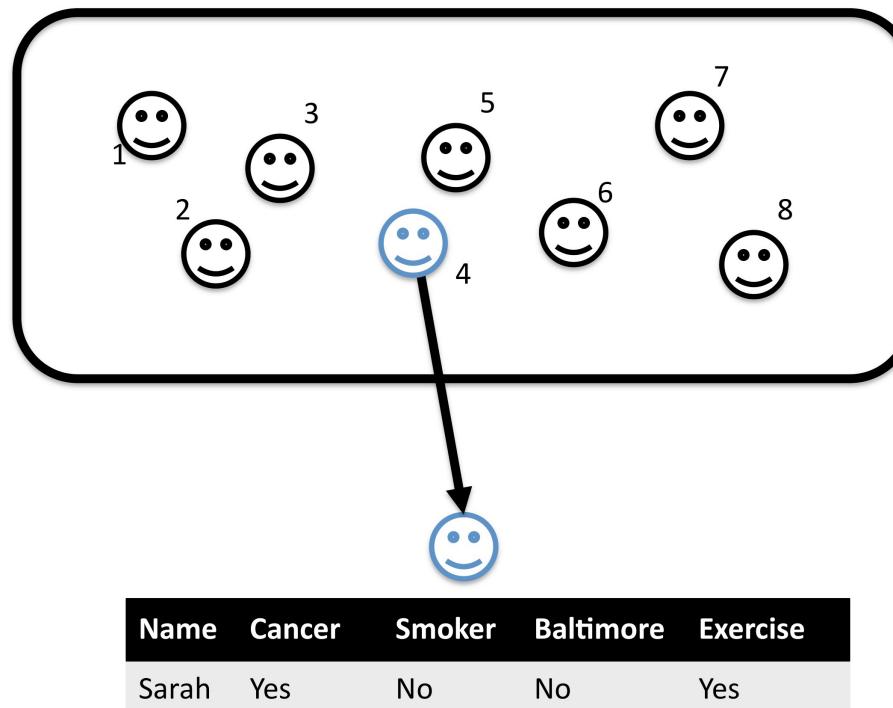
## Other types

- Prediction study (prediction)
- Studies over time
  - Cross sectional (inferential)
  - Longitudinal (inferential, predictive)
- Retrospective (inferential)

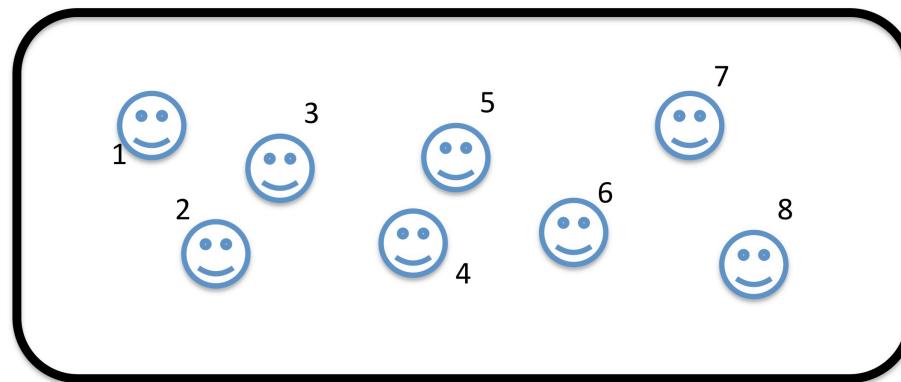
# A population



# Pick a person and measure



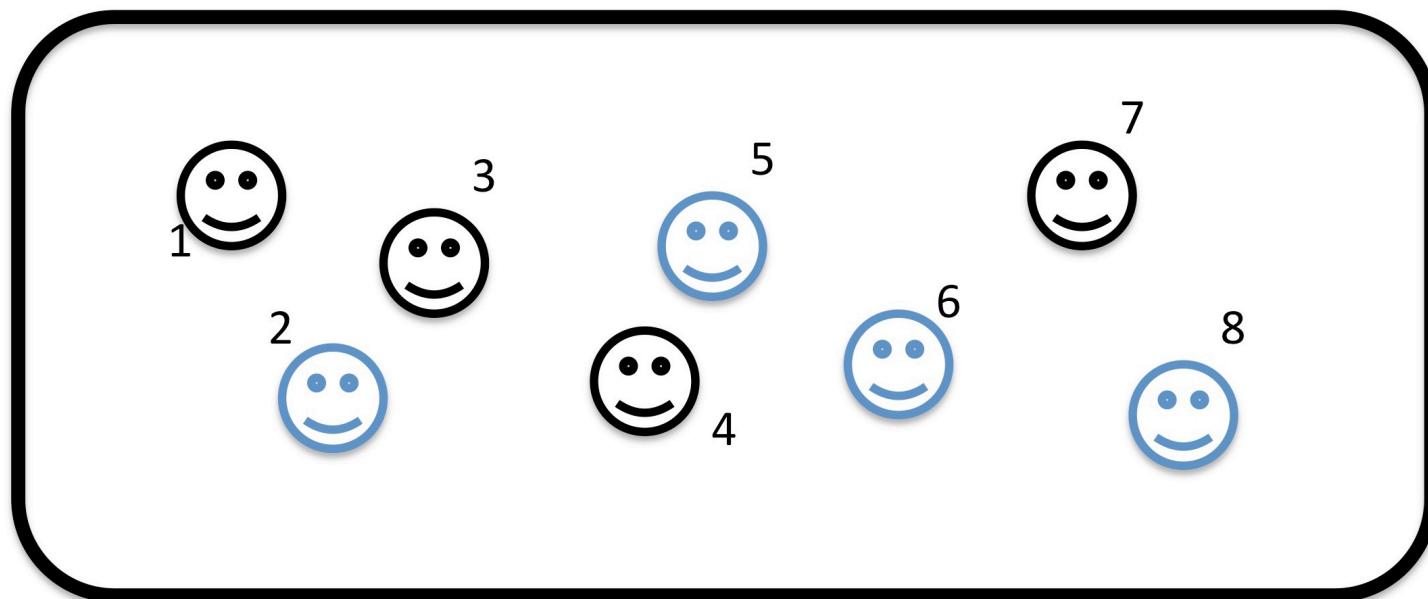
# Census



# Observational study

```
set.seed(5)  
sample(1:8, size=4, replace=FALSE)
```

```
[1] 2 5 6 8
```

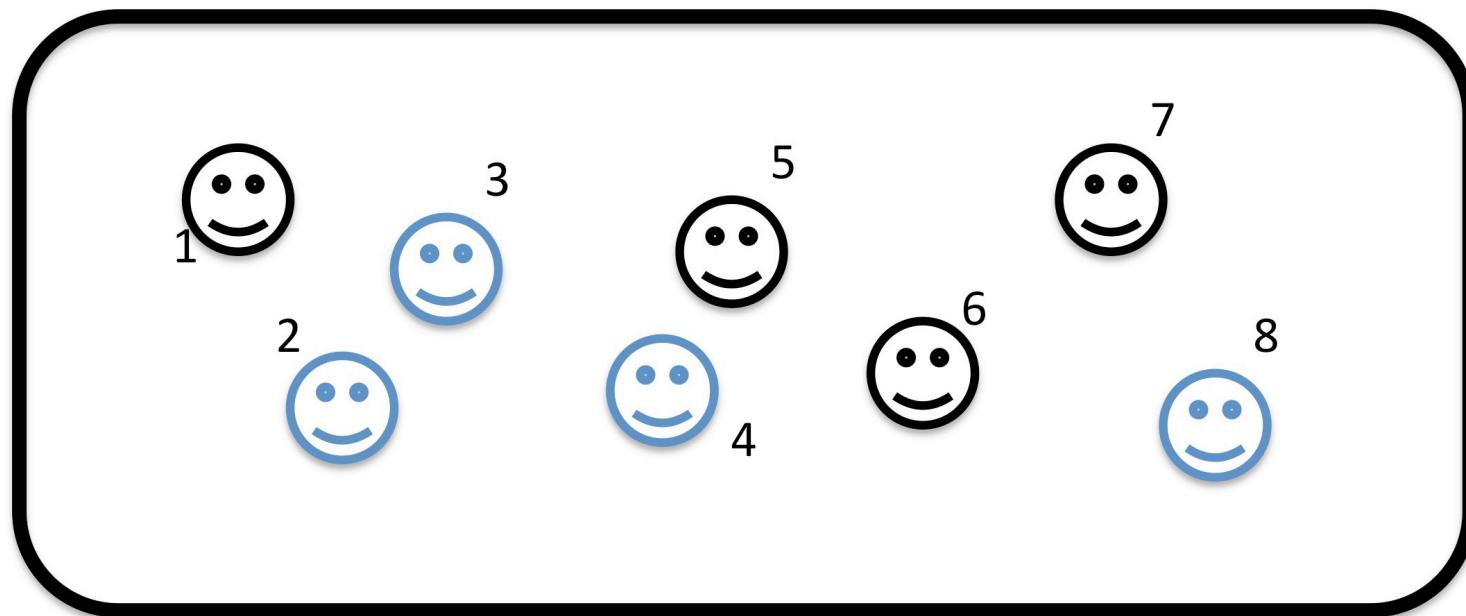


6/13

# Convenience sample

```
probs = c(5,5,5,5,1,1,1,1)/24  
sample(1:8,size=4,replace=FALSE,prob=probs)
```

```
[1] 4 1 2 5
```

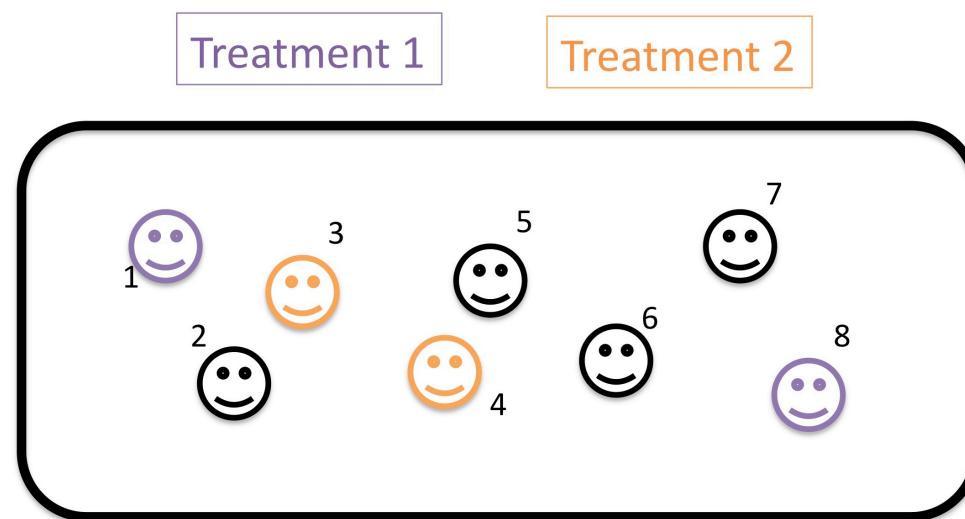


7/13

# Randomized trial

```
treat1 = sample(1:8,size=2,replace=FALSE); treat2 = sample(2:7,size=2,replace=FALSE)  
c(treat1,treat2)
```

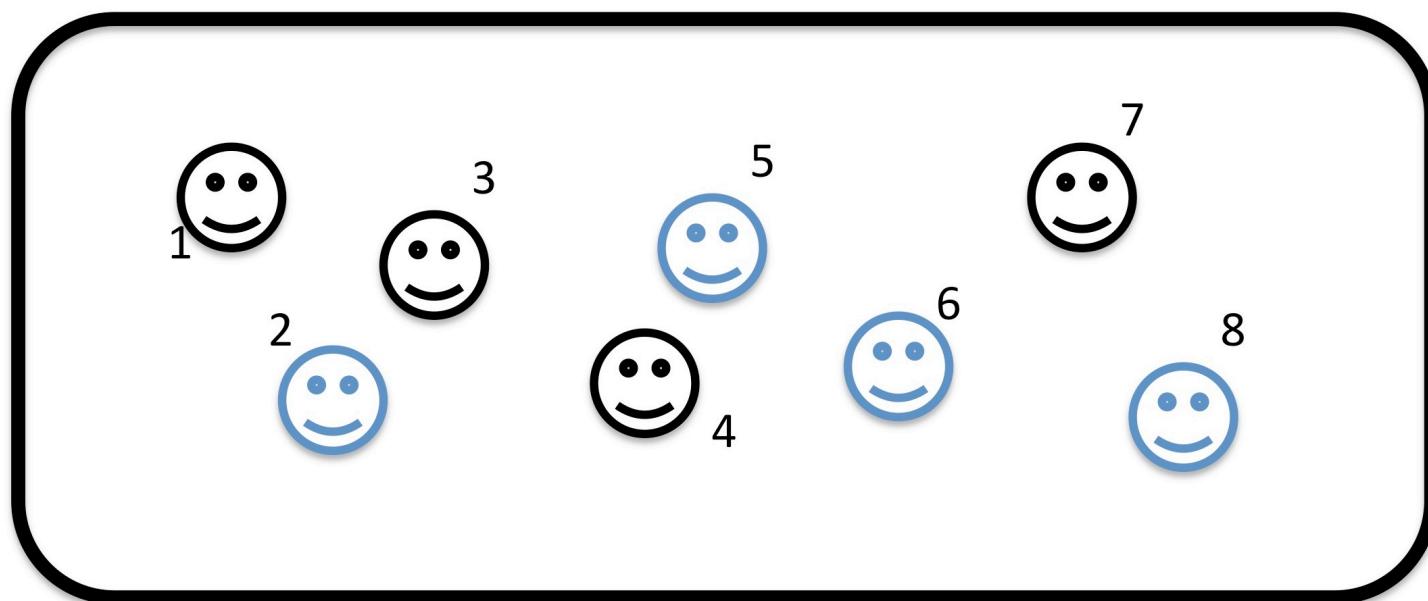
```
[1] 8 1 3 4
```



# Prediction study: train

```
set.seed(5)  
sample(1:8, size=4, replace=FALSE)
```

```
[1] 2 5 6 8
```

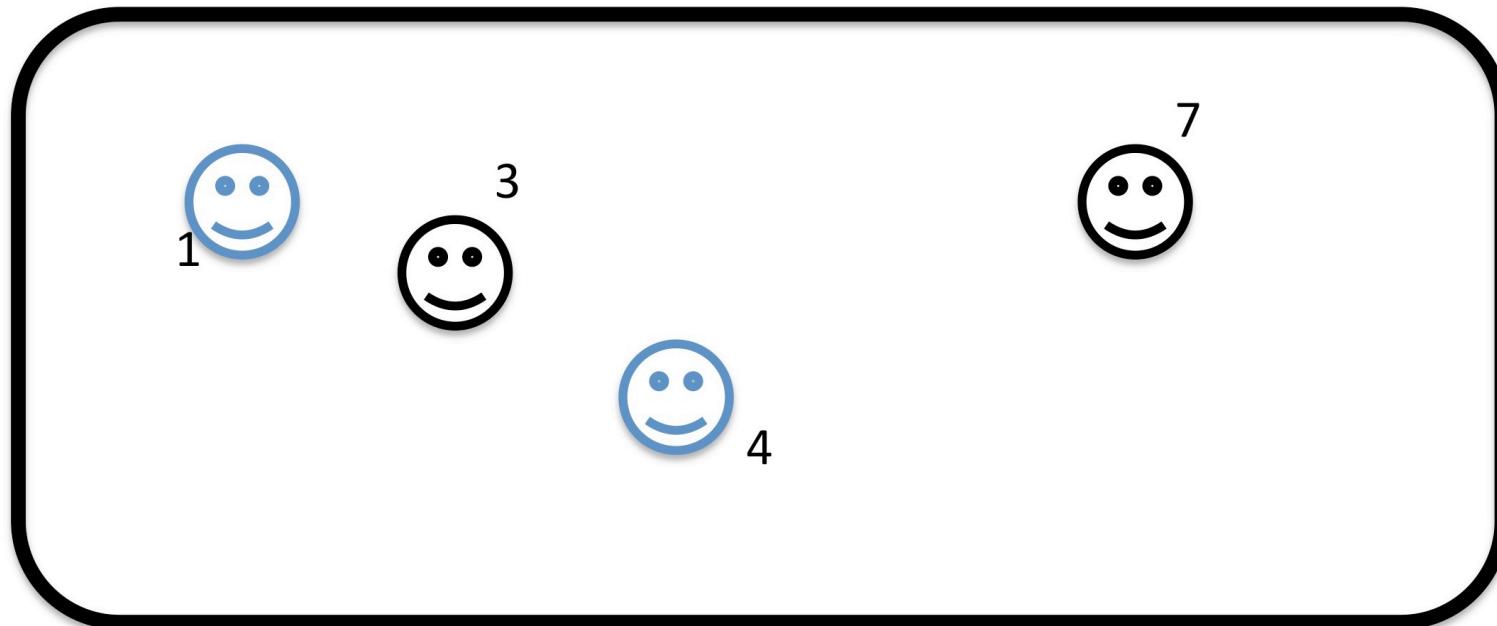


9/13

# Prediction study: test

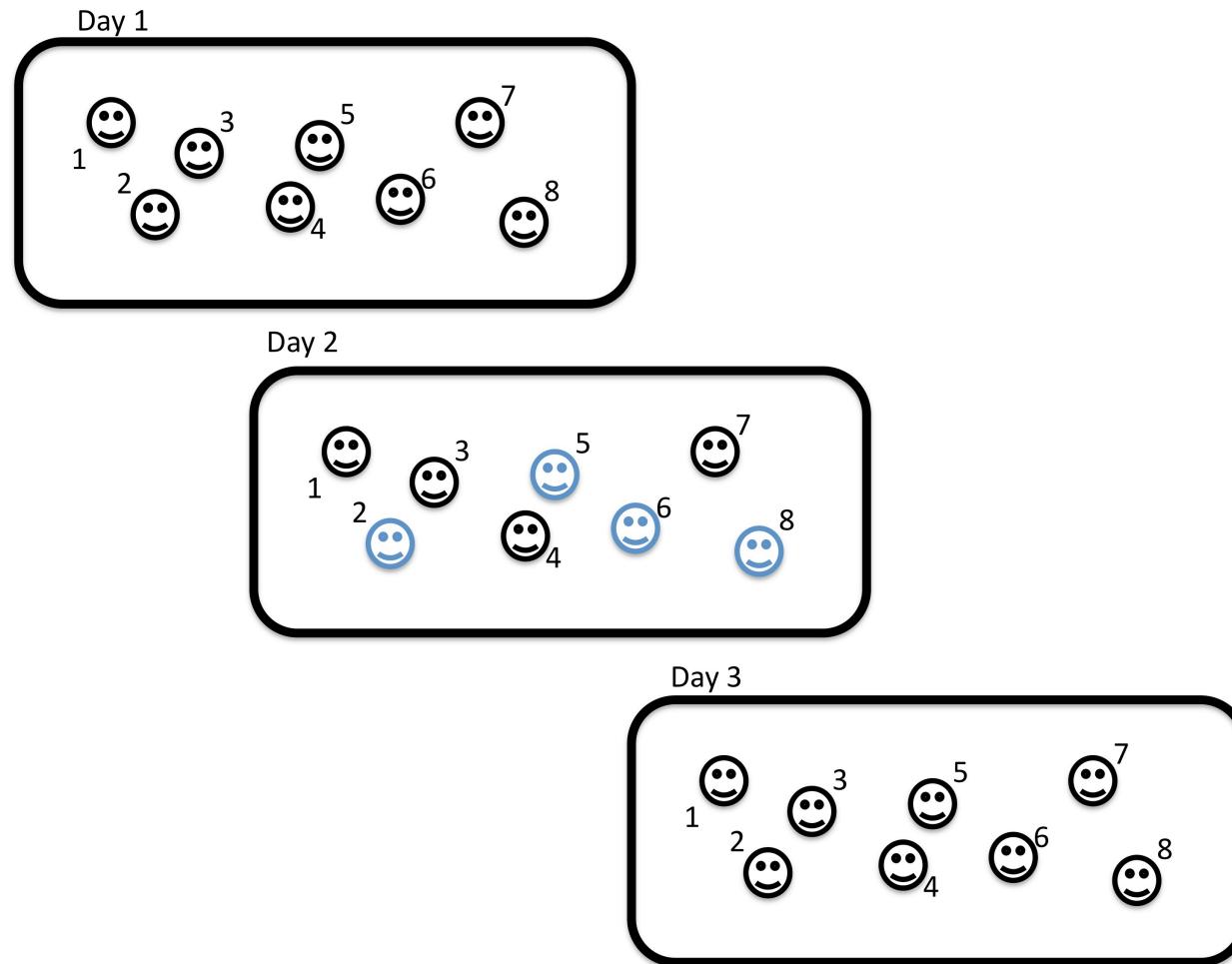
```
sample(c(1,3,4,7),size=2,replace=FALSE)
```

```
[1] 1 4
```

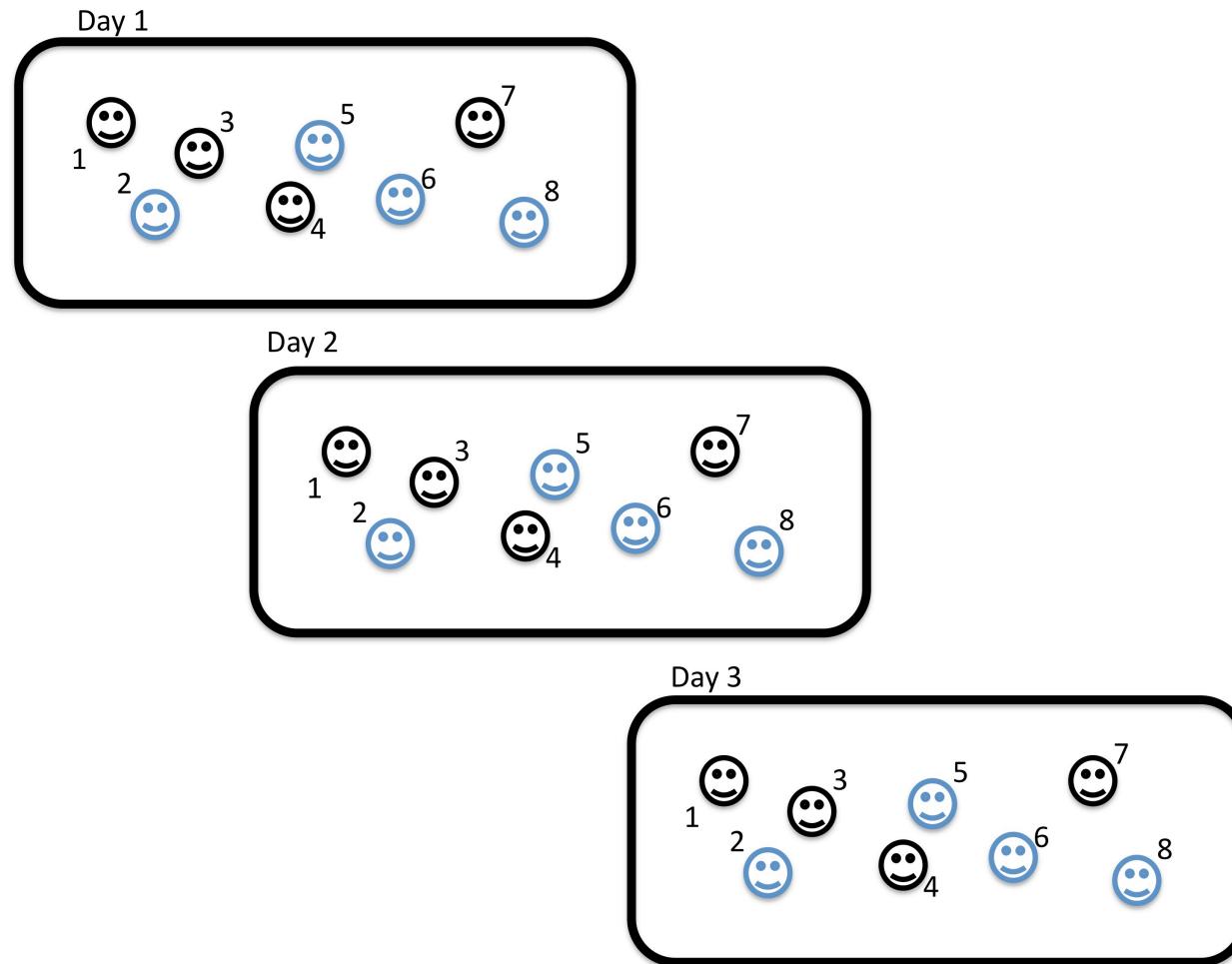


10/13

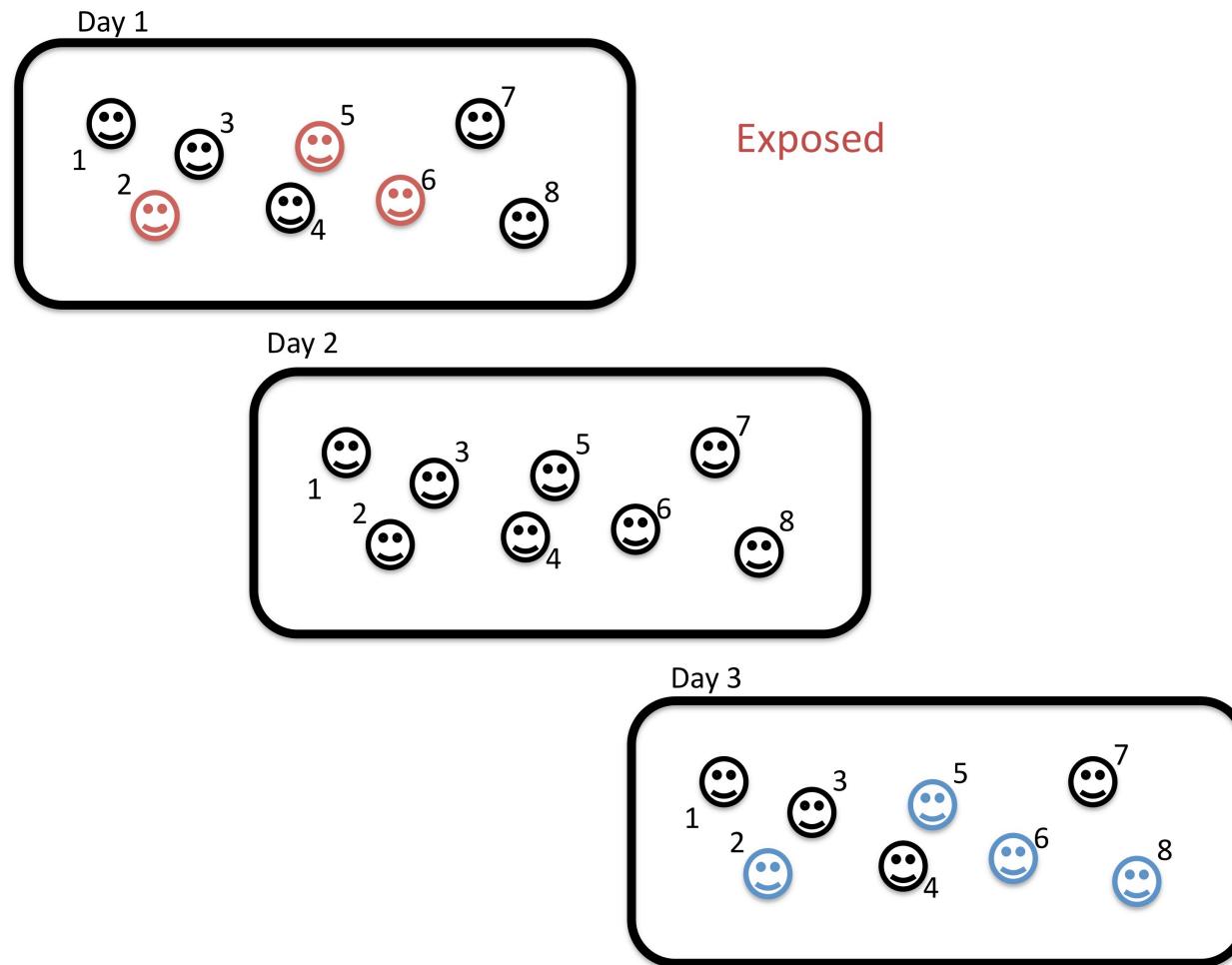
# Study over time: cross-sectional



# Study over time: longitudinal



# Study over time: retrospective



# Structure of a Data Analysis

## Part 1

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

# 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

# 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

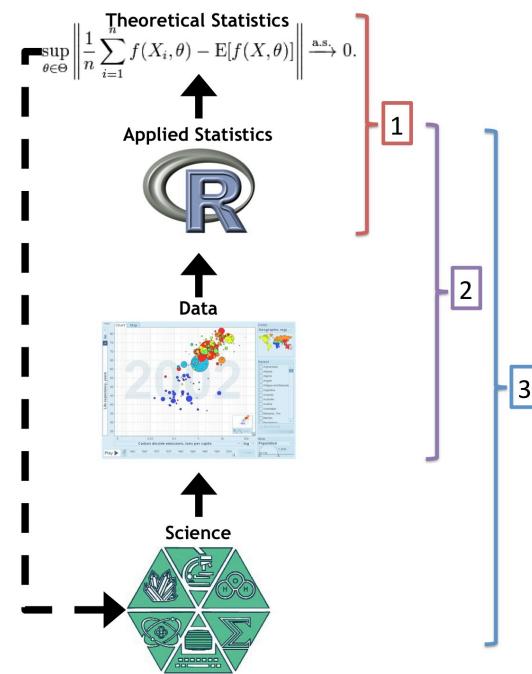
# The key challenge in data analysis

“ 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

# Defining a question



1. Statistical methods development
2. Danger zone!!!
3. Proper data analysis

5/16

# An example

## Start with a general question

Can I automatically detect emails that are SPAM that are not?

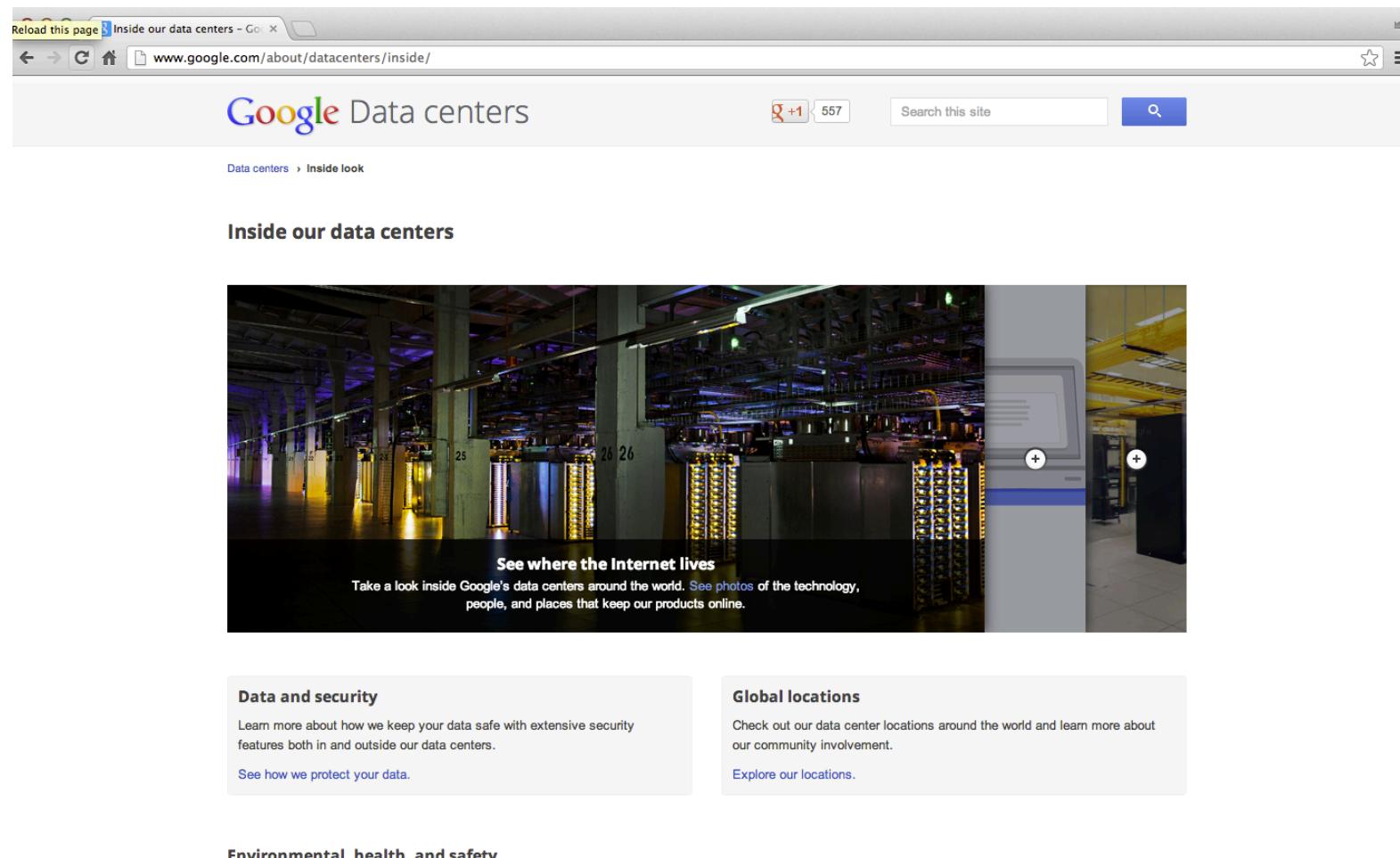
## Make it concrete

Can I use quantitative characteristics of the emails to classify them as SPAM/HAM?

# Define the ideal data set

- The data set may depend on your goal
  - Descriptive - a whole population
  - Exploratory - a random sample with many variables measured
  - Inferential - the right population, randomly sampled
  - Predictive - a training and test data set from the same population
  - Causal - data from a randomized study
  - Mechanistic - data about all components of the system

# Our example



A screenshot of a web browser displaying the Google Data Centers Inside look page. The URL in the address bar is [www.google.com/about/datacenters/inside/](http://www.google.com/about/datacenters/inside/). The page title is "Google Data centers". Below the title, the breadcrumb navigation shows "Data centers > Inside look". The main content area features a large image of a data center with server racks and a caption: "See where the Internet lives. Take a look inside Google's data centers around the world. See photos of the technology, people, and places that keep our products online." Below this, there are two sections: "Data and security" and "Global locations".

**Data and security**  
Learn more about how we keep your data safe with extensive security features both in and outside our data centers.  
[See how we protect your data.](#)

**Global locations**  
Check out our data center locations around the world and learn more about our community involvement.  
[Explore our locations.](#)

Environmental, health, and safety

<http://www.google.com/about/datacenters/inside/>

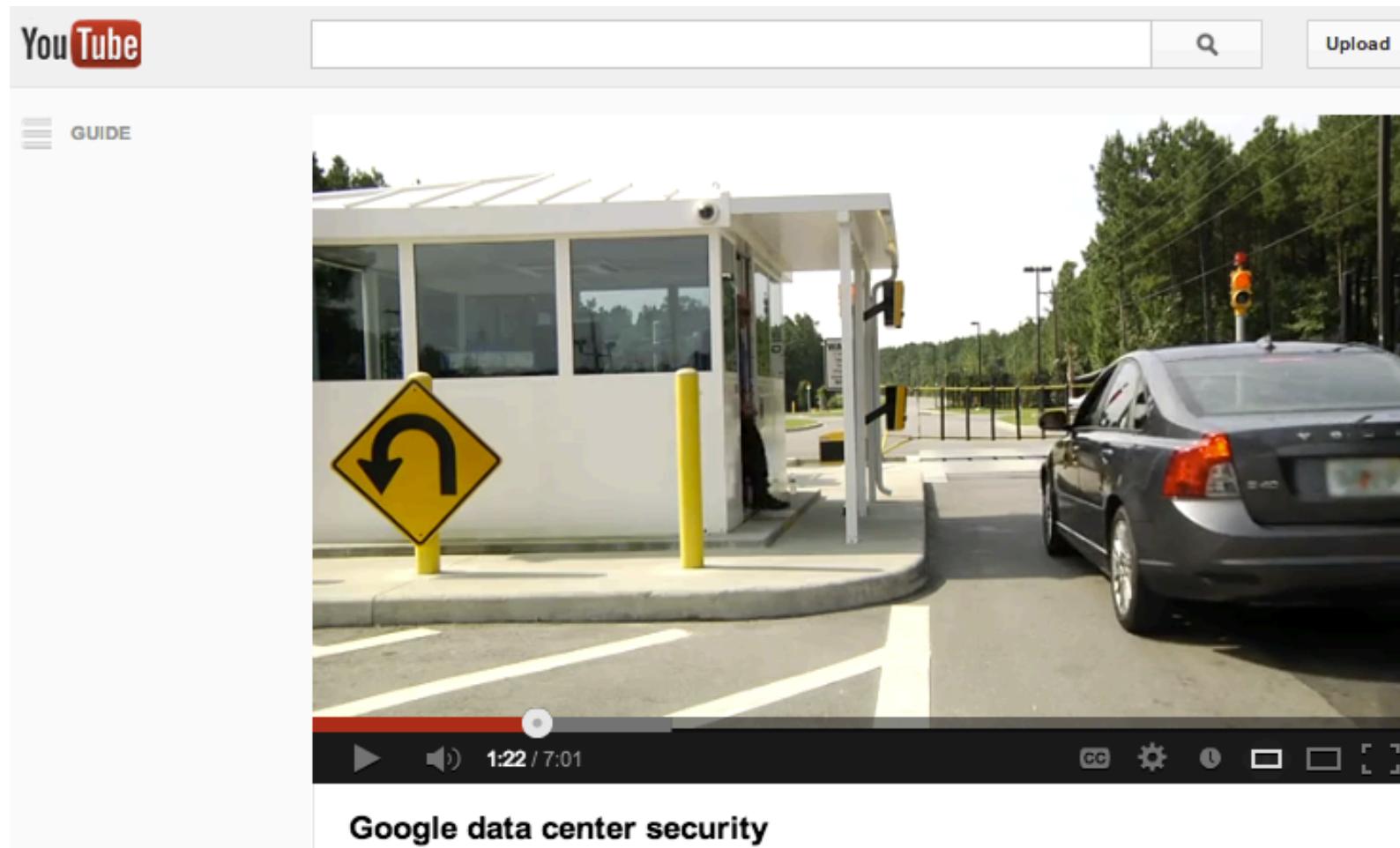
8/16

# Determine what data you can access

- Sometimes you can find data free on the web
- Other times you may need to buy the data
- Be sure to respect the terms of use
- If the data don't exist, you may need to generate it yourself

9/16

# Back to our example



Google data center security

10/16

# A possible solution

Screenshot of a web browser showing the UCI Machine Learning Repository page for the Spambase dataset.

**UCI Machine Learning Repos** archive.ics.uci.edu/ml/datasets/Spambase

**Machine Learning Repository**  
Center for Machine Learning and Intelligent Systems

**Spambase Data Set**

**Download:** [Data Folder](#), [Data Set Description](#)

**Abstract:** Classifying Email as Spam or Non-Spam

**Deleted Items**

| ID | Name           | Subject   |
|----|----------------|---|
| 1  | CarLoPro       | Get the car of your dreams with CarLoPro! Help! |
| 2  | StatHelp       | How Old Are You Really? Take the FreeAge Test   |
| 3  | PrizeDraw      | Win a \$1000000000 Giveaway!                    |
| 4  | BernieM        | Unsubscribe                                     |
| 5  | Illustratop    | Special Illustratop Member Offer                |
| 6  | Acme Credit    | Process Credit Cards for Zero Up Front Cost     |
| 7  | Acme Credit    | Process Credit Cards for Zero Up Front Cost     |
| 8  | Quick Cash A   | Get A \$5000 Cash Advance                       |
| 9  | LoyaltyRewards | Brontford embroidery                            |
| 10 | Wells Fargo    | Open a New Account                              |
| 11 | Cash Back      | Get a \$5000 Starbucks Gift Card on us          |
| 12 | GoldCard       | Free No Attention to the Man Behind the Camera  |
| 13 | TravelMedic    | Get ready for Monday CYCLOPS BETTER             |

|                                   |                |                              |      |                            |            |
|-----------------------------------|----------------|------------------------------|------|----------------------------|------------|
| <b>Data Set Characteristics:</b>  | Multivariate   | <b>Number of Instances:</b>  | 4601 | <b>Area:</b>               | Computer   |
| <b>Attribute Characteristics:</b> | Integer, Real  | <b>Number of Attributes:</b> | 57   | <b>Date Donated</b>        | 1999-07-01 |
| <b>Associated Tasks:</b>          | Classification | <b>Missing Values?</b>       | Yes  | <b>Number of Web Hits:</b> | 66346      |

**Source:**

Creators:

Mark Hopkins, Erik Reeber, George Forman, Jaap Suermondt  
Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304

Donor:

George Forman (gforman at nosspam hpl.hp.com) 650-857-7835

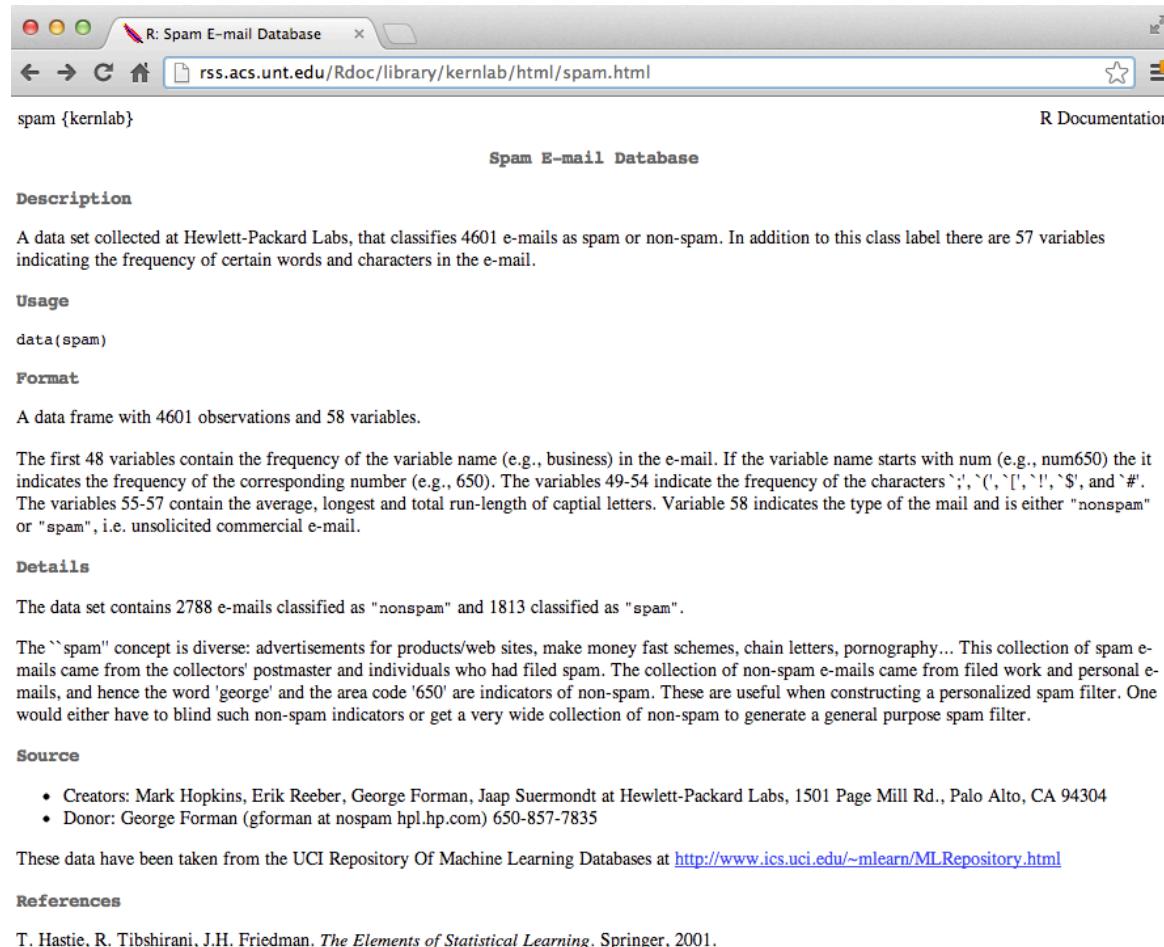
<http://archive.ics.uci.edu/ml/datasets/Spambase>

11/16

# Obtain the data

- Try to obtain the raw data
- Be sure to reference the source
- Polite emails go a long way
- If you will load the data from an internet source, record the url and time accessed

# Our data set



The screenshot shows a web browser window titled "R: Spam E-mail Database". The URL in the address bar is "rss.acs.unt.edu/Rdoc/library/kernlab/html/spam.html". The page content is the R documentation for the "spam" dataset. It includes sections for "Description", "Usage", "Format", "Details", "Source", and "References". The "Description" section states: "A data set collected at Hewlett-Packard Labs, that classifies 4601 e-mails as spam or non-spam. In addition to this class label there are 57 variables indicating the frequency of certain words and characters in the e-mail." The "Usage" section shows the command "data(spam)". The "Format" section notes it is a data frame with 4601 observations and 58 variables. The "Details" section provides a detailed description of the 58 variables, mentioning they represent word frequencies and character counts. The "Source" section lists creators and a donor, and the "References" section cites "The Elements of Statistical Learning" by T. Hastie, R. Tibshirani, J.H. Friedman.

spam {kernlab}

R Documentation

**Spam E-mail Database**

**Description**

A data set collected at Hewlett-Packard Labs, that classifies 4601 e-mails as spam or non-spam. In addition to this class label there are 57 variables indicating the frequency of certain words and characters in the e-mail.

**Usage**

`data(spam)`

**Format**

A data frame with 4601 observations and 58 variables.

The first 48 variables contain the frequency of the variable name (e.g., business) in the e-mail. If the variable name starts with num (e.g., num650) the it indicates the frequency of the corresponding number (e.g., 650). The variables 49-54 indicate the frequency of the characters `;`, `(`, `[`, `!`, `\$`, and `#`. The variables 55-57 contain the average, longest and total run-length of captial letters. Variable 58 indicates the type of the mail and is either "nonsspam" or "spam", i.e. unsolicited commercial e-mail.

**Details**

The data set contains 2788 e-mails classified as "nonsspam" and 1813 classified as "spam".

The "spam" concept is diverse: advertisements for products/web sites, make money fast schemes, chain letters, pornography... This collection of spam e-mails came from the collectors' postmaster and individuals who had filed spam. The collection of non-spam e-mails came from filed work and personal e-mails, and hence the word 'george' and the area code '650' are indicators of non-spam. These are useful when constructing a personalized spam filter. One would either have to blind such non-spam indicators or get a very wide collection of non-spam to generate a general purpose spam filter.

**Source**

- Creators: Mark Hopkins, Erik Reeber, George Forman, Jaap Suurmond at Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304
- Donor: George Forman (gforman at nospam hpl.hp.com) 650-857-7835

These data have been taken from the UCI Repository Of Machine Learning Databases at <http://www.ics.uci.edu/~mlearn/MLRepository.html>

**References**

T. Hastie, R. Tibshirani, J.H. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

<http://rss.acs.unt.edu/Rdoc/library/kernlab/html/spam.html>

13/16

# Clean the data

- Raw data often needs to be processed
- If it is pre-processed, make sure you understand how
- Understand the source of the data (census, sample, convenience sample, etc.)
- May need reformatting, subsampling - record these steps
- **Determine if the data are good enough** - if not, quit or change data

# Our cleaned data set

```
# If it isn't installed, install the kernlab package  
library(kernlab)  
data(spam)  
dim(spam)
```

```
[1] 4601    58
```

<http://rss.acs.unt.edu/Rdoc/library/kernlab/html/spam.html>

15/16

# Subsampling our data set

We need to generate a test and training set (prediction)

```
set.seed(3435)
trainIndicator = rbinom(4601, size=1, prob=0.5)
table(trainIndicator)
```

```
trainIndicator
```

```
0      1
2314 2287
```

```
trainSpam = spam[trainIndicator==1, ]
testSpam = spam[trainIndicator==0, ]
dim(trainSpam)
```

```
[1] 2287   58
```

16/16

# Structure of a Data Analysis

## Part 2

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

# 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

# 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

3/24

# An example

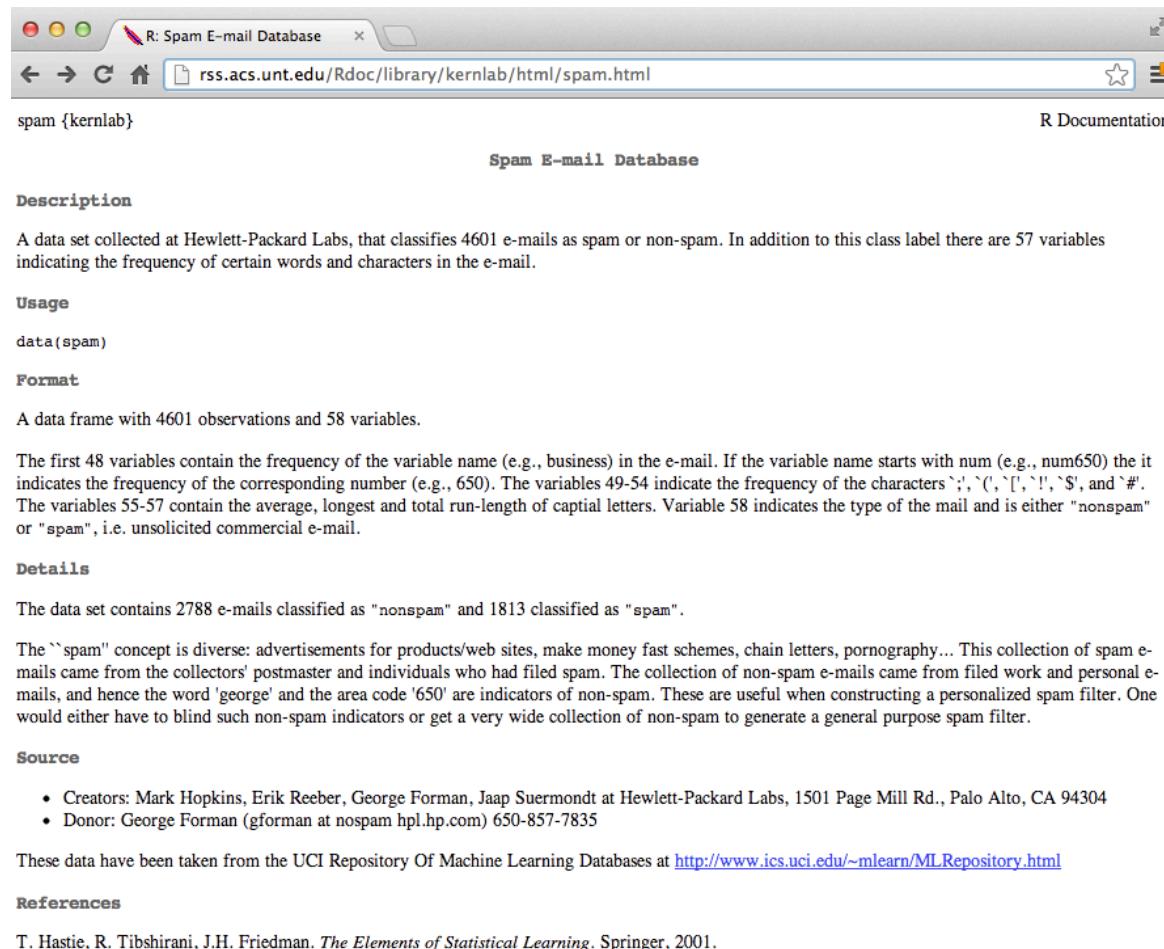
## Start with a general question

Can I automatically detect emails that are SPAM that are not?

## Make it concrete

Can I use quantitative characteristics of the emails to classify them as SPAM/HAM?

# Our data set



The screenshot shows a web browser window titled "R: Spam E-mail Database". The URL in the address bar is "rss.acs.unt.edu/Rdoc/library/kernlab/html/spam.html". The page content is the R documentation for the "spam" dataset from the "kernlab" package. The documentation includes sections for "Description", "Usage", "Format", "Details", "Source", and "References". The "Description" section states: "A data set collected at Hewlett-Packard Labs, that classifies 4601 e-mails as spam or non-spam. In addition to this class label there are 57 variables indicating the frequency of certain words and characters in the e-mail." The "Usage" section shows the command "data(spam)". The "Format" section notes it is a data frame with 4601 observations and 58 variables. The "Details" section provides a detailed description of the 58 variables, mentioning they represent word frequencies and character counts. The "Source" section lists creators and a donor, and mentions the data was taken from the UCI Repository Of Machine Learning Databases. The "References" section cites "T. Hastie, R. Tibshirani, J.H. Friedman. *The Elements of Statistical Learning*. Springer, 2001."

<http://rss.acs.unt.edu/Rdoc/library/kernlab/html/spam.html>

5/24

# Subsampling our data set

We need to generate a test and training set (prediction)

```
# If it isn't installed, install the kernlab package
library(kernlab)
data(spam)
# Perform the subsampling
set.seed(3435)
trainIndicator = rbinom(4601, size = 1, prob = 0.5)
table(trainIndicator)

## trainIndicator
##    0     1
## 2314 2287

trainSpam = spam[trainIndicator == 1, ]
testSpam = spam[trainIndicator == 0, ]
```

6/24

# Exploratory data analysis

- Look at summaries of the data
- Check for missing data
- Create exploratory plots
- Perform exploratory analyses (e.g. clustering)

7/24

# Names

```
names(trainSpam)
```

```
## [1] "make"           "address"        "all"
## [4] "num3d"          "our"            "over"
## [7] "remove"         "internet"       "order"
## [10] "mail"           "receive"        "will"
## [13] "people"         "report"         "addresses"
## [16] "free"           "business"       "email"
## [19] "you"            "credit"         "your"
## [22] "font"           "num000"          "money"
## [25] "hp"              "hpl"            "george"
## [28] "num650"          "lab"             "labs"
## [31] "telnet"          "num857"          "data"
## [34] "num415"          "num85"           "technology"
## [37] "num1999"         "parts"           "pm"
## [40] "direct"          "cs"              "meeting"
## [43] "original"        "project"         "re"
## [46] "edu"             "table"           "conference"
## [49] "charSemicolon"   "charRoundbracket" "charSquarebracket"
## [52] "charExclamation" "charDollar"       "charHash"
```

8/24

# Head

```
head(trainSpam)
```

```
##      make address all num3d our over remove internet order mail receive
## 1  0.00    0.64  0.64     0  0.32  0.00   0.00       0  0.00  0.00   0.00
## 7  0.00    0.00  0.00     0  1.92  0.00   0.00       0  0.00  0.64   0.96
## 9  0.15    0.00  0.46     0  0.61  0.00   0.30       0  0.92  0.76   0.76
## 12 0.00    0.00  0.25     0  0.38  0.25   0.25       0  0.00  0.00   0.12
## 14 0.00    0.00  0.00     0  0.90  0.00   0.90       0  0.00  0.90   0.90
## 16 0.00    0.42  0.42     0  1.27  0.00   0.42       0  0.00  1.27   0.00
##      will people report addresses free business email  you credit your font
## 1  0.64    0.00    0     0  0.32       0  1.29  1.93   0.00  0.96   0
## 7  1.28    0.00    0     0  0.96       0  0.32  3.85   0.00  0.64   0
## 9  0.92    0.00    0     0  0.00       0  0.15  1.23   3.53  2.00   0
## 12 0.12    0.12    0     0  0.00       0  0.00  1.16   0.00  0.77   0
## 14 0.00    0.90    0     0  0.00       0  0.00  2.72   0.00  0.90   0
## 16 0.00    0.00    0     0  1.27       0  0.00  1.70   0.42  1.27   0
##      num000 money hp hpl george num650 lab labs telnet num857 data num415
## 1      0  0.00  0  0     0     0  0  0       0  0  0.00   0
## 7      0  0.00  0  0     0     0  0  0       0  0  0.00   0
## 9      0  0.15  0  0     0     0  0  0       0  0  0.15   0
```

9/24

# Summaries

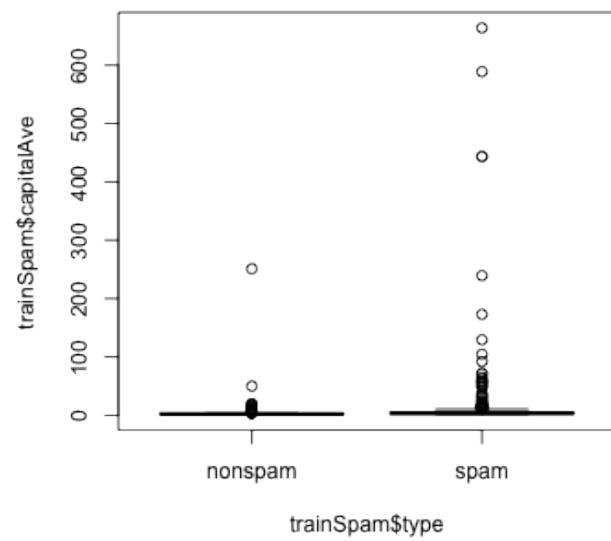
```
table(trainSpam$type)
```

```
##  
## nonspam     spam  
##    1381      906
```

10/24

# Plots

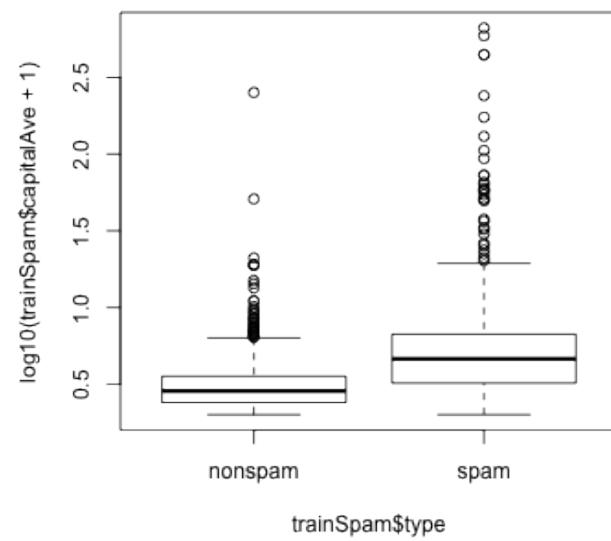
```
plot(trainSpam$capitalAve ~ trainSpam$type)
```



11/24

# Plots

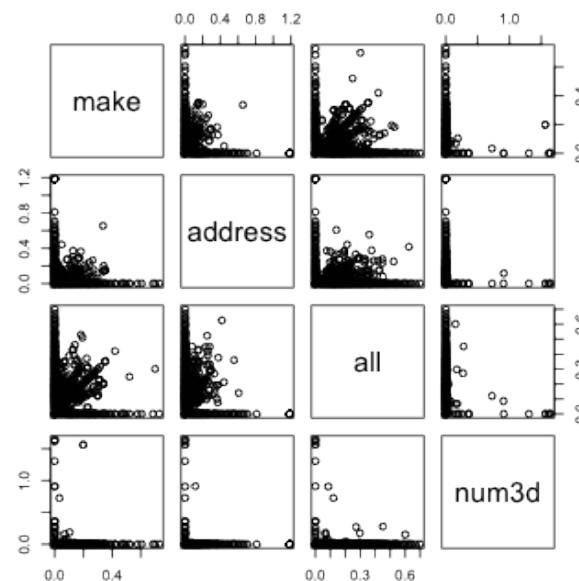
```
plot(log10(trainSpam$capitalAve + 1) ~ trainSpam$type)
```



12/24

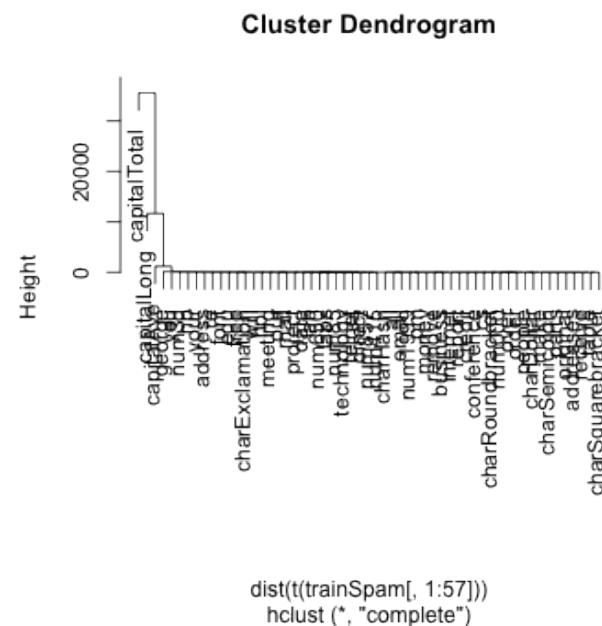
# Relationships between predictors

```
plot(log10(trainSpam[, 1:4] + 1))
```



# Clustering

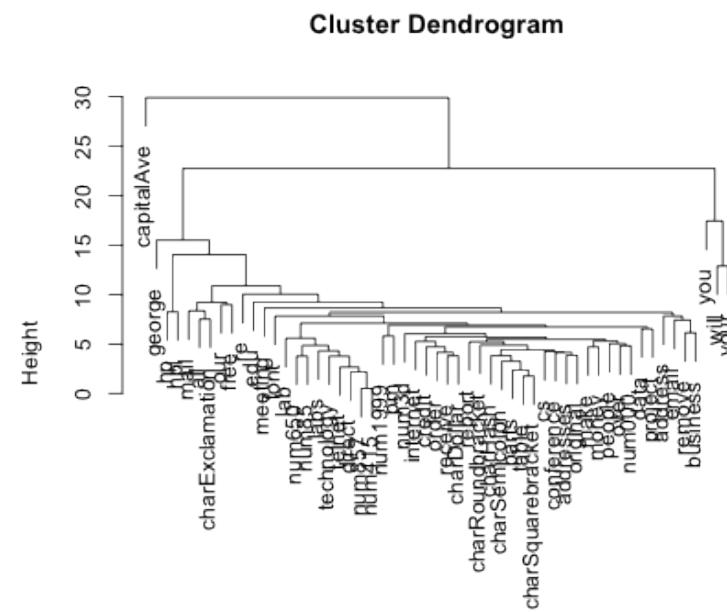
```
hCluster = hclust(dist(t(trainSpam[, 1:57])))  
plot(hCluster)
```



14/24

# New clustering

```
hClusterUpdated = hclust(dist(t(log10(trainSpam[, 1:55] + 1))))  
plot(hClusterUpdated)
```



```
dist(t(log10(trainSpam[, 1:55] + 1)))  
hclust (*, "complete")
```

15/24

# Statistical prediction/modeling

- Should be informed by the results of your exploratory analysis
- Exact methods depend on the question of interest
- Transformations/processing should be accounted for when necessary
- Measures of uncertainty should be reported

16/24

# Statistical prediction/modeling

```
trainSpam$numType = as.numeric(trainSpam$type) - 1
costFunction = function(x, y) {
  sum(x != (y > 0.5))
}
cvError = rep(NA, 55)
library(boot)
for (i in 1:55) {
  lmFormula = as.formula(paste("numType~", names(trainSpam)[i], sep = ""))
  glmFit = glm(lmFormula, family = "binomial", data = trainSpam)
  cvError[i] = cv.glm(trainSpam, glmFit, costFunction, 2)$delta[2]
}

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

17/24

# Get a measure of uncertainty

```
predictionModel = glm(numType ~ charDollar, family = "binomial", data = trainSpam)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
predictionTest = predict(predictionModel, testSpam)
predictedSpam = rep("nonspam", dim(testSpam)[1])
predictedSpam[predictionModel$fitted > 0.5] = "spam"
table(predictedSpam, testSpam$type)
```

```
##
## predictedSpam nonspam spam
##      nonspam    1346   458
##      spam        61   449
```

```
(61 + 458)/(1346 + 458 + 61 + 449)
```

18/24

# Interpret results

- Use the appropriate language
  - describes
  - correlates with/associated with
  - leads to/causes
  - predicts
- Give an explanation
- Interpret coefficients
- Interpret measures of uncertainty

# Our example

- The fraction of characters that are dollar signs can be used to predict if an email is Spam
- Anything with more than 6.6% dollar signs is classified as Spam
- More dollar signs always means more Spam under our prediction
- Our test set error rate was 22.4%

20/24

# Challenge results

- Challenge all steps:
  - Question
  - Data source
  - Processing
  - Analysis
  - Conclusions
- Challenge measures of uncertainty
- Challenge choices of terms to include in models
- Think of potential alternative analyses

# Synthesize/write-up results

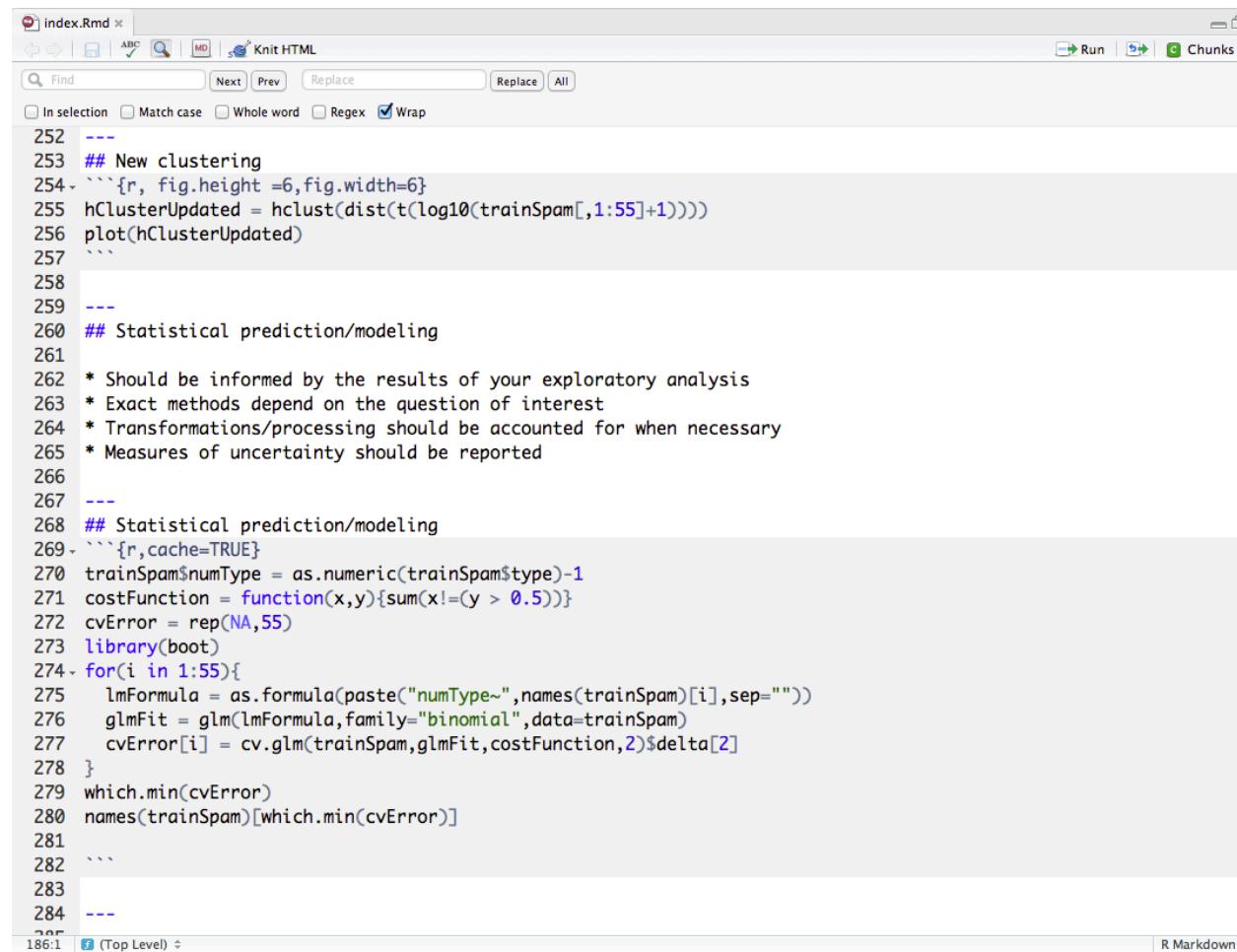
- Lead with the question
- Summarize the analyses into the story
- Don't include every analysis, include it
  - If it is needed for the story
  - If it is needed to address a challenge
- Order analyses according to the story, rather than chronologically
- Include "pretty" figures that contribute to the story

# In our example

- Lead with the question
  - Can I use quantitative characteristics of the emails to classify them as SPAM/HAM?
- Describe the approach
  - Collected data from UCI -> created training/test sets
  - Explored relationships
  - Choose logistic model on training set by cross validation
  - Applied to test, 78% test set accuracy
- Interpret results
  - Number of dollar signs seems reasonable, e.g. "Make money with Viagra \$ \$ \$ \$!"
- Challenge results
  - 78% isn't that great
  - I could use more variables
  - Why logistic regression?

23/24

# Create reproducible code



```
index.Rmd x
ABC MD Knit HTML
Find Next Prev Replace All
In selection Match case Whole word Regex Wrap
252 ---
253 ## New clustering
254 ````{r, fig.height =6,fig.width=6}
255 hClusterUpdated = hclust(dist(t(log10(trainSpam[,1:55]+1))))
256 plot(hClusterUpdated)
257 ```
258 ---
259 ---
260 ## Statistical prediction/modeling
261
262 * Should be informed by the results of your exploratory analysis
263 * Exact methods depend on the question of interest
264 * Transformations/processing should be accounted for when necessary
265 * Measures of uncertainty should be reported
266
267 ---
268 ## Statistical prediction/modeling
269 ````{r,cache=TRUE}
270 trainSpam$numType = as.numeric(trainSpam$type)-1
271 costFunction = function(x,y){sum(x!=y > 0.5)}
272 cvError = rep(NA,55)
273 library(boot)
274 for(i in 1:55){
275   lmFormula = as.formula(paste("numType~",names(trainSpam)[i],sep=""))
276   glmFit = glm(lmFormula,family="binomial",data=trainSpam)
277   cvError[i] = cv.glm(trainSpam,glmFit,costFunction,2)$delta[2]
278 }
279 which.min(cvError)
280 names(trainSpam)[which.min(cvError)]
281
282 ```
283 ---
284 ---
285
186:1 f (Top Level) R Markdown
```

24/24

# Summarizing data

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

# Why summarize?

- Data are often too big to look at the whole thing
- The first step in an analysis is to find problems
- When you do these summaries you should be looking for
  - Missing values
  - Values outside of expected ranges
  - Values that seem to be in the wrong units
  - Mislabeled variables/columns
  - Variables that are the wrong class

2/20

# Earthquake data

The screenshot shows a web browser window for the DATA.GOV website. The URL is <https://explore.data.gov/Geography-and-Environment/Worldwide-M1-Earthquakes-Past-7-Days/7tag-iwnu>. The page title is "Worldwide M1+ Earthquakes, Past 7 Days". The main content area displays a table with the following data:

| Category         | Value  |
|------------------|--------|
| Community Rating | ★★★★★  |
| Your Rating      | ★★★★★  |
| Raters           | 12     |
| Visits           | 179823 |
| Downloads        | 182476 |
| Comments         | 7      |
| Contributors     | 0      |

On the right side of the page, there is a sidebar with the heading "Data.gov Program Management Office" and the text "created Feb 17, 2011" and "updated Jan 09, 2013". There are also links for "External Link", "CSV 103KB", and "KML 12.2KB". The bottom of the page includes a navigation bar with links to Home, About, FAQ, Contact Info, Data Policy, Accessibility, Privacy Policy, and Sitemap.

<https://explore.data.gov/Geography-and-Environment/Worldwide-M1-Earthquakes-Past-7-Days/7tag-iwnu>

3/20

# Earthquake data

```
fileUrl <- "http://earthquake.usgs.gov/earthquakes/catalogs/eqs7day-M1.txt"
download.file(fileUrl, destfile = "./data/earthquakeData.csv", method = "curl")
dateDownloaded <- date()
dateDownloaded
```

```
[1] "Sun Jan 27 00:23:22 2013"
```

```
eData <- read.csv("./data/earthquakeData.csv")
```

4/20

# Looking at data - the whole thing

eData

|    | Src | Eqid     | Version | Datetime                              |
|----|-----|----------|---------|---------------------------------------|
| 1  | nc  | 71929481 | 1       | Sunday, January 27, 2013 05:03:01 UTC |
| 2  | ci  | 15278017 | 0       | Sunday, January 27, 2013 04:59:04 UTC |
| 3  | ak  | 10645573 | 1       | Sunday, January 27, 2013 04:55:09 UTC |
| 4  | nc  | 71929476 | 0       | Sunday, January 27, 2013 04:51:48 UTC |
| 5  | nn  | 00401016 | 9       | Sunday, January 27, 2013 04:45:19 UTC |
| 6  | ak  | 10645564 | 1       | Sunday, January 27, 2013 04:16:45 UTC |
| 7  | hv  | 60459531 | 2       | Sunday, January 27, 2013 04:15:57 UTC |
| 8  | ak  | 10645555 | 1       | Sunday, January 27, 2013 04:14:35 UTC |
| 9  | ci  | 15278009 | 0       | Sunday, January 27, 2013 04:07:44 UTC |
| 10 | us  | c000ewb3 | 7       | Sunday, January 27, 2013 04:05:42 UTC |
| 11 | ci  | 15278001 | 0       | Sunday, January 27, 2013 03:54:27 UTC |
| 12 | hv  | 60459521 | 1       | Sunday, January 27, 2013 03:50:13 UTC |
| 13 | hv  | 60459516 | 2       | Sunday, January 27, 2013 03:43:56 UTC |
| 14 | ak  | 10645533 | 1       | Sunday, January 27, 2013 03:25:17 UTC |
| 15 | ak  | 10645528 | 1       | Sunday, January 27, 2013 03:18:17 UTC |
| 16 | us  | c000ewax | 6       | Sunday, January 27, 2013 03:17:57 UTC |
| 17 | ci  | 15277993 | 0       | Sunday, January 27, 2013 02:47:04 UTC |

5/20

# Looking at data - dim(), names(), nrow(), ncol()

```
dim(eData)
```

```
[1] 1057    10
```

```
names(eData)
```

```
[1] "Src"        "Eqid"       "Version"     "Datetime"   "Lat"  
[6] "Lon"        "Magnitude"  "Depth"       "NST"        "Region"
```

```
nrow(eData)
```

```
[1] 1057
```

6/20

# Looking at the data - quantile(),summary()

```
quantile(eData$Lat)
```

```
0%      25%      50%      75%      100%
-61.30  35.56  38.77  52.58  67.66
```

```
summary(eData)
```

| Src           | Eqid           | Version      | Datetime                               | Lat             |
|---------------|----------------|--------------|--|-----------------|
| ak : 330      | 00400150:      | 1 2 : 379    | Monday, January 21, 2013 11:00:00 UTC: | 2 Min. : -61.3  |
| nc : 247      | 00400153:      | 1 0 : 195    |  |                 |
| ci : 145      | 00400155:      | 1 1 : 168    |  |                 |
| nn : 92       | 00400156:      | 1 9 : 97     |  |                 |
| us : 89       | 00400157:      | 1 3 : 82     |  |                 |
| pr : 40       | 00400159:      | 1 4 : 43     |  |                 |
| (Other) : 114 | (Other) : 1051 | (Other) : 93 | Friday, January 25, 2013 00:06:25 UTC: | 1 1st Qu.: 35.6 |

7/20

# Looking at data - class()

```
class(eData)
```

```
[1] "data.frame"
```

```
sapply(eData[1],class)
```

| Src       | Eqid      | Version  | Datetime | Lat       | Lon       | Magnitude |
|-----------|-----------|----------|----------|-----------|-----------|-----------|
| "factor"  | "factor"  | "factor" | "factor" | "numeric" | "numeric" | "numeric" |
| Depth     | NST       | Region   |          |           |           |           |
| "numeric" | "integer" | "factor" |          |           |           |           |

8/20

# Looking at data - unique(),length(),table()

```
unique(eData$Src)
```

```
[1] nc ci ak nn hv us pr uw nm mb uu  
Levels: ak ci hv mb nc nm nn pr us uu uw
```

```
length(unique(eData$Src))
```

```
[1] 11
```

```
table(eData$Src)
```

|  | ak  | ci  | hv | mb | nc  | nm | nn | pr | us | uu | uw |
|--|-----|-----|----|----|-----|----|----|----|----|----|----|
|  | 330 | 145 | 29 | 10 | 247 | 2  | 92 | 40 | 89 | 40 | 33 |

9/20

# Looking at data - table()

```
table(eData$Src,eData$Version)
```

|    | 0  | 1  | 2   | 3  | 4  | 5  | 6  | 7  | 8  | 9  | A | B | D | E |
|----|----|----|-----|----|----|----|----|----|----|----|---|---|---|---|
| ak | 0  | 93 | 211 | 26 | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 |
| ci | 64 | 0  | 67  | 7  | 3  | 3  | 1  | 0  | 0  | 0  | 0 | 0 | 0 | 0 |
| hv | 0  | 14 | 11  | 0  | 2  | 2  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 |
| mb | 0  | 0  | 10  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 |
| nc | 91 | 46 | 51  | 37 | 10 | 4  | 3  | 1  | 1  | 1  | 1 | 1 | 0 | 0 |
| nm | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 2 | 0 | 0 | 0 |
| nn | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 92 | 0 | 0 | 0 | 0 |
| pr | 40 | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 |
| us | 0  | 0  | 2   | 0  | 14 | 13 | 24 | 13 | 11 | 4  | 4 | 2 | 1 | 1 |
| uu | 0  | 0  | 15  | 6  | 14 | 3  | 2  | 0  | 0  | 0  | 0 | 0 | 0 | 0 |
| uw | 0  | 15 | 12  | 6  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 |

# Looking at data - any(), all()

```
eData$Lat[1:10]
```

```
[1] 38.83 36.04 65.23 39.56 37.26 62.10 19.41 63.51 32.91 -5.17
```

```
eData$Lat[1:10] > 40
```

```
[1] FALSE FALSE TRUE FALSE FALSE TRUE FALSE TRUE FALSE FALSE
```

```
any(eData$Lat[1:10] > 40)
```

```
[1] TRUE
```

11/20

# Looking at data - all()

```
eData$Lat[1:10] > 40
```

```
[1] FALSE FALSE TRUE FALSE FALSE TRUE FALSE TRUE FALSE FALSE
```

```
all(eData$Lat[1:10] > 40)
```

```
[1] FALSE
```

12/20

# Looking at subsets - &

```
eData[eData$Lat > 0 & eData$Lon > 0,c("Lat","Lon")]
```

|     | Lat    | Lon    |
|-----|--------|--------|
| 51  | 5.486  | 127.05 |
| 56  | 39.749 | 77.30  |
| 58  | 38.295 | 46.81  |
| 110 | 34.571 | 24.10  |
| 129 | 51.130 | 179.35 |
| 134 | 9.438  | 126.10 |
| 146 | 38.426 | 73.36  |
| 153 | 49.728 | 155.69 |
| 155 | 43.337 | 18.77  |
| 160 | 29.379 | 132.20 |
| 175 | 44.280 | 10.53  |
| 193 | 31.763 | 50.95  |
| 239 | 4.998  | 95.96  |
| 325 | 53.564 | 142.75 |
| 348 | 38.608 | 73.49  |
| 359 | 27.771 | 56.41  |
| 385 | 49.825 | 87.60  |

13/20

# Looking at subsets - |

```
eData[eData$Lat > 0 | eData$Lon > 0,c("Lat","Lon")]
```

|    | Lat     | Lon     |
|----|---------|---------|
| 1  | 38.8292 | -122.81 |
| 2  | 36.0403 | -117.35 |
| 3  | 65.2271 | -149.51 |
| 4  | 39.5573 | -121.99 |
| 5  | 37.2587 | -114.07 |
| 6  | 62.1046 | -150.70 |
| 7  | 19.4065 | -155.26 |
| 8  | 63.5132 | -150.83 |
| 9  | 32.9112 | -116.25 |
| 10 | -5.1704 | 102.94  |
| 11 | 35.5633 | -118.53 |
| 12 | 19.2960 | -155.38 |
| 13 | 19.9262 | -155.54 |
| 14 | 62.1638 | -149.58 |
| 15 | 63.2917 | -149.24 |
| 16 | 34.2925 | -106.71 |
| 17 | 33.6293 | -116.69 |

14/20

# Peer review experiment data

- Data on submissions/reviews in an experiment

The screenshot shows a web browser displaying a PLOS ONE article. The URL in the address bar is [www.plosone.org/article/info:doi/10.1371/journal.pone.0026895](http://www.plosone.org/article/info:doi/10.1371/journal.pone.0026895). The page features a banner advertisement for 'Simplify your research with automatic and continuous dosing'. The main header includes the PLOS ONE logo, navigation links for 'Articles', 'For Authors', 'About Us', and a search bar. Below the header, article statistics are displayed: 6,497 views, 2 citations, 61 academic bookmarks, and 108 social shares. The title of the article is 'Cooperation between Referees and Authors Increases Peer Review Accuracy' by Jeffrey T. Leek, Margaret A. Taub, and Fernando J. Pineda. The article summary section includes a diagram illustrating the peer review process, showing 'Closed/Private' and 'Open/Public' review types. To the right, there are buttons for 'Download', 'Print', and 'Share', and a 'Comments' section which lists 'Media Coverage of This Article' posted by 'PLoS\_ONE\_Group'.

<http://www.plosone.org/article/info:doi/10.1371/journal.pone.0026895>

# Peer review data

```
fileUrl1 <- "https://dl.dropbox.com/u/7710864/data/reviews-apr29.csv"
fileUrl2 <- "https://dl.dropbox.com/u/7710864/data/solutions-apr29.csv"
download.file(fileUrl1, destfile = "./data/reviews.csv", method = "curl")
download.file(fileUrl2, destfile = "./data/solutions.csv", method = "curl")
reviews <- read.csv("./data/reviews.csv"); solutions <- read.csv("./data/solutions.csv")
head(reviews, 2)
```

```
  id solution_id reviewer_id      start      stop time_left accept
1  1          3    1304095698 1304095758       1754      1
2  2          4    1304095188 1304095206       2306      1
```

```
head(solutions, 2)
```

```
  id problem_id subject_id      start      stop time_left answer
1  1        156    1304095119 1304095169      2343      B
2  2        269    1304095119 1304095183      2329      C
```

16/20

# Find if there are missing values - `is.na()`

```
is.na(reviews$time_left[1:10])
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE
```

```
sum(is.na(reviews$time_left))
```

```
[1] 84
```

```
table(is.na(reviews$time_left))
```

|     | FALSE | TRUE |
|-----|-------|------|
| 115 | 84    |      |

17/20

# Important table()/NA issue

```
table(c(0,1,2,3,NA,3,3,2,2,3))
```

```
0 1 2 3  
1 1 3 4
```

```
table(c(0,1,2,3,NA,3,3,2,2,3),useNA="ifany")
```

```
0      1      2      3 <NA>  
1      1      3      4      1
```

18/20

# Summarizing columns/rows - rowSums(),rowMeans(),colSums(),colMeans()

- Important parameters: *x, na.rm*

```
colSums(reviews)
```

|                        | <code>id</code> | <code>solution_id</code> | <code>reviewer_id</code> | <code>start</code> | <code>stop</code> |
|------------------------|-----------------|--------------------------|--------------------------|--------------------|-------------------|
|                        | 19900           | 19929                    | 5064                     | NA                 | NA                |
| <code>time_left</code> |                 | accept                   |                          |                    |                   |
|                        | NA              |                          | NA                       |                    |                   |

19/20

# Summarizing columns/rows - rowSums(),rowMeans(),colSums(),colMeans()

```
colMeans(reviews,na.rm=TRUE)
```

```
    id solution_id reviewer_id      start      stop
1.000e+02  1.001e+02   2.545e+01  1.304e+09  1.304e+09
time_left     accept
1.114e+03  6.435e-01
```

```
rowMeans(reviews,na.rm=TRUE)
```

```
[1] 3.726e+08 3.726e+08 3.726e+08 3.726e+08 3.726e+08 3.726e+08
[7] 3.726e+08 1.300e+01 3.726e+08 3.726e+08 3.726e+08 3.726e+08
[13] 3.726e+08 3.726e+08 3.726e+08 3.726e+08 1.967e+01 3.726e+08
[19] 3.726e+08 1.933e+01 3.726e+08 3.726e+08 3.726e+08 2.433e+01
[25] 2.367e+01 2.367e+01 3.726e+08 3.726e+08 3.726e+08 3.726e+08
[31] 3.726e+08 3.726e+08 3.726e+08 3.726e+08 3.133e+01 3.726e+08
[37] 3.267e+01 3.726e+08 3.400e+01 3.726e+08 3.200e+01 3.726e+08
```

20/20

# Types of Data Analysis Questions

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

# Types of Data Analysis 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

Screenshot of the 2010 Census homepage ([www.census.gov/2010census/](http://www.census.gov/2010census/))

The page features a navigation bar with links to "2010 Census Home", "Press & Media", "Partners", "ABOUT", "DATA", "CONNECT", and "MULTIMEDIA".

A main section titled "A Look at Your Community" displays a map of Alabama's Congressional Districts (AL - Congressional District) with population density shading. A callout box highlights the "TOTAL POPULATION" legend, showing ranges from 0 to 720,000. Another callout box shows racial demographic data for the district.

Below this, there are four numbered buttons (1, 2, 3, 4) and a "See More" link.

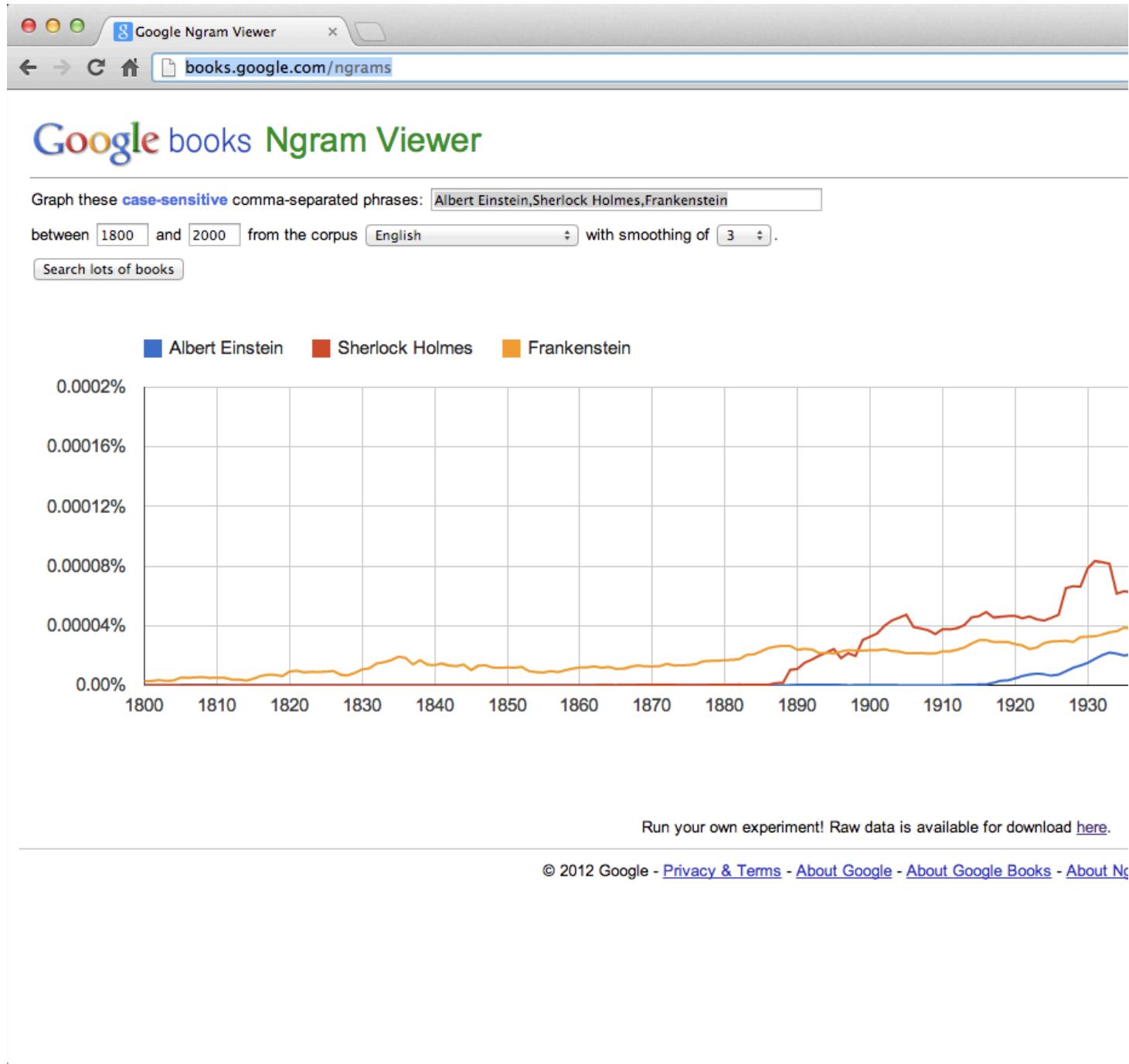
On the left, there are three sections: "Population Finder" (with a dropdown menu for selecting a state), "Interactive Map" (with a link to explore 2010 Census statistics), and "Census Briefs and Reports".

On the right, there is a detailed "2010 Census: District of Columbia Profile" section titled "Population by Sex and Age". It includes a map of the District of Columbia and a population pyramid chart showing the distribution of the population by sex and age groups (e.g., 85+ Years).

[www.census.gov/2010census/](http://www.census.gov/2010census/)

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

# Descriptive analysis



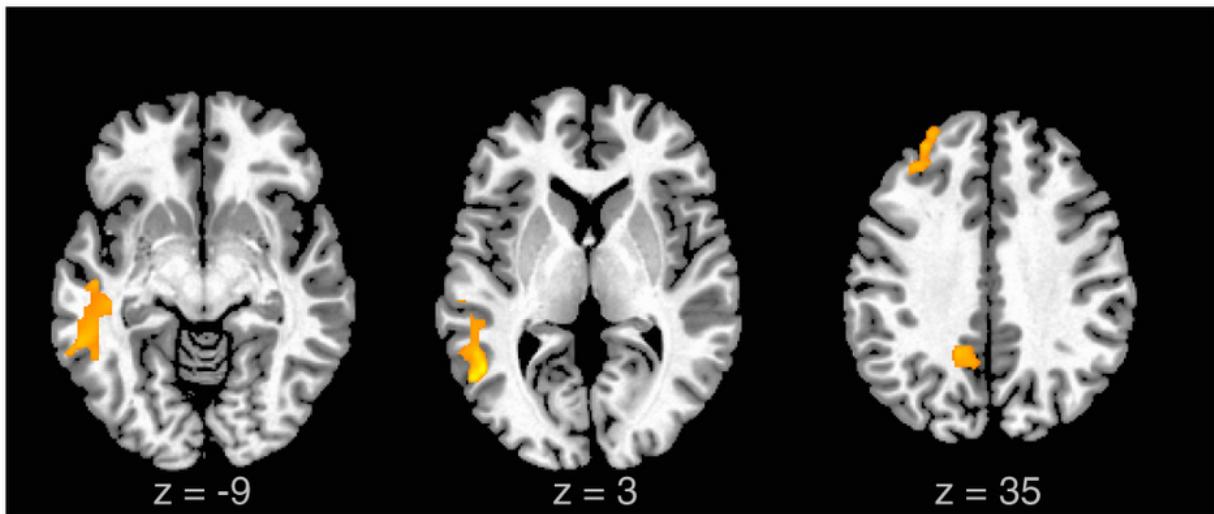
<http://books.google.com/ngrams>

# 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

The screenshot shows the homepage of the Sloan Digital Sky Survey (SDSS). The header features the SDSS logo and the text "Sloan Digital Sky Survey" and "Mapping the Universe". A sidebar on the left contains links to various sections: Home, SDSS-III, SDSS Data DR9, SDSS Data DR8, SDSS Data DR7, Science, Press Releases, Education, Image Gallery, Legacy Survey, SEGUE, Supernova Survey, Collaboration, Publications, Contact Us, and Search. The main content area is titled "The Sloan Digital Sky Survey" and describes the survey's history, data releases, and current status. It mentions the survey's deep, multi-color images covering more than a quarter of the sky, containing over 930,000 galaxies and 120,000 quasars. It also discusses the Third Sloan Digital Sky Survey (SDSS-III), which began in 2008 and includes BOSS spectroscopy. The page also highlights the largest color image of the sky ever made and the enormous data flow from the telescope.

**The Sloan Digital Sky Survey**

The Sloan Digital Sky Survey (SDSS) is one of the most ambitious and influential surveys in the history of astronomy. Over 2000-2005; SDSS-II, 2005-2008), it obtained deep, multi-color images covering more than a quarter of the sky and created than 930,000 galaxies and more than 120,000 quasars.

SDSS data have been released to the scientific community and the general public in annual increments, with the final public in October 2008. That release, [Data Release 7](#), is available through this website.

Meanwhile, SDSS is continuing with the [Third Sloan Digital Sky Survey \(SDSS-III\)](#), a program of four new surveys using observations in July 2008 and released [Data Release 8](#) in January 2011 and [Data Release 9](#) in August 2012. SDSS-III will continue through 2014.

[Data Release 9](#) contains the first release of BOSS spectroscopy to the public as well as several significant updates to the current data release.

[Data Release 8](#) contains all images from the SDSS telescope - [the largest color image of the sky ever made](#). It also includes stars and galaxies, and spectra of nearly two million. All the images, measurements, and spectra are available free online. You can look up [data for individual objects](#), or [search for objects](#) anywhere in the sky based on any criteria.

The SDSS used a dedicated 2.5-meter telescope at Apache Point Observatory, New Mexico, equipped with two powerful 12-megapixel camera imaged 1.5 square degrees of sky at a time, about eight times the area of the full moon. A pair of spectrographs spectra of (and hence distances to) more than 600 galaxies and quasars in a single observation. A custom-designed set of software enabled the enormous data flow from the telescope. The two key technologies that enabled the SDSS, optical fibers and the digital imaging system, were awarded the [2009 Nobel Prize in Physics](#).

During its first phase of operations, 2000-2005, the SDSS imaged more than 8,000 square degrees of the sky in five optical bands and selected from 5,700 square degrees of that imaging. It also obtained repeated imaging (roughly 30 scans) of the southern Galactic cap.

With new financial support and an expanded collaboration including 25 institutions around the globe, SDSS-II carried out the following:

- [The Sloan Legacy Survey](#) completed the original SDSS imaging and spectroscopic goals. The final dataset includes 2,200 square degrees of imaging and spectra of 930,000 galaxies, 120,000 quasars, and 225,000 stars.
- [SEGUE](#) (the Sloan Extension for Galactic Understanding and Exploration) probed the structure and history of the Milky Way.

<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 States: An Analysis of 545 U.S. Counties from 1990 to 2007

Correia, Andrew W.<sup>a</sup>; Pope, C. Arden III<sup>b</sup>; Dockery, Douglas W.<sup>b</sup>; Schwartz, James D.<sup>c</sup>; Francesca, Barbara A.<sup>d</sup>

**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

## Five Thirty Eight Forecast

Updated 10:10 AM ET on Nov. 6

**President**  
Nov. 6 Forecast

President  
Now-cast

Senate  
Nov. 6 Forecast

Barack Obama

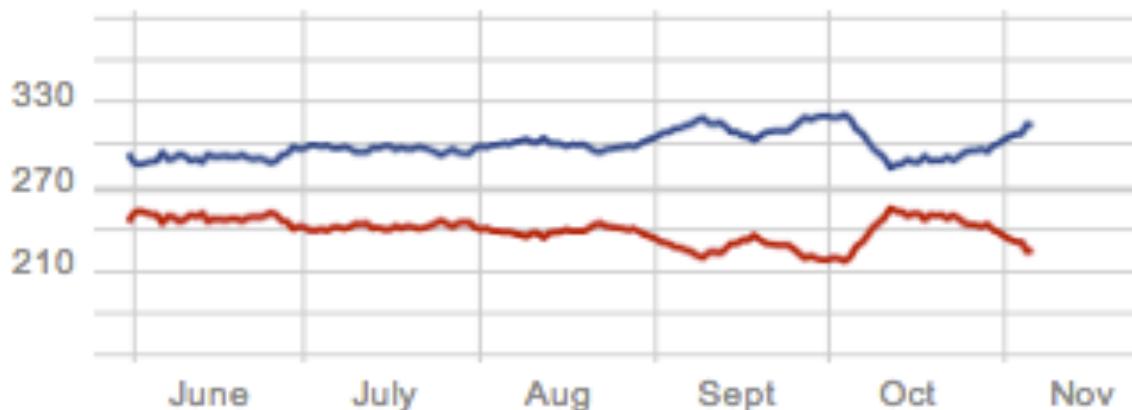
Mitt Romney

**313.0**  
+14.0 since Oct. 30

**Electoral  
vote**

**225.0**  
-14.0 since Oct. 30

270 to win



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

# Predictive analysis

FREE REPORT: Top 10 Stocks for 2013

**Forbes** • **New Posts** • **Most Popular**  
Best Cover Letter Ever? • **Lists** • 30 Under 30

62.8k  
[f Share](#)

13.7k  
[Tweet](#)

5.6k  
[in Share](#)

353  
[Submit](#)

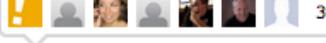
3.5k  
[g +1](#)

1.9k  
[reddit](#)

 **Kashmir Hill**, Forbes Staff  
Welcome to The Not-So Private Parts where technology & privacy collide  
[+ Follow](#) (1,089) [Follow](#) 174k

TECH | 2/16/2012 @ 11:02AM | 1,913,626 views

## How Target Figured Out A Teen Girl Was Pregnant Before Her Father Did

 307 comments, 167 called-out [+ Comment Now](#) [+ Follow Comments](#)

Every time you go shopping, you share intimate details about your consumption patterns with retailers. And many of those retailers are studying those details to figure out what you like, what you need, and which coupons are most likely to make you happy. Target, for example, has figured out how to data-mine its way into your womb, to figure out whether you have a baby on the way long before you need to start buying diapers.



<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

[HOME](#)
[ARTICLES & MULTIMEDIA](#)
[ISSUES](#)
[SPECIALTIES & TOPICS](#)
[FOR AUTHORS](#)
[CME](#)

### 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 I. 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](#).

[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



## 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) has established the Design Guide Implementation Team (DGIT) consisting of FHWA division offices, State highway departments, members of the AASHTO Pavement Committee, and other organizations and experts to assist in the development of the upcoming guide and to help potential users implement the guide. To introduce the guide and to discuss implementation issues, the DGIT has developed a one-day workshop series. Seven of these workshops will be held across the country starting on May 25, 2004, in Biloxi, MS. The remaining four workshops will be held in Vancouver, WA (June); Inland Empire, CA (July); Hawaii (July); Mystic, CT (August); and Kansas City, KS (September); and Phoenix, AZ (October).

[http://www.fhwa.dot.gov/resourcecenter/teams/pavement/pave\\_3pdg.pdf](http://www.fhwa.dot.gov/resourcecenter/teams/pavement/pave_3pdg.pdf)



# What is data?

Jeffrey Leek, Assistant Professor of Biostatistics  
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

# 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](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

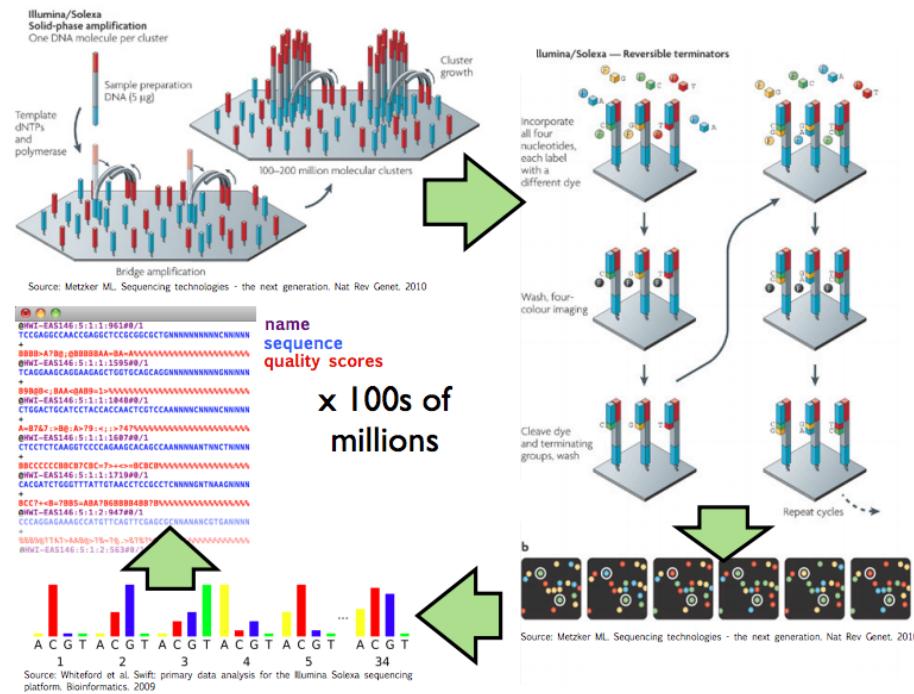
[http://en.wikipedia.org/wiki/Computer\\_data\\_processing](http://en.wikipedia.org/wiki/Computer_data_processing)

# An example of a processing pipeline



[http://www.illumina.com.cn/support/sequencing/sequencing\\_instruments/hiseq\\_1000.asp](http://www.illumina.com.cn/support/sequencing/sequencing_instruments/hiseq_1000.asp)

# An example of a processing pipeline



[http://www.cbcn.umd.edu/~hcorrada/CMSC858B/lectures/lect22\\_seqIntro/seqIntro.pdf](http://www.cbcn.umd.edu/~hcorrada/CMSC858B/lectures/lect22_seqIntro/seqIntro.pdf)

# What do raw data look like?

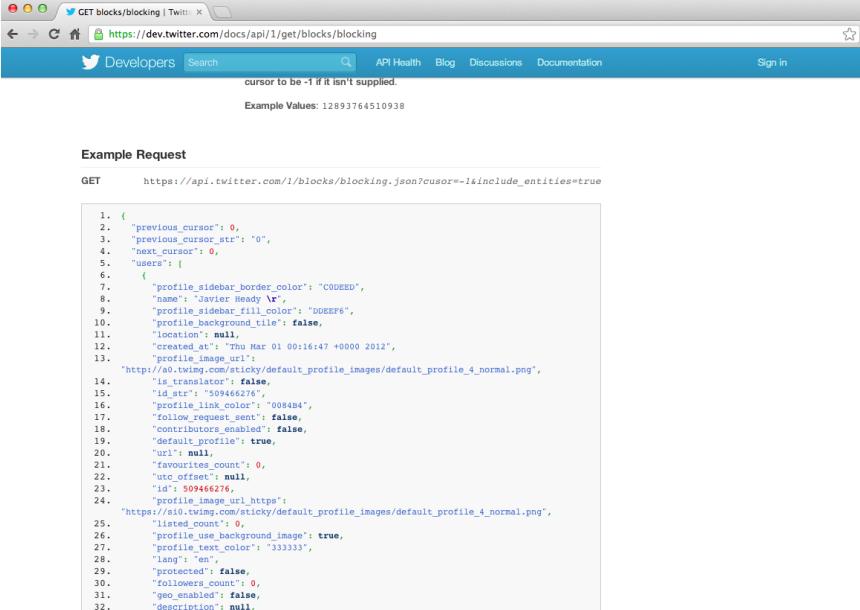
```

@HWI-EAS121:4:100:1783:550#0/1
CGTTACGAGATCGGAAGAGCGGGTTCAGCAGGAATGCCGAGACGGATCTGTATGCCGTCTGCTCCGTGACAAGACAGGG
+HWI-EAS121:4:100:1783:550#0/1
aaaaaa`b_aa`aa`YaX]aZ`aZM^Z]YRa]YSG[[ZREQLHESDHNDHNMEEDDM PENITKFLFEEDDDHEJQMEDDD
@HWI-EAS121:4:100:1783:1611#0/1
GGGTGGGCATTTCACACTCGCAGTATGGGTTGCCGCACGACAGGCAGCGGTCA GCGCTTGGCCTGGCCTTCGGAAA
+HWI-EAS121:4:100:1783:1611#0/1
a``^`_ ``^a``^a_`_ja_]`a____`_ ``^`]X_]XTV_\]NX_XVX]]_TTTG[VTHPN]VFDZ
@HWI-EAS121:4:100:1783:322#0/1
CGTTTATGTTTTGAATATGTCTTATCTAACGGTTATTTAGATGTTGGTCTTATTCTAACGGTCATATATTTCTA
+HWI-EAS121:4:100:1783:322#0/1
abaa``^aaaaabbbaaabbbbbbb`bbbb_bbbbbbbb`bbbaV^_a``a``]``aT]a__V\]]_]^a`]a_abbaV_
@HWI-EAS121:4:100:1783:1394#0/1
GGGTCTTATTGGCTCTGGTGATCCCCCATATTCTCCGGTTGTGGTTAACCGATCATCGCGCATTACTCCGGCTGC
+HWI-EAS121:4:100:1783:1394#0/1
````[aa\b``^`]aabbb][`a_abbb`a``bbbbbabaaaab_Vza_``_bab_X`[a\HV[_]_[^_X\T_VQQ
@HWI-EAS121:4:100:1783:207#0/1
CCCTGGGAGATCGGAAGAGCGGGTTCAGCAGGAATGCCGAGACCGATCTGTATGCCGTCTCTGTTGAAAAAAACAA
+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
GGGTAATTCAAGGGACAATGTAATGGCTGCACAAAAAAATACATCTTCATGTTCCATTGCACCATTGACAAATACATATT
+HWI-EAS121:4:100:1783:455#0/1
abb_babbabaabbbbbbbbbbba``b`\abbabbabbabbbaabbbbb`bb`ab_O_bab_Q_bbabaa_a

```

[http://brianknaus.com/software/srtoolbox/s\\_4\\_1\\_sequence80.txt](http://brianknaus.com/software/srtoolbox/s_4_1_sequence80.txt)

# What do raw data look like?



The screenshot shows a web browser window with the URL [https://dev.twitter.com/docs/api/1/get\(blocks/blocking](https://dev.twitter.com/docs/api/1/get(blocks/blocking)). The page title is "GET blocks/blocking | Twitter". The main content area displays an "Example Request" for a GET request to [https://api.twitter.com/1/blocks/blocking.json?cursor=-1&include\\_entities=true](https://api.twitter.com/1/blocks/blocking.json?cursor=-1&include_entities=true). Below the URL, there is a large block of JSON code representing the response from the API. The code is as follows:

```
1. {
2.   "previous_cursor": 0,
3.   "previous_cursor_str": "0",
4.   "next_cursor": 0,
5.   "users": [
6.     {
7.       "profile_sidebar_border_color": "CODEED",
8.       "name": "Twitter Meeny",
9.       "profile_sidebar_fill_color": "DDEEF6",
10.      "profile_background_tile": false,
11.      "location": null,
12.      "created_at": "Thu Mar 01 00:16:47 +0000 2012",
13.      "profile_image_url": "http://si0.twimg.com/sticky/default_profile_images/default_profile_4_normal.png",
14.      "id": 509466276,
15.      "id_str": "509466276",
16.      "profile_link_color": "008484",
17.      "follow_request_sent": false,
18.      "contributors_enabled": false,
19.      "default_profile": true,
20.      "url": null,
21.      "favourites_count": 0,
22.      "utc_offset": null,
23.      "id": 509466276,
24.      "profile_image_url_https": "https://si0.twimg.com/sticky/default_profile_images/default_profile_4_normal.png",
25.      "list_id": 0,
26.      "profile_use_background_image": true,
27.      "profile_text_color": "333333",
28.      "lang": "en",
29.      "protected": false,
30.      "followers_count": 0,
31.      "geo_enabled": false,
32.      "description": null,
```

[https://dev.twitter.com/docs/api/1/get\(blocks/blocking](https://dev.twitter.com/docs/api/1/get(blocks/blocking)

# What do raw data look like?

| ALLERGIES                        |                                                    | MEDICATION HISTORY               |                                                                      |
|----------------------------------|----------------------------------------------------|----------------------------------|----------------------------------------------------------------------|
| Last Updated: 01 Dec 2011 @ 0851 |                                                    | Last Updated: 11 Apr 2011 @ 1737 |                                                                      |
| Allergy Name:                    | TRIMETHOPRIM                                       | Medication:                      | AMLODIPINE BESYLATE 10MG TAB                                         |
| Location:                        | DAYT29                                             | Instructions:                    | TAKE ONE TABLET BY MOUTH TAKE ONE-HALF TABLET FOR GRAPEFRUIT JUICE-- |
| Date Entered:                    | 09 Mar 2011                                        | Status:                          | Active                                                               |
| Action:                          |                                                    | Refills Remaining:               | 3                                                                    |
| Allergy Type:                    | DRUG                                               | Last Filled On:                  | 20 Aug 2010                                                          |
| A Drug Class:                    | ANTI-INFECTIVES, OTHER                             | Initially Ordered On:            | 13 Aug 2010                                                          |
| Observed/Historical:             | HISTORICAL                                         | Quantity:                        | 45                                                                   |
| Comments:                        | The reaction to this allergy was MILD (NO SQUELAE) | Days Supply:                     | 90                                                                   |
| Allergy Name:                    | TRAMADOL                                           | Pharmacy:                        | DAYTON                                                               |
| Location:                        | DAYT29                                             | Prescription Number:             | 2718953                                                              |
| Date Entered:                    | 09 Mar 2011                                        |                                  |                                                                      |
| Action:                          | URINARY RETENTION                                  |                                  |                                                                      |
| Allergy Type:                    | DRUG                                               |                                  |                                                                      |
| A Drug Class:                    | NON-OPIOID ANALGESICS                              |                                  |                                                                      |
| Observed/Historical:             | HISTORICAL                                         |                                  |                                                                      |
| Comments:                        | gradually worsening difficulty emptying bladder    |                                  |                                                                      |

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

# What do processed data look like?

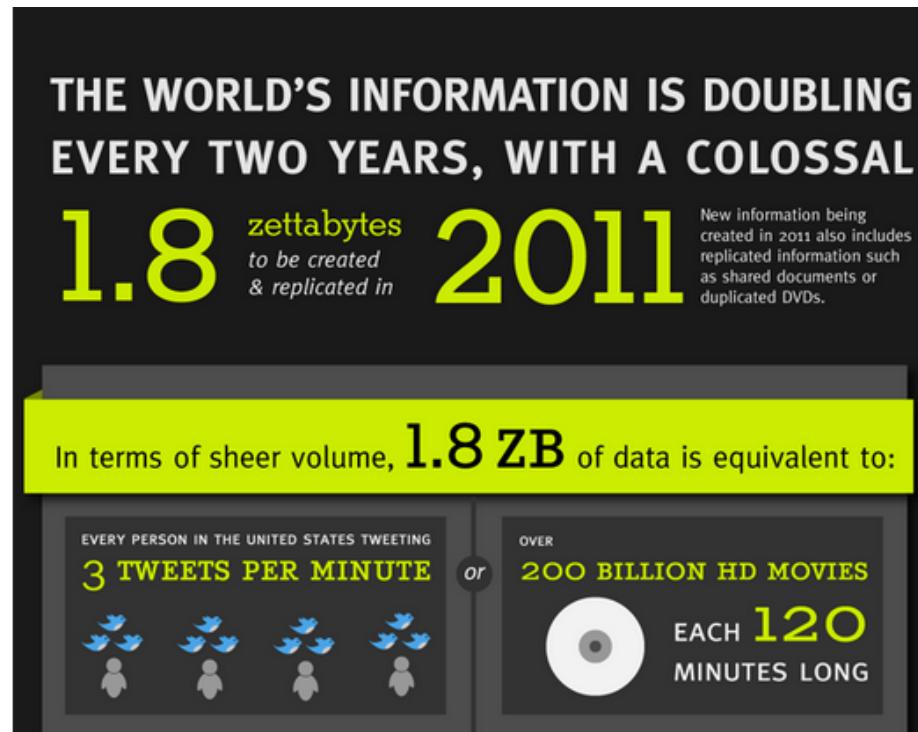
|    | A  | B          | C          | D          | E          | F         | G      | H | I | J | K | L | M | N | O | P |
|----|----|------------|------------|------------|------------|-----------|--------|---|---|---|---|---|---|---|---|---|
| 1  | id | problem_id | subject_id | start      | stop       | 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  | 142        | 13         | 1307119995 | 1307120019 | 2360      | B      |   |   |   |   |   |   |   |   |   |
| 6  | 5  | 273        | 14         | 1307119996 | 1307120008 | 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 | 1307120052 | 2293      | B      |   |   |   |   |   |   |   |   |   |
| 12 | 11 | 494        | 17         | 1307120013 | 1307120075 | 2310      | D      |   |   |   |   |   |   |   |   |   |
| 13 | 12 | 357        | 13         | 1307120021 | 1307120118 | 2367      | 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         | 1307120080 | 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         | 1307120164 | 1307120181 | 2090      | D      |   |   |   |   |   |   |   |   |   |
| 25 | 24 | 121        | 19         | 1307120252 | 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         | 1307120311 | 1307120368 | 2000      | B      |   |   |   |   |   |   |   |   |   |
| 31 | 30 | 502        | 16         | 1307120322 | 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         | 1307120371 | 1307120397 | 1984      | B      |   |   |   |   |   |   |   |   |   |
| 37 | 36 | 995        | 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      |   |   |   |   |   |   |   |   |   |

1. Each variable forms a column
2. Each observation forms a row
3. Each table/file stores data about one kind of observation (e.g. people/hospitals).

<http://vita.had.co.nz/papers/tidy-data.pdf>

Leek, Taub, and Pineda 2011 PLoS One

# How much is there?

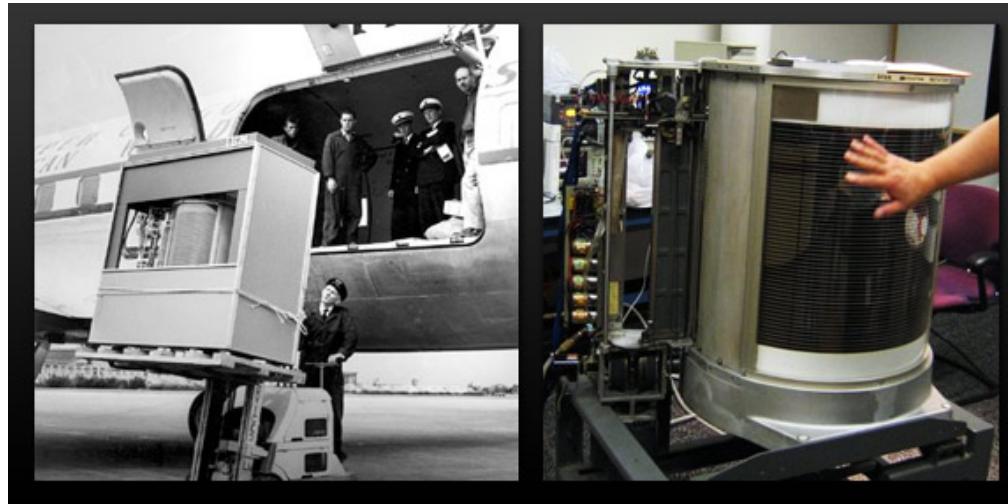


<http://mashable.com/2011/06/28/data-infographic/>

# 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, rather than the actions and characteristics of individuals. The dataset contains summary properties of 30 billion conversations among 240 million people. From the data, we construct a communication graph with 180 million nodes and 1.3 billion undirected edges, creating the largest 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 claim that people are separated by ``six degrees of separation'' and find that the average path length among Messenger users is 6.6. We also find that people tend to communicate more with each other when they have similar age, language, and location, and that 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

“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