1 Introduction

1.1 Concepts

- Summary Statistics and Exploratory Data Analysis
- Distributions
- Sampling and asymptotics
  - Probability and Random Variables
  - CLT
  - Estimates and Standard Errors
  - Confidence intervals
  - Law of rare events
- Testing
  - Hypergeometric
  - p-values
- Two variables
  - Bivariate normal
  - Correlation
  - Confounding
  - Conditional expectation
  - Regression
- Experiments and ANOVA
  - Explain observational versus experiment
- Modeling
  - Poisson
  - GLM
  - MLE
  - Mixture models
  - Smoothing

1.2 Datasets

- Maternal Smoking and infant health (Chapter 1)
- Who plays video games (Chapter 2)
- DNA patterns (Modeling, Discovery, Testing)
- Election Data (Sampling, ANOVA, smoothing)
- A Mouse Model for Down Syndrome (ANOVA)
- Baseball Standings (Modeling, Simulation, Prediction)
- Genotype calling (Mixture model)
- Graduate School admissions (Confounding)
- Paper helicopter design (experimental design)
- Regression example (confounding)

Before we start, set up some global R graphics parameters

```
> library(RColorBrewer)
> set.seed(1)
> par(mar = c(2.5, 2.5, 1.6, 1.1), mgp = c(1.5, 0.5, 0))
> palette(brewer.pal(8, "Dark2"))
> datadir = "http://www.biostat.jhsph.edu/bstcourse/bio751/data"
```

2 Summary statistics and distributions

A distribution describes a list of numbers

Examples: 2,3,3,6,7

We can define the distribution function: the proportion of each outcome. For example,

$$\Pr(X = 3) = 2/5$$

We use X to denote an arbitrary member and Pr to denote proportion. Later this will denote probability. We use N to denote the number of unique elements and use $x_1, \ldots, x_n$ to denote the possible outcomes.

Example. Let's look at Heights.

```
> dat = read.csv(file.path(datadir, "USheights_subsample.csv"))
> x = dat$Height[dat$Gender == 1]
> length(x)
```

```
[1] 12621
```

How do we summarize these numbers? How do we explain to a martian the height of US men? What should they expect?

Note, the measurements are so precise that they are all unique (this data is artificial since height is never this precise)

```
> length(unique(x))
```

```
[1] 12621
```

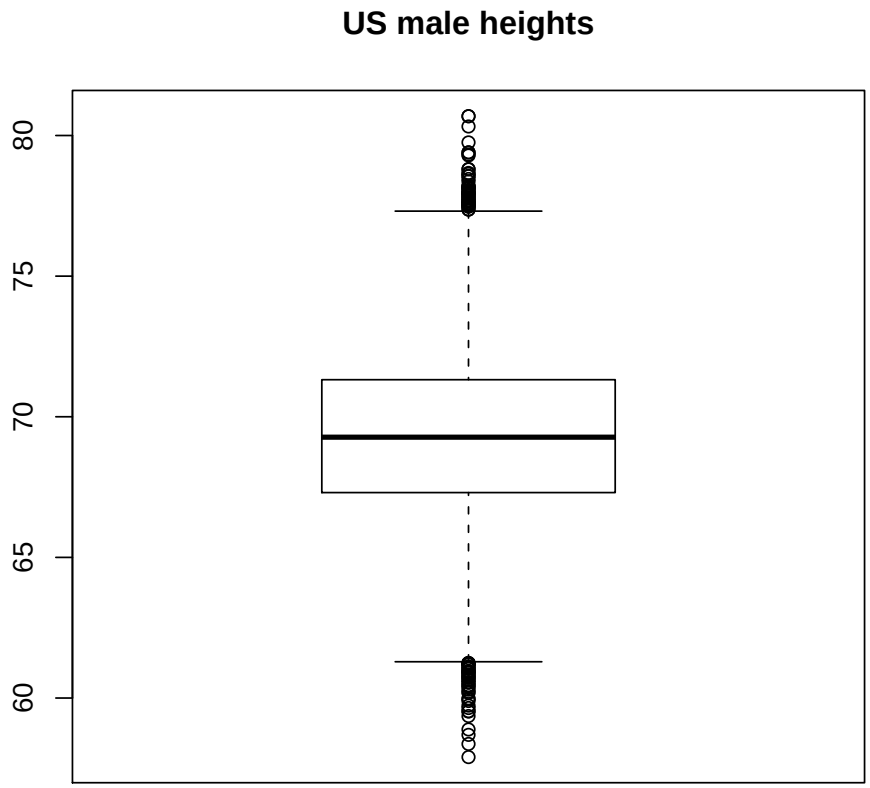
So we plot the proportion of each outcome we simply get a bunch of $1/N$. Not very informative. We might as well report the numbers.

A popular 5 number summary are: smallest, 25-th percentile, 50-th percentile or median, 75-th percentile and largest. The box and whiskers plot, or boxplot for short, shows these graphically.

```
> quantile(x, c(0:4)/4)
```

```
      0%      25%      50%      75%     100%
57.90035 67.30421 69.27401 71.31775 80.69008
```

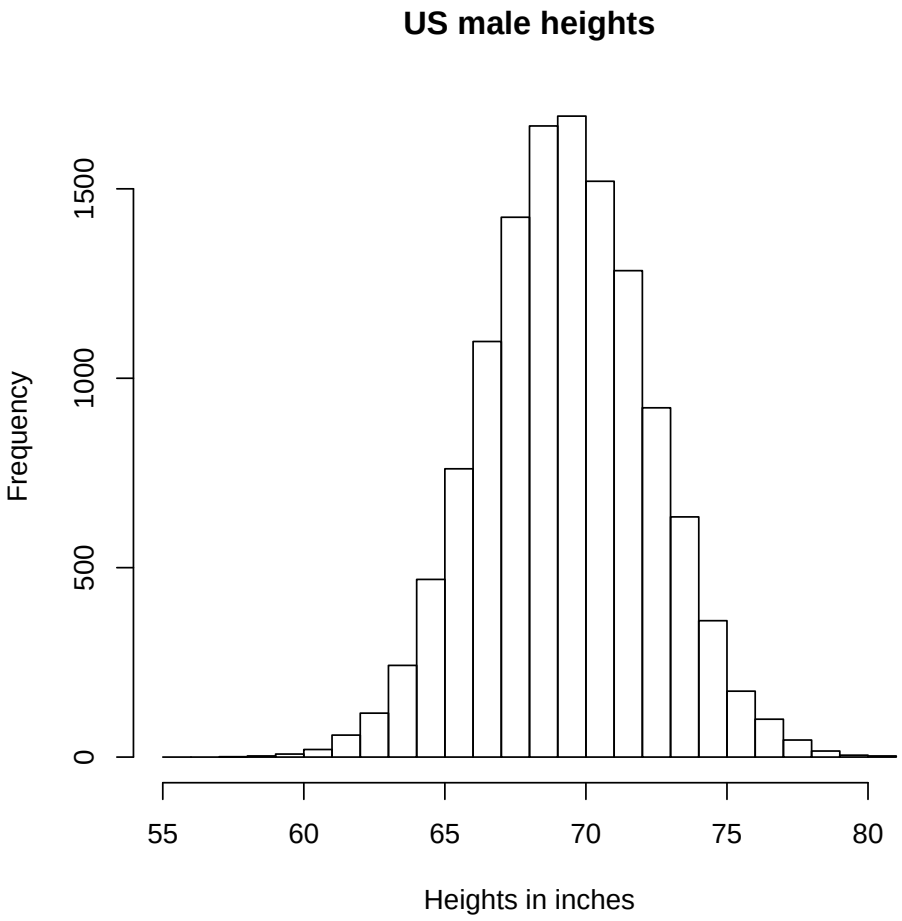
```
> boxplot(x, main = "US male heights")
```



The boxplot defines outliers, for more on that read the help file.

Another useful plot is the histogram. One simply defines intervals and counts the number of outcomes in each.

```
> hist(x, main = "US male heights", xlab = "Heights in inches",
+     breaks = seq(55, 81))
```



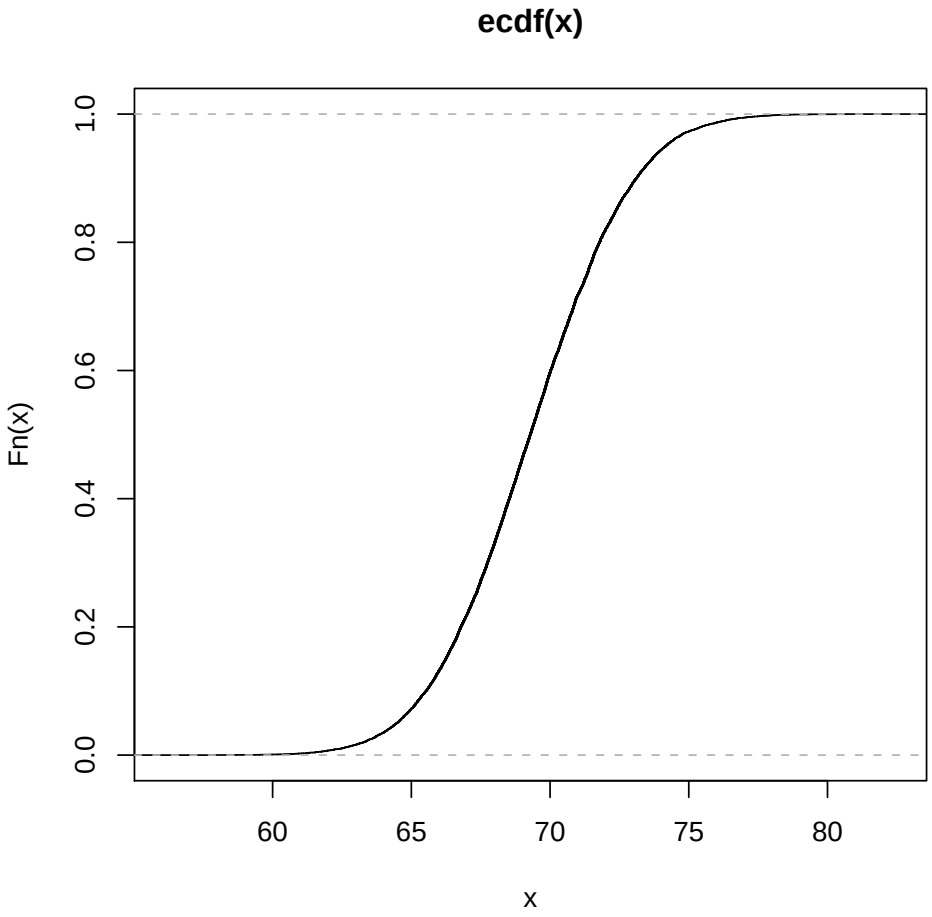
An important concept in statistics is the cumulative distribution function (CDF)

$$F(x) = \Pr(X \leq x) = \sum_{x_i \leq x} Pr(X = x_i)$$

Note that this is like area under the curve of the histogram when the total area is 1.

We picture of $F(x)$ serves a summary as well.

```
> plot(ecdf(x))
```



Note that we if we know $F(x)$ we can quickly compute the proportion in any interval

$$\Pr\{X \in (a, b]\} = F(b) - F(a)$$

Currently we have no mathematical function but rather a numerical table.

2.1 Functions of F

There are two functions/properties of F that come up over and over (you will learn various reasons later).

The mean or average is defined as

$$E[X] = \sum_{i=1}^{N} x_i \Pr(X = x_i)$$

Note that if each x_i has same proportion, i.e. if they are unique, then

$$\mathrm{E}[X] = \frac{1}{N} \sum_{i=1}^{N} x_i$$

The variance is defined as

$$\mathrm{var}[X] = \sum_{i=1}^{N} (x_i - E[X])^2 \; Pr(X = x_i)$$

Similarly, if unique

$$\mathrm{var}[X] = \frac{1}{N} \sum_{i=1}^{N} (x_i - E[X])^2$$

```
> mean(x)
```

```
[1] 69.28988
```

```
> sd(x)
```

```
[1] 2.962913
```

Note that using we can easily show the following properties. If we rescale and shift all the

outcomes by

$$\mathrm{E}[a + bX] = a + b\mathrm{E}[X] \text{ and } \mathrm{var}[a + bX] = b^2\mathrm{E}[X]$$

You will use this often. So memorize it.

Note: I much rather describe the standard deviation which is defined as the $\sqrt{(}Var[X])$. Note it is in the same units as $E[X]$.

2.2 Normal distribution approximation

The normal approximation does not tell us $\Pr(X = x)$. It is not a probability function.

It is a mathematical trick that permits us to to define the probability of intervals that happens to approximate distributions such as weight, height, IQ, and test scores.

We use $\Phi(x)$ to denote the CDF:

$$\Phi(x) = \int_{-\infty}^{x} \phi(x)dx$$

with

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \exp(\frac{1}{2}x^2).$$

$\phi(x)$ is refereed to as a *d*ensity function. It is not a probability function! Statisticians also refer to these as a continuous probability distributions.

In R, ϕ and Φ are

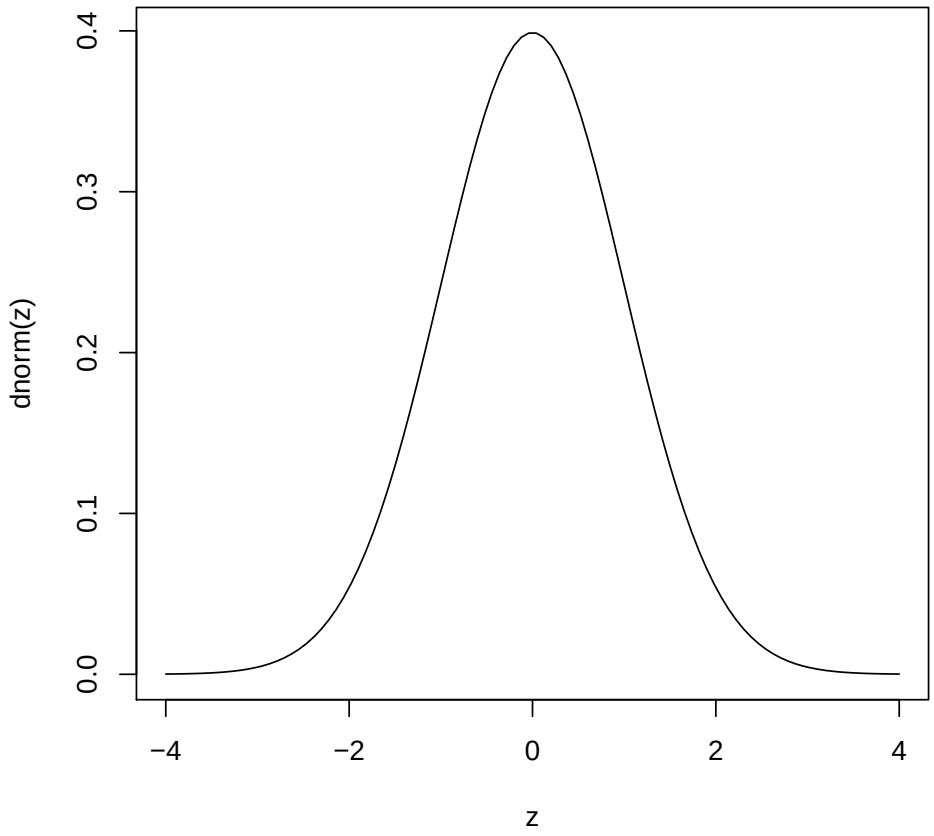
```
> dnorm(0)
```

```
[1] 0.3989423
```

```
> pnorm(0)

[1] 0.5

> pnorm(1) - pnorm(-1)

[1] 0.6826895

> pnorm(2) - pnorm(-2)

[1] 0.9544997

> pnorm(3) - pnorm(-3)

[1] 0.9973002
```

This is called the bell curve sometimes because of its shape:

```
> z = seq(-4, 4, len = 100)
> plot(z, dnorm(z), type = "l")
```



Question: Can this function work on weight and height?

Note: no units. So we need to do more to make it useful.

So, what we do is we shift and scale. For example, to approximate US male heights we need to scale by 3 inches and shift by 69 inches. If X is follows a normal distribution then US male heights can be approximated by

$$Y = (3 \text{ inches}) \times X + 69 \text{ inches}$$

For continuous probability functions we can define the mean and variance by defining very thin intervals, thinking of the variables as discrete and letting the size of the bins go to 0.

The summation then turns into an integral. For the normal distribution we have:

$$\mathrm{E}[X] = \int_{-\infty}^{\infty} x\phi(x)dx \text{ and } \mathrm{var}[X] = \int_{-\infty}^{\infty} (x - \mathrm{E}[X])^2\phi(x)dx$$

Note $\mathrm{E}[X] = 0$ and $\mathrm{var}[X] = 1$. In most books and papers, when a variable is scaled and shifted we use μ and σ. Then $Y = \sigma X + \mu$ has mean and variance μ and σ^2.

Things you should memorize is that if a variable follows a normal distribution then 68% of data is within 1 SD from average, 95% is within 2, and almost all, 99.7%, are within 3 SD.

These are standard units.

A useful property is that once we know the average and SD we know the distribution. So just 2 numbers describes everything in a very intuitive way.

Note this is why it is so common to see data summarized as $\mathrm{E}[X] \pm \mathrm{SD}[X]$. If in fact, the data is approximated by a normal distribution then this is all you need to know to entire distribution.

2.3 Quantiles

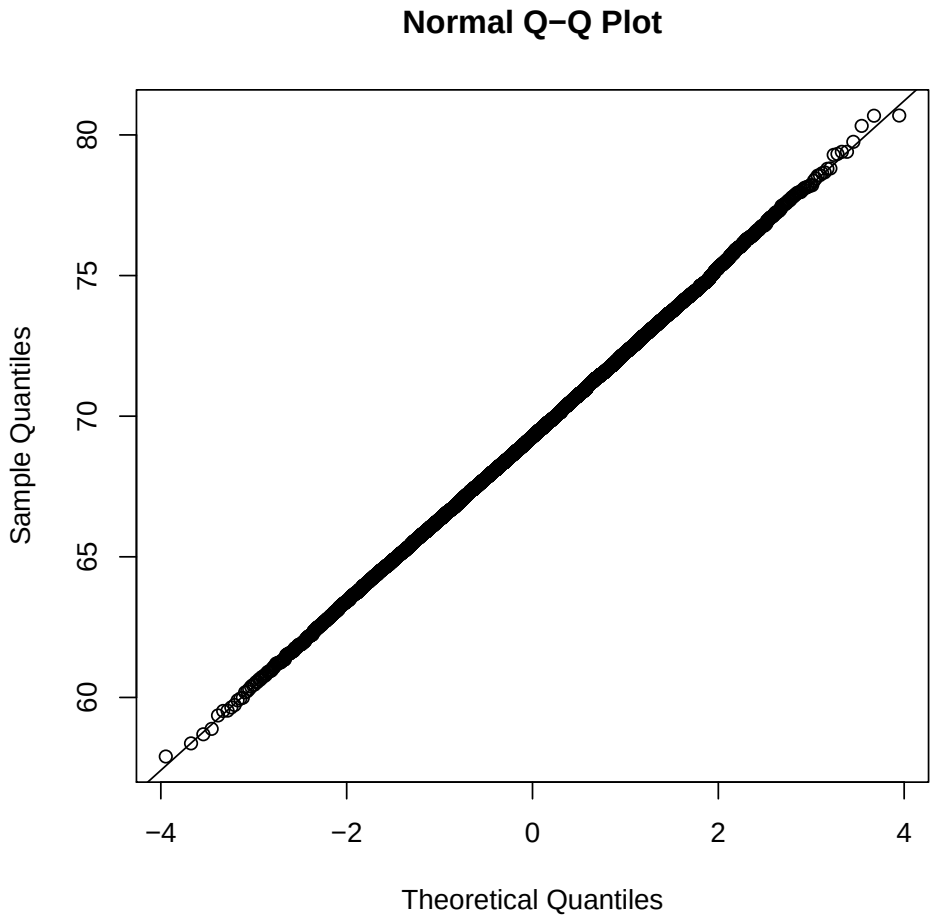
If we divide an ordered list of numbers into q groups, the break points are called the q-quantiles. Some have special names. If we divide into four they are **quartiles**. If we divide into 100 **percentiles**. Sometime the ordered data are refered to as quantiles.

We define the $0 < q < 100$ percentile as the value z_q for which a q% of data is smaller than z_q.

Formally we define $q - th$ percentile as $F(z_q) = q$.

Note that if two lists of number have the same distribution then their quantiles must agree. Therefore, we can use this

```
> qqnorm(x)
> qqline(x)
```

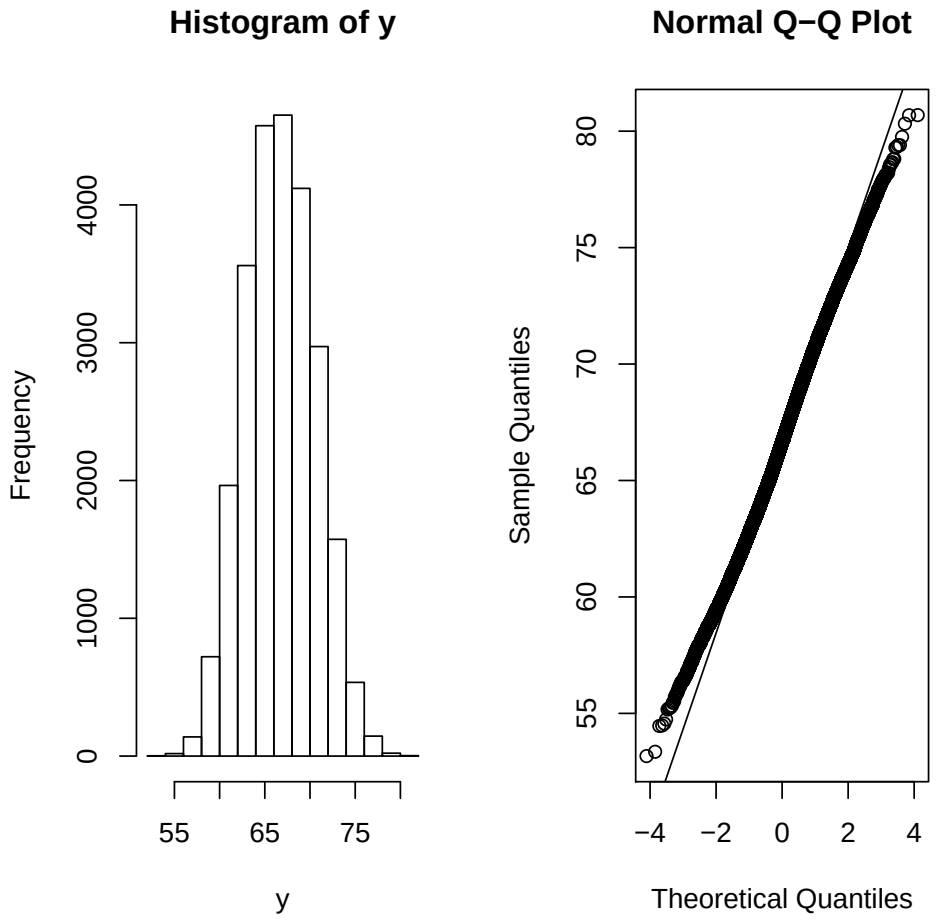


These are almost exactly on. Why? Because I generated using a normal.

How about all heights? Is it normal?

```
> y = dat$Height
```

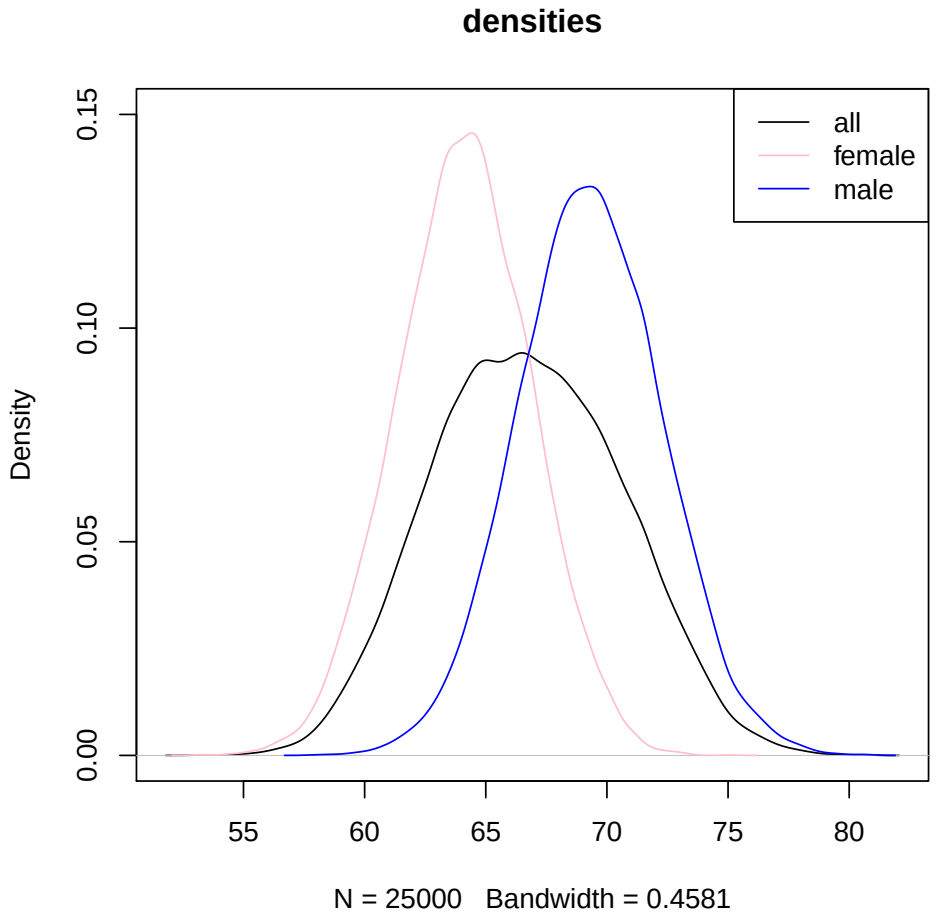
```
> par(mfrow = c(1, 2))
> hist(y)
> qqnorm(y)
> qqline(y)
```



Not quite. This distribution is too fat (kurtosis to high to be precise). The reason is that it is a mixture of two normals: Men and women.

```
> y = dat$Height
> z = split(y, dat$Gender)
> par(mfrow = c(1, 1))
> plot(density(y), col = "black", , ylim = c(0, 0.15), main = "densities")
```

```
> lines(density(z[[1]]), col = "pink")
> lines(density(z[[2]]), col = "blue")
> legend("topright", c("all", "female", "male"), lty = 1, col = c("black",
+     "pink", "blue"))
```



Note: densities are very similar to histograms: it is like drawing a curve through top of histogram bar tops.

In practice, the most common comparison is observed data to the theoretical normal distribution. However, note that the 100% percentile is the largest number in our list and for the theoretical normal the 100% percentile is ∞. To avoid this problem we compute the following percentiles: $1/(N+1) \times 100, 2/(N+1) \times 100, \ldots, N/(N+1) \times 100$ for both distributions.

Homework 1 discussion

Does smoking cause infant mortality?

Does not smoking cause infant mortality?

See tables and pictures from Chapter 1 in book.

Smoking -> small size <- infant killing diseases

If the healthier the heavier, then it might not be fair to compare babies of sames size but rather same standard units.

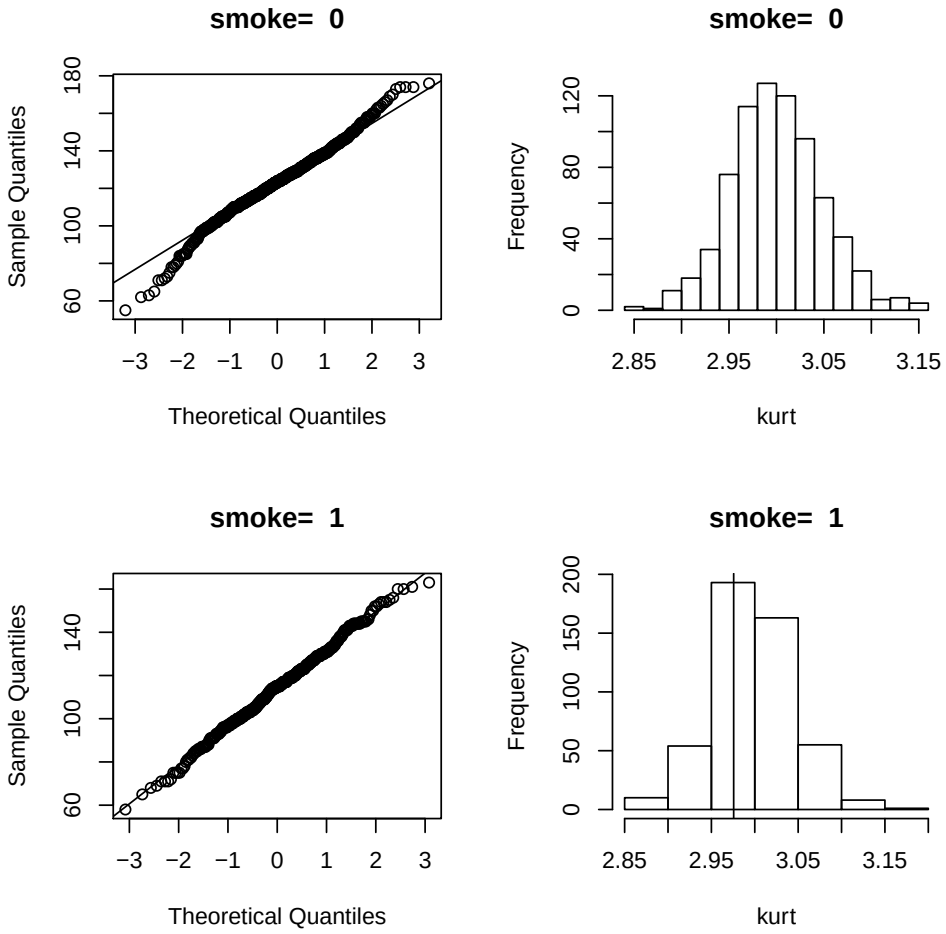
```
> dat = read.table(file.path(datadir, "babies.data"), header = TRUE)
> names(dat)
```

```
[1] "bwt"       "gestation" "parity"    "age"       "height"    "weight"
[7] "smoke"
```

```
> par(mfrow = c(2, 2))
> for (i in c(0, 1)) {
+     y = dat$bwt[dat$smoke == i]
+     qqnorm(y, main = paste("smoke= ", i))
+     qqline(y)
+     z = (y - mean(y))/sd(y)
+     print(mean(z^4))
+     M = 10000
+     N = length(y)
+     ystar = matrix(rnorm(M * N, mean(y), sd(y)), N, M)
+     zstar = (ystar - rowMeans(ystar))/apply(ystar, 1, sd)
+     kurt = rowMeans(zstar^4)
```

```
+     hist(kurt, main = paste("smoke= ", i))
+     abline(v = mean(z^4))
+ }
```

```
[1] 4.026186
[1] 2.975698
```



3 Probability, Sampling, and CLT

We introduce the concept of a random variable. We denote them with upper case letters, e.g. X, and think that they are the outcome of a random process. For example picking a number from an urn (like in the lottery).

So if we have an urn with the following numbers

$$\{2, 3, 3, 6, 7\}$$

Then $\Pr(X = 3) = 2/5$, etc... One way to think about this is that if we repeat the process over and over again, then 3 will come up 2/5 of the time.

We define the probability function of a random variable as the probability function of the list of possibilities. The mean and variance are defined using the definitions above. We will soon see how these are useful.

The random variables that we deal with as applied statistics are all *discrete*. But, often *continuous* random variables are useful approximations.

In general, we define the distribution of a random variable with its cumulative distribution function: $\Pr(X \in (a, b]) = F(b) - F(a)$. Mathematical statistics makes this more general: $\Pr(X \in A)$ with A any subset...

3.1 Bernoulli trials

Working example: A casino owner asks us how many times does a person have to bet \$1 on black so that the casino is guaranteed to take \$10 from this person.

First question? What do you mean by guaranteed? Let's say 99.99%.

To answer this question we will define a random variable X to denote the outcome of the game: player wins (1) or not (0). The probability function is defined by one number: the proportion of outcomes that lead to a win. We denote this number with $\Pr(X = 1) = p$,

In roulette, the possible outcomes are 18 reds, 18 blacks and 2 greens. So $p = 9/19$.

The *expected value* of the random variable is defined as the mean of the distribution function. In this case

$$\mathrm{E}[X] = 0 \times (1 - p) + 1 \times p = p$$

Similarly the variance is

$$\mathrm{var}[X] = (0 - p)^2 \times (1 - p) + (1 - p)^2 \times p = p(1 - p)$$

Later it will become apparent why we care about the expected value and variance.

We can play this game over and over again an observe various Xs. The roulette works the same no matter what happens before so the Xs don't depend on each other. We denote these as *independent* random variables. In this case they also have the same distribution so they are *independent* and *identically* distributed (IID).

Some properties to know about independent random variables: If X and Y are independent.

$$\mathrm{E}[X + Y] = E[X] + E[Y] \text{ and } \mathrm{var}[X + Y] = \mathrm{var}[X] + \mathrm{var}[Y]$$

The quantity we care about here is the sum of n random variables, n being the number of games we have played. Then the number of times the player has won is

$$S_n = \sum_{i=1}^{n} X_i$$

Note the expected values and variance are

$$\mathrm{E}[S_n] = np \text{ and } \mathrm{var}[S_n] = np(1-p)$$

Note that for the proportion of times we win we get:

$$\mathrm{E}[S_n/n] = p \text{ and } \mathrm{var}[S_n] = p(1-p)/n$$

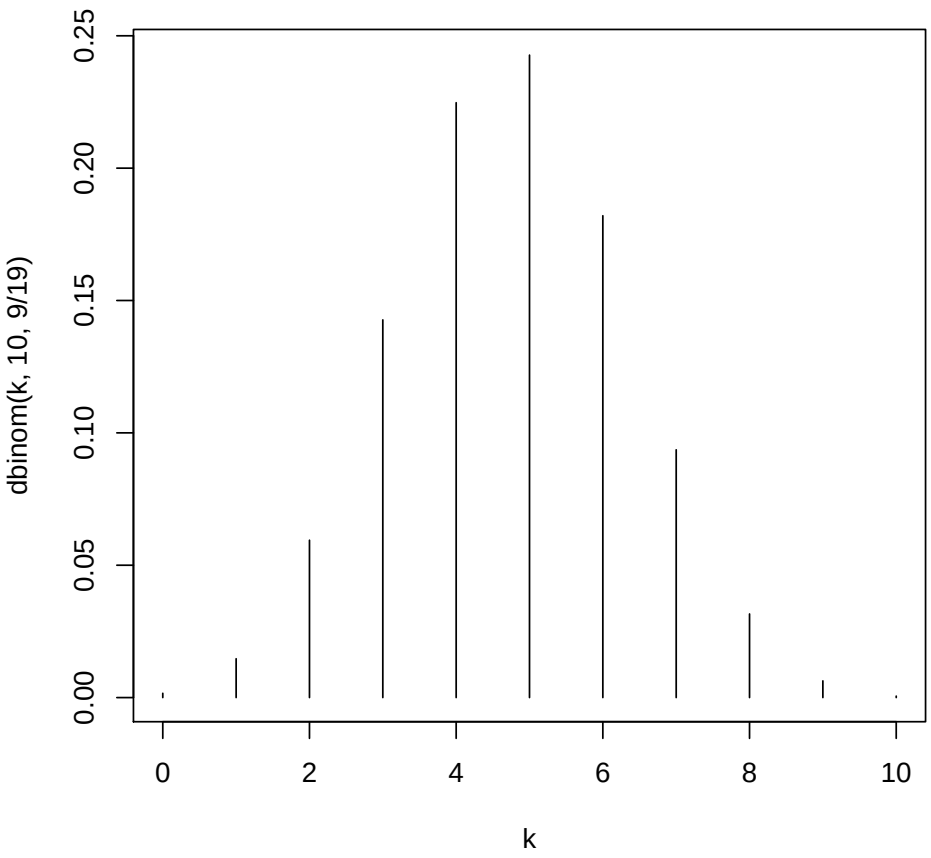
We will get back to this soon.

Fact: The distribution of S_n is relatively easy to figure out. It follows a binomial distribution:

$$\Pr(S_n = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

In R we get this by simply using dbinom. Here is the distribution function for the sum of 10 roulette games:

```
> k = c(0:10)
> plot(k, dbinom(k, 10, 9/19), type = "h")
```



Note that losing 6 is substantially more likely that winning 6.

3.2 Central Limit Theorem

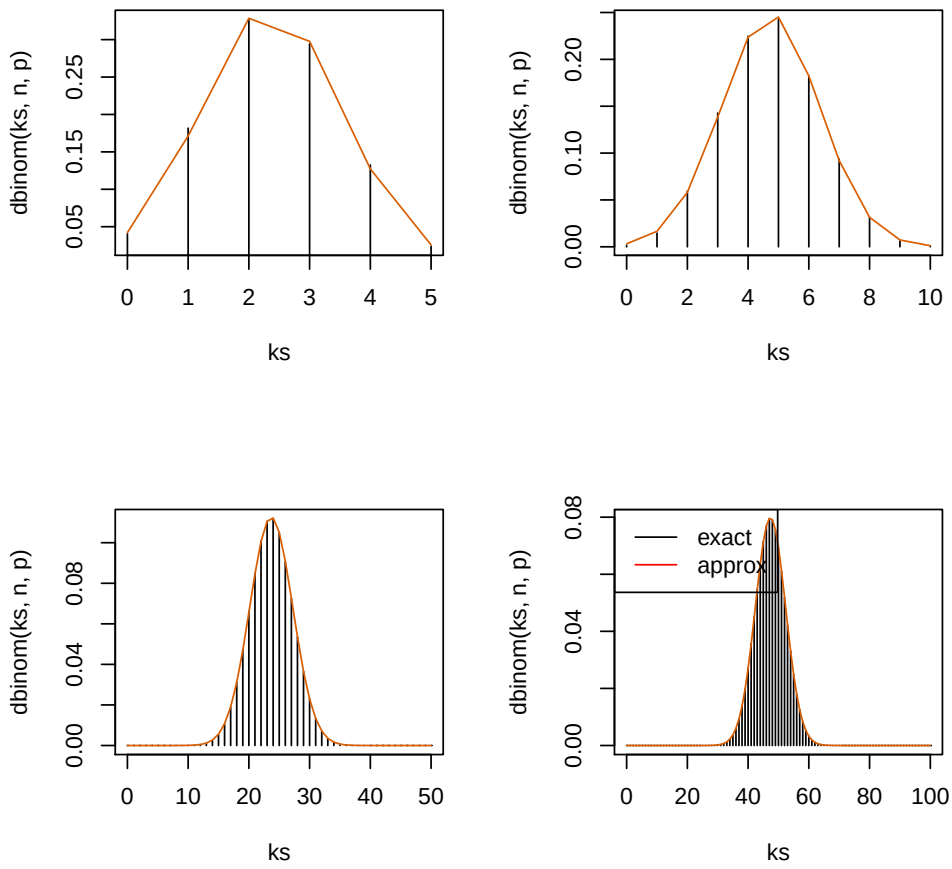
You will learn the central limit theorem (CLT) which states that the sum of Bernoulli random variables is very well approximated by a normal distribution

$$Y \equiv (S_n - np)/\sqrt{np(1-p)} \sim N(0,1)$$

This last symbol means that it follows the distribution, i.e. $\Pr(Y \leq y) = \Phi(y)$.

Here we compare binomial to normal for various n

```
> par(mfrow = c(2, 2))
> p = 9/19
> for (n in c(5, 10, 50, 100)) {
+     ks = 0:n
+     plot(ks, dbinom(ks, n, p), type = "h")
+     mu = n * p
+     sd = sqrt(n * p * (1 - p))
+     normapprox = pnorm(ks + 0.5, mu, sd) - pnorm(ks - 0.5, mu,
+         sd)
+     lines(ks, normapprox, col = 2)
+ }
> legend("topleft", c("exact", "approx"), col = c("black", "red"),
+     lty = 1)
```



Now, it is easy to answer the casino's question:

Casino winnings are $2n - S_n$, so we can use central limit theorem to answer

$$\Pr(n - 2S_n \geq 100)$$

Since we know the approximate distribution of S_n

Note, in this case we know the exact distribution of S_n (binomial) but it is much more convenient to use the normal distribution.

To avoid solving a quadratic formula we can use the computer to find the solution.

```
> zscore = function(n, p = 18/38, a = 100) ((n - a)/2 - n * p)/sqrt(n *
+     p * (1 - p))
> par(mfrow = c(1, 1))
> n = seq(1, 10000)
> plot(n, zscore(n), type = "l")
> abline(h = qnorm(0.9999))
> n[which.min(abs(zscore(n) - qnorm(0.9999)))]
```

```
[1] 8347
```

3.3 Estimation

Example of applications:

- How many voters will vote Democratic?
- What is the average medical expenditure for employees of JHSPH?

Define:

- Population
- Population Size: N
- Parameters:
  - population average, usually denoted with μ
  - population SD, usually denoted with σ
- Simple Random Sample

ASK: What is random what is not?

Questions:

- How good is the sample average?
- How close to population average?
- If we take the sample again, how similar will it be?

Sample distribution. The sample average is a random variable. Theoretical Statistics deals with finding the distribution.

Why is knowing the distribution useful? We can say how different our estimate could have been.

Example 1: Voters

What is the population distribution?

What is the population parameter of interest?

The population is a bunch of 0s (not democrats) and 1s (democrats)

The proportion of 1s, say p, is the population average.

Take a random sample (with replacement) of size n: $\{X_1, \ldots, X_n\}$ Note

$$\Pr(X_i) = p, \text{ for all} i$$

Estimate: Start with sample average.

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Take sampling with replacement.

What is the distribution of the sample average?

We know

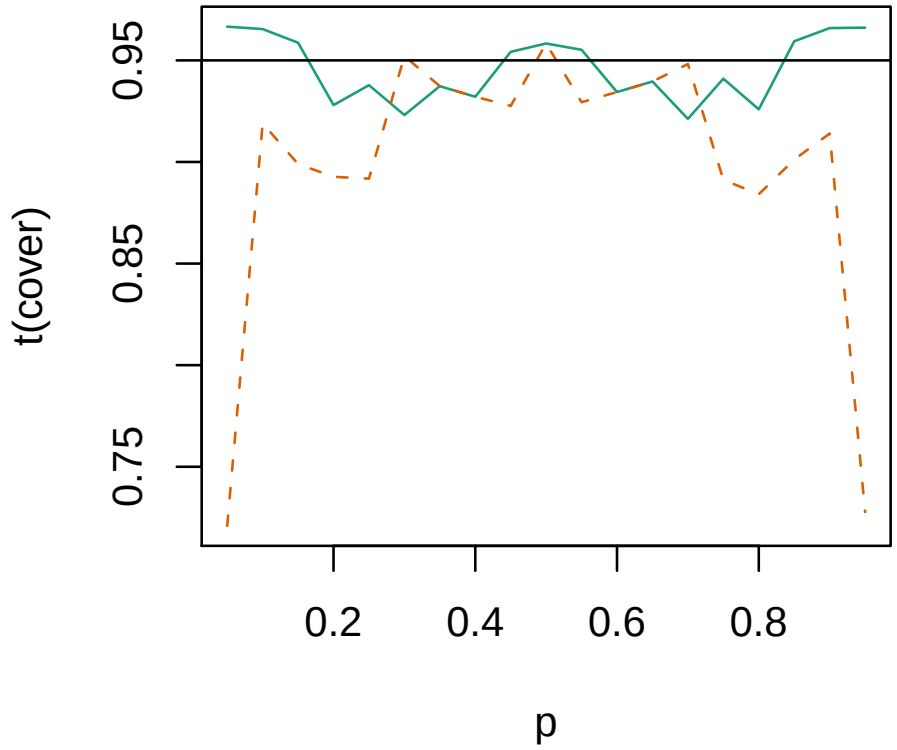
$$n\bar{X}$$

is binomial, just like roulette except we don't know p. We know for large n looks like normal.

So, $\bar{X} \pm 4\sqrt{np(1-p)}$ is a 95% confidence interval. We don't know p so we stick in $\bar{X}$ and it still works.

Check with simulation. Does the 95% actually cover the true p 95% of time

```
> p = seq(0.05, 0.95, 0.05)
> getcover = function(p, n = 25, B = 10000) {
+     X = sample(c(0, 1), B * n, replace = TRUE, prob = c(1 - p,
+         p))
+     X = matrix(X, B, n)
+     phat = rowMeans(X)
+     se1 = sqrt(p * (1 - p)/n)
+     se2 = sqrt(phat * (1 - phat)/n)
+     c(mean(abs(phat - p) < 1.96 * se1), mean(abs(phat - p) <
+         1.96 * se2))
+ }
> cover = sapply(p, getcover)
> matplot(p, t(cover), type = "l")
> abline(h = 0.95)
```



Question: And without replacement whatâĂŹs the distribution?

Example 2: Medical expenditures

What is the population distribution?

In general, we can write: $\{x_1, \ldots, x_N\}$

The population mean is simly $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$, similarly the population variance is $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$

What is the population parameter of interest?

Take a random sample (with replacement) of size n: $\{X_1, \ldots, X_n\}$

The *sample average*, denoted with $\bar{X}$ is a random variable defined by

$$\bar{X} \equiv \frac{1}{N} \sum_{i=1}^{n} X_i$$

What is the distribution of $\bar{X}$?

What is the standard error of $\bar{X}$?

Note: we sometimes refer to this a precision.

Note that as n gets larger, our estimate gets more precise.

What else can we say?

Do we know distribution of $\bar{X}$

3.3.1 Confidence intervals

CLT actually gives us distribution of

How close is $\bar{X}$ to mu?

Trick questions: The prob of parameter in the interval is 95%? Draw it.

It's the other way around. The interval is random not the population parameter.

Note that $(\bar{X} - \mu)/\text{SE}(\bar{X})$ is approx $N(0, 1)$.

$$\Pr\{|\bar{X} - \mu| \leq 2\text{SE}(\bar{X})\} = 0.95$$

$\bar{X} \pm 2\text{SE}(\bar{X})$ is called a 95% confidence interval.

Unlike the poll, the population distribution is very complicated because we don't know what the xs are.

But wait!! Can we write $\text{SE}(\bar{X})$?

We don't know σ. A common estimate is the sample variance. What is it?

$$s^2 = 1/n \sum_{i=1}^{n} (X_i - \bar{X})^2$$

The heuristic explanation is that the sample distribution is similar to the population distribution so they should have similar variance.

Quick trick question: Is the population average the same as the sample average?

Note: It turns out dividing by $1/(n-1)$ is a bit better. YouâĂŹll learn this later or maybe homework.

Now is

$$\frac{\bar{X} - \mu}{s/\sqrt{n}}$$

Still normal? The CLT says yesâĂę You will learn that later as well.

This statistic, estimate minus expected value divide by an estimate of its standard error, is maybe the most widely used statistic. This is referred sometimes as Wald statistic, test

statistic, z-score, depending on the situation. Theoretical statistics is all about showing that this statistic is always approximately normal.

Here is the confidence interval:

```
> X = scan(file.path(datadir, "medicalcost.csv"))
> mean(X) + c(-2, 2) * sd(X)/sqrt(length(X))
```

```
[1] 5531.509 9925.579
```

3.3.2 Without replacement

What about samples without replacement (in practice much more common)?

It turns it CLT works for dependent data under certain restrictions. You learn this in your theory class.

3.4 Law of rare events

The lottery. Every person either wins or loses. The evens are IID. n is huge, so is the number of people that win the lottery should be normal according to CLT. Is it? Why not?

CLT works better when

When $n \to \infty$ and $p \to 0$ but $np \to \alpha$ then S_n converges to something else: Poisson.

How do we interpret asymptotics in practice?

3.5 Bootstrap

Is the distribution of $\bar{X}$ really normal? If we donâĂŹt trust CLT. What if we want to figure out

What can we do?

Simulation? We donâĂŹt know the population distribution.

The population distribution $\{x_1, \ldots, x_N\}$ and the sample distribution $\{X_1, \ldots, X_N\}$ should be "similar".

We imitate the

So we take a simple random sample (with replacement) from the sample.

And form the statistic, for example the mean.

For the b-th bootstrap sample, we consider:

$$\bar{X}^b = \sum_{i=1}^{n} X_i^b$$

We do this many times, say B, and now we have

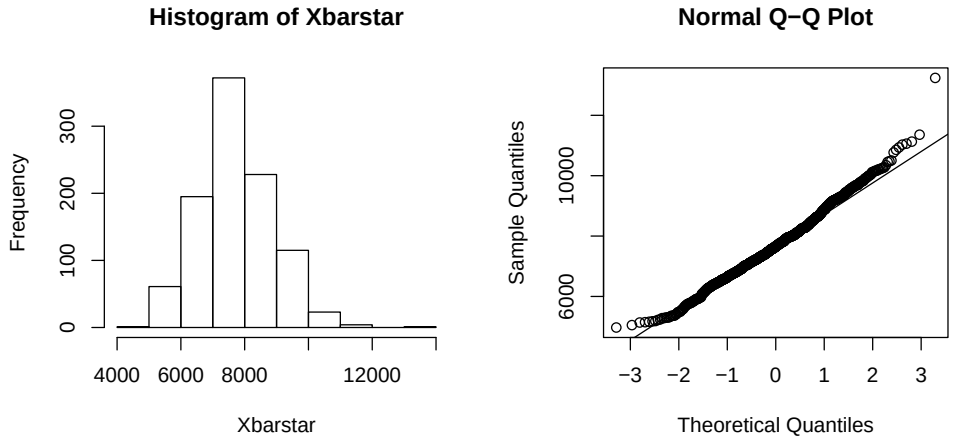
$$\bar{X}^1, \ldots, \bar{X}^B$$

A distribution of $\bar{X}$s. There is theory showing that the properties of the bootstrap version of the distribution are similar to the real distribution.

```
> X = scan("data/medicalcost.csv")
```

```
> B = 1000
> Xbarstar = sapply(1:B, function(i) mean(sample(X, replace = TRUE)))
> par(mfrow = c(1, 2))
> hist(Xbarstar)
> qqnorm(Xbarstar)
> qqline(Xbarstar)
> quantile(Xbarstar, c(0.025, 0.975))
```

```
    2.5%    97.5%
5646.738 9912.549
```



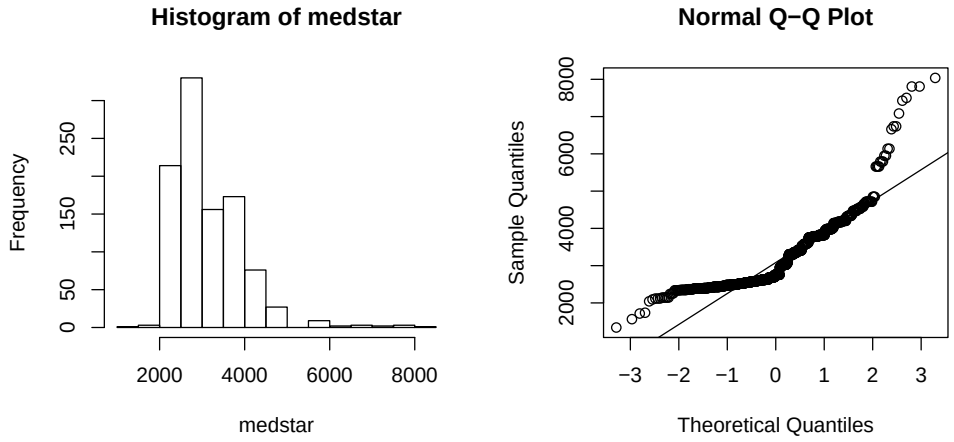
Looks very normal. Confidence interval is different.

What if we are asked to estimate median? No useful CLT. So we can use bootstrap.

```
> X = scan("data/medicalcost.csv")
> median(X)
```

```
[1] 2693.79
```

```
> B = 1000
> medstar = sapply(1:B, function(i) median(sample(X, replace = TRUE)))
> par(mfrow = c(1, 2))
> hist(medstar)
> qqnorm(medstar)
> qqline(medstar)
> quantile(medstar, c(0.025, 0.975))


    2.5%    97.5%
2347.805 4584.039
```



Note, it is not even close to normal.

Notes on probability

This is to help you understand probability class.

Two die:

$\Pr(1) = 1/36, \Pr(7) = 1/6$

What is the sample space for the pair? $\{(1,1),(1,2),\ldots,(6,6)\}$

What would be an example of a sigma field?

What is the measure or probability function?

$\Pr(x) = 1/36$, for all x in the sample

What element of the sigma field defines the event: The sum is 7?

$E = \{(1,6),(2,5),(3,4),(4,3),(5,2),(6,1)\}$

$\Pr(E) = \sum_{x \in E} f(x) = 1/36 + \ldots + 1/36 = 1/6$

Thing to know:

- Cards
- Independence

Collins Case? Very common mistake to assume independence.

4 Testing

Fisher's story: In 1834 John Bennet Lawes, a chemist student , left school to return home. Noticed bone meal did not always help. Sometimes acid helped more. He started a research facility at Rothamsted to study this: R.A Fisher joined the staff.

One day at tea time one of the biologist claimed she liked milk then tea. Fished did not believe she could tell the difference.

We want "Results of the experiment should be completely controlled by the laws of chance."

This is how applied statisticians use probability to test. Sometimes this is hard. You will see some examples later. For now they will be relatively easy.

Pepsi challenge.

What is the strategy? The NULL?

Is once enough? Once the chance of guessing?

Hypothesis testing

Define the NULL.

This is not what Fisher did, but itâĂŹs easier.

Give them 8 cups. Flip a coin each time.

What are chances of picking correct k times?

$\Pr(X = k)\binom{8}{k}(1/2)^8$

Using this method, we might have few of one or the other.

Another way

We serve 4 and 4.

What data do we keep?

Make two by two table.

Mention sufficient statistics. Here its just # of correct.

What are chances of $0, 1, 2, \ldots, 4$ correct?

Hypergeometic: Enumerate all possibilities:

What are all the possibilities: $\binom{8}{4}$.

Number of white: $m = 4$, Number of black: $n = 4$, Choose at random: $k = 4$

Correct or white balls is x

Total ways $\binom{m+n}{k}$

Total way to choose x white balls $\binom{m}{x}\binom{n}{k-x}$

$\binom{4}{x}\binom{4}{k-x}/\binom{8}{4}$

$\binom{8}{4} = 70, \binom{4}{0} = 1, \binom{4}{1} = 4$

For $x = 0, 1, 2, 3, 4$ the chances are

```
> k = 0:4
> dhyper(k, 4, 4, 4)
```

```
[1] 0.01428571 0.22857143 0.51428571 0.22857143 0.01428571
```

which is 1/70, 16/70, 36/70, 16/70, 1/70

4.1 p-values

4.1.1 Pepsi challenge

You see these all over the scientific literature. As a collaborator you will many times find yourself simply computing p-values. What do these mean?

Say someone made only 1 mistake. Then $\Pr(1 or less mistakes) = 17/70$ UNDER NULL!

What about $\Pr(1 or less | if person can do it?)$ This is probability model is way more complicated.

If 1/70 is not good enough, How else can we do it?

Where did chance come from?

Where did chance come from in AJE papers?

4.1.2 Comparing two population averages

Talk about Distributions, parameters, SD

Use the weights data (smoke, not smoke are different?)

We need to be specific. Are the population averages different? Note: if normal and SD the same this tells us everything.

We have two samples now. And two sample averages, say $\bar{Y}$ and $\bar{X}$.

We define a null hypothesis: population averages not different: $\mu_X = \mu_Y$.

Observe $\bar{Y} - \bar{X}$.

```
> dat = read.table(file.path(datadir, "babies.data"), header = TRUE)
> Y = dat$bwt[dat$smoke == 1]
> X = dat$bwt[dat$smoke == 0]
> mean(Y) - mean(X)
```

```
[1] -8.937666
```

Now what? Can a difference of -8.9 grams happen by chance?

Looking at the SDs and ns we have an idea

```
> sd(X)/sqrt(length(X))
```

```
[1] 0.638726
```

```
> sd(Y)/sqrt(length(Y))
```

```
[1] 0.8226793
```

```
> t.test(Y, X)


        Welch Two Sample t-test

data:  Y and X
t = -8.5813, df = 1003.197, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -10.98148  -6.89385
sample estimates:
mean of x mean of y
 114.1095  123.0472
```

What is the distribution of our statistic under the null? CLT tells us: $N(0, \sigma_X^2/n_X + \sigma_Y^2/n_Y)$

So the following will be $N(0,1)$

$$\frac{\bar{Y} - \bar{X}}{\sqrt{(\sigma_x^2/n_x + \sigma_y^2/n_y)}}$$

This statistic is referred to as a t-statistic.

p-value answers: what is chance of seeing something as extreme as what we saw:

$$\Pr(\text{t-stat}| > \text{observed t-stat}|\text{Null})$$

Note we have to estimate σs to obtain a number. CLT still works if n large.

If n small, there is still hope. Note p-values are given in biology papers using 3 mice.

If the population distributions are normal, we can figure out the distribution of the t-statistic. It is not normal: has heavier tails. It follows a distribution refereed to as a t-distribution. It has parameter refereed to as degrees of freedom that related to the number of data points used in the test. Usually number of data points minus number of parameters estimated.

When degrees of freedom goes to 30, the t is just like Normal.

Please look up how to compute the degrees of freedom.

5 Two variables

Most papers in Epidemiology journals are about the assciation between two outcomes: coffee drinking and strokes for example. The great majority of these conclusion do not come from deductinve reasoning based on molecular biology or physiology. Instead they are empricially derived from observational data. For example, for a group of people, for which the number of cups of coffee are recorded, is followed for various years and stroke events recorded as well. This section describes the statistical procedures that are used in these studies. We will start from the very begining.

5.1 Prediction motivation

If I ask you to guess a the weight of a US adult males?

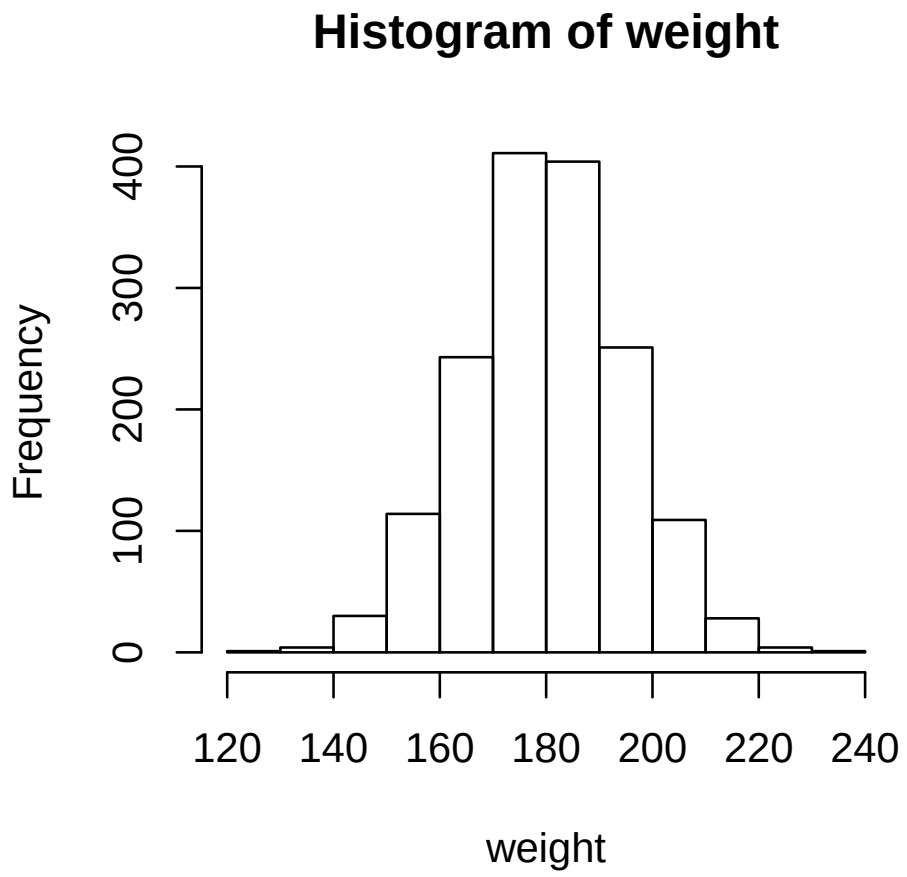
The population average is your best prediction: most likely and minimizes squared error.

```
> dat = read.csv(file.path(datadir, "USMaleHeightsAndWeights.csv"))
> attach(dat)
> hist(weight)
> print(mean(weight))
```

```
[1] 180.4081
```

```
> print(sd(weight))
```

```
[1] 14.97504
```



But what if I tell you this person is 71 inches, then what is your prediction?

```
> print(mean(height))
```

```
[1] 68.11187
```

```
> print(sd(height))
```

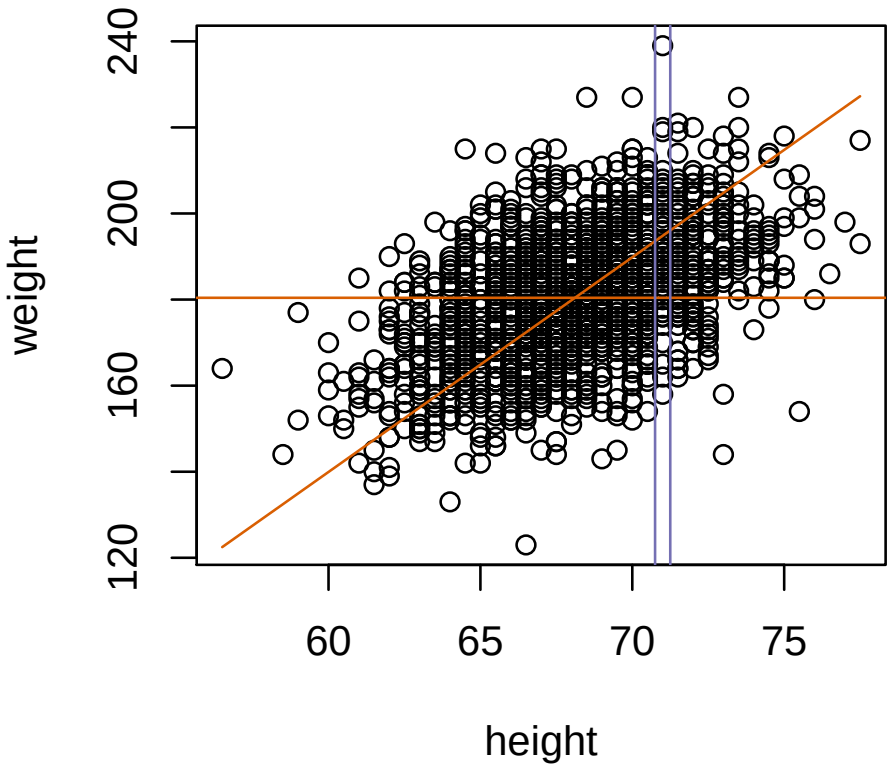
```
[1] 3.001873
```

Reminder, we can convert any list of numbers into standard units by subtracting the average and dividing by the standard deviation

$$\frac{X - \bar{X}}{SD_X}$$

So 71 inches is 3 SDs away from mean. How many SDs away from the mean do you expect in weight?

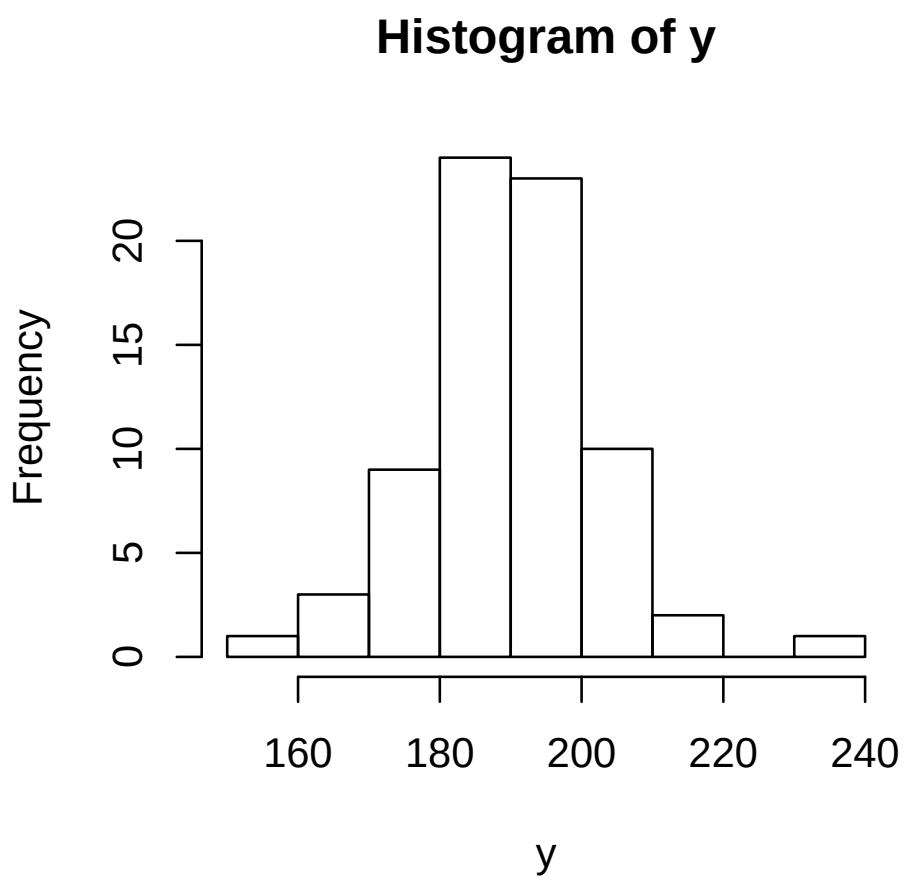
```
> plot(height, weight)
> x = seq(min(height), max(height), len = 300)
> lines(x, mean(weight) + (x - mean(height)) * sd(weight)/sd(height),
+     col = 2)
> abline(h = mean(weight), col = 2)
> abline(v = 71 + c(-1, 1) * 0.25, col = 3)
```



Let's look at the weights of all the men that are 71 inches

```
> y = weight[round(height) == 71]
> hist(y)
> mean(y)
```

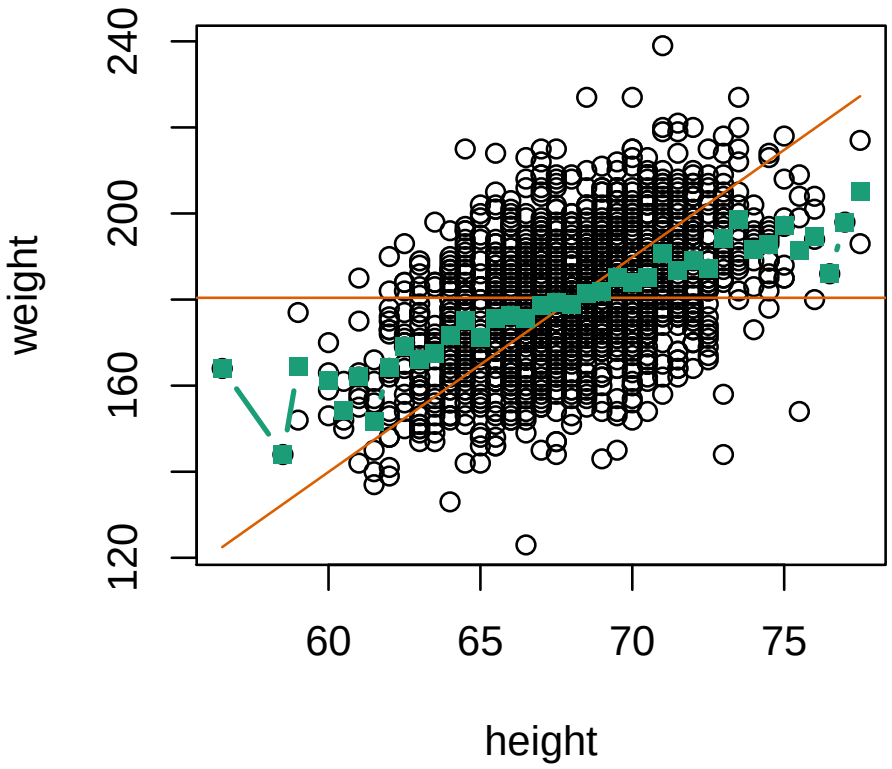
```
[1] 190.6712
```



This is called a conditional mean: E$[Weight|Height = 71]$. It minimizes mean squared error just like other means.

Let's do it for every height

```
> plot(height, weight)
> x = seq(min(height), max(height), len = 300)
> lines(x, mean(weight) + (x - mean(height)) * sd(weight)/sd(height),
+     col = 2)
> abline(h = mean(weight), col = 2)
> heights = sort(unique(height))
> preds = sapply(heights, function(h) mean(weight[height == h]))
> lines(heights, preds, type = "b", col = 1, pch = 15, lwd = 2)
```

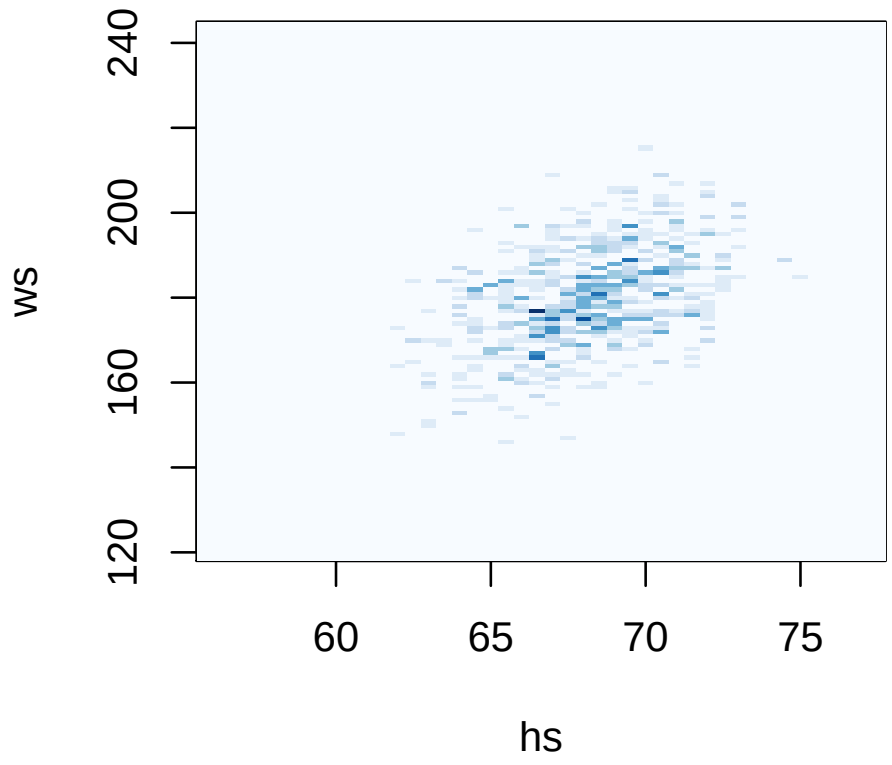


Note its lower than the SD lines. This is because height does not predict perfectly.

To define this expectation precisely we will need to define *joint distributions*. It's not that complicated. Now we have a list of pairs and we assign a probability to each pair.

Bivariate normal data look like a football.

```
> tab = table(height, weight)
> hs = as.numeric(rownames(tab))
> ws = as.numeric(colnames(tab))
> image(hs, ws, tab, col = brewer.pal(9, "Blues"))
```



Their is a bivariate version of the normal distribution. It gives us probabilities for any 2 dimensional set, for example rectangles:

$$\Pr[X \in (a,b) \text{and} Y \in (c,d)] = \int_c^d \int a^b \phi(x,y) dx dy$$

here the joint distribution is

$$\phi(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y} \exp \frac{1}{2}(x-\mu_X \, y-\mu_Y) \begin{pmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{pmatrix}^{-1} \begin{pmatrix} x-\mu_X \\ y-\mu_Y \end{pmatrix}$$

This implies that

$$\mathrm{E}[Y|X=x] = \mu_Y + \sigma_Y \rho \frac{x-\mu_X}{\sigma_X}$$

and that

$$\text{var}[Y|X = x] = (1 - \rho^2)\sigma_Y^2$$

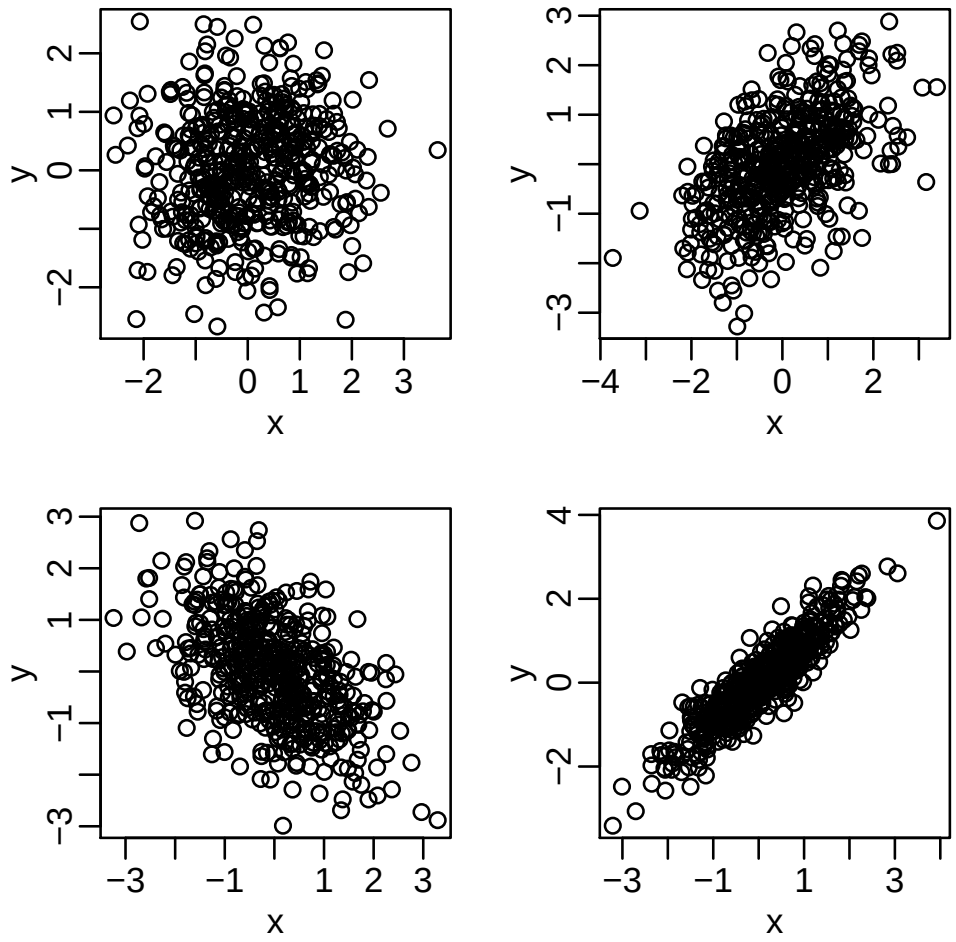
ρ is called the correlation. Note for every standard unti change in X we predict ρ standard unit changes in Y

$$\frac{\text{E}[Y|X = x] - \mu_Y}{\sigma_Y} = \rho \frac{x - \mu_X}{\sigma_X}$$

Note that the more correlation, the more you use X to predict Y. Also the better the prediction.

In the plots you can see the spread get smaller

```
> library(mvtnorm)
> mypar(2, 2)
> rho = 0
> plot(rmvnorm(500, c(0, 0), matrix(c(1, rho, rho, 1), 2, 2)),
+     xlab = "x", ylab = "y")
> rho = 0.5
> plot(rmvnorm(500, c(0, 0), matrix(c(1, rho, rho, 1), 2, 2)),
+     xlab = "x", ylab = "y")
> rho = -0.5
> plot(rmvnorm(500, c(0, 0), matrix(c(1, rho, rho, 1), 2, 2)),
+     xlab = "x", ylab = "y")
> rho = 0.9
> plot(rmvnorm(500, c(0, 0), matrix(c(1, rho, rho, 1), 2, 2)),
+     xlab = "x", ylab = "y")
```



For a discrete distribution the correlation is defined as

$$\sum_{i=1}^{n} \frac{x_i - \mu_X}{\sigma_X} \frac{y_i - \mu_Y}{\sigma_Y} \Pr(X = x_i, Y = y_i)$$

It can be defined for any distribution. For the normal distribution we can show that:

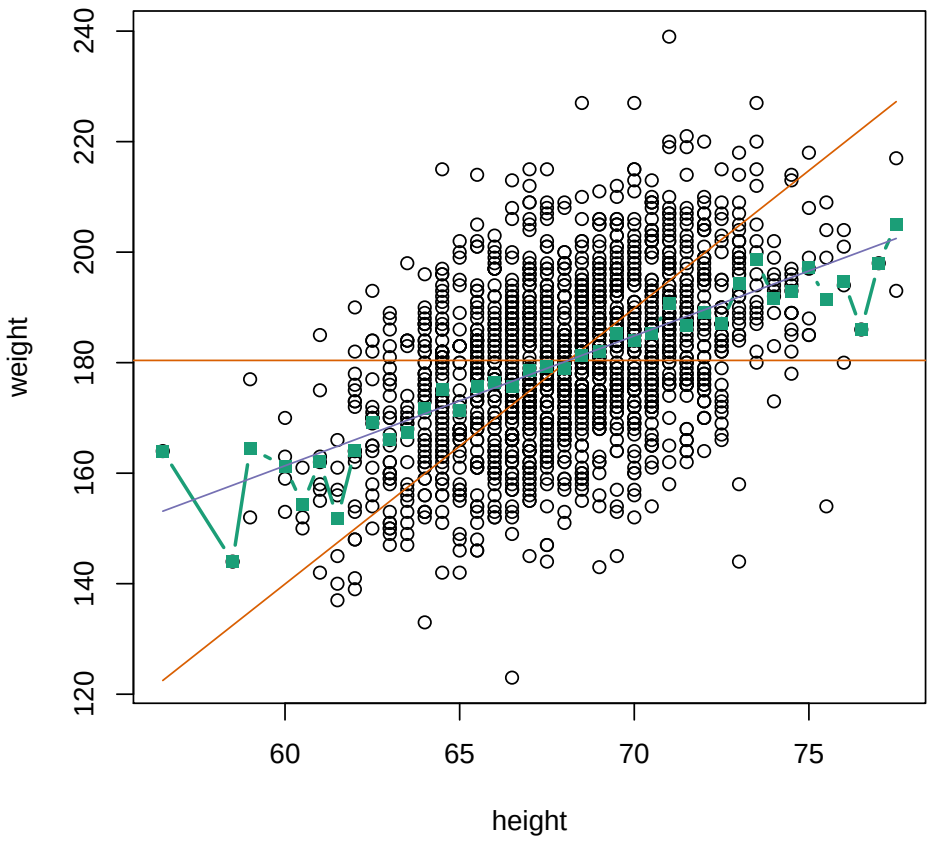
$$\rho = \int \int \frac{x - \mu_X}{\sigma_X} \frac{y - \mu_Y}{\sigma_Y} \phi(x, y) dx dy$$

If we have a sample, the sample correlation is an estimate of the population of correlation.

```
> cor(weight, height)
```

```
[1] 0.471138
```

```
> plot(height, weight)
> x = seq(min(height), max(height), len = 300)
> lines(x, mean(weight) + (x - mean(height)) * sd(weight)/sd(height),
+     col = 2)
> abline(h = mean(weight), col = 2)
> heights = sort(unique(height))
> preds = sapply(heights, function(h) mean(weight[height == h]))
> lines(heights, preds, type = "b", col = 1, pch = 15, lwd = 2)
> linepred = mean(weight) + cor(weight, height) * (x - mean(height)) *
+     sd(weight)/sd(height)
> lines(x, linepred, col = 3)
```



Note that the *regression line* appears to be a better approach that stratifying and computing conditional means.

We are *modeling*. Assuming bivariate normal.

Note we can also start by writing a model

$$Y = \alpha + \beta x + \varepsilon$$

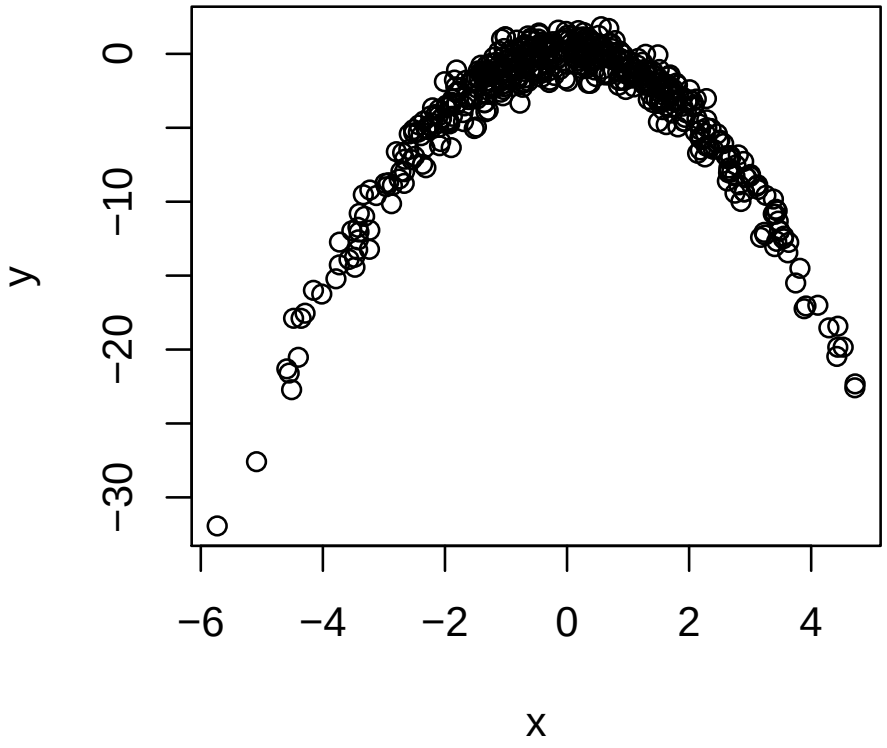
with x fixed and ε random. We can be more specific and say independent normals with mean 0 and s.d. σ. More on this later.

Watch out for non-linear relationships

```
> x = rnorm(500, 0, 2)
> y = -x^2 + rnorm(100)
> plot(x, y)
> cor(x, y)
```

```
[1] 0.04141546
```

```
> detach(dat)
```



Discuss Regression Fallacy

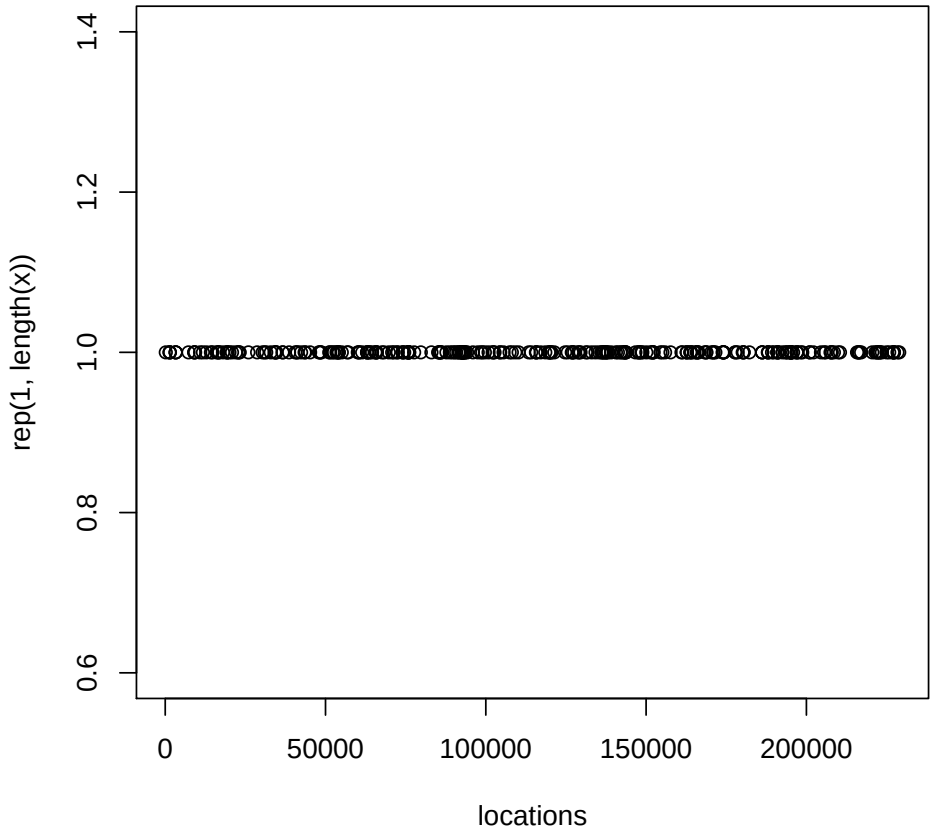
6 DNA Patterns

Are Palindromes over-reprented in a region of the genome?

It is useful for modelling counts. You demonstrated that binomial converges to Poisson when N is large and p is small.

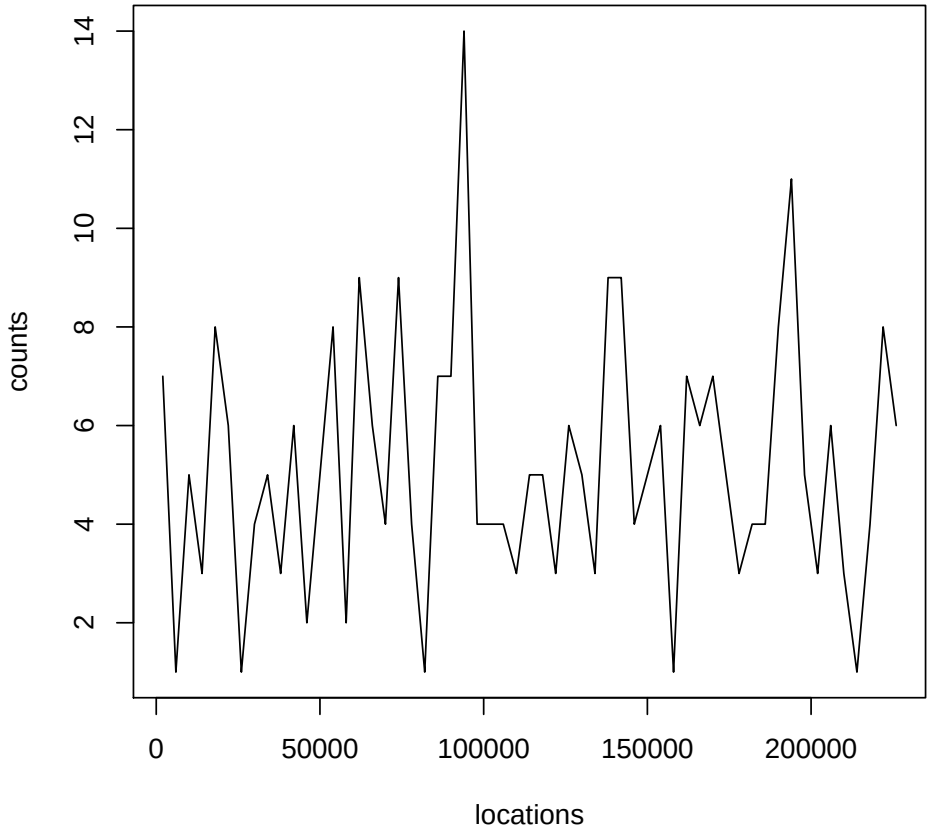
Read in the data and make and plot the locations. Not very useful.

```
> x = read.csv(file.path(datadir, "hcmv.csv"))[, 2]
> plot(x, rep(1, length(x)), xlab = "locations")
```



More useful is plot counts in bins

```
> breaks = seq(0, 4000 * round(max(x)/4000), 4000)
> tmp = cut(x, breaks)
> counts = table(tmp)
> locations = (breaks[-1] + breaks[-length(breaks)])/2
> plot(locations, counts, type = "l", ylab = )
```



Is 14 more than expected? What's random? How is it random?

Tha Poisson distribution is

$$\Pr(X = k) = \lambda^k / k! \exp(-\lambda)$$

If we make intervals of, say, bins of 4000 then we have N=4000 and p of about 0.001. So seems like asymptotics apply.

How do we estimate lambda?

Method of moments:

- estimate the mean of N.
- this will be a function of lambda
- plugin sample mean, var, etc..

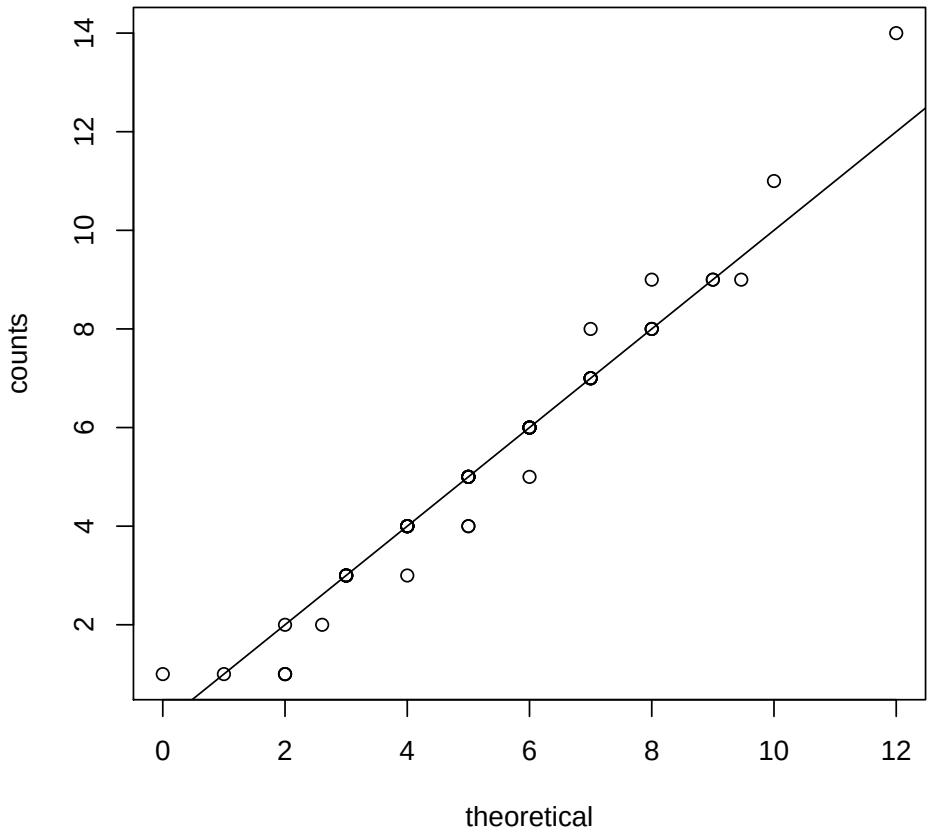
```
> lambda = length(x)/length(counts)
```

Estimate $\lambda = 296$ in 57 intervals of 4000,

Rate = 5.19 per 4000

qqplot

```
> theoretical = qpois(1:296/297, lambda)
> qqplot(theoretical, counts)
> abline(0, 1)
```



For count data we can do a Chi squared test. Compared observed to expected counts.

```
> observed = table(counts)
> expected = 57 * dpois(as.numeric(names(observed)), lambda)
> cbind(expected, observed)
```

```
     expected observed
1  1.64440234        5
2  4.26967625        2
3  7.39078462        8
4  9.59505371       10
5  9.96538912        9
```

```
6  8.62501514        8
7  6.39850747        5
8  4.15341713        4
9  2.39651359        4
11 0.58751765        1
14 0.03767202        1
```

To create the χ^2 test we compute "residuals". N_j = Counts in interval with expectation $\mu_j = \mathrm{E}(N_j)$. Because they are counts, they are roughly Poisson and $\mathrm{var}(N_j) = \mu_j$. and $\frac{N_j - \mu_j}{\sqrt{u_j}}$ is roughly normal. To construct a test for "closness" we look at sum of distances.

$$\sum_{j=1}^{J} \frac{(N_j - u_j)^2}{u_j}$$

This is the sum of normals square tha has χ^2 with J minus number of parameters estimated minus 1 degrees of freedom.

For the above to work we need large counts (CLT needs to kick in). I think rule of thumb is around 5.

```
> tmp = counts
> tmp[tmp < 2] <- 2
> tmp[tmp > 9] <- 9
> observed = table(tmp)
> expected = 57 * c(ppois(2, lambda), dpois(3:8, lambda), 1 - ppois(8,
+     lambda))
> cbind(expected, observed)
```

```
  expected observed
2 6.230737        7
3 7.390785        8
4 9.595054       10
5 9.965389        9
6 8.625015        8
```

```
7 6.398507        5
8 4.153417        4
9 4.641096        6
```

```
> 1 - pchisq(sum((observed - expected)^2/expected), 6)
```

```
[1] 0.9852182
```

Why are bins of different?

If we divide first into 10 we will not have enough counts.

If we divide second in 57, expected in each count is small.

Now, is 14 unexpected?

$$\begin{aligned}\Pr(\text{maximum count} > k) &= 1 - \Pr(\text{maximum count} < k) \\ &= 1 - \Pr(\text{all} < k) \\ &= 1 - \Pr(N < k)^{\text{intervals}}\end{aligned}$$

6.1 Maximum Likelihood Estimation

Write the likelihood function

$$\Pr(X_1 = x_1, X_2 = x_2, , X_n = x_n) = \lambda^{x_1}/x_1! \exp -\lambda \lambda^{x_2}/x_2! \exp -\lambda x \ldots \lambda^{x_n}/x_n! \exp -\lambda$$

Take log

$$\sum_{i=1}^{n} x_i \log(\lambda) n\lambda \sum_{i=1}^{n} log(x_i!)$$

Take derivative with respect to lambda, solve: MLE is mean

Similar for continuos (notice this is no longer probability) e.g. exponential ys are the waiting times

$$L(\theta) = f(y_1, , y_n; \theta)$$

$$l(\theta) = n \log(\theta) \theta \sum(xi)$$

$$n\theta = \sum_{i=1}^{n} x_i$$

7 Confounding

Admission data from Berkeley (Yaer???) showed more men were being admitted than women.

44% men admitted compared to 30% women!

```
> dat = read.csv(file.path(datadir, "admissions.csv"))
> dat$total = dat$Percent * dat$Number/100
> sum(dat$total[dat$Gender == 1]/sum(dat$Number[dat$Gender == 1]))
```

```
[1] 0.4451951
```

```
> sum(dat$total[dat$Gender == 0]/sum(dat$Number[dat$Gender == 0]))
```

```
[1] 0.3033351
```

But closer inspection shows a paradoxical results. Here are the percent admissions by Major:

```
> y = cbind(dat[1:6, c(1, 3)], dat[7:12, 3])
> colnames(y)[2:3] = c("Male", "Female")
> y
```

```
  Major Male Female
1     A   62     82
2     B   63     68
3     C   37     34
4     D   33     35
5     E   28     24
6     F    6      7
```

What's going on?

```
> y = cbind(dat[1:6, c(1, 2)], dat[7:12, 2])
> colnames(y)[2:3] = c("Male", "Female")
> y
```

```
  Major Male Female
1     A  825    108
```

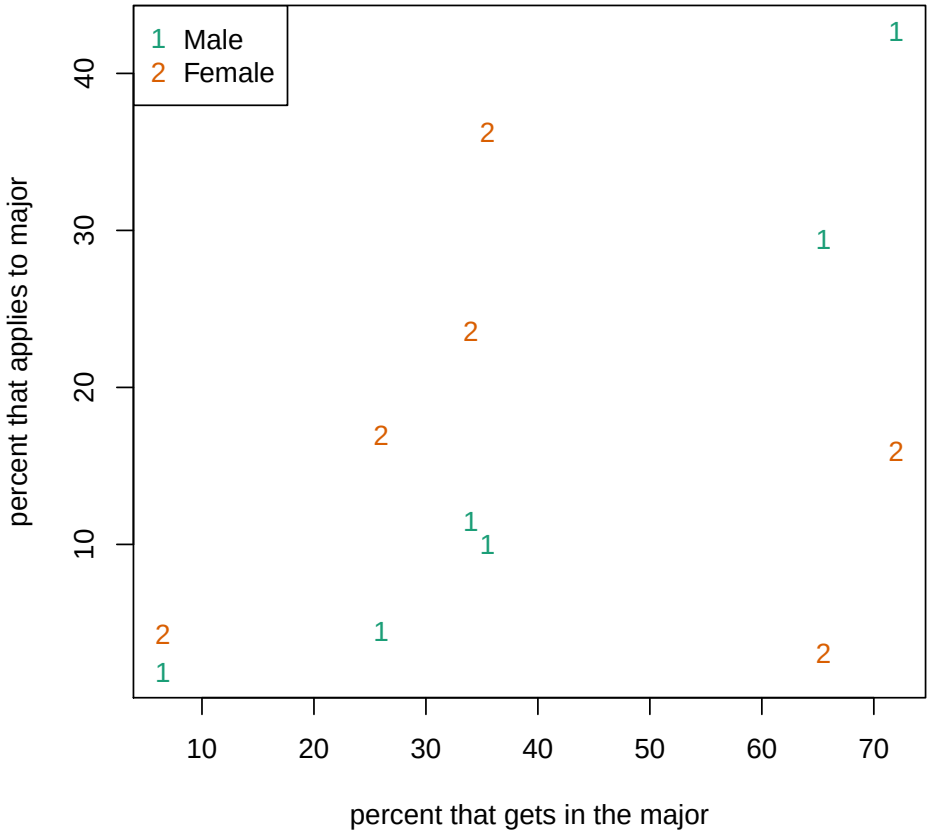
```
2     B  560     25
3     C  325    593
4     D  417    375
5     E  191    393
6     F  373    341
```

What's going?

This is called Simpson's paradox.

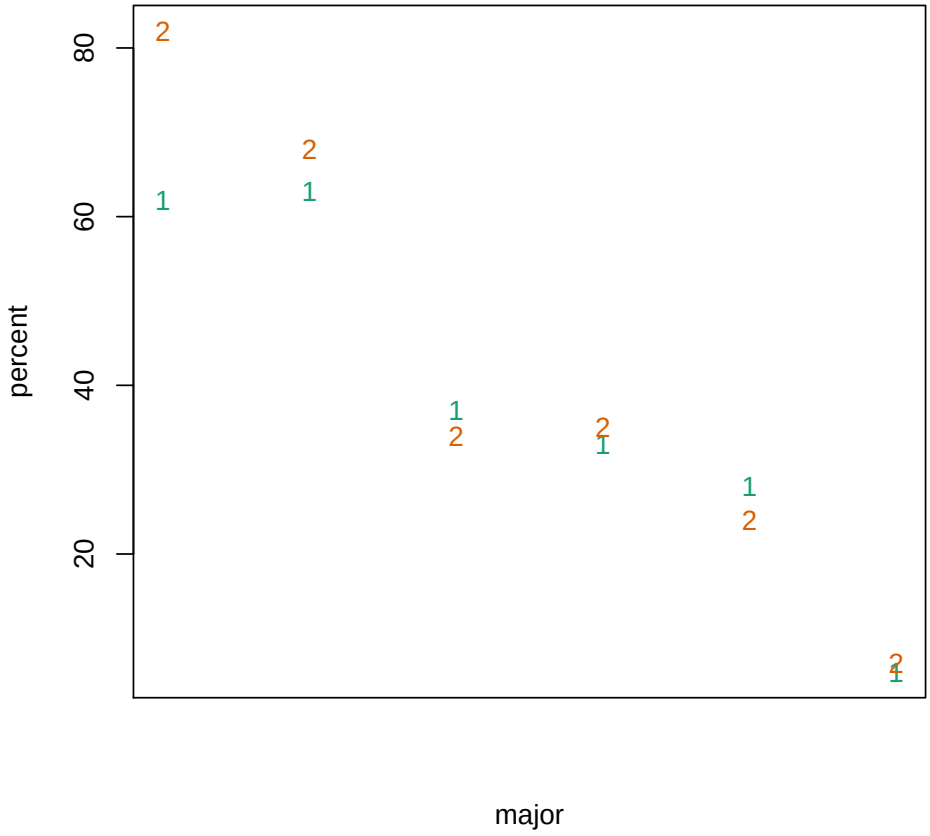
Male and easy majors are confounded.

```
> y = cbind(dat[1:6, 5], dat[7:12, 5])
> y = sweep(y, 2, colSums(y), "/") * 100
> x = rowMeans(cbind(dat[1:6, 3], dat[7:12, 3]))
> matplot(x, y, xlab = "percent that gets in the major", ylab = "percent that applies
> legend("topleft", c("Male", "Female"), col = c(1, 2), pch = c("1",
+     "2"))
```



So if we condition or stratify by major this goes away.

```
> y = cbind(dat[1:6, 3], dat[7:12, 3])
> matplot(1:6, y, xaxt = "n", xlab = "major", ylab = "percent")
```



The average difference by Major is 3.5% higher for women.

```
> mean(y[, 1] - y[, 2])
```

```
[1] -3.5
```

		1995	1996	Combined
We see this in Baseball often:	Derek Jeter	12/48 .250	183/582 .314	195/630 .310
	David Justice	104/411 .253	45/140 .321	149/551 .270

8 Regression part II

Regression is used to account for confounding. Here we see how.

Another way to find the regression line is to fit the model

$$Y_i = \alpha + \beta X_i + \varepsilon_i$$

with ε_i independent identially distributed random variables. We can use least squares to find the best fitting α and β, i.e. find the values that minimize

$$\sum_{i=1}^{n}(Y_i - \alpha - \beta X_i)^2$$

It turns out these are the same as above.

If we assume the ε are normally distributed then this is also the MLE.

Linear algebra is useful for finding least squares esimates.

Interpret Var(ε_i). Why is Var(Y) so small?

$$y_i = a + bx_i + \varepsilon_i$$

$$\underset{\sim}{y} = \underset{\sim}{X}\underset{\sim}{\beta} + \underset{\sim}{\varepsilon}$$

$$\underset{\sim}{X} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \qquad \underset{\sim}{\beta} = \begin{pmatrix} a \\ b \end{pmatrix} \underset{\sim}{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

$$E[\varepsilon] = \underset{\sim}{0} \ \mathrm{Var}(\varepsilon) = \begin{pmatrix} \sigma^2 & \vdots \\ \vdots & \sigma^2 \end{pmatrix}$$

$$
\begin{aligned}
&= I\sigma^2 \\
\Sigma(y_i - (a + bx_i))^2 &= (y - S\beta)^T(y - X\beta)
\end{aligned}
$$

Take derivative to find $\hat{\beta}$.

$$
\begin{aligned}
-2X^T(y - X\hat{\beta}) &= 0 \\
(X^TX)\hat{\beta} &= X^Ty \\
\hat{\beta} &= (X^TX)^{-1}X^Ty
\end{aligned}
$$

If solution exists.

Example:

$$
\begin{aligned}
X^TX &= \begin{pmatrix} n & \Sigma x_i \\ \Sigma x_i & \Sigma x_i^2 \end{pmatrix} \\
(X^TX)^{-1} &= \begin{pmatrix} \Sigma x_i^2 & -\Sigma x_i \\ -\Sigma x_i & n \end{pmatrix} \circ \frac{1}{n\Sigma x_i^2 - (\Sigma x_i)^2} \\
X^Ty &= \begin{pmatrix} \Sigma y_i \\ \Sigma X_i y_i \end{pmatrix},
\end{aligned}
$$

If y and x are centered, then

$$
\begin{aligned}
(X'X)^{-1} &= \begin{pmatrix} 1/n & 0 \\ 0 & 1/(n\mathrm{SD}_x^2) \end{pmatrix}^{-1} \\
X^ty &= \begin{pmatrix} 0 \\ \mathrm{SD}_x\mathrm{SD}_y \times nr \end{pmatrix} \\
(X'X)^{-1}X^Ty &= (0\ r\mathrm{SD}_y/\mathrm{SD}_x)' \\
\mathrm{E}(\hat{\beta} &= (X'X)^{-1}X^TX\beta
\end{aligned}
$$

$$= \beta$$

$$\begin{aligned}
\mathrm{Var}(AX) &= A\mathrm{Var}(X)A^T \\
\mathrm{Var}(\hat{\beta}) &= \mathrm{Var}\left((X^1X)^{-1}X^Ty\right) \\
&= (X^1X)^{-1}X^T(\sigma^2 I)X(X^1X)^{-1} \\
&= \sigma^2(X^1X)^{-1}
\end{aligned}$$

$\Uparrow$

Why correlation not here?

$$\mathrm{Var}\begin{pmatrix} \hat{a} \\ \hat{b} \end{pmatrix} = \begin{matrix} \sigma^2/n \\ \sigma^2/n\mathrm{SD}_x^2 \end{matrix}$$

$\uparrow$

$\overbrace{\text{What does this say about designing } X?}$

Some points: $\hat{\beta}$ is best linear unbiased estimate

CLT applies so $\hat{\beta} \sim N(\beta, (X^TX)^{-1}\sigma^2)$

In CLT what are asymptotic os X?

What about SE $(\hat{y})$?

$\hat{y}_i = \hat{a} + \hat{b}x_i$

$\mathrm{Var}(\underset{\sim}{x}_i^T\ \hat{\beta}) = \underset{\sim}{x}_i^T\ (X^1X)^{-1}\sigma^2$

$\mathrm{Var}(X\hat{beta})$

$\text{Var}(X(X'X)^{-1}X^T y)$

$$X(X^1X)^{-1}X^T X(X^1X)^{-1}X'\sigma^2$$
$$X(X^1X)X^T\sigma^2$$

Goodness of Fit

Use residuals $\hat{\varepsilon}_i = y_i - \hat{y}_i = y_i - (\hat{a} + \hat{b}x_i)$

Notes

$$\begin{aligned}
\text{E}[\hat{\varepsilon}] &= \text{E}[y - X\hat{\beta}] = \\
&= \text{E}[y - X(X^1X)^{-1}X'y] \\
&= \text{E}(I - X(X^1X)^1X^1)\text{E}(y) \\
&= (I - X(X^1X)^{-1}X^T)X\beta \\
&= X\beta - X\beta = 0
\end{aligned}$$

How do we estimate σ^2

$$\sum_{i=1}^{n} \hat{\varepsilon}_i^2 = \varepsilon^T\varepsilon$$

Note

$\text{E}[\varepsilon^T\varepsilon] = (N-p)\sigma^2$

So $\text{E}\Big[\frac{1}{N-p}\sum_{i=1}^{n}\hat{\varepsilon}_i^2\Big] = \sigma^2$

Regression for Confounding

Say we observe correlation between y and z but suspect it's due to x being confounder

$$\mathrm{E}[y|x.z] = f(x, z)$$

Check if f increasing function of x for all z.

Easy example: $Z = \begin{cases} 0 \\ 1 \end{cases}$

$$f(x, 1)\ ,\ f(x, 0)$$

Model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 z_i + \varepsilon_i$$

Interpret β_2

Note: Z_i is indictor of dummy variable

$$\underset{\sim}{X} = \begin{pmatrix} 1 & x_1 & 0 \\ \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & 0 \\ 1 & x_n & \cdot \end{pmatrix}$$

$[(X^T X)^{-1} X^T y]$ still works!

8.0.1 Example: Baby weights

```
> dat = read.table(file.path(datadir, "babies.data"), header = TRUE)
```

```
> for (i in 1:ncol(dat)) {
+     print(names(dat)[i])
+     print(table(dat[, i]))
+ }
```

```
[1] "bwt"

 55  58  62  63  65  68  69  71  72  73  75  77  78  79  80  81  82  83  8
  1   1   1   1   2   1   1   5   2   1   5   2   3   1   2   3   2   1
 86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 10
  4   7   5   2   5  10   4  10   5   4  12  13  13  16  17  14  19  18
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 12
 12  16  15  20  28  16  24  24  30  36  30  35  21  31  31  24  28  33
126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 14
 24  29  28  34  23  25  21  19  18  13  22  15  18  15   9  12  10  13
146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 16
 10   6   4   3   8   3   6   2   5   6   2   1   5   1   5   1   1   3
166 167 169 170 173 174 176
  1   1   1   1   1   3   1
[1] "gestation"

148 181 204 223 224 225 228 229 232 233 234 235 236 237 238 239 240 241 24
  1   1   1   1   1   2   1   1   3   1   3   1   1   2   3   1   1   2
244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 26
  3   4   6   3   3   4   3   3   5   1   6   9   7   4   6   3   9   7
264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 28
 10  12  15  20  22  18  34  26  24  37  38  40  39  42  44  37  44  42
284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 30
 40  38  41  20  34  27  31  23  29  25  17  13  11  12   9   9   9   5
```

```
304 305 306 307 308 309 310 311 312 313 314 315 316 318 319 320 321 323 32
  3   4   6   3   4   1   1   1   2   3   1   3   2   4   2   1   1   2
329 330 336 338 351 353 999
  1   2   1   1   1   1  13
[1] "parity"

  0   1
921 315
[1] "age"

15 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
 1  7 15 53 58 67 79 93 86 77 90 85 70 66 63 46 39 42 33 30 26 29 18 24 11
42 43 44 45 99
 4  6  1  1  2
[1] "height"

 53  54  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  7
  1   1   1   1  10  26  55 105 131 166 183 182 153 105  54  20  13   6
[1] "weight"

 87  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 10
  1   2   3   1   1   4   2   5   4   3   6   6  18   3  10  15  13  27
108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 12
 16  10  66   9  29  14  10  42  16  21  32  10  55   9  22  15  21  68  1
128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 14
 18  10  78   3  23   9   9  62  11  15   8   4  41   2  10   6   4  42
148 149 150 151 152 153 154 155 156 157 158 159 160 162 163 164 165 168 16
  9   4  31   1   2   2   5  18   5   3   1   4  16   4   1   2   8   1
171 174 175 176 177 178 180 181 182 185 189 190 191 192 196 197 198 200 20
```

```
  1   1   8   1   2   2   7   1   2   4   1   4   1   1   1   1   1   2
215 217 220 228 250 999
  3   1   1   1   1  36
[1] "smoke"

  0   1   9
742 484  10
```

```
> dat[dat == 999] <- NA
> dat$height[dat$height == 99] <- NA
> dat$age[dat$age == 99] <- NA
> dat[dat[, 7] == 9, ] <- NA
```

TO keep it simple we take out the NAs

```
> Index = which(rowMeans(is.na(dat)) == 0)
> dat = dat[Index, ]
```

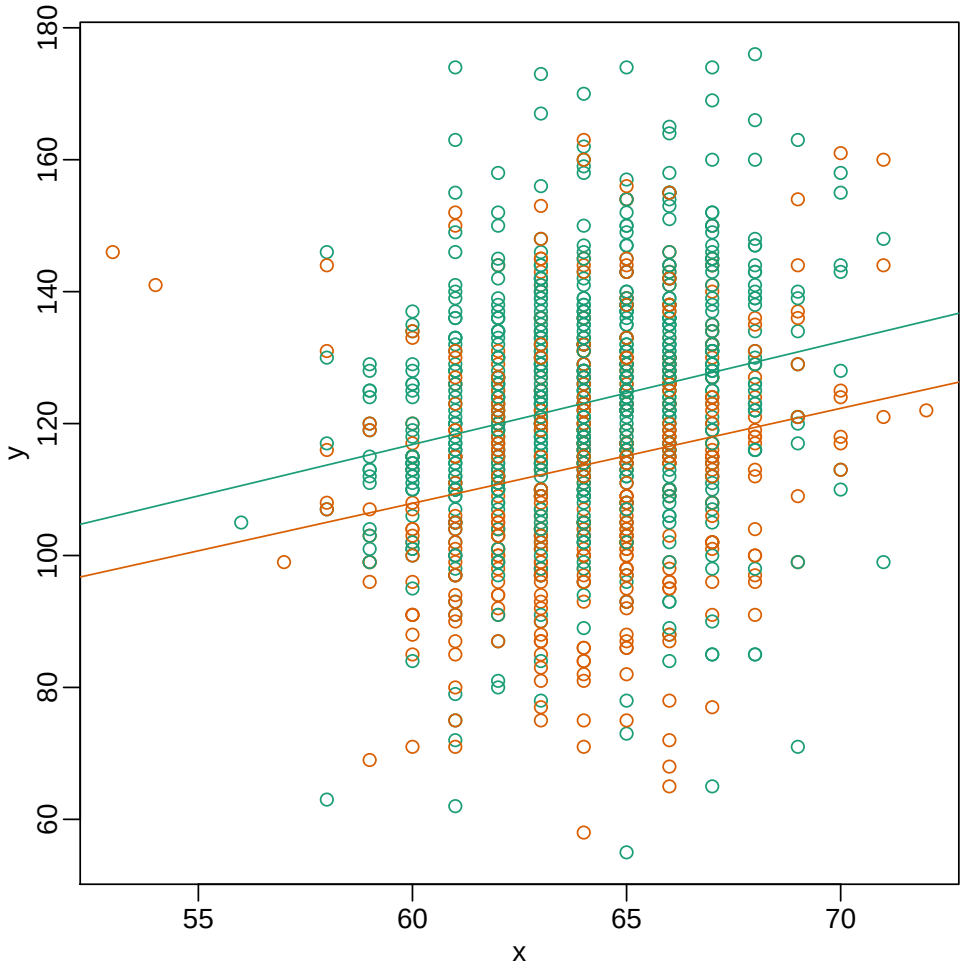
```
> round(cor(dat, use = "complete"), 2)
```

```
            bwt gestation parity   age height weight smoke
bwt        1.00      0.41  -0.04  0.03   0.20   0.16 -0.25
gestation  0.41      1.00   0.08 -0.05   0.07   0.02 -0.06
parity    -0.04      0.08   1.00 -0.35   0.04  -0.10 -0.01
age        0.03     -0.05  -0.35  1.00  -0.01   0.15 -0.07
height     0.20      0.07   0.04 -0.01   1.00   0.44  0.02
```

```
weight     0.16      0.02  -0.10  0.15   0.44   1.00 -0.06
smoke     -0.25     -0.06  -0.01 -0.07   0.02  -0.06  1.00
```

Height a bit confounded with smoking. Clearly height affects weight: tall mothers have heavier babies.

```
> y = dat$bwt
> x = dat$height
> z = dat$smoke
> mypar()
> plot(x, y, col = z + 1)
> abline(lm(y ~ x, subset = z == 1), col = 2)
> abline(lm(y ~ x, subset = z == 0), col = 1)
```



Here are regression fits and plots:

```
> fit1 = lm(bwt ~ smoke, data = dat)
> fit2 = lm(bwt ~ height + smoke, data = dat)
> summary(fit1)


Call:
lm(formula = bwt ~ smoke, data = dat)

Residuals:
```

```
     Min       1Q   Median       3Q      Max
-68.0853 -11.0853   0.9147  11.1808  52.9147

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 123.0853     0.6645 185.221   <2e-16 ***
smoke        -9.2661     1.0628  -8.719   <2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 â

Residual standard error: 17.77 on 1172 degrees of freedom
Multiple R-squared: 0.06091,        Adjusted R-squared: 0.06011
F-statistic: 76.02 on 1 and 1172 DF,  p-value: < 2.2e-16


> summary(fit2)


Call:
lm(formula = bwt ~ height + smoke, data = dat)

Residuals:
      Min        1Q    Median        3Q       Max
-69.57403 -10.08386  -0.05437  10.93089  55.46530

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  26.4350    12.8648   2.055   0.0401 *
height        1.5098     0.2007   7.522 1.07e-13 ***
smoke        -9.4029     1.0386  -9.053  < 2e-16 ***
```

```
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 â

Residual standard error: 17.36 on 1171 degrees of freedom
Multiple R-squared: 0.1042,        Adjusted R-squared: 0.1027
F-statistic:  68.1 on 2 and 1171 DF,  p-value: < 2.2e-16


> summary(lm(y ~ z))


Call:
lm(formula = y ~ z)

Residuals:
     Min       1Q   Median       3Q      Max
-68.0853 -11.0853   0.9147  11.1808  52.9147

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 123.0853     0.6645 185.221   <2e-16 ***
z            -9.2661     1.0628  -8.719   <2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 â

Residual standard error: 17.77 on 1172 degrees of freedom
Multiple R-squared: 0.06091,        Adjusted R-squared: 0.06011
F-statistic: 76.02 on 1 and 1172 DF,  p-value: < 2.2e-16


> summary(lm(y ~ x))
```

```
Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-65.8675 -10.4335   0.6545  11.4360  59.0446

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  24.7963    13.3004   1.864   0.0625 .
x             1.4780     0.2075   7.123 1.84e-12 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 â

Residual standard error: 17.95 on 1172 degrees of freedom
Multiple R-squared: 0.0415,        Adjusted R-squared: 0.04068
F-statistic: 50.74 on 1 and 1172 DF,  p-value: 1.838e-12


> summary(lm(y ~ x + z))


Call:
lm(formula = y ~ x + z)

Residuals:
      Min        1Q    Median        3Q       Max
-69.57403 -10.08386  -0.05437  10.93089  55.46530

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  26.4350    12.8648   2.055   0.0401 *
x             1.5098     0.2007   7.522 1.07e-13 ***
z            -9.4029     1.0386  -9.053  < 2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 â

Residual standard error: 17.36 on 1171 degrees of freedom
Multiple R-squared: 0.1042,        Adjusted R-squared: 0.1027
F-statistic:  68.1 on 2 and 1171 DF,  p-value: < 2.2e-16
```

Model A

Smoking and baby weights

$$y_i = \beta_o + \varepsilon \quad \longrightarrow \quad \begin{array}{l}\text{unexplained} \\ \text{variability}\end{array}$$

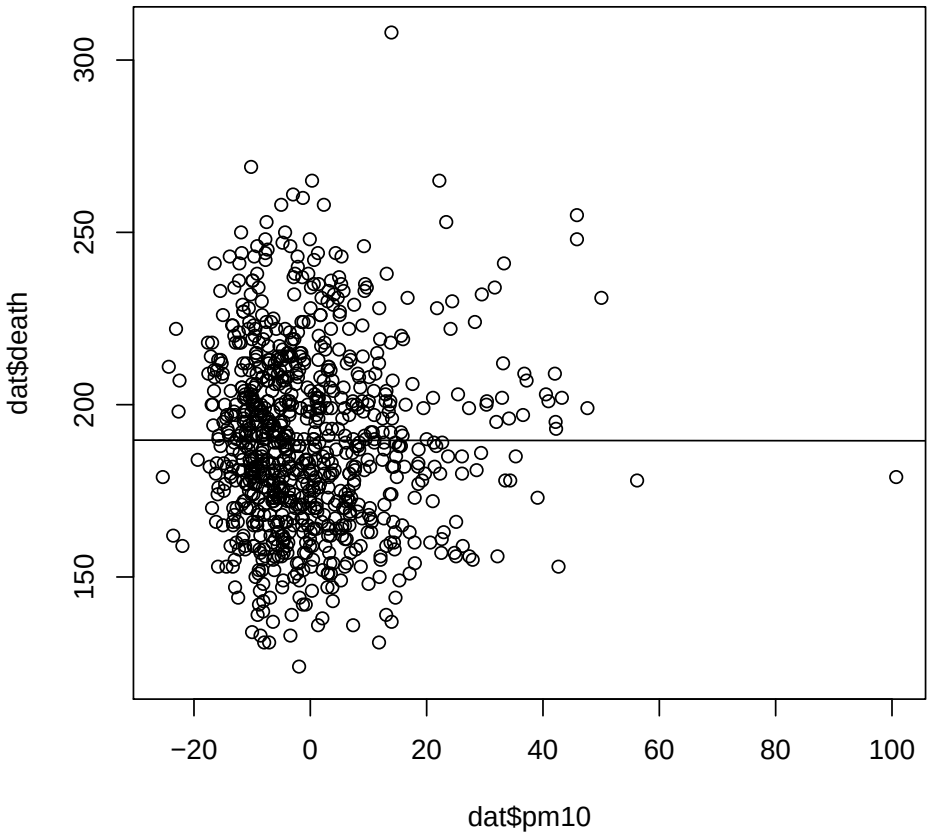
r.m.s is what?

With smoking it goes down from 18.1 to 17.8 or 3% drop.

8.1 Example 2

Does pollution cause mortality?

```
> dat = read.csv(file.path(datadir, "ny-pm10.csv"))
> plot(dat$pm10, dat$death)
> dat = dat[which(!is.na(dat$pm10)), ]
> abline(lm(death ~ pm10, data = dat))
```



The correlation is actually negative! But this could be confounded by temperature

```
> mypar(1, 2)
> plot(dat$temp, dat$pm10)
```

```
> plot(dat$temp, dat$death)
```

Note cold weather appears to kill (more likely viruses) and produce less pollution! How do we correct?

The idea of stratifyng regressions has been extended for cases with continous confounder. It is a bit harder to defende because the linear relationship must be the same for all continous values. Note that if you fix z the slope β_1 does not change.

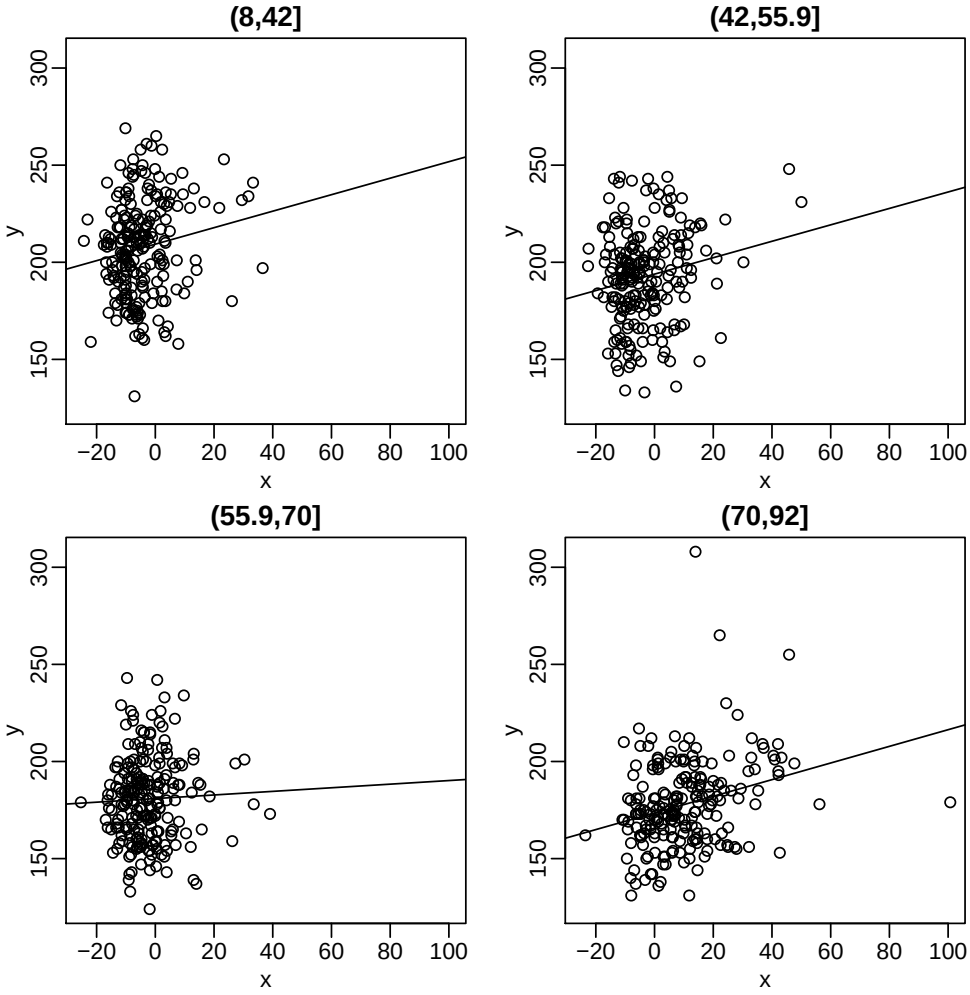
The model is

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \varepsilon_i$$

with x_2 also continuous.

We can check

```
> Indexes = split(1:nrow(dat), cut(dat$temp, quantile(dat$temp,
+     c(0, 0.25, 0.5, 0.75, 1))))
> mypar(2, 2)
> for (i in seq(along = Indexes)) {
+     Index = Indexes[[i]]
+     x = dat$pm10[Index]
+     y = dat$death[Index]
+     plot(x, y, xlim = range(dat$pm10), ylim = range(dat$death),
+         main = names(Indexes)[i])
+     abline(lm(y ~ x))
+ }
```



Note once we stratify, pm10 appears to be positively associated with mortality! If we believe the line above is about the same slope,

```
> summary(lm(death ~ temp + pm10, data = dat))


Call:
lm(formula = death ~ temp + pm10, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-72.247 -15.459   0.873  14.127 137.652
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 236.07073    2.95095  79.998  < 2e-16 ***
temp         -0.83790    0.05139 -16.304  < 2e-16 ***
pm10          0.45440    0.06785   6.697 3.83e-11 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 â

Residual standard error: 22.91 on 861 degrees of freedom
Multiple R-squared: 0.2359,        Adjusted R-squared: 0.2341
F-statistic: 132.9 on 2 and 861 DF,  p-value: < 2.2e-16
```

8.2 GLM

We have $Y_1, \ldots, Y_n$ and covariate $X_1, \ldots, X_n$. But Y's are binary.

$$Y - i = \beta_0 \beta_1 x_i + \varepsilon$$

$\varepsilon \sim 11$?? is hard to conceive. Why?

Scientists, especialy Epi., like to interpret β_1

We generalize the LM (GLM) by the following model

$$g(\mathrm{E}[y|X]) = \beta_o + \beta_1 x$$

ε use to be random, now what?

g is called LINK Function

More notation

$$\eta = \beta_0 + \beta_1 x \text{ more general } \sum_{j=0}^{p} \beta_j X_j X\beta$$

is linear predictor.

For binary data, most popular link function is logic: "logistic regression" comes from this

$$g(x) = \log \frac{P}{1-p} \; \frac{\mathrm{E}(y|x)}{1 - \mathrm{E}[y|X]}$$

NOTE: $g : (0,1) \to \Re$. Also called log logs.

Why? Interprettion.

For every increase in 1 unit of x, logs odds increase β.

$$\begin{aligned} \ell(\underset{\sim}{p}, y) &= \sum_{i=1}^{n} (y_i \log p_i - (1-y_i)\log(1-p_i)) \\ &= \sum_{i=1}^{n} \left[y_i \log \frac{p_i}{1-p_i} - \log(1-p_i) \right] \\ \log \frac{p_i}{1-p_i} &= \beta_0 + \beta_1 x_i = \eta_i \end{aligned}$$

$$\ell(\beta; y| \;=\; \sum_{i=1}^{n} y_i(\beta_0 + \beta_1 x_i) - \sum_{i=1}^{n} \log\left(1 + e^{\beta_0+\beta_1 x}\right)$$

Take derivative and solve?

No closed form!

Let's try it

$$\begin{aligned}
\ell(\underset{\sim}{\beta}\, \underset{\sim}{y}) &= \sum_{i=1}^{N} y_i \beta^T x_i - \log\left(1 + e^{\beta^T x}\right) \\
\frac{\partial \ell}{\partial \beta_j} &= \sum_{i=0}^{N} y_i x_{ij} - \sum \frac{x_{ij} e^{\beta^T x_i}}{1 + e^{\beta^T x_i}} \\
&= \sum y_i x_{ij} - \sum p_i x_{ij} \\
&= \sum_{i=1}^{N} x_{ij}(y_i - p_i). \\
\frac{\partial \ell}{\partial \beta} &= \underset{\sim}{X}^T\, (\underset{\sim}{y} - \underset{\sim}{p}) \\
\frac{\partial \ell}{\partial \beta \partial \beta^T} X^T W X & \\
& w_{ii} = p_i(1 - p_i)
\end{aligned}$$

Newton Raphson

$$\begin{aligned}
\beta^{\text{new}} &= \beta^{\text{old}} + (X^T w X)^{-1} X^T (y - p) \\
\text{if } Z &= x\beta^{\text{old}} + W^{-1}(y - p)
\end{aligned}$$

Then

$$\beta^{\text{new}} = (X^T w x)^{-1} X^T W_Z$$

Weighted version of regression

w depends on β

Z is adjusted response. We minimize $Z - X\beta)^T w(Z - X\beta)$. Think

$Z = X\beta + \varepsilon$ with $\text{Var}(\varepsilon) = w^{1/2}$

Alg call IRLS

$\frac{y-p}{p(1-p)}$ can be plotted act as if "normal" How about goodness-of-fit

We use $\hat{\pi} = g^{-1}(\hat{n}))$ and compare the maximum achievable. In this case $\pi_i = y_i$.

$D = 1 \sum y_i \log\left(\frac{\hat{p_i}}{y_i}\right) + (1 - y_i) \log\left(\frac{1-p_i}{1-y_i}\right)$

This is related to entopy.

9 Analysis of Variance (ANOVA)

Similar to regression but we want to know which "factors" affect variability

We will use data to illustrate. The following experiment is described in chapeter 11. Here is a summary. We are trying to determine if adding pieces of DNA to te mouse genome makes them heavier (weight is one of the symptoms of trisomy). We are trying four different pices of DNA.

```
> dat = read.table(file.path(datadir, "mouse.data"), header = TRUE,
+     as.is = TRUE, comment.char = "")
> colnames(dat)
```

```
[1] "DNA"    "line"   "tg"     "sex"    "age"    "weight" "cage"
```

```
> table(dat[, c(1, 3)])
```

```
   tg
DNA   0   1
  1  73 104
  2  88  70
  3  16  21
  4  79  81
```

Here is a model that puts all varaibility in the random part:

$$y_i \;=\; \beta_0 + \varepsilon_i$$

$$\text{RSS} \;=\; \sum_{i=1}^{n}(y_i - \beta_0)^2$$

As usual we assume the εs are IID.

Note, that sex variability is included in the ε. Is this correct? If sex randomly assigned, then yes! But if we can explain it why not do it?

$$y_i = \beta_0 + \beta_1 \cdot x_i + \varepsilon_i$$

Notes:

- x_i is dummy. 1 if male.
- Var(ε) no longer includes sex variability.
- Be careful with dummy variables in R. R might think its a number. With 0 and 1s numbers are fine though.

Easy to show that LSE are:

$$\begin{aligned}\hat{\beta}_0 &= \bar{Y}_F \\ \hat{\beta}_1 &= \bar{Y}_M - \bar{Y}_F\end{aligned}$$

Use R to see this:

```
> lm(weight ~ sex, data = dat)$coef
```

```
(Intercept)         sex
   26.01011     5.90234
```

```
> tapply(dat$weight, dat$sex, mean)
```

```
       0        1
26.01011 31.91245
```

How about fragments?

$$y_i = \beta_0 + \beta_1 x_{i,1}^{\swarrow\ 14166} + \beta_2 x_{2,i}^{\swarrow 15277} + \beta_3 x_{i,3}^{\swarrow 220E8} + \beta_4 x_{i,4}^{\swarrow 265E6} + \varepsilon \leftarrow \quad \text{sex here!}$$

Fancy way of saying: each group 1 mean

Lets try it in R. We need to create a dummy variable. The *DNA* column stratifies mosue into fragment that we tried to integrate. The *tg* tells us if it actually was integrated.

```
> fragment = rep("No trisomy", nrow(dat))
> DNAlevels = c("141G6", "152F7", "230E8", "285E6")
> tmpIndex = dat$tg == 1
> fragment[tmpIndex] = DNAlevels[dat$DNA[tmpIndex]]
> dat$fragment = factor(fragment, levels = c("No trisomy", DNAlevels))
> summary(lm(weight ~ fragment, data = dat))
```

```
Call:
lm(formula = weight ~ fragment, data = dat)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-7.7214 -2.9672 -0.2415  2.5557 13.1328

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    28.3672     0.2388 118.797  < 2e-16 ***
fragment141G6   0.9944     0.4443   2.238   0.0256 *
fragment152F7   2.6542     0.5153   5.151 3.67e-07 ***
fragment230E8   2.0566     0.8672   2.371   0.0181 *
fragment285E6  -0.2746     0.4871  -0.564   0.5731
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 â

Residual standard error: 3.821 on 527 degrees of freedom
Multiple R-squared: 0.06219,        Adjusted R-squared: 0.05507
F-statistic: 8.736 on 4 and 527 DF,  p-value: 7.816e-07
```

But right now, we do not actually care about the individual *effect sizes*. We want to know is fragment a factor that affects variability.

Sum of Squares

Does fragment variation matter? **Note:** Different from which fragment matters.

Fit model and break up total var

$$\sum_{i=1}^{n}(y_i-\bar{y})^2 = \sum_{i=1}^{n}\overbrace{(\hat{y}_i-\bar{y})^2}+\sum_{i=1}^{n}\overbrace{(y_i-\hat{y}_i)^2}$$
$$= \underbrace{\sum_{g=1}^{G} n_g(\bar{y}_g-\bar{y})^2}_{\text{between group SS}} + \underbrace{\sum_{g=1}^{G}\sum_{(g)}(y_i-\bar{y}_g)^2}_{\text{residual SS}}$$

```
> summary(aov(weight ~ fragment, data = dat))
```

```
             Df Sum Sq Mean Sq F value    Pr(>F)
fragment      4  510.1 127.522  8.7363 7.816e-07 ***
Residuals   527 7692.5  14.597
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 â
```

This analysis demonstrates that fragment is a significant factor.

Note, the same model is fit using the same technique (LSE). But we are interested in something different.

Note we get a p-value for an F-test. What is that?

Mean Square (M.S.) MS defined as $\frac{\text{SS}}{\text{d.f.}}$

d.f. Group - 1 for group
N - group for residual

UNDER NULL $\beta_1 = \ldots = \beta_p - 0$

Group MS / Residual MS follows an F-stat with $G - 1$, $N - G$ degrees of freedom.

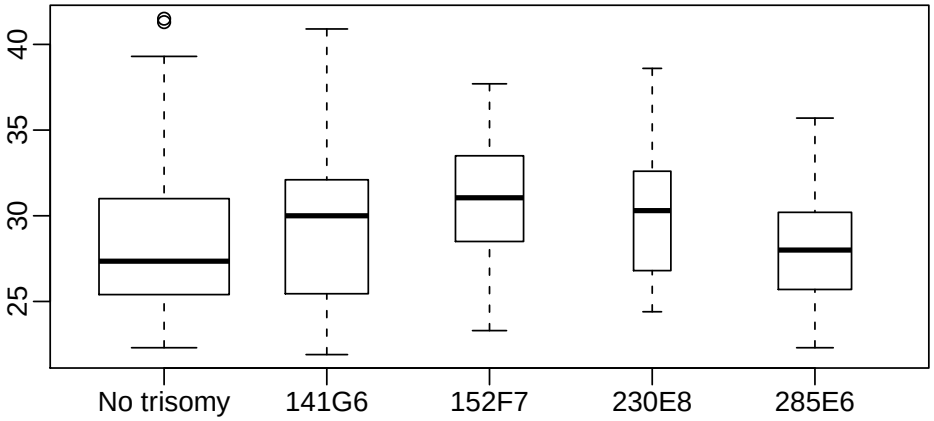
Then variance estimates of between group and within group should be the same. They are both estimating σ^2. The ratio of the M.S.s should be the same. Under null and assuming errors are normal it follows an F-distribution.

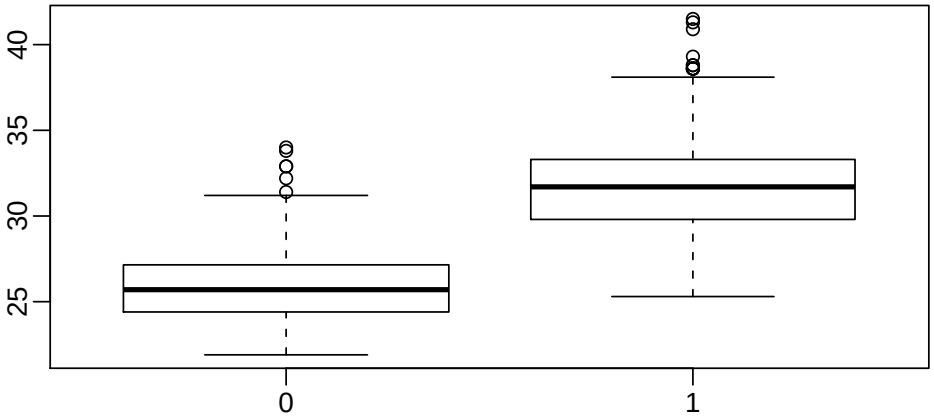
This is called one-way classification ANOVA.

9.1 Exploratory plots

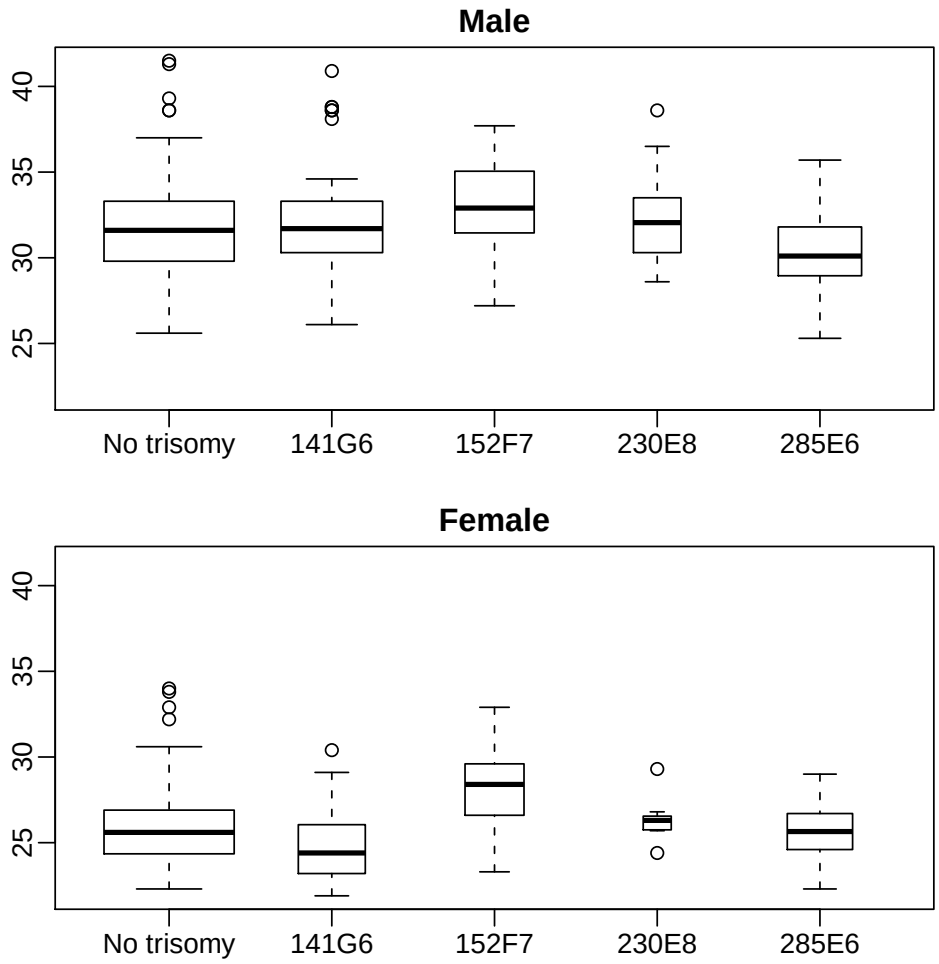
Stratify!

```
> YLIM = range(dat$weight)
> mypar(2, 1)
> boxplot(split(dat$weight, dat$fragment), varwidth = TRUE, ylim = YLIM)
> boxplot(split(dat$weight, dat$sex), varwidth = TRUE, ylim = YLIM)
```






```
> Index = dat$sex == 1
> mypar(2, 1)
> boxplot(split(dat$weight[Index], dat$fragment[Index]), varwidth = TRUE
+     main = "Male", ylim = YLIM)
> boxplot(split(dat$weight[!Index], dat$fragment[!Index]), varwidth = TR
+     main = "Female", ylim = YLIM)
```



Alternative Parameterization

If we don't like having a *reference*, e.g. the no trisomy group in the previous example, we can have a mean for each group.

$i =$ group $j =$ replicate

$Y_{ij} = \mu + a_i + \varepsilon_{ij}$

There are an infinite LSE solutions for $\mu, a_1, \ldots, a_p$

Not identifable

Assume $\sum a_i = 0$ then OK

Two-way Classification

Say an investigator wants to learn about the effects of sex an fragment on weight. Good experimental desing permits us to examine both at once! We can fit 1 model and learn about more than one factor. Power comes from pooling variance.

Ex. Sex (X_1) and Transgenic (X_2)

To illustrate, we focus on just one fragment.

The model is:

$$\boxed{Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon}$$

The Xs are both dummy variables.

What if only affect males?

$$\boxed{Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon}$$

We usually write like this **if balanced**. We can write:

$$
\begin{aligned}
y_{ijk} &= \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + \varepsilon_{ijk} \\
\hat{\mu} &= \bar{y}_{1\cdot 1} \; \hat{\alpha}_i = \bar{y}_{i\cdot\cdot} - \bar{y} \; \hat{\beta}_j = \bar{y}_{\cdot j\cdot} - \bar{y} \\
\widehat{(\alpha\beta)}_{ij} &= \bar{y}_{ij\cdot} - \bar{y}_{i\cdot\cdot} - \bar{y}_{j\cdot\cdot} + \bar{y}
\end{aligned}
$$

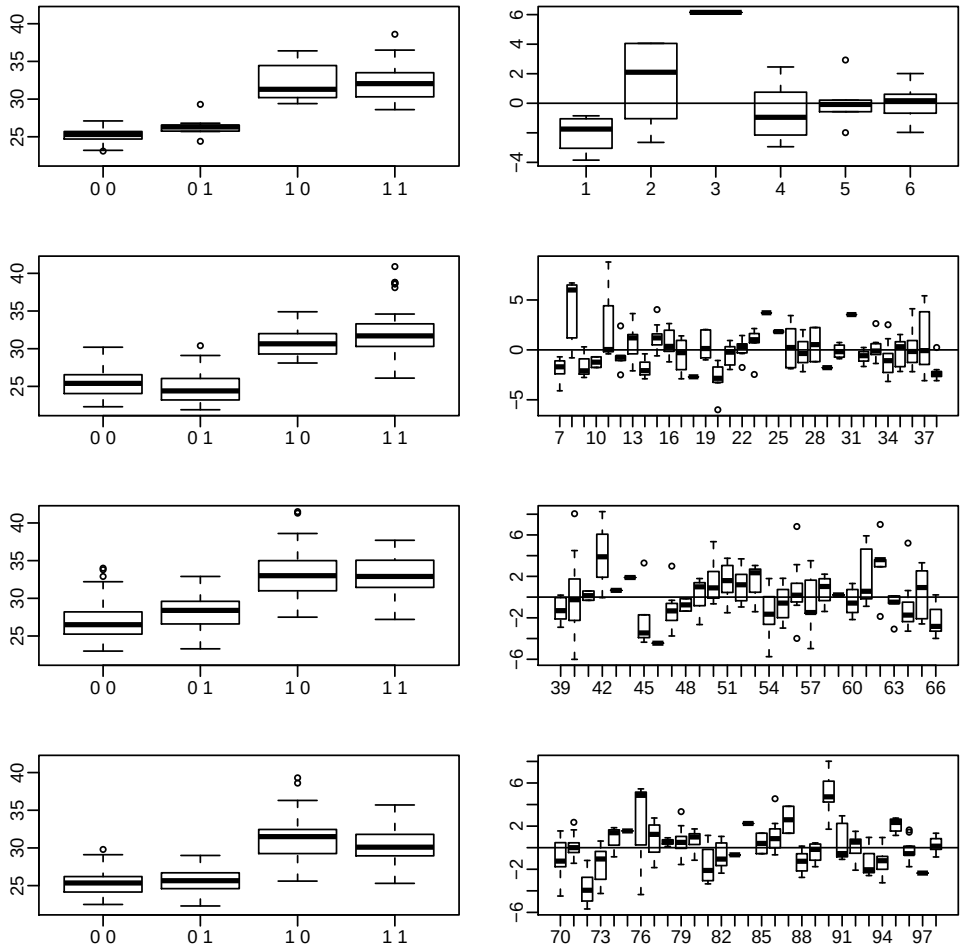
The sum of squares can be broken up into:

$$
\begin{aligned}
\sum_i \sum_j \sum_k (y_{ijk} - \bar{y})^2 &= JK \sum_i (y_{i\cdot\cdot} - \bar{y})^2 \\
&+ IK \sum_j (y_{\cdot j\cdot} - \bar{y})^2 \\
&+ K \sum_i \sum_j (y_{ij\cdot} - y_i - y_{\cdot j} + \bar{y})^2 \\
&+ \sum_i \sum_j \sum_j (y_{ij} - \hat{y}_{ijk})^2
\end{aligned}
$$

Dfs are $I - 1, J - 1, (I - 1)(J - 1), IJ(K - 1)$

```
> mypar(4, 2)
> for (i in unique(dat$DNA)) {
+     Index = dat$DNA == i
+     summary(aov(weight ~ sex + tg, data = dat, subset = Index))
+     summary(lm(weight ~ sex + tg, data = dat, subset = Index))
+     boxplot(split(dat$weight[Index], paste(dat$sex, dat$tg)[Index]),
+         ylim = YLIM)
+     summary(aov(weight ~ sex * tg, data = dat, subset = Index))
+     summary(lm(weight ~ sex * tg, data = dat, subset = Index))
+     boxplot(split(lm(weight ~ sex * tg, data = dat, subset = Index)$re
+         dat$cage[Index]))
```

```
+     abline(h = 0)
+ }
```



Notes

– Fancy way of saying:

Every group has a mean

– ANOVA breaks it in

FACTORS

– Linear models tells us effect of levels

– Droping terms gives us power

9.2 2008 Poll Data

Polls in the US are rather accurate. However, their are many polls and they never agree. Usually at least one of them gets them right. But how to know which one? Here we use ANOVA to combine data from all polls and provide an improved confidence interval.

We got this data from a nice website `http://www.fivethirtyeight.com/`

```
> tab = read.delim("http://www.biostat.jhsph.edu/bstcourse/bio751/data/p
+     as.is = TRUE)
> Index08 = grep("08", tab$Dates)
> Index07 = grep("07", tab$Dates)
> year = rep(0, nrow(tab))
> year[Index08] = "2008"
> year[Index07] = "2007"
> d = sapply(strsplit(tab$Dates, "-"), function(x) x[1])
> d = gsub("/08", "", d)
> d = paste(year, d, sep = "/")
> d = strptime(d, format = "%Y/%m/%d")
> d = d - strptime("2008/11/4", format = "%Y/%m/%d")
> tab$day = d
> week = round(d/7)
> tab$week = week
> tab$diff = tab$Obama - tab$McCain
```

Note that pollsters take many polls, but some only do one or two. So let's keep only those with more than 10

```
> tmp1 = table(week)
> N1 = 3
> keepIndex = week %in% as.numeric(names(tmp1)[tmp1 >= N1])
> tab2 = tab[keepIndex, ]
> tmp2 = table(tab2$Pollster)
> N2 = 10
> keepIndex = tab2$Pollster %in% names(tmp2)[tmp2 >= N2]
> tab2 = tab2[keepIndex, ]
```

What happens if we just take mean?

```
> mean(y) + c(-2, 2) * sd(y)/sqrt(length(y))
```

```
[1] 174.7678 180.9693
```

Would we get it right? No the difference was 7!

What are the factors here?

Let's try some ANOVA

```
> week = factor(-tab2$week)
> pollster = factor(tab2$Pollster)
> contrasts(pollster) = contr.sum
> y = tab2$diff
> lm1 = lm(y ~ week + pollster)
> summary(lm1)
```

```
Call:
lm(formula = y ~ week + pollster)

Residuals:
      Min        1Q    Median        3Q       Max
-10.09343  -1.43209  -0.04693   1.79186   8.49279

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.3962     1.3865   3.171 0.001801 **
week6        -0.1626     1.7581  -0.092 0.926426
week7        -1.9790     1.6461  -1.202 0.230921
week8        -5.5376     1.7581  -3.150 0.001928 **
week9        -3.0939     1.6892  -1.832 0.068748 .
week10       -0.8804     1.7847  -0.493 0.622404
week11       -1.7194     1.7259  -0.996 0.320530
week12       -1.4894     1.8882  -0.789 0.431302
week13       -0.7809     1.9684  -0.397 0.692088
week14       -1.1797     1.8289  -0.645 0.519761
week15        0.1615     1.7723   0.091 0.927517
week16       -0.9809     1.9684  -0.498 0.618914
week17       -1.6074     1.8731  -0.858 0.392013
week18        1.0191     1.9684   0.518 0.605300
week19       -0.1579     1.8173  -0.087 0.930857
week20        0.7387     1.8308   0.404 0.687076
week21        0.1891     1.8141   0.104 0.917120
week22       -1.2059     1.9650  -0.614 0.540233
week23       -5.2119     1.9772  -2.636 0.009158 **
week24       -3.4148     1.8742  -1.822 0.070196 .
```

```
week25       -4.9298     2.0729  -2.378 0.018492 *
week26       -2.1224     1.9551  -1.086 0.279183
week27       -4.3268     1.7773  -2.435 0.015933 *
week28       -4.7180     2.0855  -2.262 0.024932 *
week29       -3.9898     1.8813  -2.121 0.035372 *
week30       -3.5224     1.9551  -1.802 0.073350 .
week31       -5.4680     2.0855  -2.622 0.009529 **
week32       -7.3421     1.8056  -4.066 7.26e-05 ***
week33       -8.0449     2.2663  -3.550 0.000497 ***
week34       -4.8621     1.9830  -2.452 0.015208 *
week35       -3.5022     2.2644  -1.547 0.123792
week36       -3.2329     1.9471  -1.660 0.098673 .
week37       -4.0966     2.2926  -1.787 0.075713 .
week38        2.3136     2.0797   1.112 0.267502
week39        1.2432     2.1029   0.591 0.555156
week40       -3.3137     2.2732  -1.458 0.146739
week43       -5.9033     2.3079  -2.558 0.011393 *
week80        5.0940     3.4265   1.487 0.138941
week94       -1.7386     2.5912  -0.671 0.503138
pollster1     1.3439     0.9193   1.462 0.145613
pollster2     1.2205     0.9092   1.342 0.181235
pollster3     0.1883     0.8630   0.218 0.827522
pollster4     1.5098     0.7673   1.968 0.050701 .
pollster5    -1.0309     0.5713  -1.804 0.072940 .
pollster6    -0.3639     0.4738  -0.768 0.443492
pollster7    -1.6589     0.4373  -3.794 0.000205 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 â
```

```
Residual standard error: 3.057 on 172 degrees of freedom
Multiple R-squared: 0.4789,        Adjusted R-squared: 0.3426
F-statistic: 3.513 on 45 and 172 DF,  p-value: 1.800e-09
```

Now the CI is (1.6,7.2)

Note that this is not a balances experiment. So the SS don't break up nicely. As a result you get different resutls when you change order:

```
> summary(aov(y ~ week + pollster))

             Df  Sum Sq Mean Sq F value    Pr(>F)
week         38 1270.56  33.436  3.5786 6.333e-09 ***
pollster      7  206.53  29.504  3.1578  0.003635 **
Residuals   172 1607.04   9.343
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 â

> summary(aov(y ~ pollster + week))

             Df  Sum Sq Mean Sq F value    Pr(>F)
pollster      7  270.99  38.712  4.1433 0.0003087 ***
week         38 1206.10  31.740  3.3970 2.665e-08 ***
Residuals   172 1607.04   9.343
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 â
```

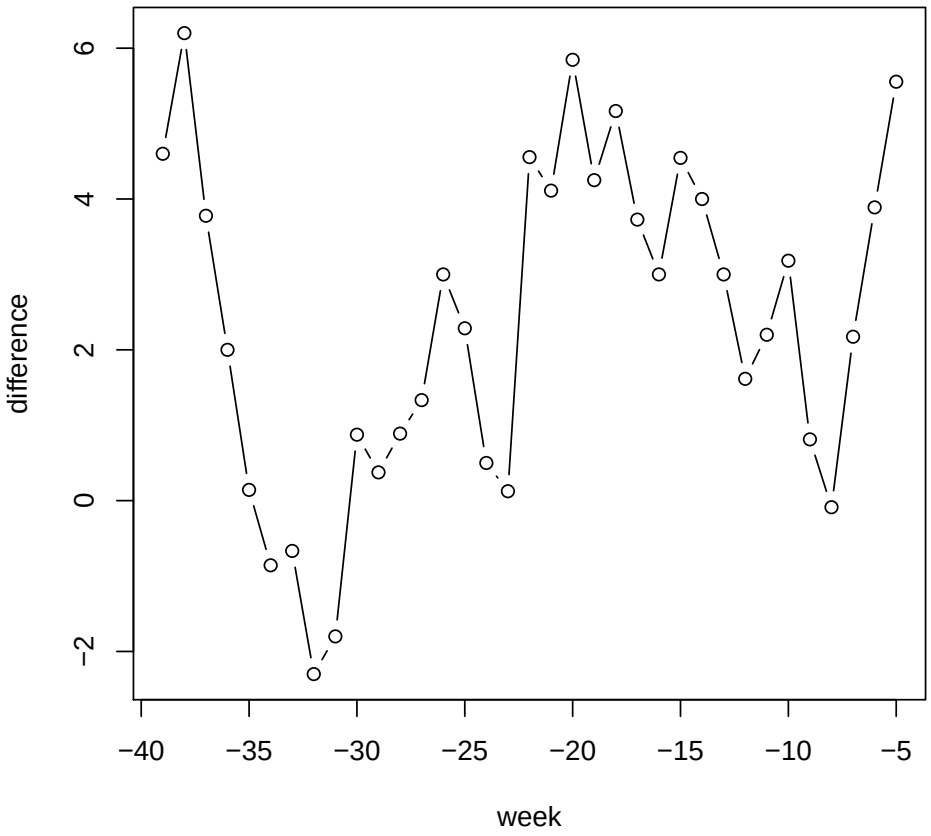
10 Smoothing

Let's look at the pool data again:

```
> tab = read.delim("http://www.biostat.jhsph.edu/bstcourse/bio751/data/p
+     as.is = TRUE)
> Index08 = grep("08", tab$Dates)
> Index07 = grep("07", tab$Dates)
> year = rep(0, nrow(tab))
> year[Index08] = "2008"
> year[Index07] = "2007"
> d = sapply(strsplit(tab$Dates, "-"), function(x) x[1])
> d = gsub("/08", "", d)
> d = paste(year, d, sep = "/")
> d = strptime(d, format = "%Y/%m/%d")
> d = d - strptime("2008/11/4", format = "%Y/%m/%d")
> tab$day = d
> week = round(d/7)
> tab$week = week
> tab$diff = tab$Obama - tab$McCain
> tab = tab[tab$week > -40, ]
```

Now, ignoring pollster, lets look at the week effect.

```
> y = tapply(tab$diff, tab$week, mean)
> x = as.numeric(names(y))
> plot(x, y, xlab = "week", ylab = "difference", type = "b")
```



Note the spikes. Do you think those are real trends? If not, what are they?

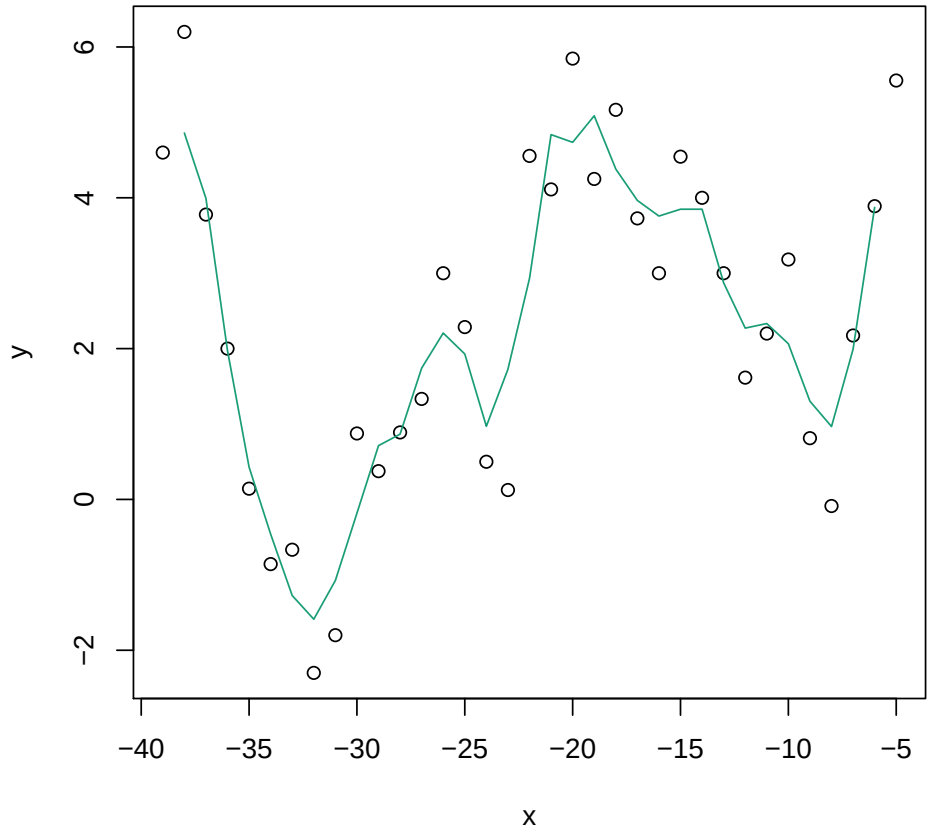
Because weeks is a quantitative covariate we may try a *smoothing technique*. In smoothing we use Taylor's approximation: any function can be locally approximated by a polynomial.

Here we are interested in

$$\mathrm{E}[\text{difference}|\text{weeks}] = f(\text{weeks})$$

Example: assume that every three weeks actually has roughly same f. Here we assume a local constant.

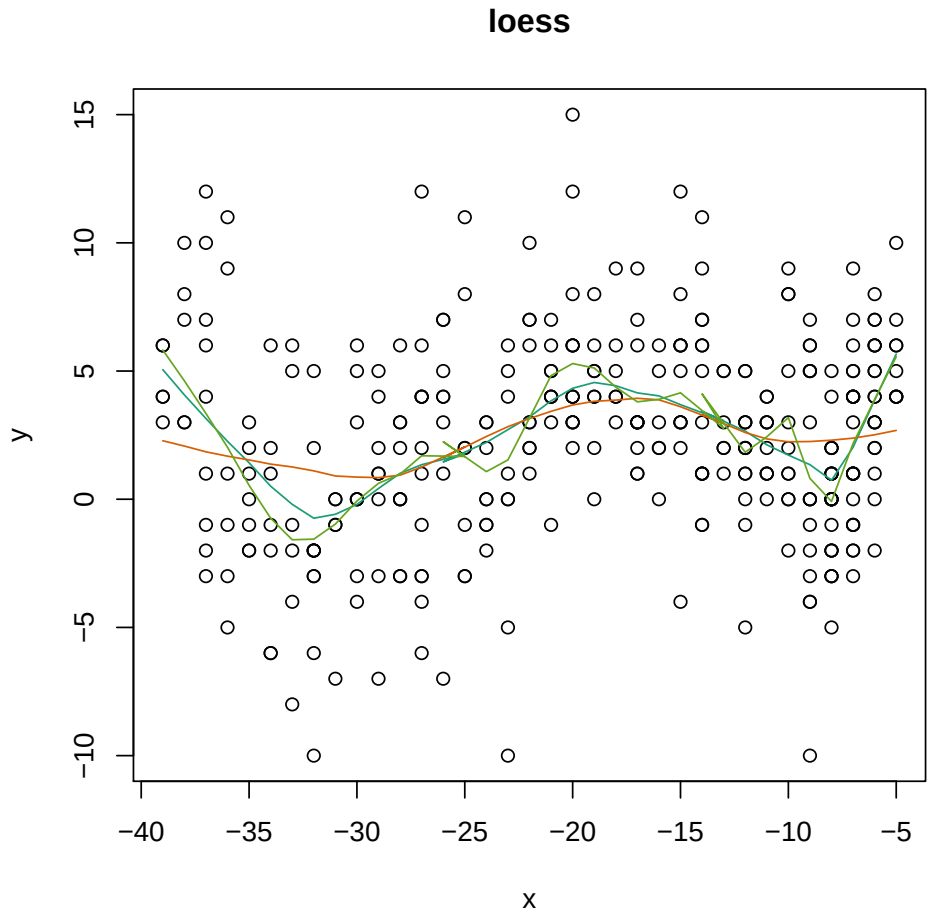
```
> s = filter(y, filter = rep(1, 3)/3)
> plot(x, y)
> lines(x, s, col = 1)
```



There are better approaches. For exwmple, we can fit lines locally (loess) or cubic splines. Why not higher order polynomials?

ADD LEGEND

```
> y = tab$diff
> x = tab$week
> plot(x, y, main = "loess")
> fit = loess(y ~ x, span = 0.25, degree = 1)
> lines(unique(x), predict(fit, unique(x)), col = 1)
> fit = loess(y ~ x, span = 0.5, degree = 1)
> lines(unique(x), predict(fit, unique(x)), col = 2)
> fit = loess(y ~ x, span = 0.2, degree = 2)
> lines(unique(x), predict(fit, unique(x)), col = 5)
```



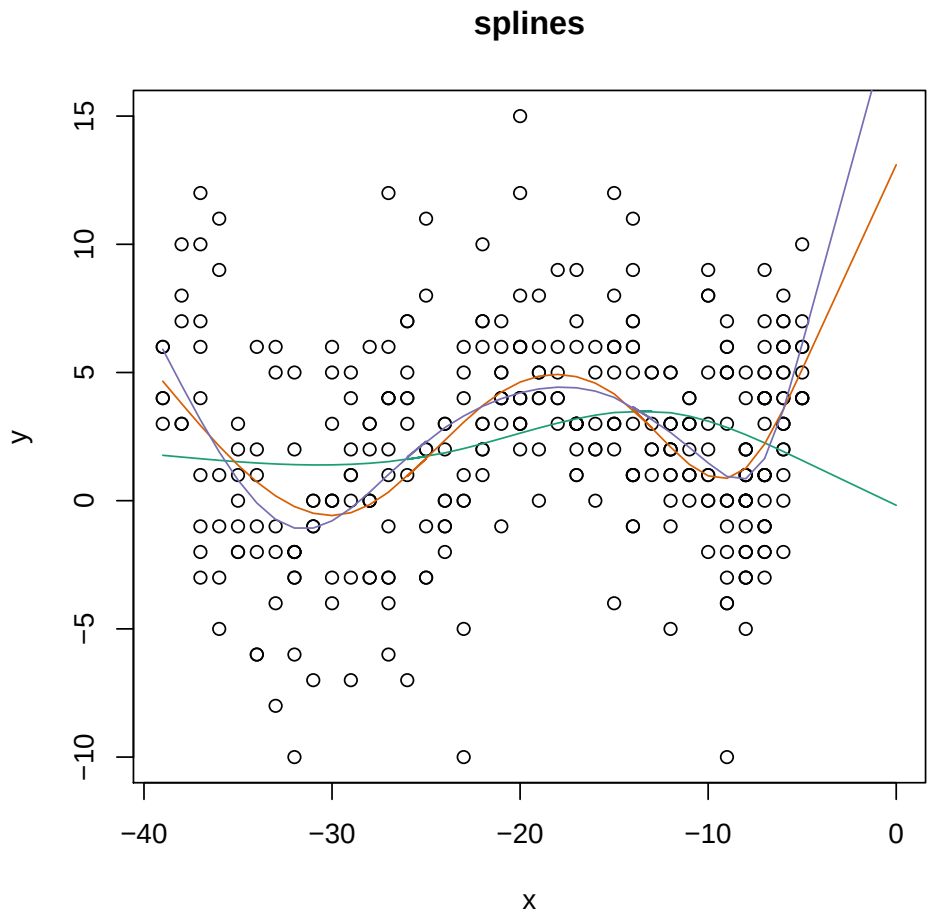
Statisticians are particularly fond of splines. Just like polynomials, they can be represented as linear functions of the covariate (in this case it's x).

$$f(x) = \sum_{j=1}^{J} \theta_j B_j(x)$$

How do we get $B_j(x)$? It is a bit intimidating but conceptually pretty easy. In any case, we can fit with least squares.

ADD LEGEND

```
> library(splines)
> xx = c(0, unique(x))
> plot(x, y, xlim = range(xx), main = "splines")
> fit = lm(y ~ ns(x, 3))
> lines(xx, predict(fit, list(x = xx)), col = 1)
> fit = lm(y ~ ns(x, 5))
> lines(xx, predict(fit, list(x = xx)), col = 2)
> fit = lm(y ~ ns(x, 7))
> lines(xx, predict(fit, list(x = xx)), col = 3)
```



Note, we can also include in models.

```
> tmp = table(tab$Pollster)
> N = 10
> keepIndex = tab$Pollster %in% names(tmp)[tmp >= N]
> tab = tab[keepIndex, ]
> y = tab$diff
> x = tab$week
> pollster = factor(tab$Pollster)
> fit = lm(y ~ ns(x, 5) + pollster)
```

```
> predict(fit, list(x = 0, pollster = pollster[4]), se.fit = TRUE)


$fit
       1
6.056862

$se.fit
[1] 2.655278

$df
[1] 180

$residual.scale
[1] 3.068298
```

11 Bayes Rule

Say a test is 99% accuracy:

If we test someone at random, and if it ??, what is the chance they have disease.
e.g. cystic fibrosis
1 in 3,900

What we know:

$\Pr(+|D) = 0.99$, $\Pr(+|H) = 0$.

$\Pr(D) = 0.00025$

We want

We want $\Pr(|+)$ not $\Pr(+|D)$

Bayes Theorem

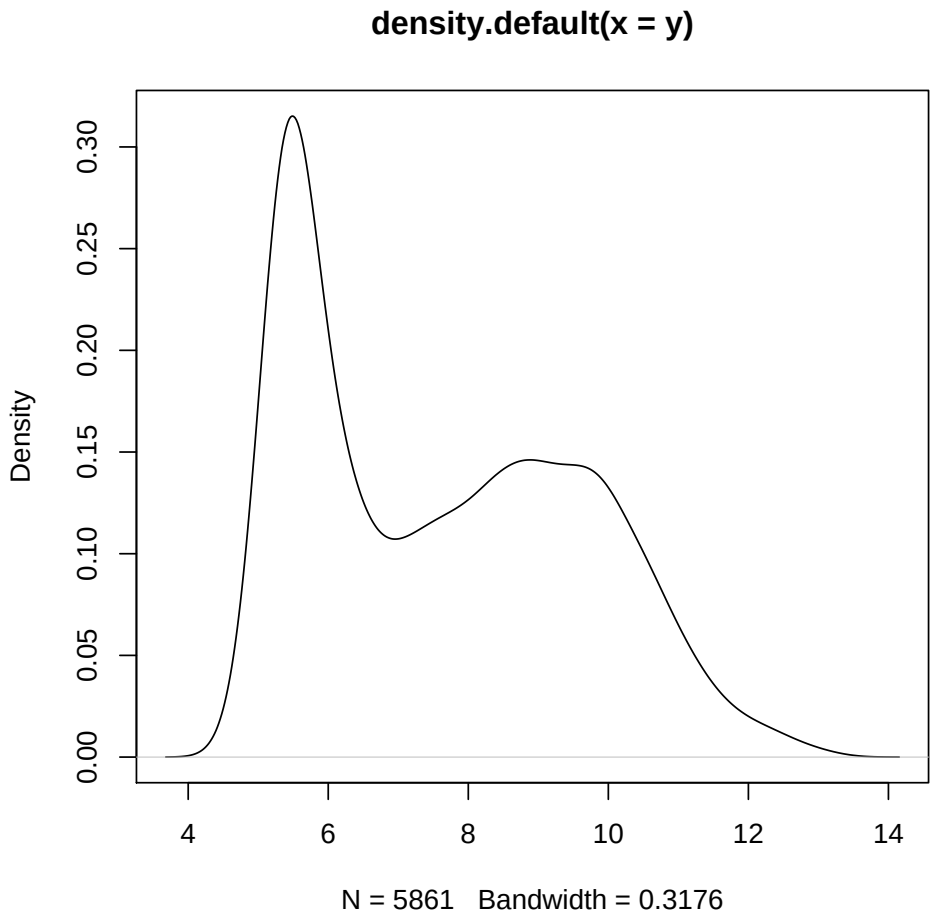
$$\begin{aligned} P(A|B) &= \frac{P(B|A)P(A)}{P(B)} \\ \text{So} & \\ \Pr(D|+) &= \frac{P(+|D) \cdot P(D)}{\Pr(+)} \\ &= \frac{\Pr(+|D) \cdot P(D)}{\Pr(+|D) \cdot P(D) + \Pr(+|D^C)\Pr(D^C)} \\ &= \frac{0.99 \cdot 0.0025}{0.99 \cdot 0.0025 + 0.01 \cdot (.9975)} \end{aligned}$$

$$= \quad 0.02 \quad \text{not} \quad 0.99$$

11.1 Yeast mutants

Each point is the growth of a yeast mutant. Which ones are alive and which one dead? With yeast mixture model.

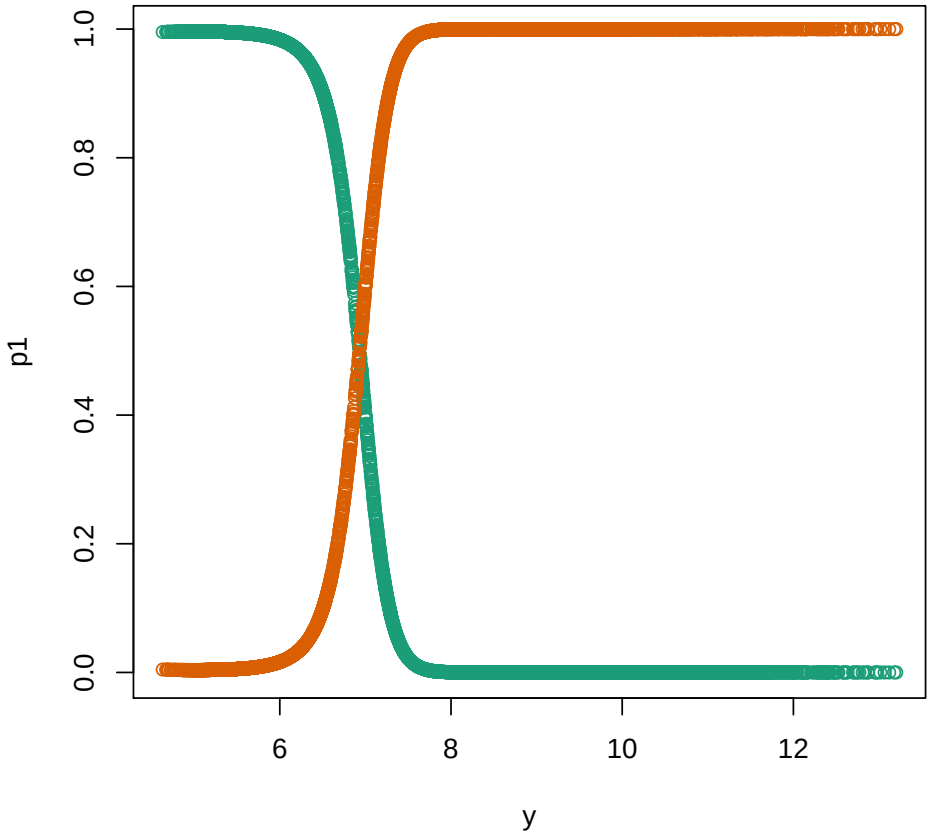
```
> dat <- read.csv(file.path(datadir, "microarray.csv"))
> y = log2(dat[, 3] - 64)
> plot(density(y))
> hist(y, nc = 35)
```



Looks like maybe two normals. But how do we fit it? Right out the likelihood.

```
> z1 = dnorm(y, 6, 0.5)
> z2 = dnorm(y, 9, 1)
> prior = 0.25
> p1 = prior * z1/(prior * z1 + (1 - prior) * z2)
> p2 = 1 - p1
> plot(y, p1, col = 1)
> points(y, p2, col = 2)
```

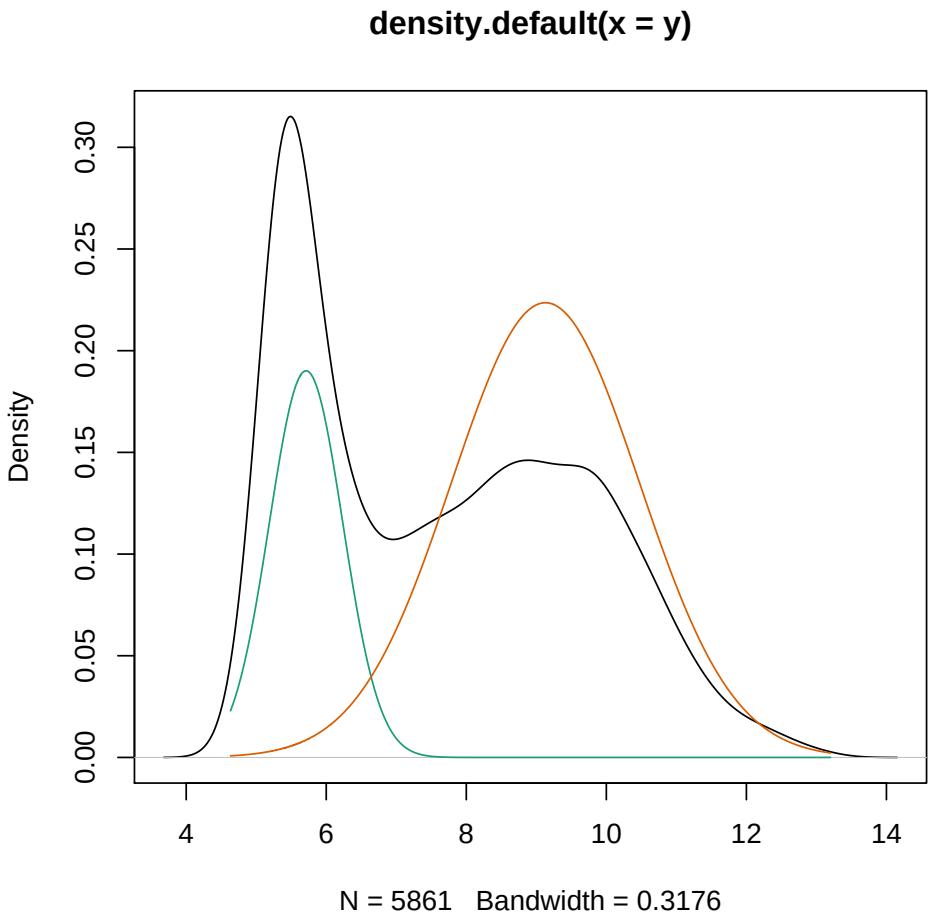
```
> mu1 = sum(p1 * y)/sum(p1)
> mu2 = sum(p2 * y)/sum(p2)
> sd1 = sqrt(sum(p1 * (y - mu1)^2)/sum(p1))
> sd2 = sqrt(sum(p2 * (y - mu2)^2)/sum(p2))
```



How does it work?

```
> plot(density(y))
> x = sort(y)
> lines(x, prior * dnorm(x, mu1, sd1), col = 1)
```

```
> lines(x, (1 - prior) * dnorm(x, mu2, sd2), col = 2)
```



$$
\begin{aligned}
\Pr(\text{Essential}) &= \pi_0 \\
\Pr(y \leq c|\text{Essential}) &= \Phi\left(\frac{c-\mu_0}{\sigma_0}\right) \\
\Pr(y \leq c|\text{Non-essential}) &= \Phi\left(\frac{c-\mu_1}{\sigma_1}\right)
\end{aligned}
$$

We can write like this

$$
y_i = (1 - Z_i)y_0 + Z_i y_1
$$

$$
\begin{aligned}
y_0 &\sim N(\mu_0, \sigma_0^2) \\
y_1 &\sim N(\mu_1, \sigma_1^2) \\
\Pr(Z = 1) &= \pi
\end{aligned}
$$

We don't see Z_i. We need to estimate $\pi, \mu_0, \sigma_1^2, \mu_1, \sigma_1^2$.

Then we can use Bayes

$$
\Pr(Z = 1|y) = \frac{\pi f_1(y)}{\pi f_1(y)(1-\pi) f_0(y)}
$$

12 Helicopter

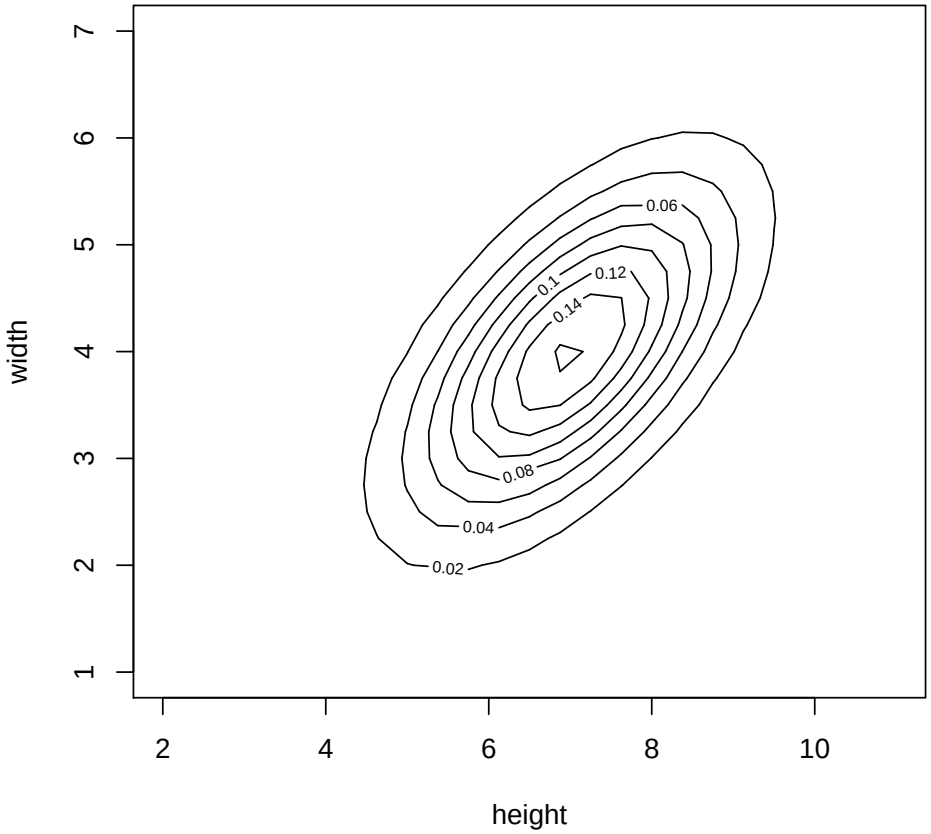
$$\text{Model}: \quad y = t(h, w) + \varepsilon \boxed{\text{What is } \varepsilon?}$$
$$\text{We want: } \max_{h,w} t(h, w)$$

t is for time.

We don't know t!

We is a plot of one possibility for t (the code is used only to make an illustrative plot, it's not the actualt truth):

```
> library(mvtnorm)
> nr = 25
> nc = 25
> h = seq(2, 11, len = nr)
> w = seq(1, 7, len = nc)
> x = expand.grid(h, w)
> z = dmvnorm(x, c(7, 4), cbind(c(1.5, 1.5 * 0.5), c(1.5 * 0.5,
+     1)))
> mat = matrix(z, nrow = nr, ncol = nc)
> contour(h, w, mat, xlab = "height", ylab = "width")
```



Linear? No way. Locally linear? **Yes**!

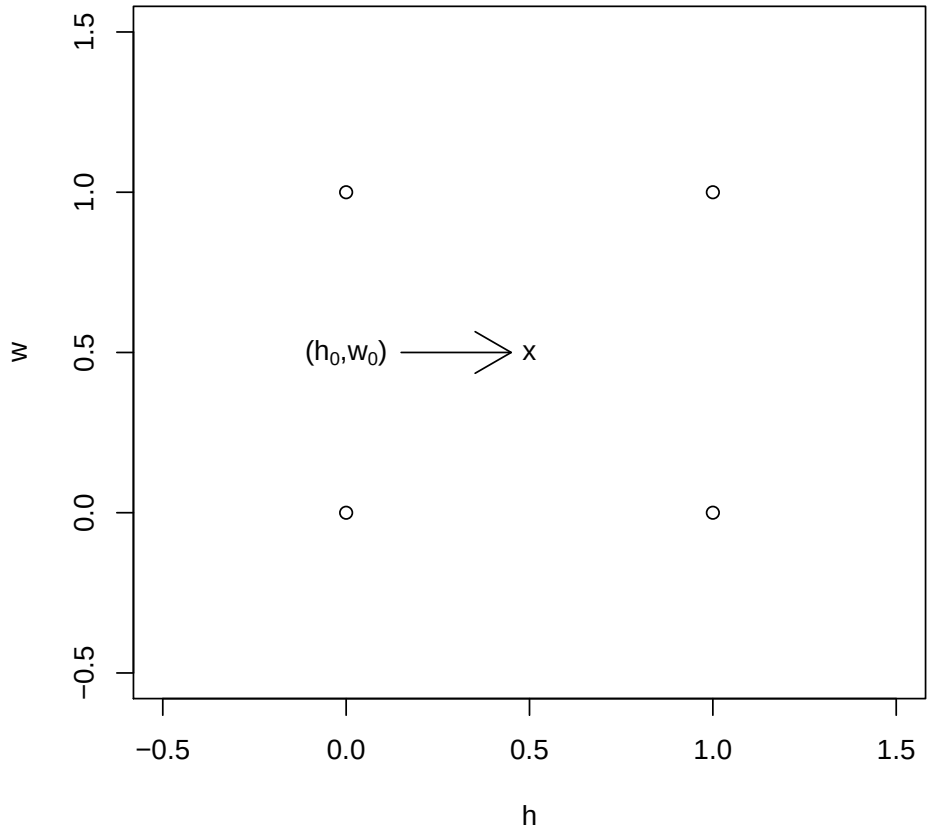
$$t(w, h) = \beta_0 + \beta_1 w + \beta_2 h + \varepsilon$$

for w, h in small regions. Should we replicate? It helps determine var(ε)

So how do we find max? Pick a starting value (h_0, w_0). Then estimate the plane around it. To do this we pick 4 points. Which 4?

```
> plot(c(0, 0, 1, 1), c(0, 1, 0, 1), xlab = "h", ylab = "w", xlim = c(-0
```

```
+     1.5), ylim = c(-0.5, 1.5))
> points(1/2, 1/2, pch = "x")
> text(0, 0.5, expression(paste("(", h[0], ",", w[0], ")", sep = "")))
> arrows(0.15, 0.5, 0.45, 0.5)
```



Regression theory tells us that the best design is to make square as big as possible.

But if too big, then local linearity no longer holds.

So we fit the model locally and get: $\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2$

Now what? Does the fitted plane give us the max? No. But it tells us which direction to follow.

The plane we fitted is

$$\hat{\beta}_0 + \hat{\beta}_1 w + \hat{\beta}_2 h = c$$

and the contour lines are

$$h = \frac{c - \hat{\beta}_0}{\hat{\beta}_2} - \frac{\hat{\beta}_1}{\hat{\beta}_2} w$$

Note that the slope is $-\hat{\beta}_1/\hat{\beta}_2$. The gradient is perpendicular, so slope is $\hat{\beta}_2/\hat{\beta}_1$. We move in that direction.

How far? We can try a few points from $h = h_0 + \frac{\hat{\beta}_2}{\hat{\beta}_1}(w - w_0)$

New center will be at

$$h - h_0 = \frac{\hat{\beta}_2}{\hat{\beta}_1}(w - w_0)$$

if $\hat{\beta}_1$ is positive make $w > w_0$.

Try a few until stop growing. At the end we must be close so instead of plane we fit a paraboloid.

```
> tmp = expand.grid(seq(0, 1, 0.5), seq(0, 1, 0.5))
> plot(tmp[, 1], tmp[, 2], xlab = "h", ylab = "w", xlim = c(-0.5,
+     1.5), ylim = c(-0.5, 1.5))
> points(1/2, 1/2, pch = "x", cex = 2)
> text(-0.25, 1/3, expression(paste("(", h[0], ",", w[0], ")",
+     sep = "")))
> arrows(-0.1, 1/3, 0.45, 0.5)
```

Now we fit the parabola:

$$y_i = \beta_0 + \beta_1 w + \beta_2 h + \beta_3 w^2 + \beta_4 h^2 + \beta_5 wh + \varepsilon$$

Fit and then maximize.