Statistics Teaching

Adam Kane

4 November 2016

```
## Warning: package 'HH' was built under R version 3.3.2
## Warning: package 'multcomp' was built under R version 3.3.2
## Warning: package 'mvtnorm' was built under R version 3.3.2
## Warning: package 'survival' was built under R version 3.3.2
## Warning: package 'TH.data' was built under R version 3.3.2
## Warning: package 'psych' was built under R version 3.3.2
## Warning: package 'multilevel' was built under R version 3.3.2
## Warning: package 'lsr' was built under R version 3.3.2
## Warning: package 'car' was built under R version 3.3.2
## Warning: package 'QuantPsyc' was built under R version 3.3.2
```

Variables

A variable is something that can take on different values e.g. height is a variable. The opposite of variables are constants e.g. the gravitational constant which has one value only.

Types of variables (NOIR)

In statistics we can consider 4 variable types as set out by Stevens (1946):

Nominal variables

are variables that have two or more categories, but which do not have an intrinsic order. For example, classifying where people live in the USA by state. In this case there will be 50 'levels' of the nominal variable.

```r
nominalVariables <- c("Alaska", "Florida", "New York", "Washington", "Texas")
```

Ordinal variables

are variables that have two or more categories just like nominal variables only the categories can also be ordered or ranked. So if you asked someone if they liked the policies of the Democratic Party and they could answer either "Not very much", "They are OK" or "Yes, a lot" then you have an ordinal variable. Why? Because you have 3 categories, namely "Not very much", "They are OK" and "Yes, a lot" and you can rank them from the most positive (Yes, a lot), to the middle response (They are OK), to the least positive (Not very much). However, whilst we can rank the levels, we cannot place a "value" to them; we cannot say that "They are OK" is twice as positive as "Not very much" for example.

```r
ordinalVariables <- c("OK", "Not very much", "OK", "Yes, a lot", "Not very much")
```

Interval variables

are variables for which their central characteristic is that they can be measured along a continuum and they have a numerical value (for example, temperature measured in degrees Celsius or Fahrenheit). So the difference between 20C and 30C is the same as 30C to 40C. However, temperature measured in degrees Celsius or Fahrenheit is NOT a ratio variable. In interval scales, addition and subtraction make sense, but multiplication and division do not. That is, 70C is not "twice as hot"" as 35C. If this is confusing, think what a negative temperature would mean, or a 0 temperature! 30C is -1 times as hot as -30C? It doesn't make sense!

```
intervalVariables <- c(30,31,29,30,29,33,34,35)
```

Ratio variables

are interval variables, but with the added condition that 0 (zero) of the measurement indicates that there is none of that variable. So, temperature measured in degrees Celsius or Fahrenheit is not a ratio variable because 0C does not mean there is no temperature. However, temperature measured in Kelvin is a ratio variable as 0 Kelvin (often called absolute zero) indicates that there is no temperature whatsoever. Other examples of ratio variables include height, mass, distance and many more. Ratio responses mean that not only is there order and spacing, but that multiplication makes sense as well. Two common examples are height and weight. A person who weighs 200 pounds weighs double what a person who weighs 100 pounds weighs.

```
ratioVariables <- c(0:10)
```

Problematic Percentages

So, are percentages nominal, ordinal, interval or ratio? Technically, they are not even ratio - you cannot double a percentage without distorting the meaning

Levels of measurement

In general it is advantageous to treat variables as the highest level of measurement for which they qualify. That is, we could treat education level as a categorical variable, but usually we will want to treat it as an ordinal variable. This is because treating it as an ordinal variable retains more of the information carried in the data. If we were to reduce it to a categorical variable, we would lose the order of the levels of the variable. By using a higher level of measurement, we will have more options in the way we analyze, summarize, and present data.

Parametric Vs Non Parametric Tests

There is a lot of confusion about parametric vs. non-parametric statistics and tests. Some of the literature that explains the difference gets pretty technical. Here is a layman's description that might not be 100% technically accurate but that will let you understand the difference.A parameter is a characteristic of a population. We often estimate parameters with statistics that come from samples. Some common parameters and statistics are the mean, the median, the standard deviation and so on.

Some tests use these parameters. For example, every variety of the t-test uses means and standard deviations. Therefore, the t-test is called a parametric test. On the other hand, some tests do not use these parameters. For example, the Mann Whitney U test uses no parameters. Therefore, it is called a non-parametric test.

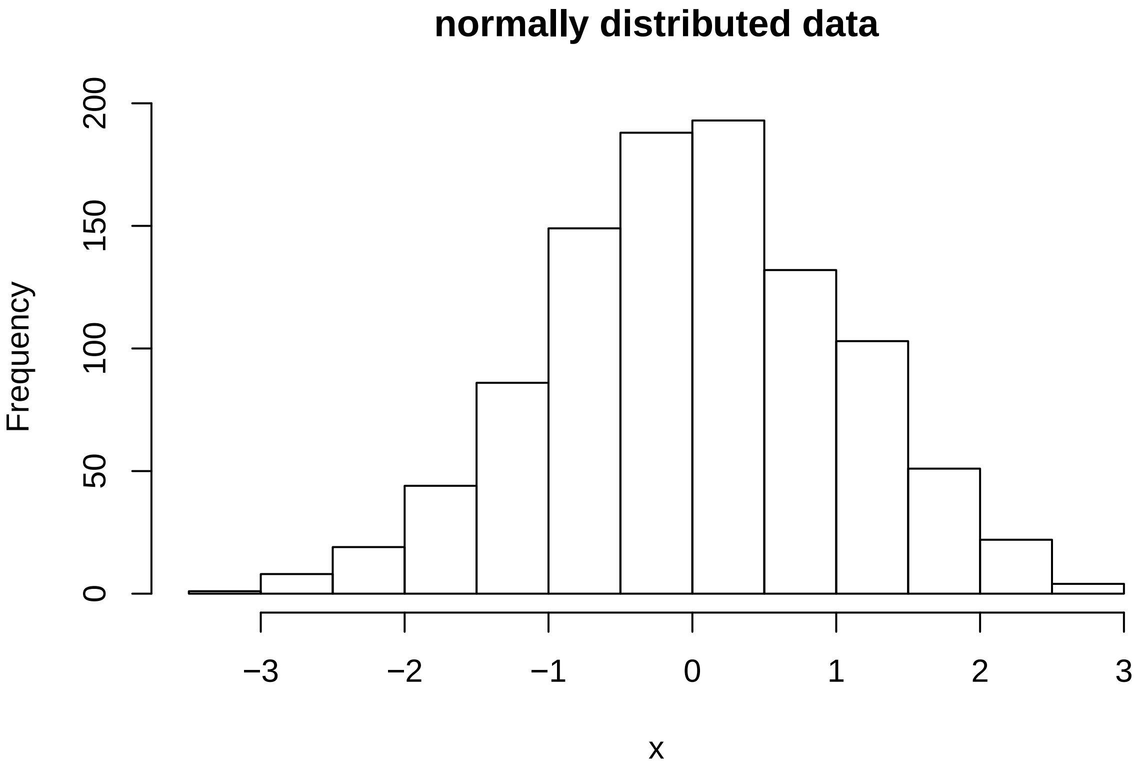
If you want to tell if a test is parametric or not, look at the formulas used in calculating it. Do they contain parameters/statistics?

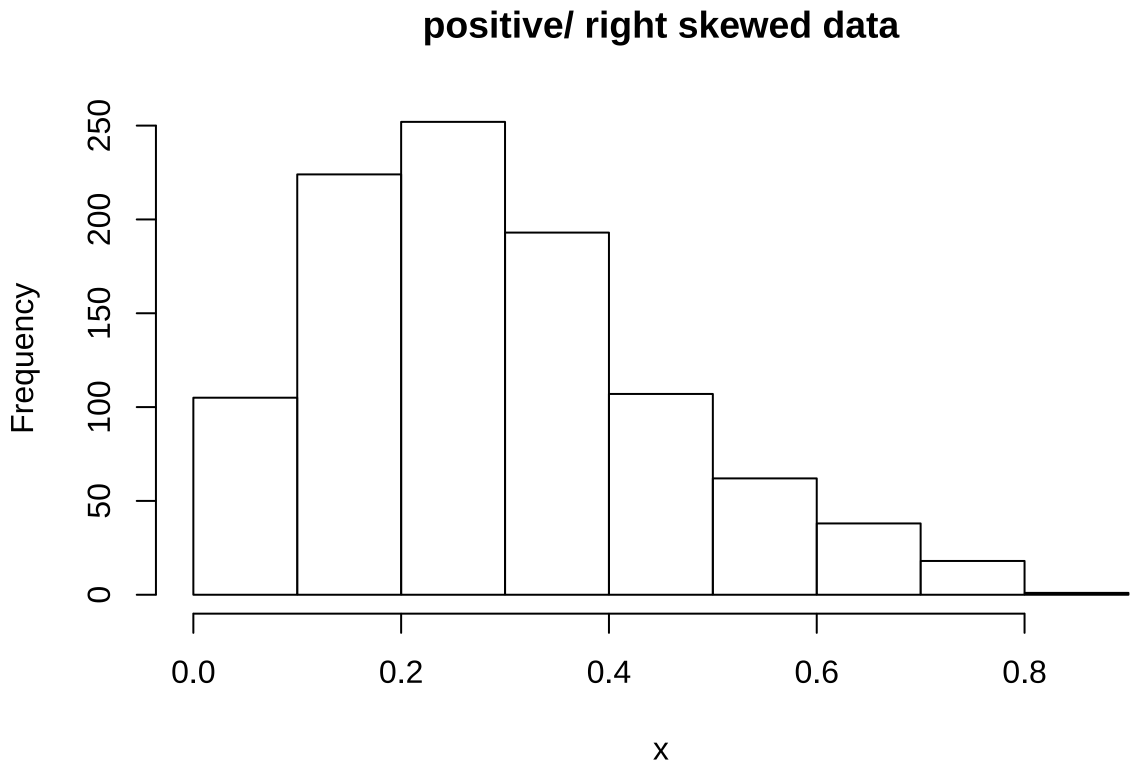
If your measurement scale is nominal or ordinal then you use non-parametric statistics

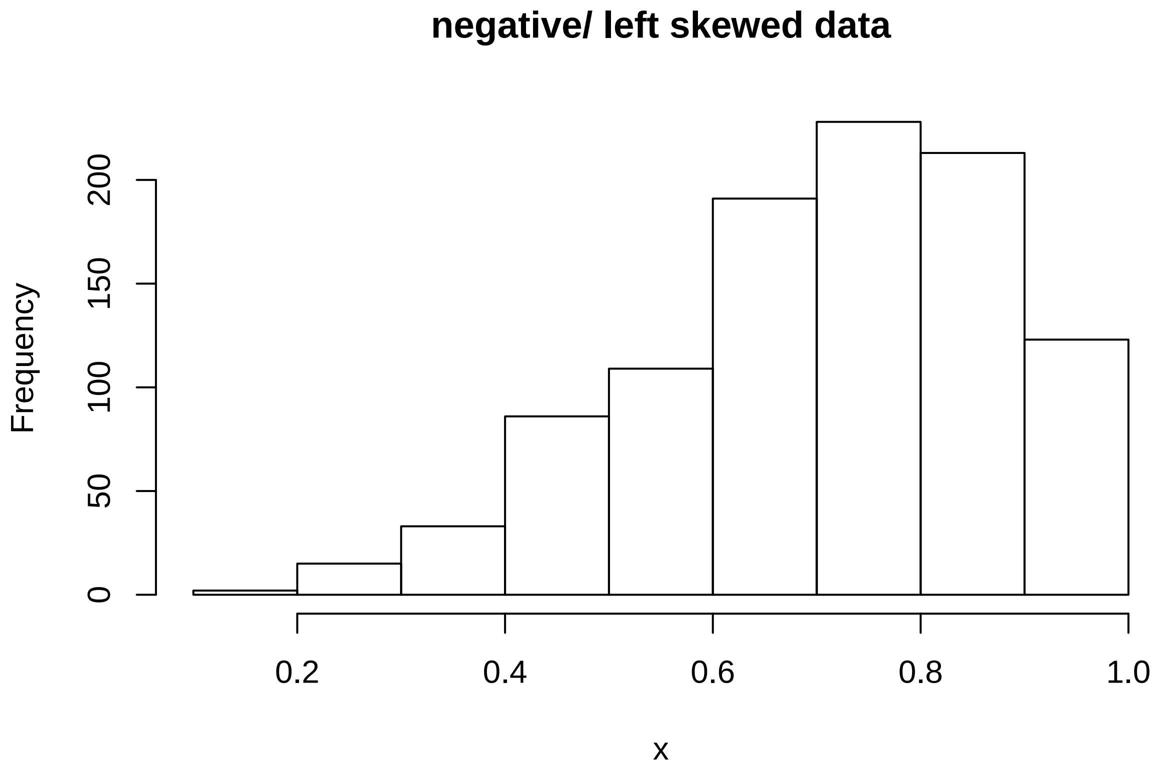
If you are using interval or ratio scales you use parametric statistics.

Histograms

A histogram is a type of graph used to display a distribution. It helps us to overcome the natural tendency to rely on summary information, such as an average. Histograms can reveal information no captured by summary statistics.




positive/ right skewed data
Frequency
0
50
100
150
200
250
0.0
0.2
0.4
0.6
0.8
x



Summary Statistics

Central Tendency

The mean

is a measure of central tendency this describes the middle or centre point of a distribution $mean = M = \frac{1}{n}\sum_{n}^{i=1} x$

The median

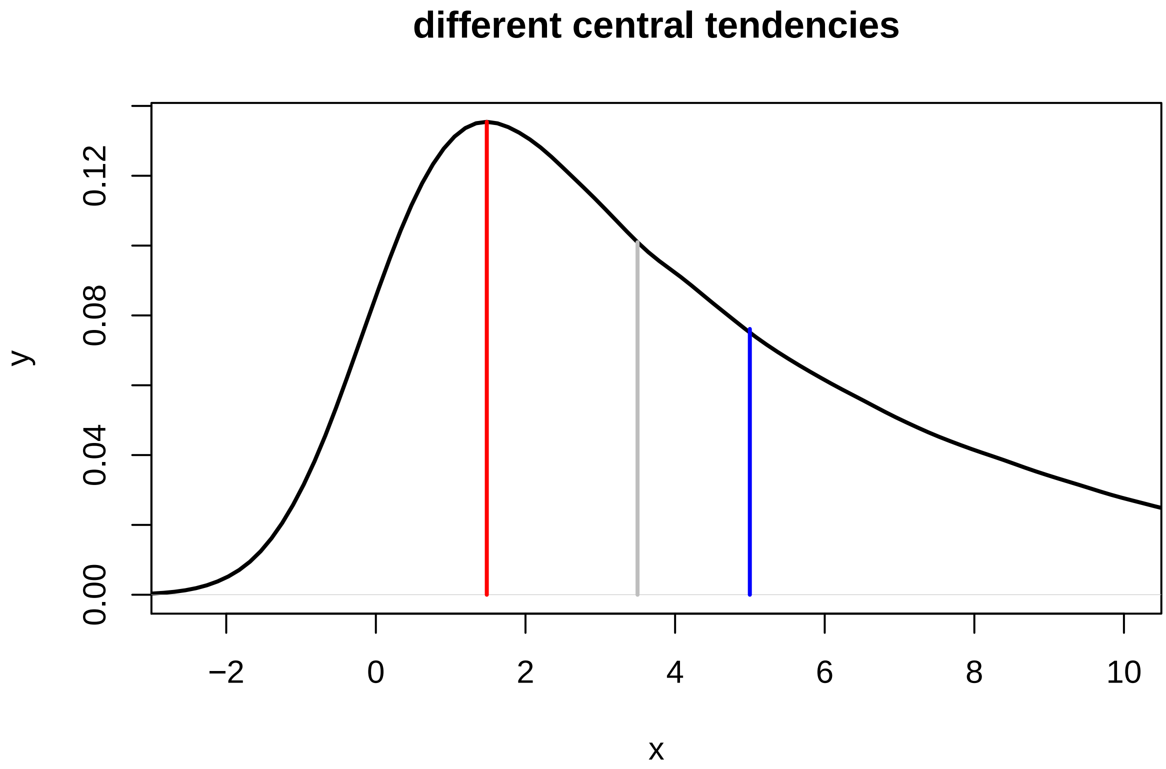
is the middle score (the score below which 50% of the distribution falls) preferred when there are extreme scores in the distribution

The mode

is the score that occurs most often in the distribution, useful for nominal variables

How distribution can affect measures of central tendency

Differing distribution may mean these three measures do not overlap Here the mean is blue, the median grey and the mode red.



Measures of Variability

a measure that describes the range and diversity of scores in a distribution

standard deviation (SD)

the average deviation from the mean in a distribution

$SD = \sqrt{\frac{[\Sigma(X-M^2)]}{N}}$ is used for descriptive statistics

$SD = \sqrt{\frac{[\Sigma(X-M^2)]}{N-1}}$ is used for inferential statistics

variance (SD^2)

sum of squared deviation scores = sum of squares are divided by the sample size

$SD^2 = \frac{[\Sigma(X-M^2)]}{N}$ is used for descriptive statistics

$SD^2 = \frac{[\Sigma(X-M^2)]}{N-1}$ is used for inferential statistics

this is also known as the mean squares

Example of Linsanity

Jeremy Lin was a basketball player who went on a scoring streak for the New York Knicks. We can calculate some summary statistics for his games.

here are the points he scored for the games he played:

```
pointsPerGame<-c(28,26,10,27,20,38,23,28,25,2)
```

we take the sum of those values and the sample size i.e. number of games he played to get the mean

```
sum(pointsPerGame)
```

```
## [1] 227
```

```
length(pointsPerGame)
```

```
## [1] 10
```

so the mean is

```
sum(pointsPerGame)/length(pointsPerGame)
```

```
## [1] 22.7
```

then the deviation scores show how much he deviated from the mean for each game i.e. it is the difference between a raw score and the mean.

```
pointsPerGame - mean(pointsPerGame)
```

```
##  [1]   5.3   3.3 -12.7   4.3  -2.7  15.3   0.3   5.3   2.3 -20.7
```

we can't get the average for the deviation scores because they sum to zero

```
deviationScores<- pointsPerGame - mean(pointsPerGame)
devsum <- sum(deviationScores)
```

```
## [1] 0
```

```
devsum/length(pointsPerGame)
```

```
## [1] 0
```

instead we square the deviation scores, sum them and divide by N to give us a score for variance.

That is to say we calculate mean squares because it is the sums of squares divided by N.

```
(pointsPerGame - mean(pointsPerGame))^2
```

```
##  [1]  28.09  10.89 161.29  18.49   7.29 234.09   0.09  28.09   5.29 428.49
```

```
devSq <- (pointsPerGame - mean(pointsPerGame))^2
devSumSq <- sum(devSq) ; devSumSq
```

```
## [1] 922.1
```

```
variance <- devSumSq/length(pointsPerGame) ; variance
```

```
## [1] 92.21
```

Squaring however does have a problem as a measure of spread and that is that the units are all squared, where as we'd might prefer the spread to be in the same units as the original data (think of squared points scored). Hence the square root allows us to return to the original units which is the standard deviation.

```r
sqrt(variance)
```

```
## [1] 9.602604
```

Standardised Scales

Z-scores

In statistics there is a standard scale the Z scale. Any score from any scale can be converted to Z scores

$Z = \frac{(X-M)}{SD}$

X = raw score, the score on the original scale

M = mean

SD = standard deviation

The mean Z-score is Z = 0

Positive Z scores are above average

Negative Z scores are below average

For example

```r
X = 99.6 # body temp for one person
M = 98.6 # the mean for the group
SD = 0.5 # the standard deviation for the group
Z=(X-M)/SD; Z
```

```
## [1] 2
```

This value of 2 means their score is 2 standard deviations above the mean

Percentile rank

The percentage of scores that fall at or below a score in a distribution Assume a normal distribution If Z = 0 then the percentile rank = 50th 50% of the distribution falls below the mean

Correlation

A statistical procedure used to measure and describe the relationship between two variables

Correlations can range between +1 and -1

+1 is perfect positive correlation

0 is no correlation (independence)

-1 is perfect negative correlation

When two variables, let's call them X and Y, are correlated, then one variable can be used to predict the other variable.

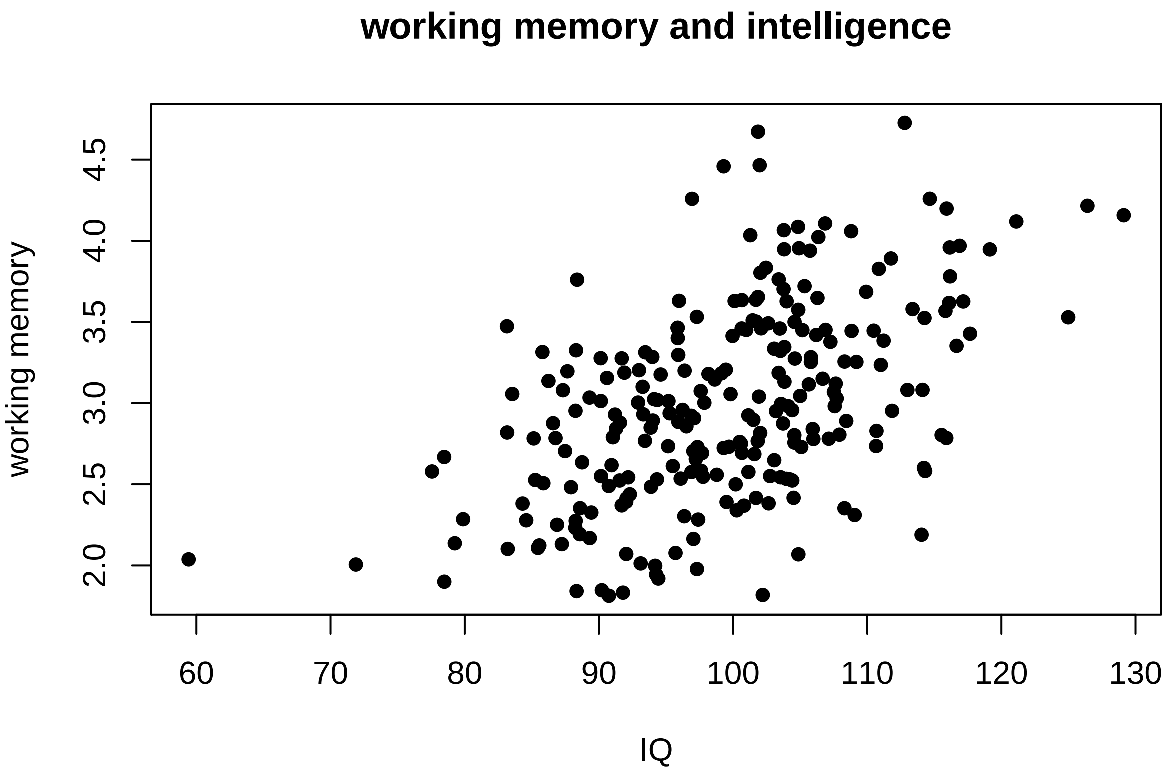
More precisely, a person's score on X can be used to predict his or her score on Y.

For example, working memory capacity is strongly correlated with intelligence, or IQ, in healthy young adults

So if we know a person's IQ then we can predict how they will do on a a test of working memeory.

We can see in this scatterplot that there is a positive correlation in our data, which is verified by the value we get for our correlation.

```r
IQ <- rnorm(250, mean = 100, sd = 10)
workingMemory <- IQ*rnorm(250, mean = 3, sd = 0.5)/100
df = data.frame(IQ, workingMemory)
plot(df$workingMemory~df$IQ, xlab="IQ" , ylab="working memory", pch = 16, main = "working memory and int
```



```r
cor.test(df$IQ, df$workingMemory)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$IQ and df$workingMemory
## t = 9.5657, df = 248, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4222865 0.6042983
## sample estimates:
##       cor
## 0.5191541
```

Warnings about correlation

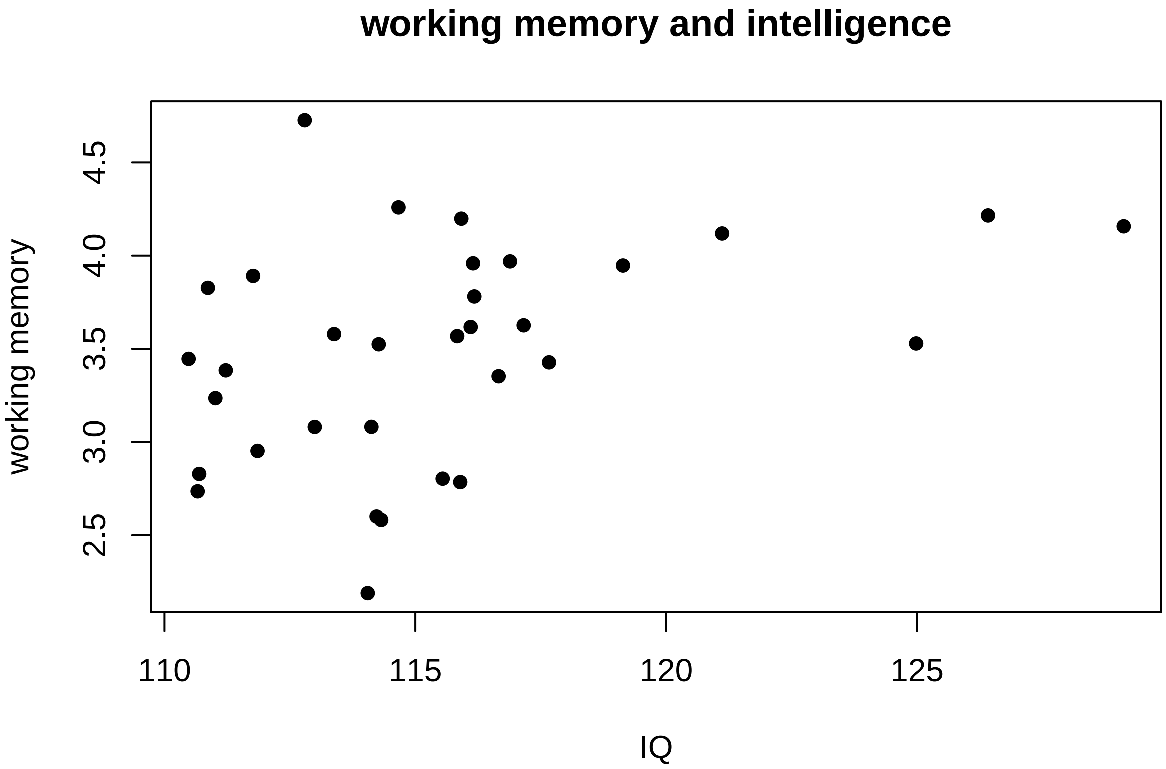
But we have to remember that correlation does not imply causation. In our example, working memory does not cause IQ and vice versa, rather there are lots of intervening variables.

The magnitude of a correlation is influenced by many factors, including: sampling (random and representative?), and the measurement of X & Y (are your measures of IQ reliable?).

When you fail to get a representative sample you can get attenutation of correlation due to a restriction of range in one of your variables. For instance, if you only select college graduates, you have preselected for higher IQ and this can reduce the correlation.

This restriction of range essentially restricts variance ultimately impacting our ability to discern covariance. In the following scatterplot and correlation measure you can see this effect.

```
dfAttenuated <- df[df$IQ >110, ]
plot(dfAttenuated$workingMemory~dfAttenuated$IQ, xlab="IQ" , ylab="working memory", pch = 16, main = "w
```



```
cor.test(dfAttenuated$IQ, dfAttenuated$workingMemory)
```

```
##
##  Pearson's product-moment correlation
##
## data:  dfAttenuated$IQ and dfAttenuated$workingMemory
## t = 2.3596, df = 31, p-value = 0.02477
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.05414736 0.64685852
## sample estimates:
```

```
##       cor
## 0.3902028
```

Finally, a correlation coefficient is a sample statistic just like the mean and won't be representative unless the correlation coefficient is 1.

Types of Correlation

There are several types of correlation coefficients, for different variable types.

The Pearson product-moment correlation coefficient (r)

This is used when both variables, X & Y, are continuous.

The Point bi-serial correlation

This is used when 1 variable is continuous and 1 is dichotomous.

The Phi coefficient

When both variables are dichotomous

Spearman rank correlation

When both variables are ordinal (ranked data)

Focus on Pearson correlation

r = the degree to which X and Y vary together, relative to the degree to which X and Y vary independently.

r = (covariance of X & Y)/(variance of X & Y)

There are a number of ways to calculate r e.g.

the raw score formula and

the Z-score formula

Remember from our calculation for variance

$variance = SD^2 = MS = (SS/N)$

To calculate SS:

For each row, calculate the deviation score

$(X - M_x)$

Square the deviation scores

$(X - M_x)^2$

Sum the squared deviation scores

$SS_x = \Sigma[(X - M_x)^2] = \Sigma[(X - M_x) * (X - M_x)]$

Sum of Cross Products

We need to calculate the sum of cross products (SP) to get r for our correlation

for each row, calculate the deviation score on X $(X - M_x)$

For each row, calculate the deviation score on Y $(X - M_y)$

Then, for each row, multiply the deviation score on X by the deviation score on Y

$(X - M_x) * (Y - M_y)$

Then sum the "cross products"

$SP = \Sigma[(X - M_x) * (Y - M_y)]$

Covariance

Covariance measures the relationship between two variables.

Covariance = COV = SP/N

The formula for covariance is: $cov(X, Y) = \frac{\Sigma(X - M_x)(Y - M_y)}{N}$

or for inferential statistics where the denominator becomes n-1

$cov(X, Y) = \frac{\Sigma(X - M_x)(Y - M_y)}{N-1}$

Covariance is not scaled, so it can't tell you the strength of that relationship. To account for this, correlation takes covariance and scales it by the product of the standard deviations of the two variables.

```
cov(df$IQ,df$workingMemory) # covariance score
```

```
## [1] 3.091844
```

```
cor.test(df$IQ,df$workingMemory) # correlation score
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$IQ and df$workingMemory
## t = 9.5657, df = 248, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4222865 0.6042983
## sample estimates:
##       cor
## 0.5191541
```

Raw score formula

The raw score formula is thus:

$r = \frac{SP_{xy}}{\sqrt{(SS_x * SS_y)}}$

remember this r value is the degree to which X and Y vary together, relative to the degree to which X and Y vary independently.

In longer form:

$SP = \Sigma[(X - M_x) * (Y - M_y)]$

$SS_x = \Sigma[(X - M_x)^2] = \Sigma[(X - M_x) * (X - M_x)]$

$SS_y = \Sigma[(Y - M_y)^2] = \Sigma[(Y - M_y) * (Y - M_y)]$

So the raw score formula to calculate the correlation coefficient r can be written out in two ways:

$r = \frac{SP_{xy}}{\sqrt{(SS_x * SS_y)}}$

or,

$r = \frac{\Sigma[(X - M_x) * (Y - M_y)]}{\sqrt{(\Sigma(X - M_x)^2 * \Sigma(Y - M_y)^2)}}$

Z-score formula

is the sum of the product of the Z-scores divided by N

$r = \frac{\Sigma(Z_x * Z_y)}{N}$

first we need to calculate the Z-scores

$Z_x = \frac{(X - M_x)}{SD_x}$

$Z_y = \frac{(Y - M_y)}{SD_y}$

where

$SD_x = \sqrt{\frac{(\Sigma(X - M_x)^2}{N}}$

$SD_y = \sqrt{\frac{(\Sigma(Y - M_y)^2}{N}}$

Proof of equivalence

here the denominator is the standard deviation

$Z_x = \frac{(X - M_x)}{\sqrt{\frac{(\Sigma(X - M_x)^2}{N}}}$

$Z_y = \frac{(Y - M_y)}{\sqrt{\frac{(\Sigma(Y - M_y)^2}{N}}}$

unpacked to it's full long form

here we have Z_x multiplied by Z_y divided by N

$$r = \frac{\frac{(X - M_x)}{\sqrt{\frac{(\Sigma(X - M_x)^2}{N}}} * \frac{(Y - M_y)}{\sqrt{\frac{(\Sigma(Y - M_y)^2}{N}}}}{N}$$

we can pack all this back together using some algebra

$r = \frac{\Sigma[(X - M_x) * (Y - M_y)]}{\sqrt{(\Sigma(X - M_x)^2 * \Sigma(Y - M_y)^2)}}$

which can be simplified further to:

$r = \frac{SP_{xy}}{\sqrt{(SS_x * SS_y)}}$

which is the raw score formula.

Variance and covariance

Variance = MS = SS/N

Covariance = COV = SP/N

Correlation is standardised covariance

it's standardised so the value is in the range -1 to +1

Note on the denominators

Correlation for descriptive statistics

Divide by N

Correlation for inferential statistics

Divide by N-1

Assumptions of Correlation

let's consider Pearson correlation

Assumptions when interpreting r:

Normal distributions for X and Y

- how to detect violations?

Plot histograms and examine summary stats

Linear relationship between X and Y

- how to detect violations?

Examine scatterplots

Homoscedasticity

- how to detect violations?

Examine scatterplots

Reliability of X and Y

Validity of X and Y

Random and representative sampling

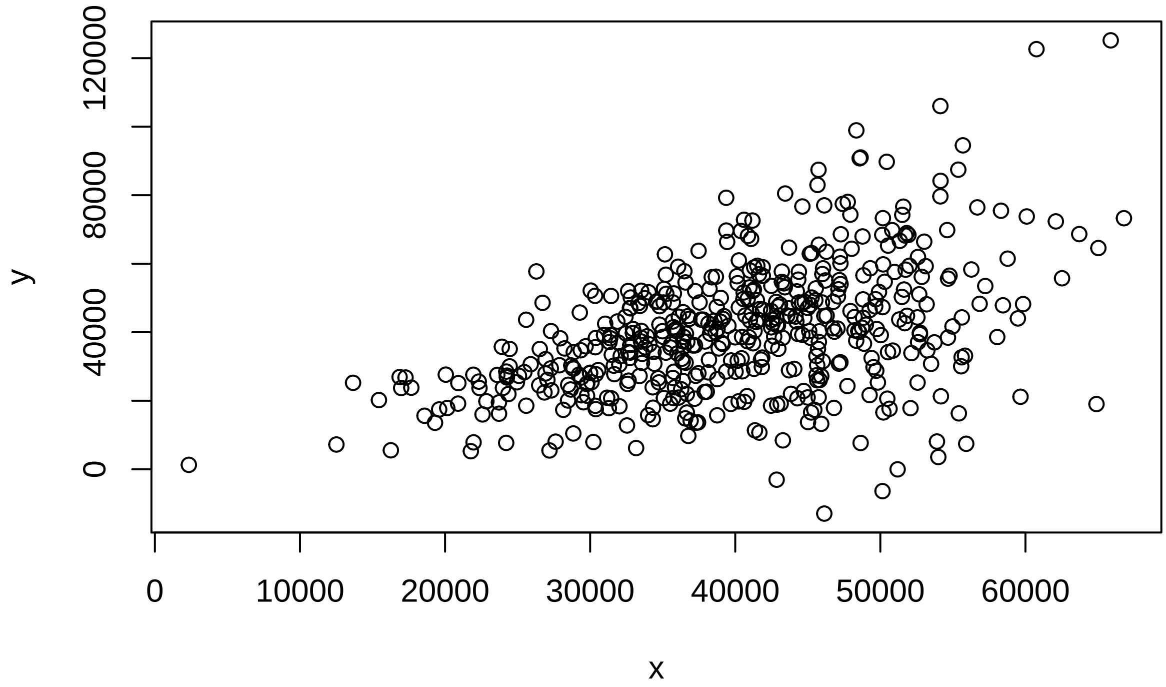
Homoscedasticity and heteroscedasticity

In a scatterplot the vertical distance between a dot and the regression line reflects the amount of predicition error (known as the "residual")

The idea of Homoscedasticity is that those residuals are not related to X. The residuals should be chance errors and not systematic.

If the residuals are related to X then we suspect some sort of confound in our study. This is termed Heteroscedasticity.

A classic example of heteroscedasticity is that of income versus expenditure on meals. As one's income increases, the variability of food consumption will increase. A poorer person will spend a rather constant amount by always eating inexpensive food; a wealthier person may occasionally buy inexpensive food and at other times eat expensive meals. Those with higher incomes display a greater variability of food consumption.



Anscombe's Quartet

Why it's critical to look at your scatterplots.

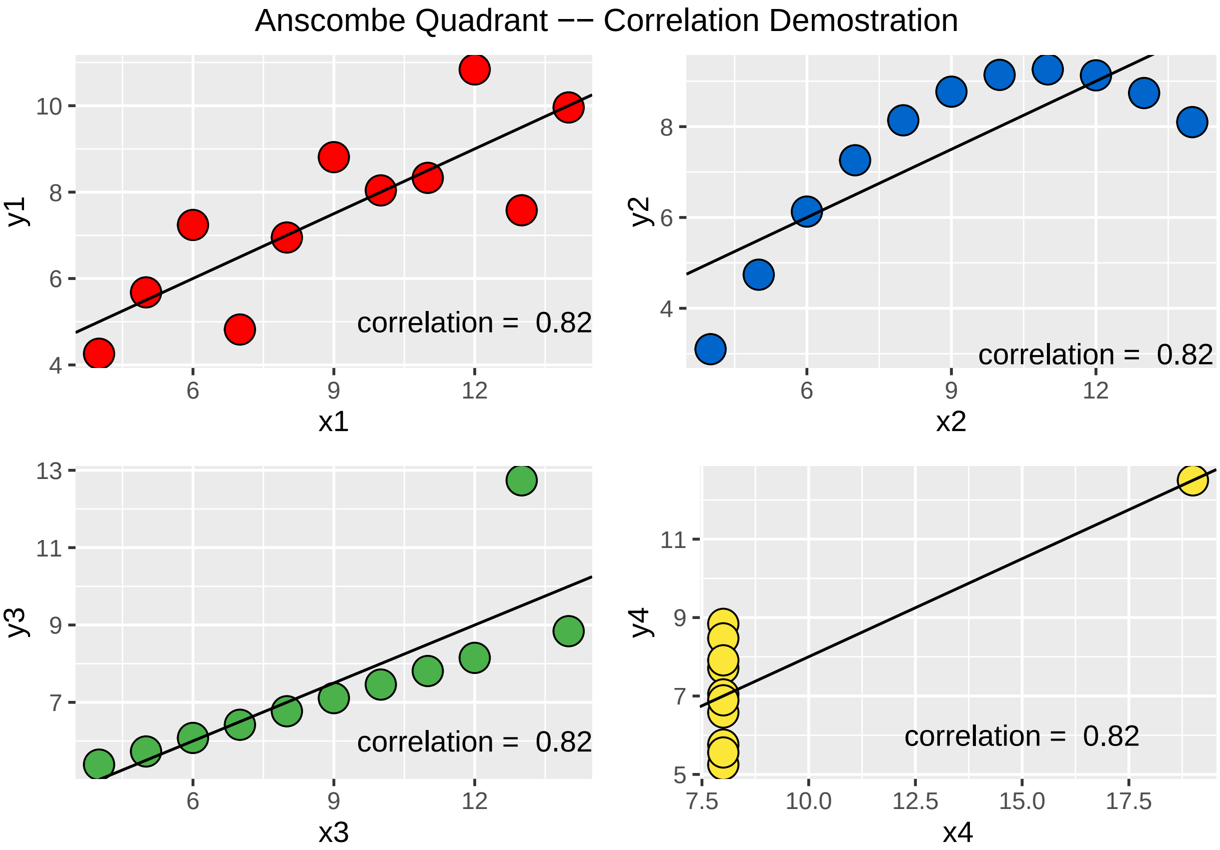
Four datasets where the correlation is exactly the same. The datasets also have the same variance. But clearly there are differences in these datasets.

```
##        x1              x2              x3              x4    
##  Min.   : 4.0   Min.   : 4.0   Min.   : 4.0   Min.   : 8  
##  1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 8  
##  Median : 9.0   Median : 9.0   Median : 9.0   Median : 8  
##  Mean   : 9.0   Mean   : 9.0   Mean   : 9.0   Mean   : 9  
##  3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.: 8  
##  Max.   :14.0   Max.   :14.0   Max.   :14.0   Max.   :19  
##        y1               y2              y3              y4        
##  Min.   : 4.260   Min.   :3.100   Min.   : 5.39   Min.   : 5.250  
##  1st Qu.: 6.315   1st Qu.:6.695   1st Qu.: 6.25   1st Qu.: 6.170  
##  Median : 7.580   Median :8.140   Median : 7.11   Median : 7.040  
##  Mean   : 7.501   Mean   :7.501   Mean   : 7.50   Mean   : 7.501  
##  3rd Qu.: 8.570   3rd Qu.:8.950   3rd Qu.: 7.98   3rd Qu.: 8.190  
##  Max.   :10.840   Max.   :9.260   Max.   :12.74   Max.   :12.500
## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##     %+%, alpha

## The following object is masked from 'package:latticeExtra':
##
##     layer
```



Measurement

Reliability - do we have reliable measurements?

If I step on a scale multiple times do I get the same weight?

But some values are harder to evaluate for reliability. Raw scores are imperfect, e.g. body temperature is suscpetible to systematic bias and chance error.

Classical test theory states that, in a perfect world, it would be possible to obtain a "true score" rather than a "raw score" (X)

X = true score + bias + error

A measure (X) is considered reliable as it approaches the true score

The problem is we don't know the true score so we estimate reliability

Methods to estimate reliability

Test/re-test

Parallel tests

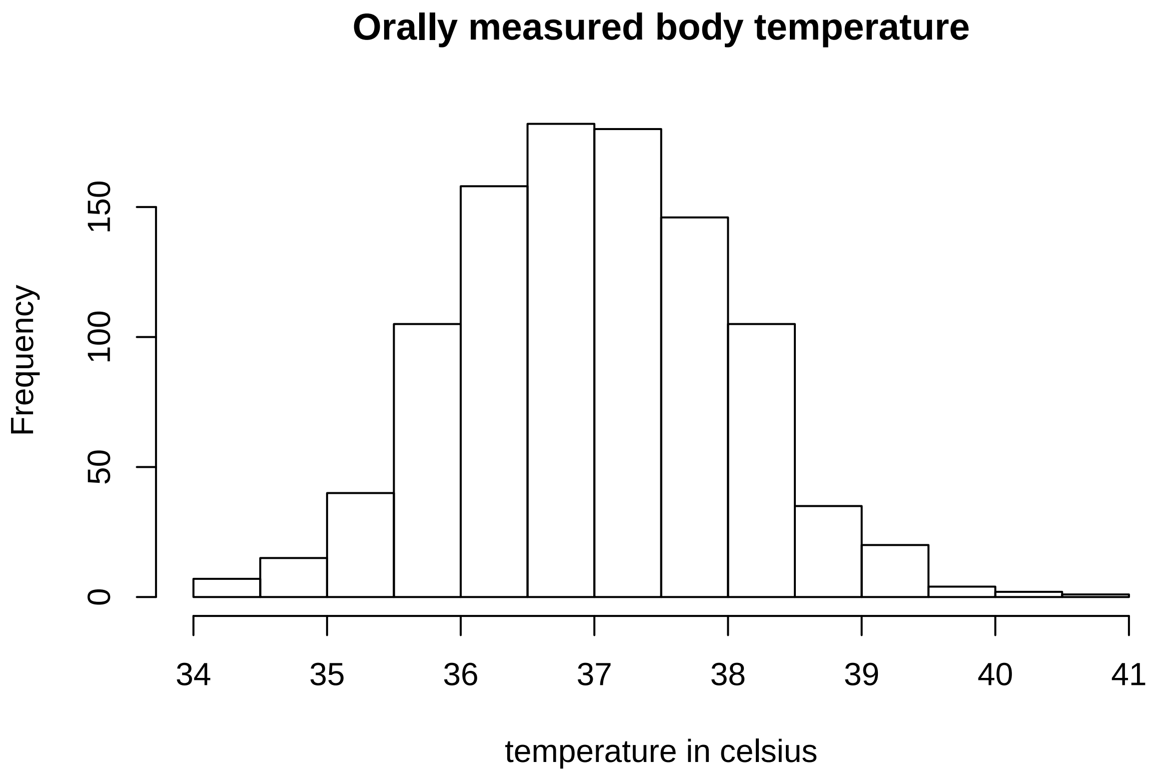
Inter-item reliability

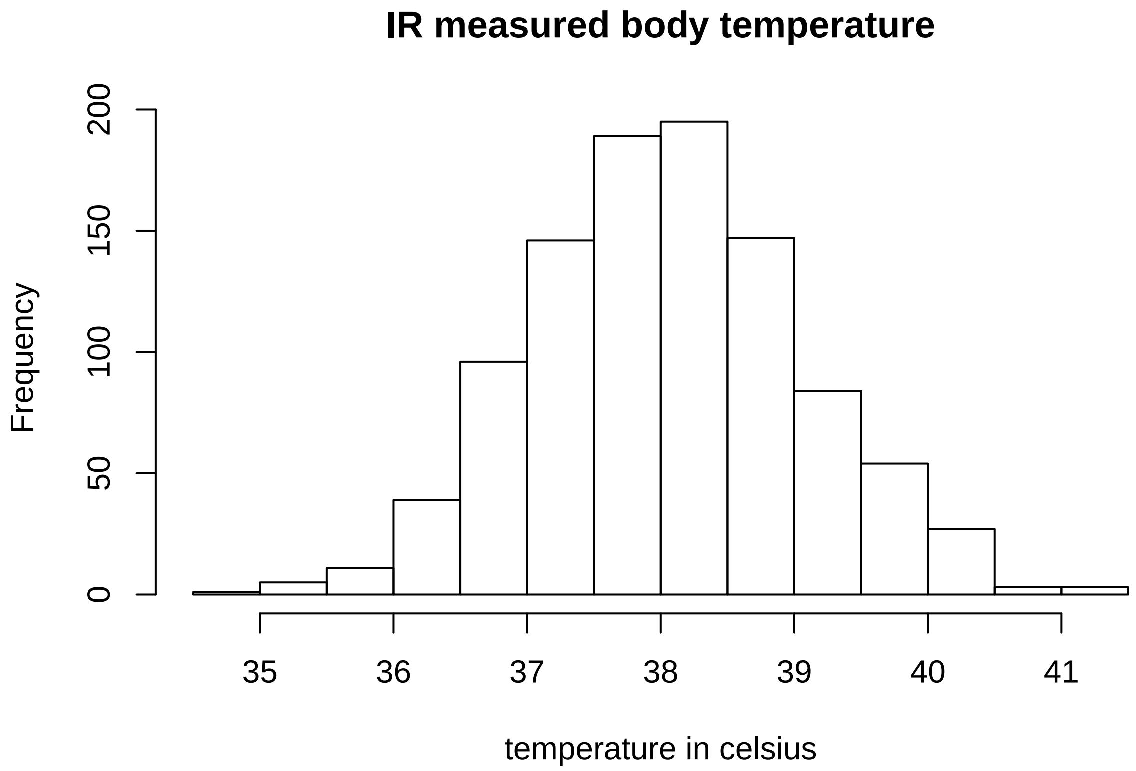
Example of body temperature

Can be measured in 3 ways,

Orally, internally, IR wand.

The IR wand has a systematic bias in that it always tends to record a higher temperature.





Test/re-test

One way to get a reliability estimate

Measure everyone twice so we'll have data for X1 and X2

Should be a strong correlation between the two measures otherwise you don't have a reliable measure

However, if the bias is uniform then we won't detect it with the test/re-test method. So in the case of the IR thermometer reading a little high, test/re-test won't work. The correlation will be high even though there is a bias.

Parallel tests

Measure body temp with the wand (X1) and with the oral thermometer(X2)

The correlation betweeen X1 and X2 is an estimate of reliability.

AND, now the bias of the wand will be revealed because you have two tests rather than one.

Inter-item

the most commonly used method in the social sciences because the focus is usually on human subjects who are difficult to work with.

Test/re-test and parallel tests are time consuming

Inter-item is therefore more cost efficient.

For example, suppose a 20 item survey is designed to measure extraversion

Randomly select 10 items to get sub-set A (X1)

The other items become sub-set B (X2)

Now we have two assessments of extraversion built into one overall survey.

If they're all getting at one personality trait then there should be a correlation between X1 and X2 which would represent an estimate of reliability. #Measurement ##Validity What is a construct?

An ideal "object" that is not directly observable as opposed to "real" observable objects

For example, "intelligence" is a construct.

How do we operationalise a construct?

The process of defining a construct to make it observable and quantifiable e.g. intelligence tests.

Construct validity

How do we assess the validity of a construct?

Let's take an example of a construct: verbal ability in children

We might operationalise this construct by using a vocabulary test.

Content validity

In the case of the vocabulary test we would ask does the test consist of words that children in the population and sample know?

Convergent validity

Does the test correlate with other, established measures of verabal ability e.g. with reading comprehension.

Divergent validity

Does the test correlate less well with measures designed to test a different type of ability e.g. spatial ability, or even more extreme, the height of the student where there should very little correlation.

Nomological validity

Are scores on the test consistent with more general theories, e.g. for child development and neuroscience. In that case a child with neural damage or disease to brain regions associated with language development should score lower on the test.

Sampling

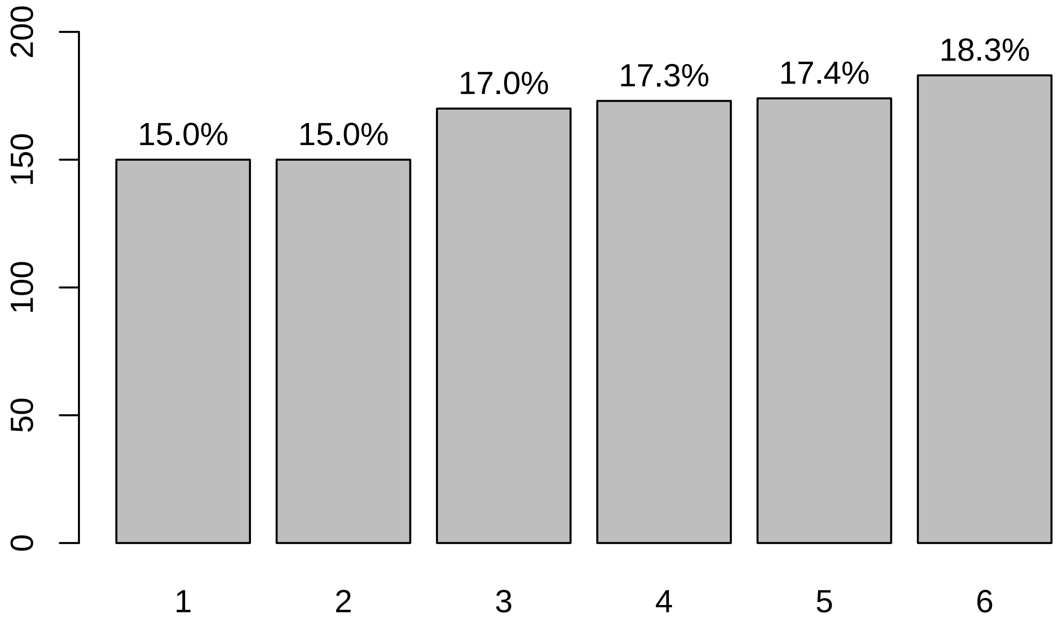
Remember, we want our sample to be random and representative.

Let's set up a die and roll it a 1000 times, then plot the results on a histogram

```
p.die <- rep(1/6,6)
sum(p.die)
```

```
## [1] 1
```

```
die <- 1:6
s <- table(sample(die, size=1000, prob=p.die, replace=T))
lbls = sprintf("%0.1f%%", s/sum(s)*100)
barX <- barplot(s, ylim=c(0,200))
text(x=barX, y=s+10, label=lbls)
```



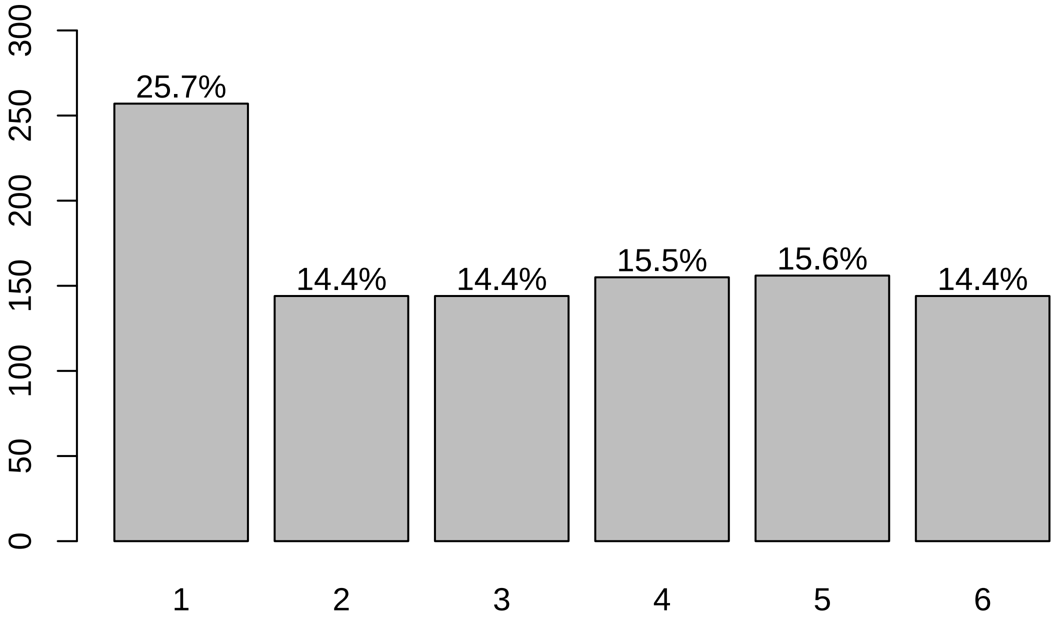
As expected each of the 6 sides came up approximately the same number of times.

We can do something similar but with a loaded die

```
p.die <- c(0.25,0.15,0.15,0.15,0.15,0.15)
sum(p.die)
```

```
## [1] 1
```

```
die <- 1:6
s <- table(sample(die, size=1000, prob=p.die, replace=T))
lbls = sprintf("%0.1f%%", s/sum(s)*100)
barX <- barplot(s, ylim=c(0,300))
text(x=barX, y=s+10, label=lbls)
```



Sampling error

The difference between the population and the sample.

This is important because we can't get everyone/everything in the popualtion.

Notice that the random histogram is not perfectly random

There is some fluctuation due to sampling error.

Problem

We don't know the population parameters

So, how do we estimate sampling error?

Estimating sampling error

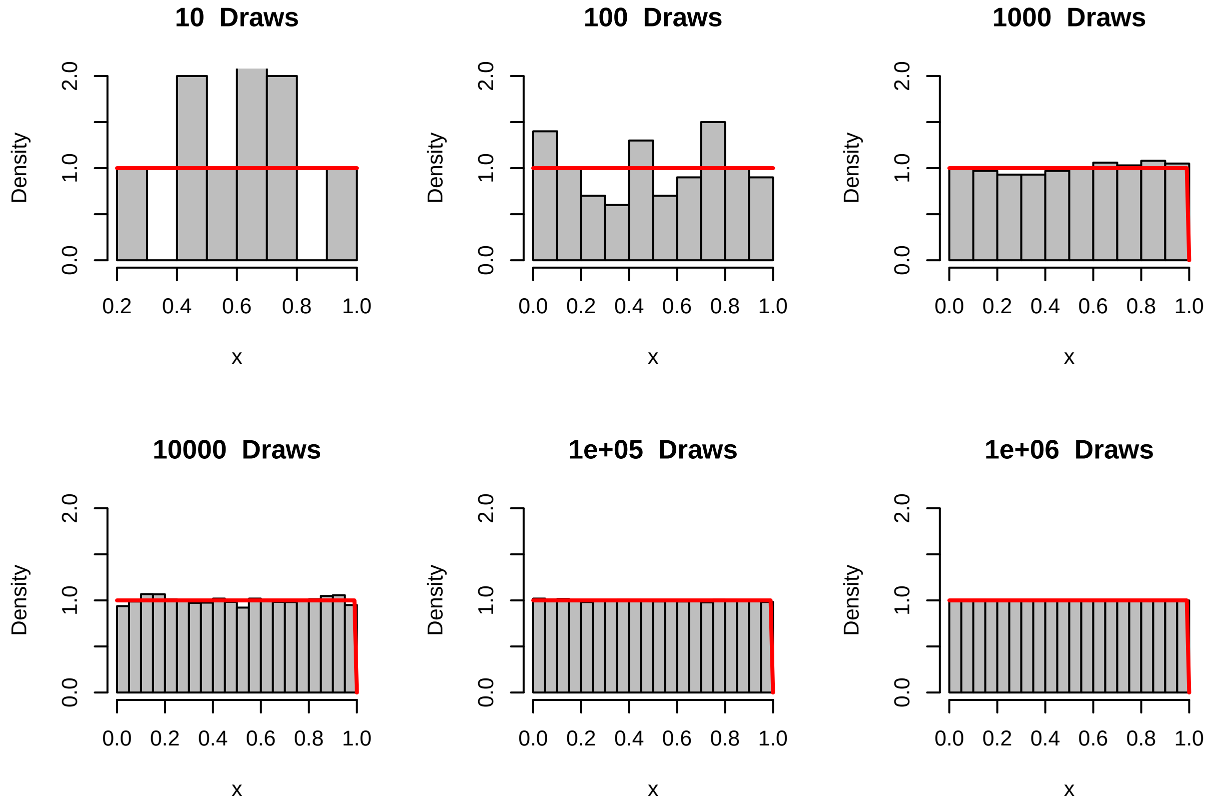
Sampling error mainly depends on the size of the sample, relative to the size of the population.

as sample size increases, sampling error decreases.

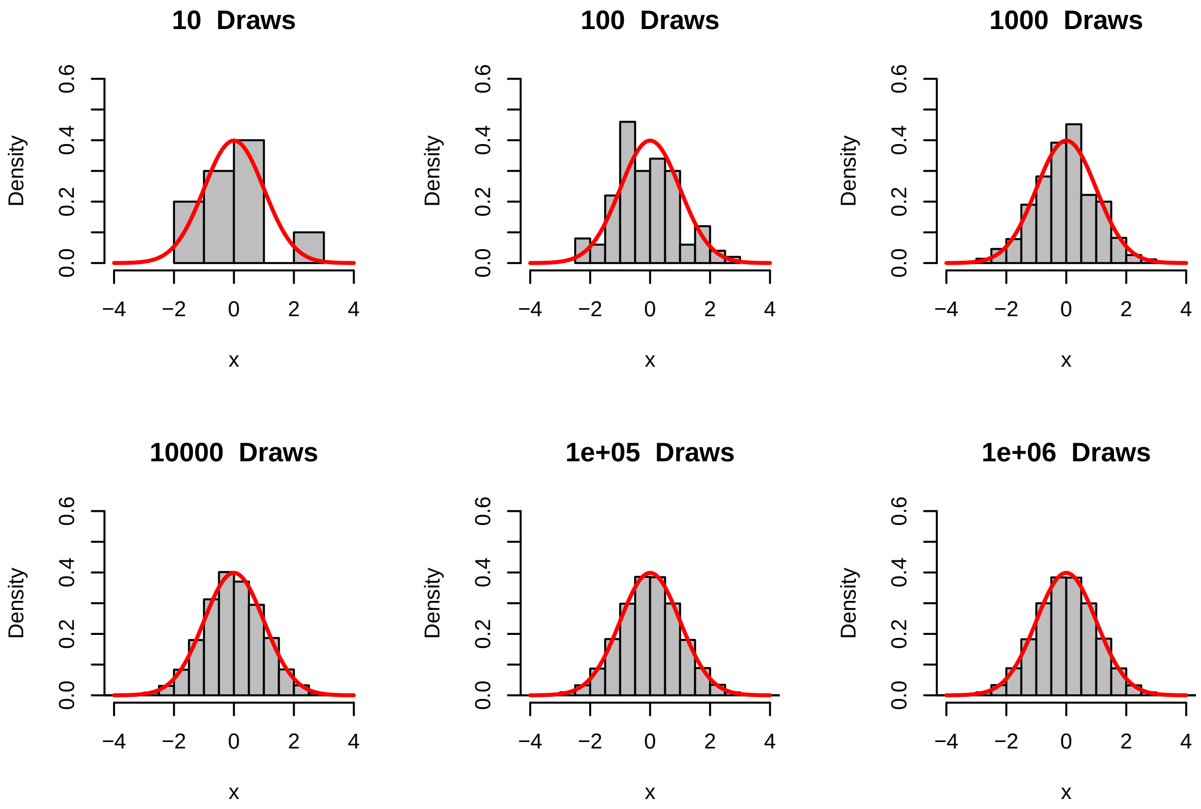
It also depends on the variance in the population (which we don't know).

as variance increases, sampling error increases.

The following histogram shows how the distribution becomes more uniform as the sample size increases illustrating the effect of sample size on sampling error.



We can check out the same effect by sampling from the normal distribtion using increasing sample sizes. As we can see the distribution becomes more normal as the sample size goes up.



Estimating sampling error

Sampling error is estimated from the size of the sample and the variance in the sample. This is under the assumption that the sample is random and representative of the population.

Standard error

Standard error is an estimate of the average amount of sampling error

$SE = \frac{SD}{\sqrt{N}}$

SE = standard error

SD = standard deviation of the sample

N = Size of the sample

We can see from here that as the sample size in the denominator increases then the standard error is going to rise.

By contrast, as the standard deviation increases (which should reflect the population standard deviation) so too will the standard error.

Here's an example of taking a sample from a population that is not very variable. Note the low standard error.

```r
population<-rnorm(1000,mean=0,sd=1)
sd(population)
```

```
## [1] 1.001635
```

```
sampleDraw<-sample(population,10)
sd(sampleDraw)
```

```
## [1] 1.077219
```

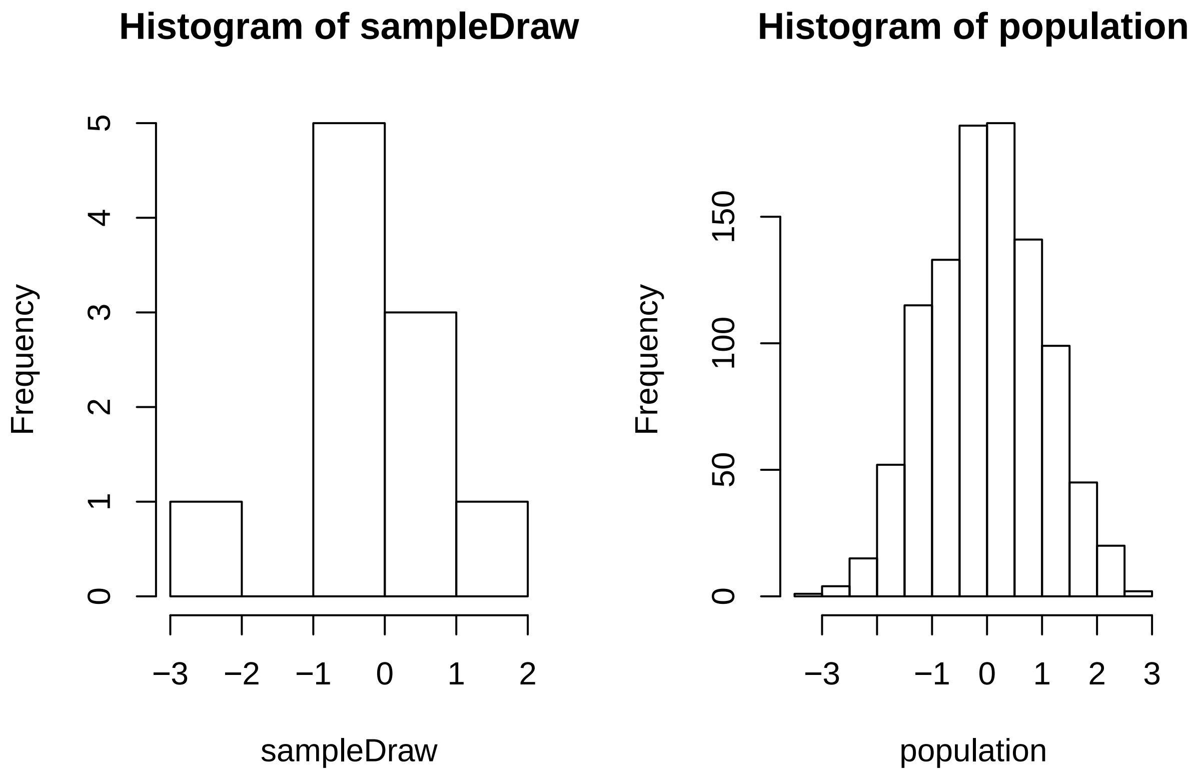
```
length(sampleDraw)
```

```
## [1] 10
```

```
sd(sampleDraw)/sqrt(length(sampleDraw))
```

```
## [1] 0.3406467
```

```
par(mfrow=c(1,2))
hist(sampleDraw)
hist(population)
```



Here's an example of taking a sample from a population that is variable. Note the higher standard error.

```
population<-rnorm(1000,mean=0,sd=10)
sd(population)
```

```
## [1] 9.71732
```

```
sampleDraw<-sample(population,10)
sd(sampleDraw)
```

```
## [1] 9.436989
```

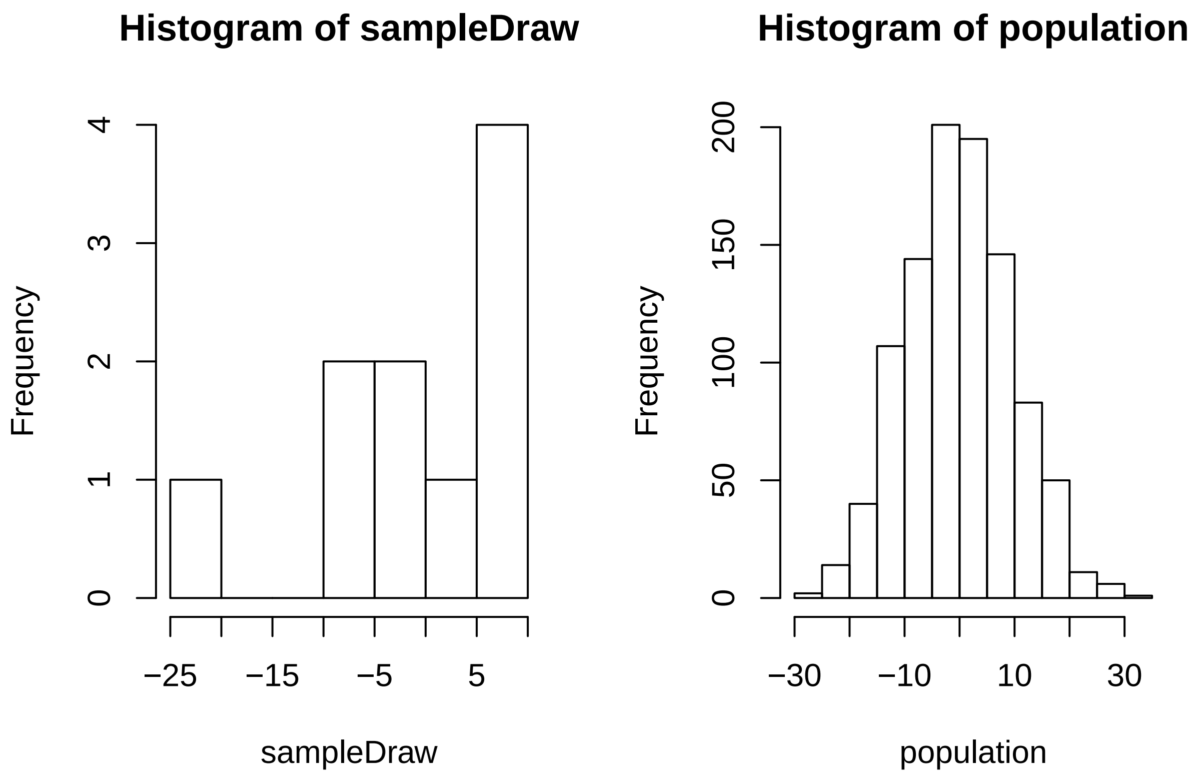
```
length(sampleDraw)
```

```
## [1] 10
```

```
sd(sampleDraw)/sqrt(length(sampleDraw))
```

```
## [1] 2.984238
```

```
par(mfrow=c(1,2))
hist(sampleDraw)
hist(population)
```



Regression

A regression is a statistical analysis used to predict scores on an outcome variable, based on scores on one or multiple predictor variables

simple regression: one predictor variable

multiple regression: multiple predictor variables

Regression equation

$Y = m + bX + e$ this is the equation of a line

Y = a linear function of X m = intercept b = slope e = residual error other notation, more commonly used for statistics $Y = B_0 + B_1X_1 + e$

Y = a linear function of X_1 $B - 0$ = intercept = regression constant B_1 = slope = regression coefficient e = residual error

Model R and R^2

R = multiple correlation coefficient

R = $r_{y'y}$

The correlation between the predicted scores and the observed scores

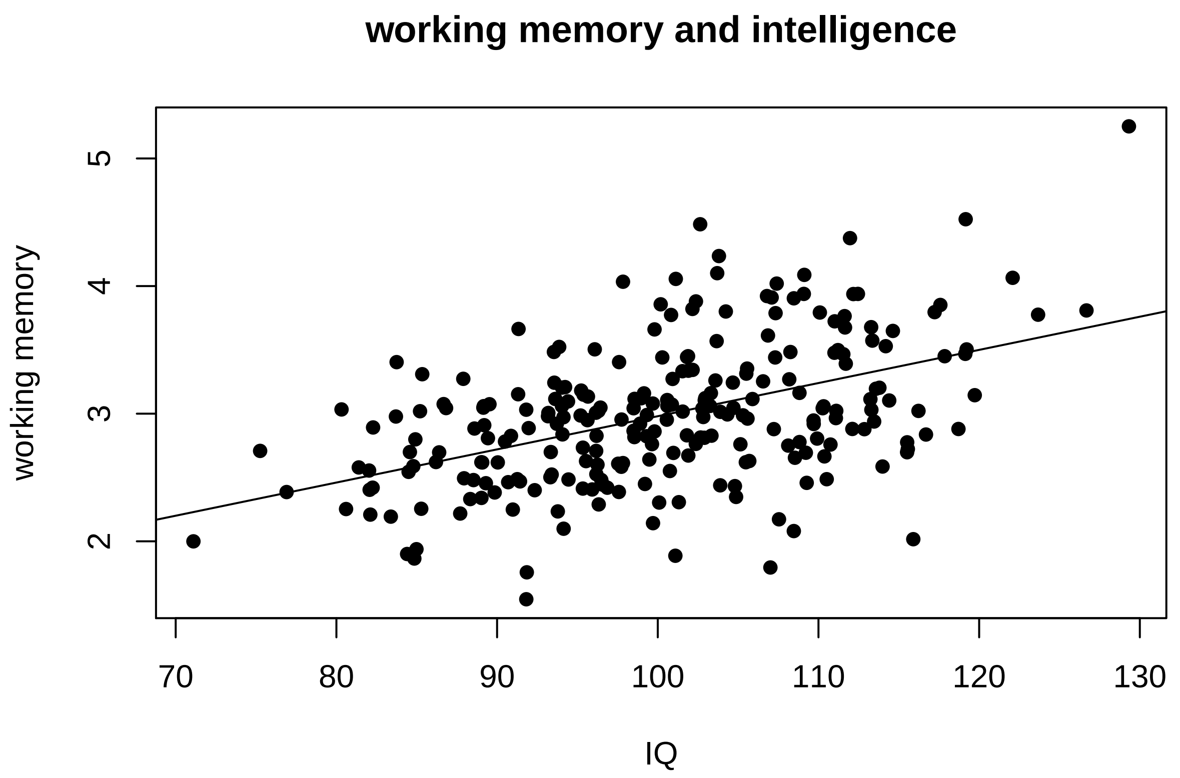
R^2 the percentage of variance in Y explained by the model.

Simple regression example

```
set.seed(20)
IQ <- rnorm(250, mean = 100, sd = 10)
workingMemory <- IQ*rnorm(250, mean = 3, sd = 0.5)/100
df = data.frame(IQ, workingMemory)
plot(df$workingMemory~df$IQ, xlab="IQ" , ylab="working memory", pch = 16, main = "working memory and in
cor.test(df$IQ, df$workingMemory)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$IQ and df$workingMemory
## t = 8.6622, df = 248, p-value = 6.037e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3806483 0.5718293
## sample estimates:
##       cor
## 0.4819546
```

```
model1 <- lm(df$workingMemory~df$IQ)
abline(model1)
```



```r
summary(model1)
```

```
##
## Call:
## lm(formula = df$workingMemory ~ df$IQ)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.37683 -0.32246 -0.00195  0.29800  1.50948
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.38105    0.30251   1.260    0.209
## df$IQ        0.02599    0.00300   8.662 6.04e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4886 on 248 degrees of freedom
## Multiple R-squared:  0.2323, Adjusted R-squared:  0.2292
## F-statistic: 75.03 on 1 and 248 DF,  p-value: 6.037e-16
```

The estimate of the slope is 0.02599, so with every one unit increase in X there is a 0.02599 increase in Y.

The estimate of the intercept is 0.38105, the predicted score on Y when X = 0.

Thus the regression equation is

$Y = 0.38105 + 0.02599(X)$

In a simple regression the R^2 value 0.2323 is equal to the correlation coefficient 0.4819546 to the power of 2.

```
0.4819546^2
```

```
## [1] 0.2322802
```

The goal with regression is to produce better models so we can generate more accurate predictions

Add more predictor variables and/or

develop better predictor variables.

Multiple Regression

Add in another predictor variables

$Y = B_0 + B_1Y_1 + B_2Y_2 + e$

Now we need to solve for B_0 & B_1 & B_2

```
set.seed(20)
IQ <- rnorm(250, mean = 100, sd = 10)
workingMemory <- IQ*rnorm(250, mean = 3, sd = 0.5)/100
salary <- IQ*rnorm(250, mean = 500, sd = 0.1)
df = data.frame(IQ, workingMemory, salary)
model2 <- lm(df$workingMemory~df$IQ+df$salary)
summary(model2)
```

```
##
## Call:
## lm(formula = df$workingMemory ~ df$IQ + df$salary)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.41027 -0.31297 -0.03154  0.29565  1.46744
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.434903   0.303143   1.435   0.1527
## df$IQ        2.410854   1.428697   1.687   0.0928 .
## df$salary   -0.004771   0.002858  -1.669   0.0963 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4868 on 247 degrees of freedom
## Multiple R-squared:  0.2408, Adjusted R-squared:  0.2347
## F-statistic: 39.18 on 2 and 247 DF,  p-value: 1.663e-15
```

The linear combination of two predictors can do better at predicting the outcome than any one predictor by itself.

Calculation of regression coefficients

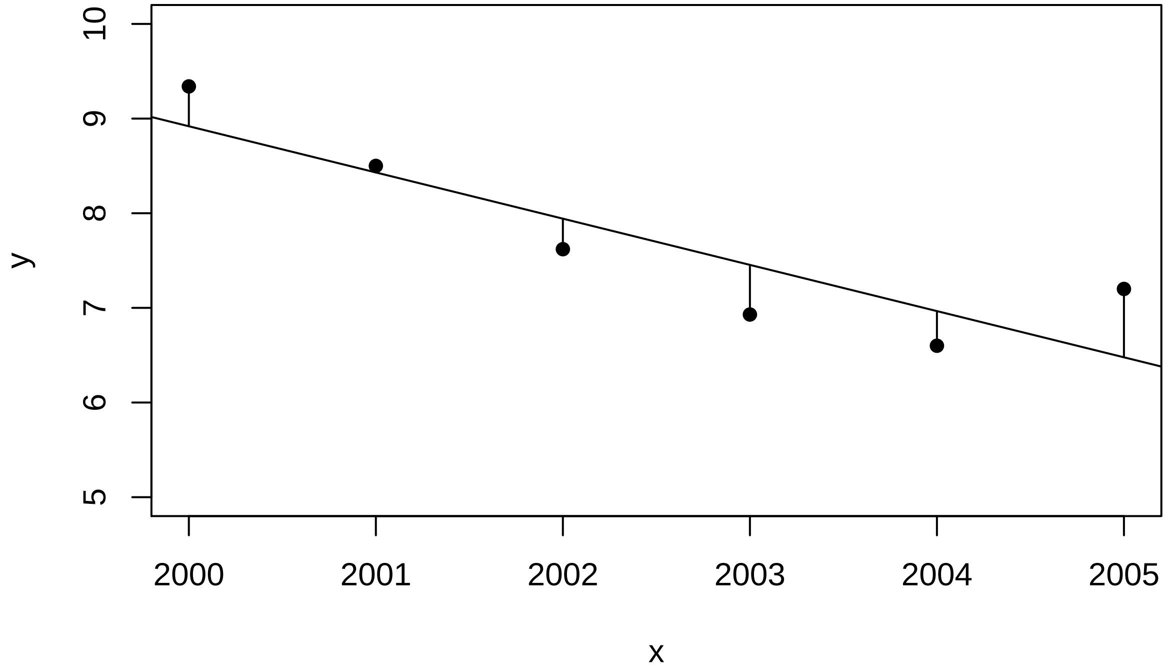
Regression equation $Y = B_0 + B_1X_1 + e$

$\hat{Y} = B_0 + B_1X_1$ where $\hat{Y}$ is the predicted score for Y

$Y - \hat{Y} = e(residual)$

The values of the coefficients (e.g. B_1) are estimated such that the regression model yields optimal predictions. What we want to do is minimise the residuals i.e. minimise the prediction error.

```
x<- c(2000,2001,2002,2003,2004,2005)
y<-c(9.34,8.50,7.62,6.93,6.60,7.2)
m1<-lm(y~x)
fitted<-predict(lm(y~x))
plot(x,y, pch =16, xlim = c(2000,2005), ylim = c(5,10))
abline(m1)
for (i in 1:6) lines(c(x[i],x[i]),c(y[i],fitted[i]))
```



ordinary least squares estimation

Minimise the sum of the squared (SS) residuals

SS.Residual = $\Sigma(Y - \hat{Y})^2$

The best fit slope is found by rotating the line until the SS.Residual is minimised. This gets the maximum likelihood estimate of the slope.

Visual approach

We have the sum of squared deviation scores (SS) in variable Y = SS.Y

We also have the sum of squared deviation scores (SS) in variable Y = SS.Y

The overlap between these two is the sum of cross products between X and Y i.e. SP.XY. So the degree to which the two variables correlate will be a measure of how much overlap there is between the two.

Here's an example with low overlap and thus low correlation.

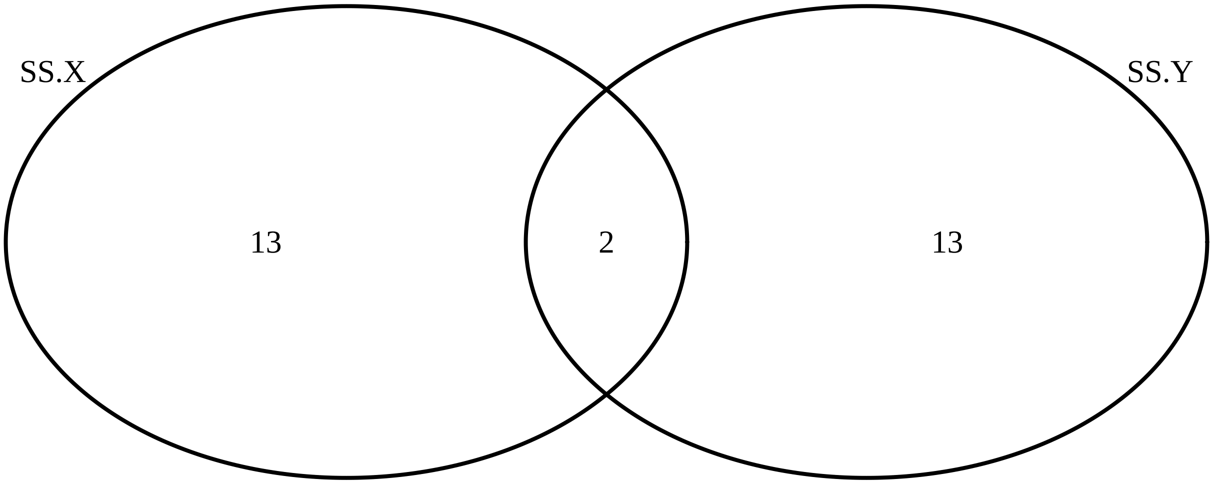
```
## Warning: package 'VennDiagram' was built under R version 3.3.2
```

```
## Loading required package: futile.logger
```

```
## Warning: package 'futile.logger' was built under R version 3.3.2
```

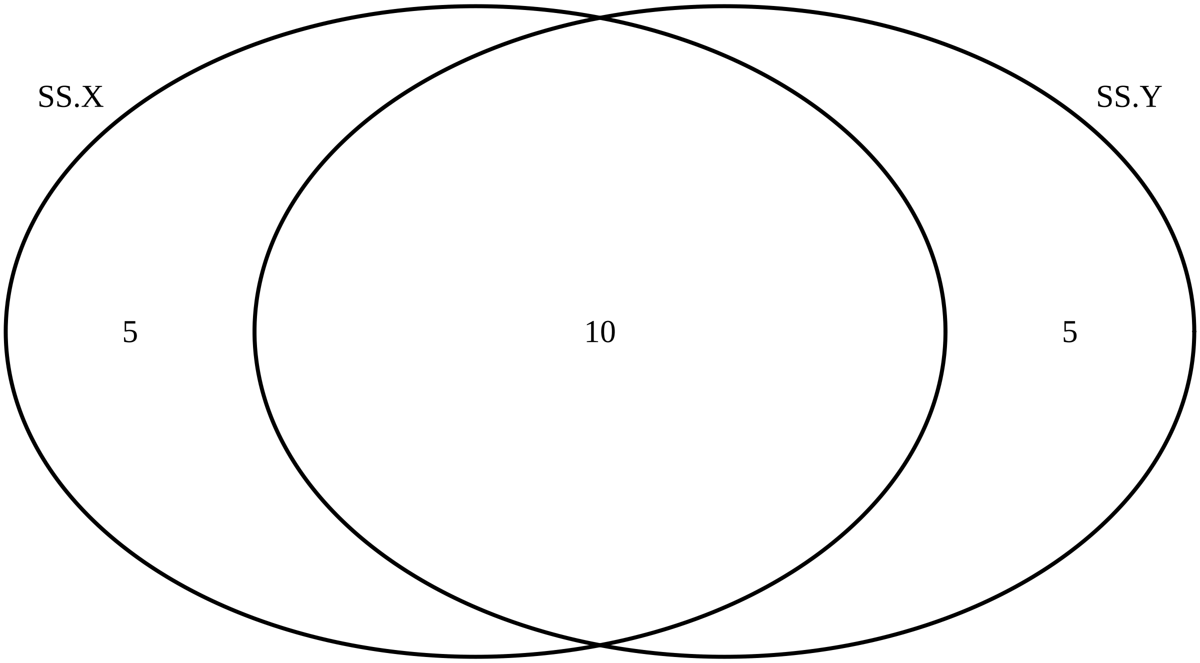
```
##
## Attaching package: 'VennDiagram'
```

```
## The following object is masked from 'package:car':
##
##     ellipse
```



```
## (polygon[GRID.polygon.176], polygon[GRID.polygon.177], polygon[GRID.polygon.178], polygon[GRID.polyg
```

Here's an example with high overlap and thus high correlation.



```
## (polygon[GRID.polygon.185], polygon[GRID.polygon.186], polygon[GRID.polygon.187], polygon[GRID.polyg
```

SP.XY can be thought of as the sum of squares of the model i.e. sum of cross products = SS of the model = SP.XY = SS.Model

Some of the variance in Y is explained by the model and some of it is unexplained, that's the residual. SS.Residual = (SS.Y - SS.Model)

Formula for the unstandardised coefficient

In a simple linear regression

$B_1 = r * (\frac{SD_y}{SD_x})$

where r is the correlation coefficient.

We divide by the standard deviations because we need to take into account the scale of Y and the scale of X. Y may be much more variable than X for instance so this division deals with that.

Formula for the standardised coefficient

Where everything is in Z-scores

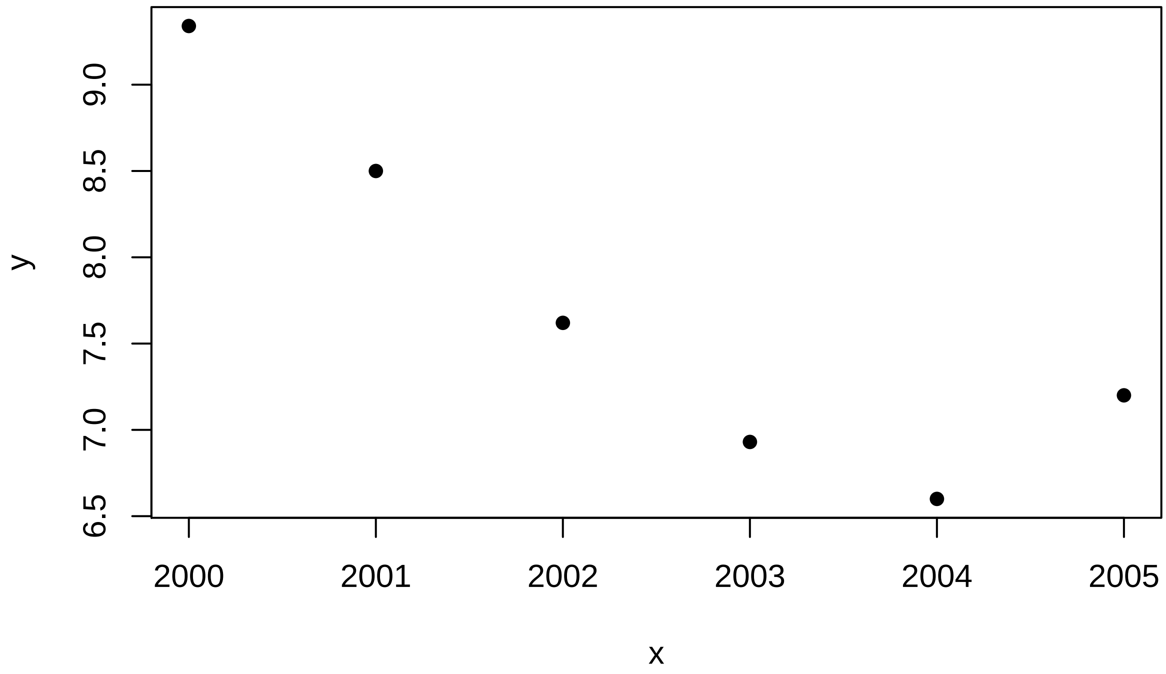
$SD_y = SD_x = 1$

$B = r * (\frac{SD_y}{SD_x})$

$\beta = r$

This is only true for simple linear regression.

```
x<- c(2000,2001,2002,2003,2004,2005)
y<-c(9.34,8.50,7.62,6.93,6.60,7.2)
plot(x,y, pch = 16)
```



```
# take the Z scores for x and y
Zx <- (x-mean(x))/sd(x)
Zy <- (y-mean(y))/sd(y)

# get the correlation for X and Y
cor(Zx,Zy)
```

```
## [1] -0.8799209
```

```
m1<-lm(Zy~Zx)
# Get the slope of y as a function of x
coef(m1)[2]
```

```
##         Zx
## -0.8799209
```

```
# also recall from earlier that the R squared value of a simple linear regression is equal to the corre
summary(m1)$r.squared
```

```
## [1] 0.7742609
```

```
cor(Zx,Zy)^2
```

```
## [1] 0.7742609
```

The correlation gives you a bounded measurement that can be interpreted independently of the scale of the two variables. The closer the estimated correlation is to ±1, the closer the two are to a perfect linear relationship. The regression slope, in isolation, does not tell you that piece of information.

The regression slope gives a useful quantity interpreted as the estimated change in the expected value of Y for a given value of X. Specifically, $\hat{\beta}$ tells you the change in the expected value of Y corresponding to a 1-unit increase in X. This information can not be deduced from the correlation coefficient alone.

Assumptions of linear regression

Normally distributed residuals

Linear relationship between X and Y

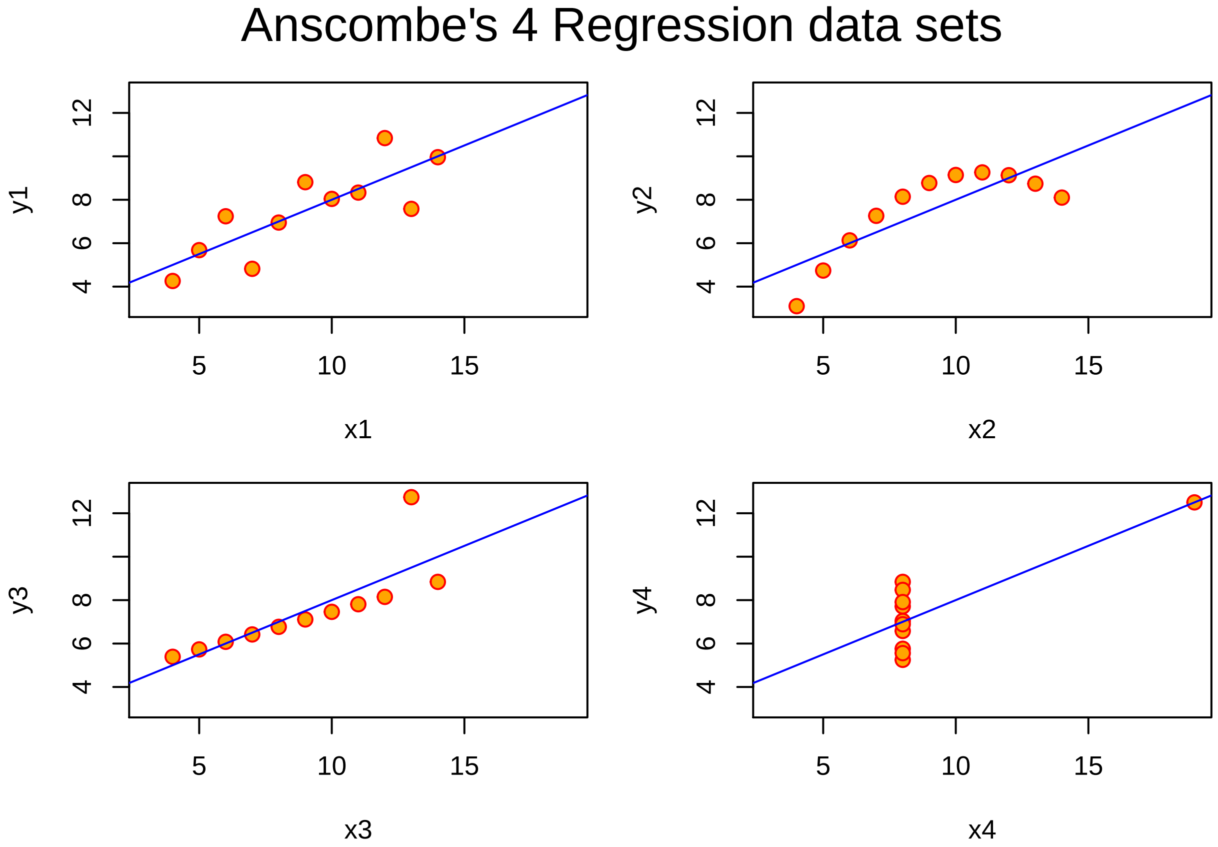
Homoscedasticity

Return to Anscombe's Quartet

The following graphs and their regression coefficients show how similar the four data sets are even for linear regression models.The regression equation for each is approximately:

$\hat{Y} = 3 + 0.5(X_1)$

```
##                   lm1      lm2       lm3       lm4
## (Intercept) 3.0000909 3.000909 3.0024545 3.0017273
## x1          0.5000909 0.500000 0.4997273 0.4999091
```



But only the top left panel looks suitable for a linear regression model. The others look like they don't satisfy the various assumptions of linear regression.

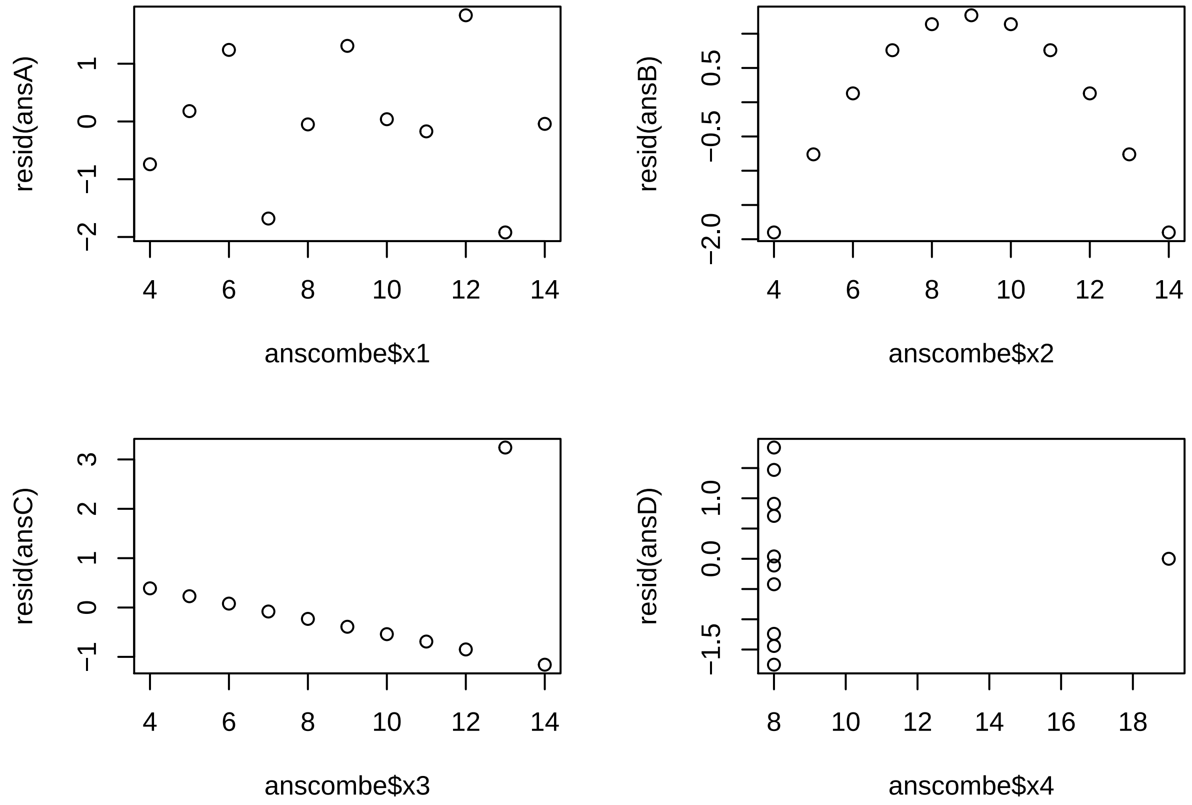
In order to test this we can save the residuals of each model.

$Y = B_0 + B_1 + e$

where

$e = (Y - \hat{Y})$

We can then look at the residuals as a function of the X predictor variable. Then we examine a scatterplot with the X variable on the X-axis and the residuals on the Y-axis.



The plot in the top left is what we are looking for. We don't want any pattern in our residual plot and this suggests no systematic error. The other plots suggest there is a relationship between X and the residual.

Null Hypothesis Significance Testing (NHST)

NHST is a procedure for hypothesis testing. We can consider NHST as a game where step 1 is the identification of the nully hypothesis and the alternative hypothesis.

Step 1

for a correlational study

H_0 = null hypothesis, e.g. r = 0

H_A = alternative hypothesis, e.g. r > 0

where r is the correlation coefficient

or for a regression model

H_0 = null hypothesis, e.g. B = 0

H_A = alternative hypothesis, e.g. B > 0

where B is the slop of the regression model

If the alternative hypothesis predicts the direction of the relationship between X & Y (positive Vs negative) it is termed a directional test (aka a one-tailed test)

Alternatively we could be agnositc and not have any idea about the direction of the relationship. In this case it would be a non-directional test (aka a two-tailed test)

The non-directional test for a regression model would be set up like this:

H_0 = null hypothesis, e.g. B = 0

H_A = alternative hypothesis, e.g. B != 0

Step 2

Assume H_0 is true, then calculate the probability of observing data with these characteristics, given that H_0 is true. This can be confusing because it's the opposite way you'd approach a study. For instance, Jonas Salk didn't predict his vaccination would have no effect.

$p = P(D|H_0)$

The probability of the data given the null hypothesis is true. This is the p-value. If the p-value is very low, then reject H_0, else retain H_0

4 possible outcomes of NHST

Either the null is true or it's false and then, as scientists, we have to successfully pick this out.

	Retain Null	Reject Null
Null is true	Correct decision	Type 2 error (miss)
Null is false	Type 1 error (false alarm)	Correct decision

NHST Overview

$p = P(D|H_0)$

Given that the null hypothesis is true, the probability of these, or more extreme data, is p.

This does not mean the probability of the null hypothesis being true in p.

In other words, $P(D|H_0)! = P(H_0|D)$

NHST so far

for correlation - is the correlation significantly different from zero?

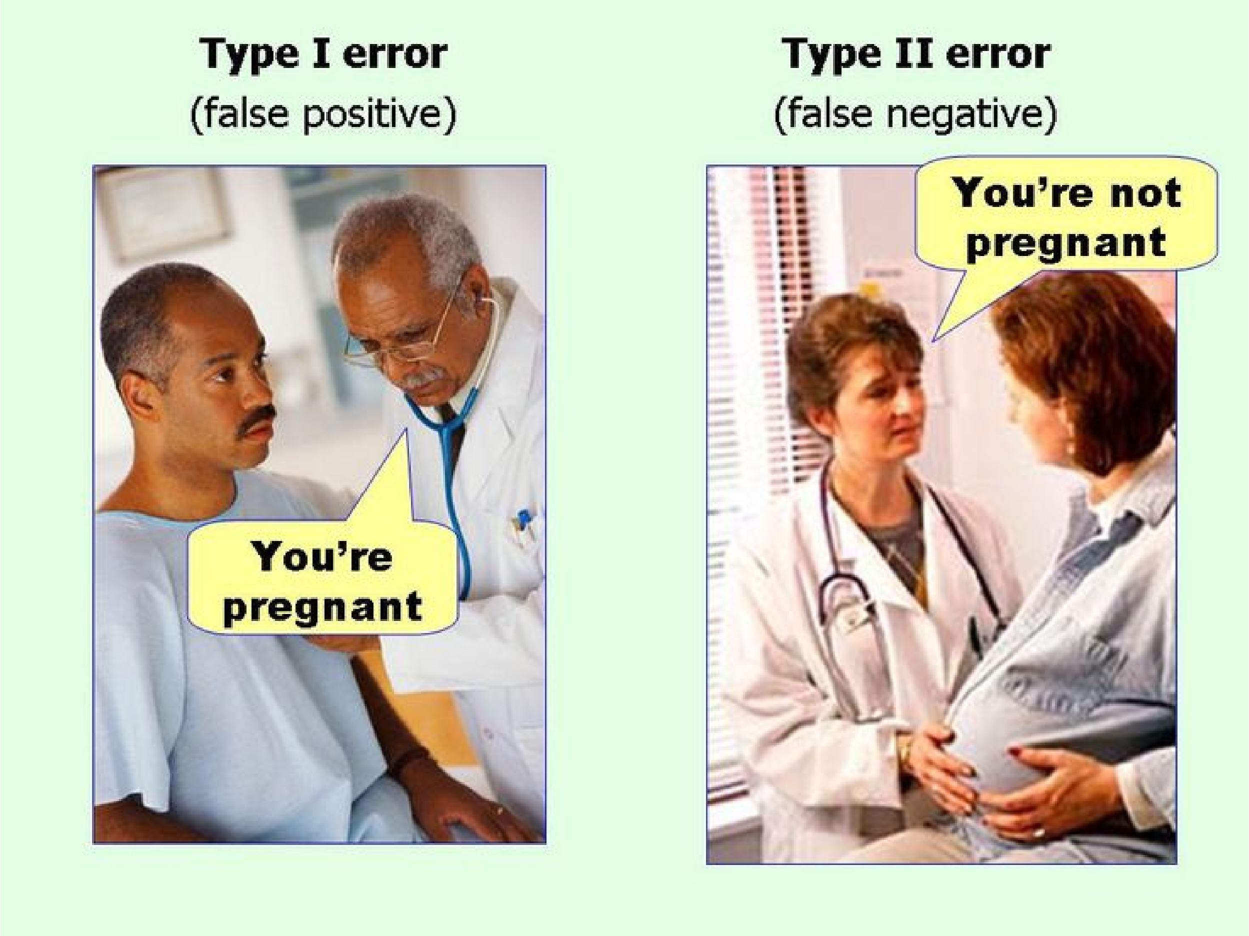

Type I error
(false positive)
You're pregnant
Type II error
(false negative)
You're not pregnant

Figure 1:

B - is the slope of the regression line for X significantly different from zero?

NHST for B - the slope of the regression line

t = B/SE

B = the unstandardised regression coefficient

SE = standard error

SE = $\sqrt{\frac{SS.Residual}{N-2}}$

NHST Problems and Remedies

1 - Biased by Sample Size

The p-value you get is based on a t-value and this t -value is affected by the standard error. N is in the denominator of the standard error and standard error is in the denominator of the t-value. Thus, if sample size goes up the standard error will go down, and if the standard error goes down the t-value will go up regardless of the slope value.

t = B/SE

B = the unstandardised regression coefficient

SE = standard error

SE = $\sqrt{\frac{SS.Residual}{N-2}}$

```r
x<-rnorm(10000,mean=100,sd=10)
y<-rnorm(10000,mean=100,sd=10)*x
df<-data.frame (x,y)
smallSample<-df[sample(nrow(df), 10), ]
m1<-lm(smallSample$y~smallSample$x)
pVal <- anova(m1)$'Pr(>F)'[1];pVal
```

```
## [1] 0.150482
```

```r
bigSample<-df[sample(nrow(df), 100), ]
m2<-lm(bigSample$y~bigSample$x)
pVal <- anova(m2)$'Pr(>F)'[1];pVal
```

```
## [1] 2.122148e-15
```

2 - Arbitrary Decision Rule

The cut-off value (alpha) is arbitrary

$p < 0.05$ is considered standard but still arbitrary.

Problems arise when p is close to 0.05 but not less than 0.05. p-hacking etc.

3 - Yokel local test

Many researchers use NHST because it is the only approach they know. NHST encourages weak hypothesis testing.

4 - Error prone

Type 1 errors - The probability of Type 1 errors increases when researchers conduct multiple NHSTs, especially when it's on the same dataset. Have to correct for these multiple tests.

Type 2 errors - Many fields of research are plagued by a large degree of sampling error because we can only get a relatively small sample relative to the population, which makes it difficult to detect an effect, even when the effect exists.

Shady Logic

Modus tollens operates like this:

If p then q

Not q

Therefore, not p

Equivalently, in the language of statistics:

If the null hypothesis is correct, then these data can not occur

The data have occurred

Therefore, the null hypothesis is false

But in NHST the language is more probabilistic than this:

If the null is correct, then these data are highly unlikely.

These data have occurred

Therefore, the null is highly unlikely

To take an equivalent example:

If a person plays football, then he or she is probably not a professional player

This person is a professional player

Therefore, he or she probably does not play football.

NHST Remedies

Remedy for Bias by sample size

Supplement all NHSTs with estimates of effect size to get at the magnitude of the effect. For example, in regression, report standardised regression coefficients and the model R-squared.

Remedy for Arbitrary decision rule

Again supplement all NHSTs with estimates of effect size to get at the magnitude of the effect. Also, avoid phrases such as "marginally signifcant" or "highly significant".

Remedy for Yokel local test

Learn other forms of hypothesis testing. Consider multiple alternative tests and use model selection.

Remedy for NHST being error prone

Replicate significant effects to avoid long-term impact of type 1 errors

Obtain large and representative samples to avoid type 2 errors.

Remedy for Shady logic

Simply remember, $p = P(D|H_0)$

Or avoid NHST, and instead,

Report confidence intervals only or,

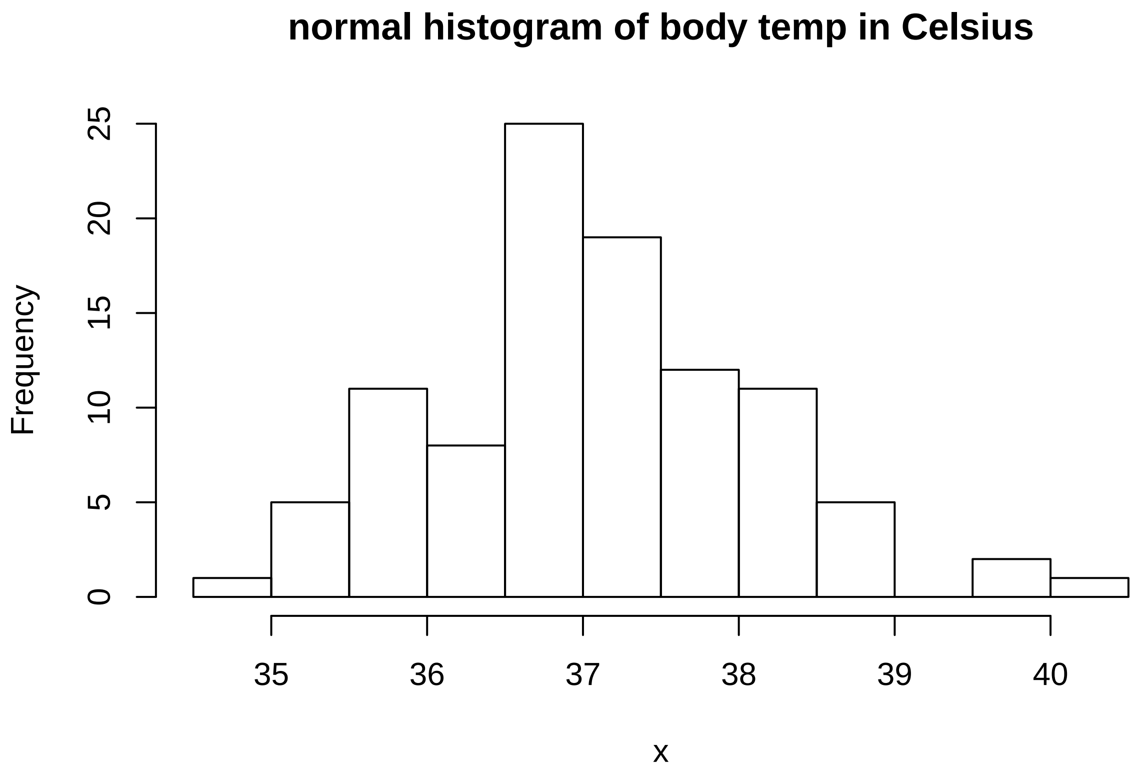
Apply Bayesian inference

Central Limit Theorem

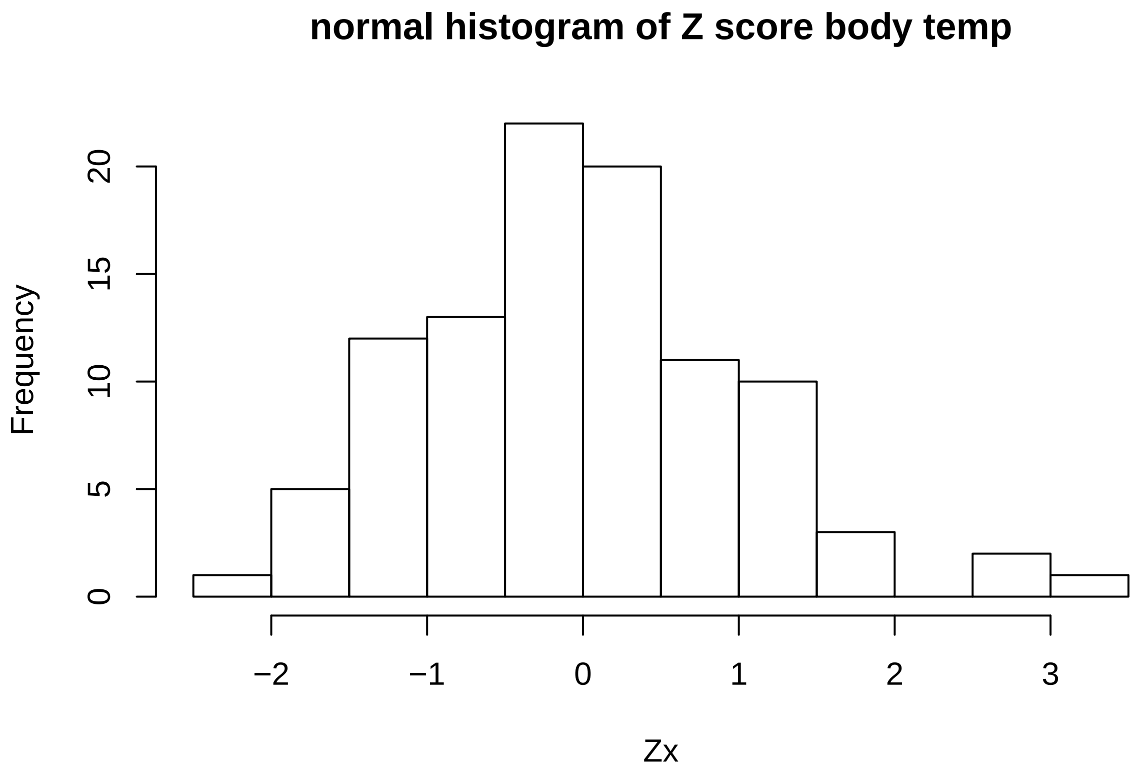
Review of histograms

Histograms are used to display distributions e.g. the body temperature of a random sample of healthy people.

```r
x <- rnorm(100, mean = 37,sd=1)
Zx <- (x-mean(x))/sd(x)
hist(x, main = "normal histogram of body temp in Celsius")
```

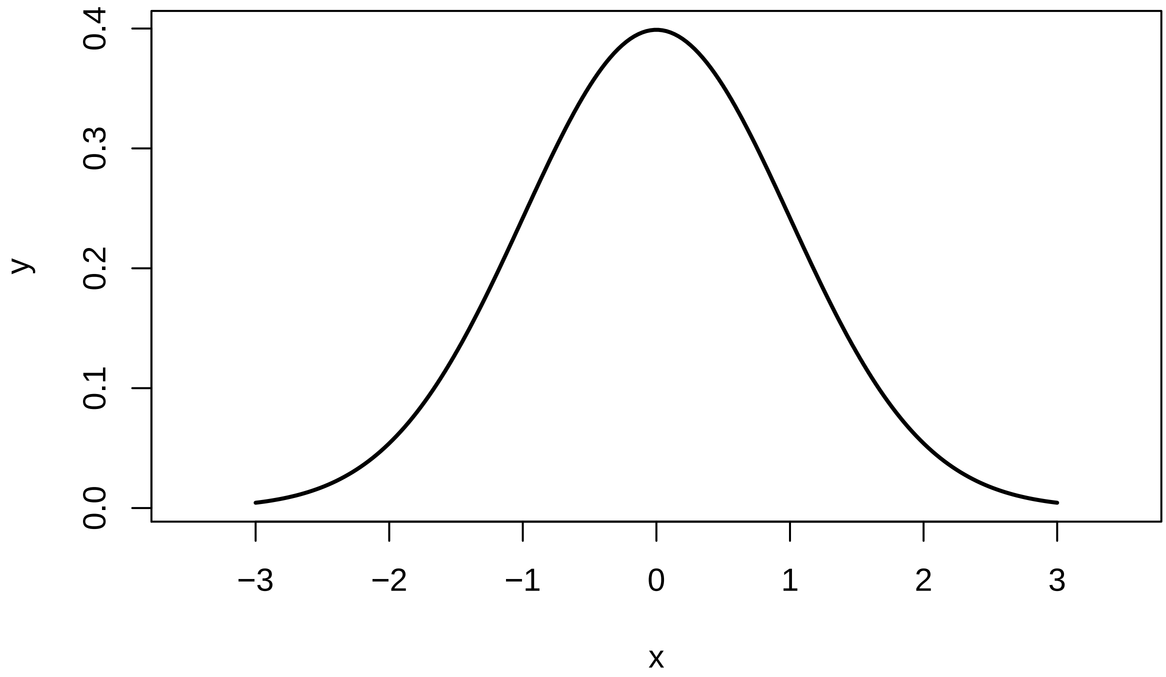


```
hist(Zx, main = "normal histogram of Z score body temp")
```



If a distribution is perfectly normal then the properties of the distribution are known.

```
x<-seq(-3,3,length=200)
s = 1
mu = 0
y <- (1/(s * sqrt(2*pi))) * exp(-((x-mu)^2)/(2*s^2))
plot(x,y, type="l", lwd=2, col = "black", xlim = c(-3.5,3.5))
```



Things to note about a normal distribution.

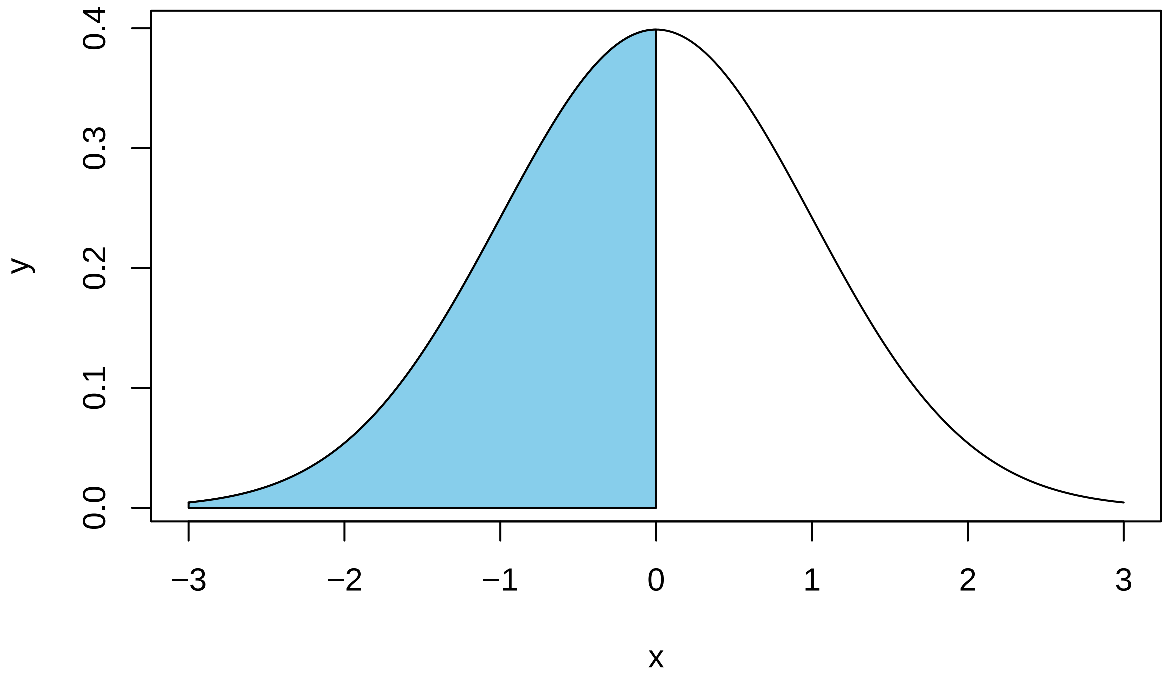
50% of the data fall above the mean and 50% below.

The majority of the data fall between 2 standard deviations above and below the mean. If you are higher or lower than these values then you are an extreme point.

This allows for predictions about the distribution because we know predictions aren't certain rather they are probabilistic.

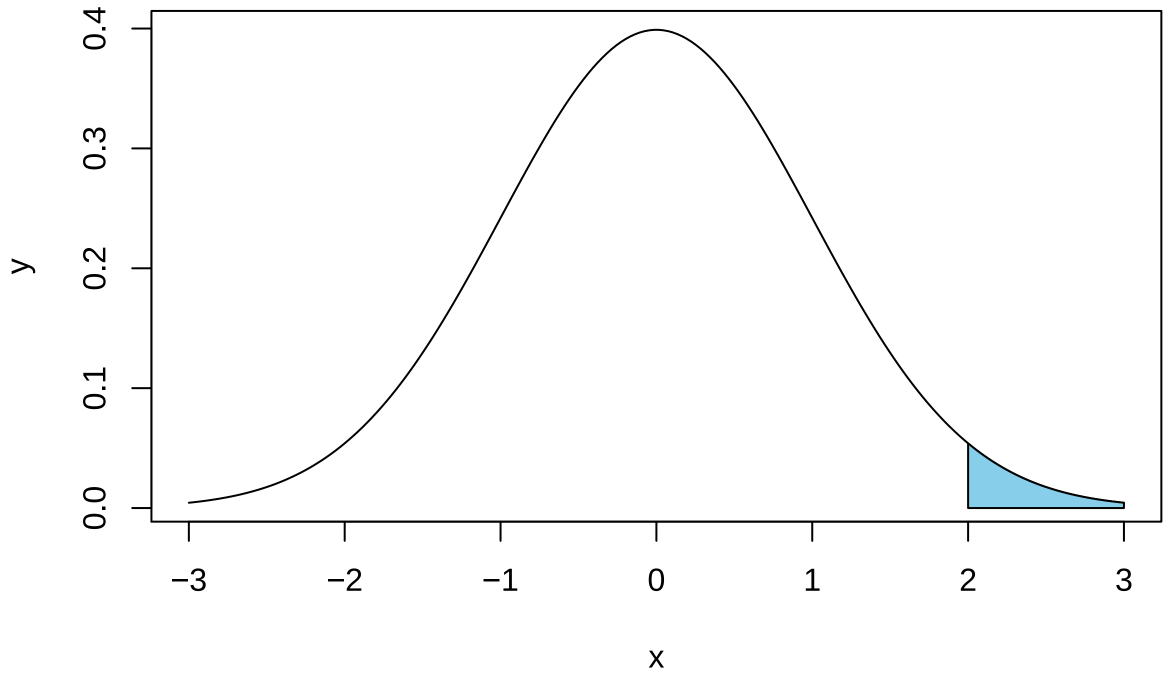
For example if one person is randomly selected from the sample, what is the probability that his or her body temperature is less than Z = 0 (or X = 37 for degrees celsius)? It's, p = 0.50.

```
x=seq(-3,3,length=200)
y=dnorm(x,mean=0,sd=1)
plot(x,y,type="l")
x=seq(-3,0,length=100)
y=dnorm(x,mean=0,sd=1)
polygon(c(-3,x,0),c(0,y,0),col="skyblue")
```



Alternatively, if one person is randomly selected from the sample, what is the probability that his or her body temperature is greater than Z = 2 (or X = 38 for degrees celsius)? It's, p = 0.20.

```
x=seq(-3,3,length=200)
y=dnorm(x,mean=0,sd=1)
plot(x,y,type="l")
x=seq(2,3,length=100)
y=dnorm(x,mean=0,sd=1)
polygon(c(2,x,3),c(0,y,0),col="skyblue")
```



If this sample is healthy, then no one should have a fever.

I detected a person with a fever

Therefore, this sample is not 100% healthy.

Sampling Distributions

But rather than using NHST on distributions of individuals we use NHST on distributions of sample statistics.

A sampling Distributions is a distribution of sample statistics, obtained from multiple samples, for example, a distribution of sample means, of sample correlations, of sample regression coefficients.

Sampling distributions are important in statistics because they provide a major simplification on the route to statistical inference. More specifically, they allow analytical considerations to be based on the sampling distribution of a statistic, rather than on the joint probability distribution of all the individual sample values.

It's important to realise that a sampling distribution is hypothetical. Instead we will only have a single sample.

Let's assume a mean is calculated from a sample, obtained randomly from the population.

Assume a certain sample size, N

Now assume we had multiple random samples (which is not what we do in practice), all of size N, and therefore many sample means. These would all differ a little bit because of sampling error.

Collectively, they form a sampling distribution.

Marrying sampling distributions and probability

If one sample is obtained from a normal healthy population, what is the probability that the sample mean is less than Z = 0? Again, it's p = 0.50.

If one sample is obtained from a normal healthy population, what is the probability that the sample mean is less than Z = 2? Again, it's p = 0.20.

In the latter case, if this population is healthy, then no one sample should have a high mean body temperature.

I obtained a very high sample mean.

Therefore, the population is not healthy.

Central Limit Theorem

Three principles

1. The mean of a sampling distribution is the same as the mean of the population
2. The standard deviation of the sampling distribution is the square root of the variance of the sampling distribution $\sigma^2 = \frac{\sigma^2}{N}$
3. The shape of a sampling distribution is approximately normal if either (a) N >= 30 or (b) the shape of the population is normal.

NHST and the Central Limit Theorem

Multiple regression

Assume the null is true

Conduct a study

Calculate B, SE and t

where t = B/SE

the p-value is a function of t and sample size

Conceptually, the t-value is a ratio of what we observed (e.g. the slope of the regression line) relative to what we would expect due to chance (e.g. the slope is zero). A ratio of 1 would be something around the null.

If the null hypothesis is true, then no one sample should have a very low or very high slope. Thus, if I obtain a very high slope I should reject the null. But what does 'very high' mean?

The very low or very high values depend on the normal distribution. Remember, the shape of a sampling distribution is approximately normal if either (a) N >= 30 or (b) the shape of the population is normal.

That means I can make probability judgements about the outcome. Note, that the third principle of the central limit theorem didn't say you get a normal distribution, rather it said you approximate one.

Instead, we get a t-distribution which comes from a family that are dependent on the sample size. As your sample size gets smaller your t-distribution gets a little wider which means you need a larger t-value to get out into the extremes to get a low p-value.

This all means that our ideas of 'very high' or 'very low' come from p being < 0.05.

Remember, that sampling error, and therefore standard error, is largely determined by sample size.

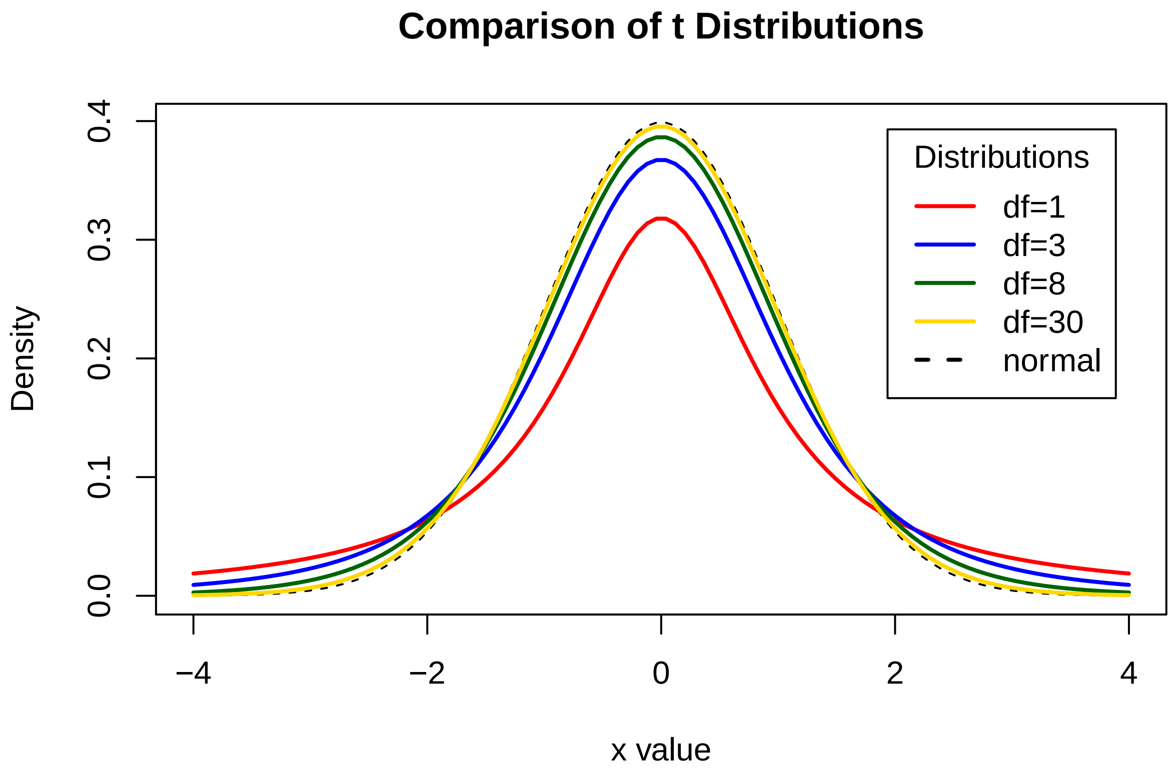
Standard error is the standard deviation of the sampling distribution. As samples get larger they're going to squeeze in around the mean and the standard error will decrease. It's important to note that NHST is biased by sample size.

t = B/SE

B = the unstandardised regression coefficient

SE = standard error

SE = $\sqrt{\frac{SS.Residual}{N-2}}$



As sample size increases, the actual mean approximates zero and the standard error shrinks as a function of sample size.

```r
meanSamplingDist <- vector(length=length(i))
SESamplingDist <- vector(length=length(i))
for(i in 1:6){
  x <- rnorm(10**i)
  Zx <- (x-mean(x)/sd(x))
  SE <- sd(Zx)/sqrt(length(Zx))
  meanSamplingDist[i]<-(mean(Zx))
  SESamplingDist[i]<- (SE)
}

meanSamplingDist # the mean of the sampling distribution
```

```
## [1]  7.563941e-03 -2.984371e-03  1.063952e-03  2.671753e-05  3.730276e-06
## [6]  2.679761e-07
```

```
SESamplingDist # the standard error of the sampling distribution
```

```
## [1] 0.330676323 0.101355430 0.032241270 0.010105143 0.003155695 0.001000965
```

Confidence Intervals

Confidence intervals around sample means

All sample statistics e.g. a sample mean, are point estimates i.e. one point from a sample distribution. More specifically, a sample mean represents a single point in a sampling distribution. Any one sample will never be perfect.

The logic of confidence intervals is to report a range of values, rather than a single value. In other words, report an interval estimate rather than a point estimate.

We can define a confidence interval as an interval estimate of a population parameter, based on a random sample. The degree of confidence, e.g. 95%, represents the probability that the interval captures the true population parameter.

The main argument for interval estimates is the reality of sampling error. Sampling error implies that point estimates will vary from one study to the next (we use standard error to get a measure of sampling error). A researcher will therefore be more confident about accuracy with an interval estimate.

Below we can see that if we take a sample mean from a normal distribution we get different values each time due to sampling error. However, because our sample size is relatively big (30), this is quite small.

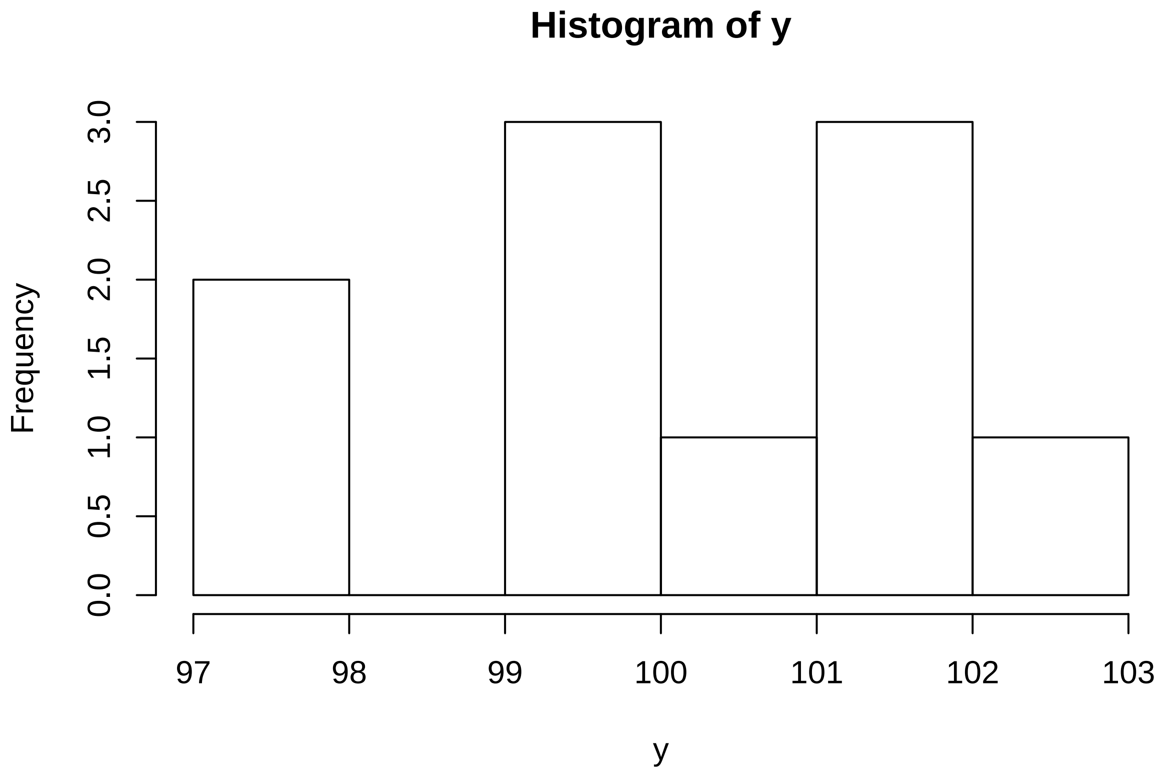
```
x<-rnorm(10000,100,10)
mean(x)
```

```
## [1] 99.88437
```

```
y <- replicate(10, {
  mm <- sample(x,30)
  mean(mm)
  print(mean(mm))
  })
```

```
## [1] 97.37197
## [1] 99.4733
## [1] 101.094
## [1] 99.44923
## [1] 99.12837
## [1] 101.6704
## [1] 102.8188
## [1] 101.1469
## [1] 100.2482
## [1] 97.0105
```

```
hist(y)
```



But if we reduce sample size there's more fluctuation in the point estimates i.e. the sample means.

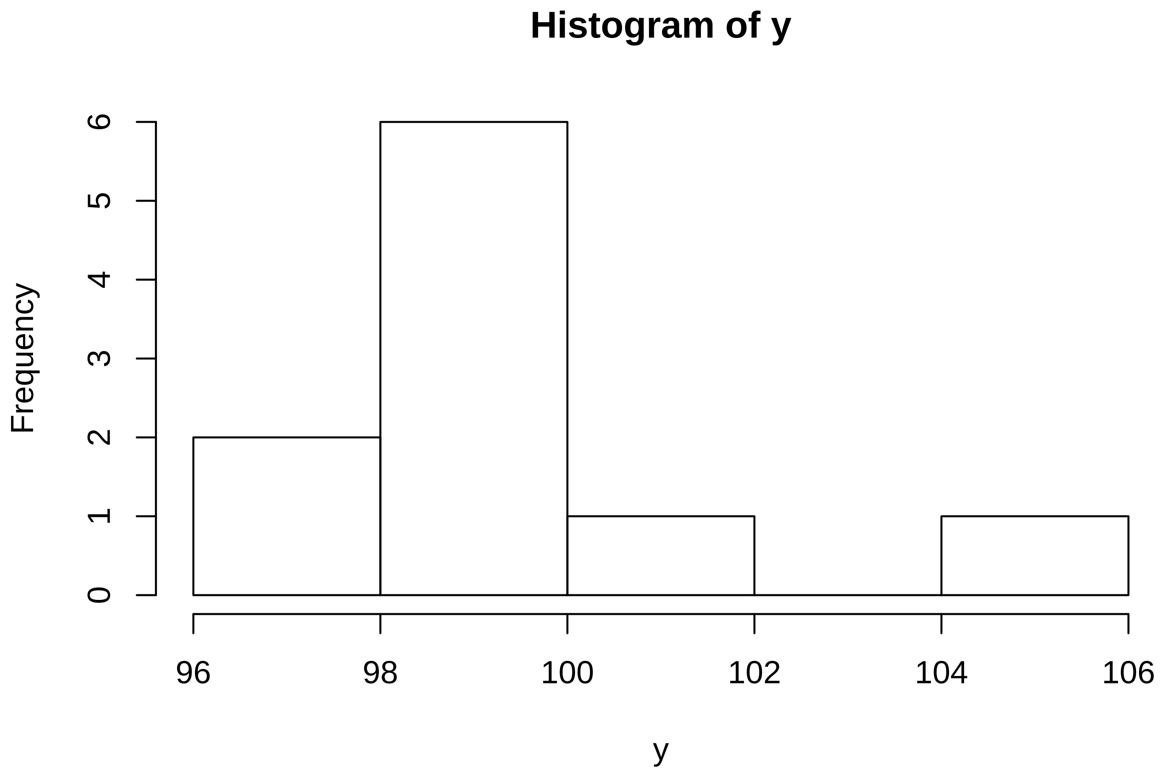
```
x<-rnorm(10000,100,10)
mean(x)
```

```
## [1] 99.97348
```

```
y <- replicate(10, {
  mm <- sample(x,10)
  mean(mm)
  print(mean(mm))
  })
```

```
## [1] 99.95794
## [1] 101.0147
## [1] 98.89737
## [1] 99.2869
## [1] 96.23901
## [1] 96.96515
## [1] 105.7327
## [1] 98.71632
## [1] 98.62212
## [1] 99.19111
```

```
hist(y)
```



The width of a confidence interval is influenced by sample size (as sample size increases you can be more confident that the interval will contain the estimate) and variance in the population (and sample).That is to say that standard error influences the width of the confidence intervals.

We can see this below in that the sample of 10 points from the original data gives a much wider confidence interval than the sample of 30 data points. The width of the intervals can be embarrassingly large.

```
x<-rnorm(10000,100,10)
mean(x)
```

```
## [1] 99.94706
```

```
smallSample <- sample(x,10)
largeSample <- sample(x,30)
t.test(smallSample)
```

```
##
##  One Sample t-test
##
## data:  smallSample
## t = 26.685, df = 9, p-value = 7.049e-10
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##   85.15837 100.93368
## sample estimates:
## mean of x
##  93.04603
```

```r
t.test(largeSample)
```

```
##
##  One Sample t-test
##
## data:  largeSample
## t = 66.603, df = 29, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##   97.45037 103.62494
## sample estimates:
## mean of x
##  100.5377
```

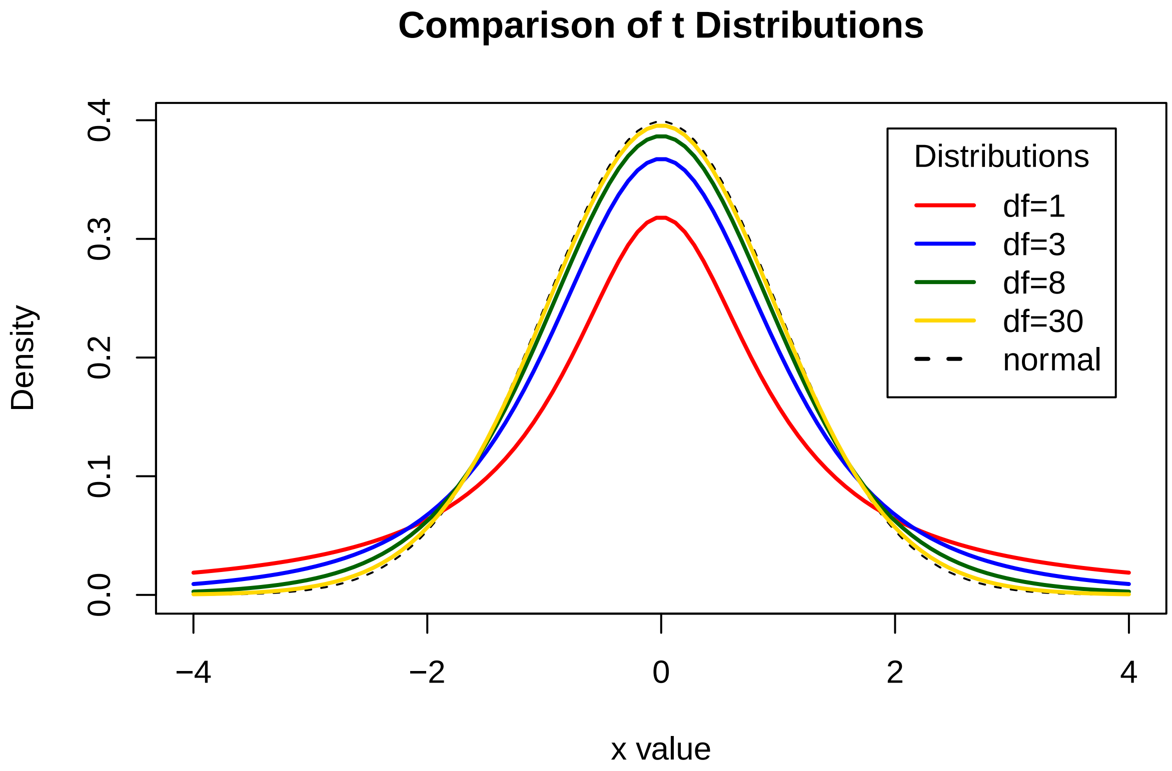
How to calculate the confidence interval

Upper bound = M + t(SE)

Lower bound = M - t(SE)

$SE = \frac{SD}{\sqrt(N)}$

t depends on the level of confidence desired (e.g. 95% which corresponds to an alpha of 0.05) and the sample size. The sampling distribution we assume depends on the exact size of your sample where the width of your sampling distribution gets a little bit wider with smaller samples. Smaller samples are going to give you more standard error so if you want 95% of that distribution then you'll have to go further out in the distribution because you'll need a slightly higher t-value.



Let's do a worked example where we have a sample of 1000 men drawn from a population whose weights were measured. We find that the average man in our sample is 180 pounds, and the standard deviation of the sample is 30 pounds. What is the 95% confidence interval?

```
sampleWeights<-rnorm(1000,180,30)
mean(sampleWeights)
```

```
## [1] 180.9167
```

```
#first find the SE
standardError<-sd(sampleWeights)/sqrt(length(sampleWeights));standardError
```

```
## [1] 0.9539764
```

```
#Find critical value. The critical value is a factor used to compute the margin of error. To express th

alpha<-1-95/100; alpha
```

```
## [1] 0.05
```

```
criticalProbability<-1-alpha/2;criticalProbability
```

```
## [1] 0.975
```

```
degreesOfFreedom<-length(sampleWeights)-1; degreesOfFreedom
```

```
## [1] 999
```

```
#The critical value is the t statistic having 999 degrees of freedom and a cumulative probability equal
critcalValue<-qt(.975,df=999);critcalValue
```

```
## [1] 1.962341
```

```
#Compute margin of error (ME): ME = critical value * standard error = 1.96 * 0.95 = 1.86

confidenceInterval<-critcalValue*standardError;confidenceInterval
```

```
## [1] 1.872028
```

```
#The range of the confidence interval is defined by the sample statistic +/- margin of error. And the u
mean(sampleWeights) + confidenceInterval
```

```
## [1] 182.7887
```

```
mean(sampleWeights) - confidenceInterval
```

```
## [1] 179.0447
```

```
#compare with inbuilt function
t.test(sampleWeights)
```

```
##
##  One Sample t-test
##
## data:  sampleWeights
## t = 189.64, df = 999, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  179.0447 182.7887
## sample estimates:
## mean of x
##  180.9167
```

We can do the exact same calculation again only this time with a smaller sample size to see what effect that will have on the confidence intervals.

```
## [1] 183.2713

## [1] 2.797939

## [1] 0.05

## [1] 0.975

## [1] 99

## [1] 1.962341

## [1] 5.490512

## [1] 188.7618

## [1] 177.7808

##
##  One Sample t-test
##
## data:  sampleWeights
## t = 65.502, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  177.7196 188.8230
## sample estimates:
```

```
## mean of x
##  183.2713
```

Confidence Intervals around regression coefficients

All sample statistics, e.g. a regression coefficient (B), are point estimates and can have associated confidence intervals.

More specifically, a regression coefficient, represents a single point in a sampling distribution. It would be better to report interval estimates.

In regression, $t = \frac{B}{SE}$ and this t will come from a distribution of t.

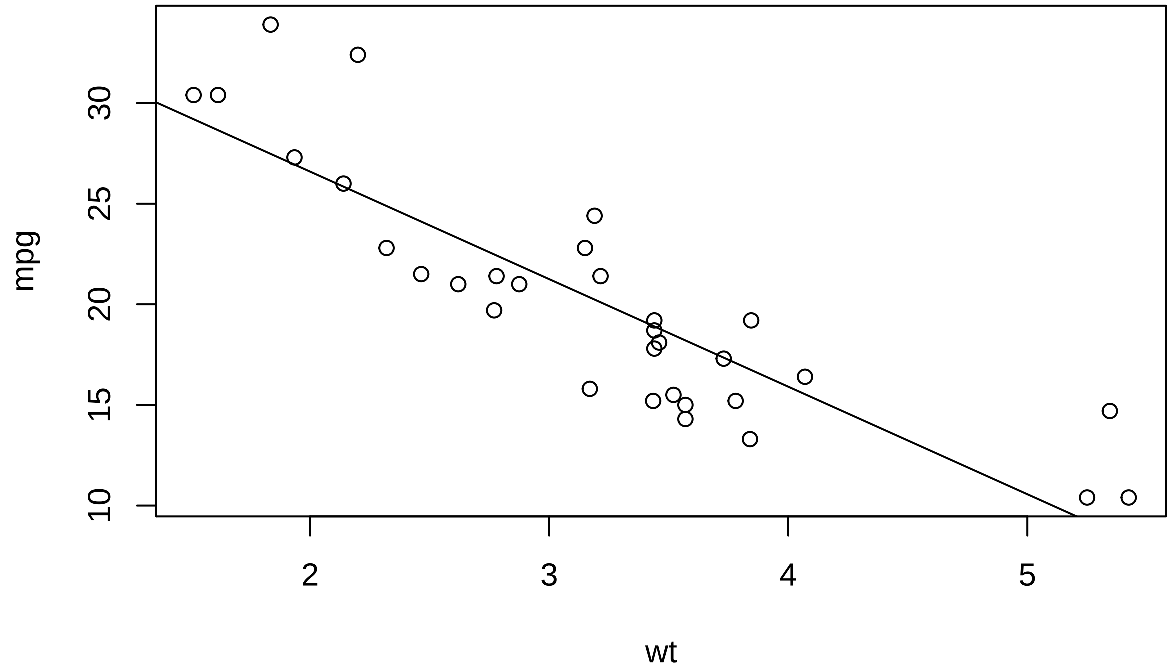
Let's run through an example of how to get the confidence interval for the slope of a regression line.

First off we need to get the slope of the line which is our point estimate sample statistic around which we'll construct our confidence interval

```
#library(ISwR)
fit <- lm(mpg ~  wt, data = mtcars)
summary(fit)
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## wt           -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.046 on 30 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7446
## F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

```
plot(mpg ~  wt, data = mtcars)
abline(fit)
```



```r
slope<-coef(fit)["wt"]
```

Then we select a confidence interval, here we'll use 95%.

Then we have to find the standard error of our regression slope:

$$SE_{slope} = \frac{\sqrt{\frac{\Sigma(Y-Y')^2}{N}}}{\sqrt{\Sigma(x-\bar{x})^2}}$$

```r
#library(ISwR)
fit <- lm(mpg ~  wt, data = mtcars)
mtcars$mpg-fitted(fit)
```

```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710
##          -2.2826106          -0.9197704          -2.0859521
##      Hornet 4 Drive   Hornet Sportabout             Valiant
##           1.2973499          -0.2001440          -0.6932545
##          Duster 360           Merc 240D            Merc 230
##          -3.9053627           4.1637381           2.3499593
##            Merc 280           Merc 280C          Merc 450SE
##           0.2998560          -1.1001440           0.8668731
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood
##          -0.0502472          -1.8830236           1.1733496
## Lincoln Continental   Chrysler Imperial            Fiat 128
##           2.1032876           5.9810744           6.8727113
##         Honda Civic      Toyota Corolla       Toyota Corona
##           1.7461954           6.4219792          -2.6110037
##    Dodge Challenger         AMC Javelin          Camaro Z28
##          -2.9725862          -3.7268663          -3.4623553
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2
##           2.4643670           0.3564263           0.1520430
##        Lotus Europa      Ford Pantera L        Ferrari Dino
##           1.2010593          -4.5431513          -2.7809399
##       Maserati Bora          Volvo 142E
##          -3.2053627          -1.0274952
```

```
(mtcars$mpg-fitted(fit))^2
```

```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710
##         5.210311365         0.845977581         4.351196241
##      Hornet 4 Drive   Hornet Sportabout             Valiant
##         1.683116864         0.040057604         0.480601837
##          Duster 360           Merc 240D            Merc 230
##        15.251857449        17.336715379         5.522308649
##            Merc 280           Merc 280C          Merc 450SE
##         0.089913646         1.210316727         0.751469030
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood
##         0.002524781         3.545777963         1.376749259
## Lincoln Continental   Chrysler Imperial            Fiat 128
##         4.423818910        35.773250845        47.234160512
##         Honda Civic      Toyota Corolla       Toyota Corona
##         3.049198454        41.241816442         6.817340533
##    Dodge Challenger         AMC Javelin          Camaro Z28
##         8.836268903        13.889532530        11.987904418
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2
##         6.073104857         0.127039726         0.023117073
##        Lotus Europa      Ford Pantera L        Ferrari Dino
##         1.442543495        20.640223569         7.733626788
##       Maserati Bora          Volvo 142E
##        10.274349735         1.055746376
```

```
sum((mtcars$mpg-fitted(fit))^2)
```

```
## [1] 278.3219
```

```
sum((mtcars$mpg-fitted(fit))^2)/(length(mtcars$mpg)-2)
```

```
## [1] 9.277398
```

```
sqrt(sum((mtcars$mpg-fitted(fit))^2)/(length(mtcars$mpg)-2))
```

```
## [1] 3.045882
```

```
mtcars$wt-mean(mtcars$wt)
```

```
##  [1] -0.59725 -0.34225 -0.89725 -0.00225  0.22275  0.24275  0.35275
##  [8] -0.02725 -0.06725  0.22275  0.22275  0.85275  0.51275  0.56275
## [15]  2.03275  2.20675  2.12775 -1.01725 -1.60225 -1.38225 -0.75225
## [22]  0.30275  0.21775  0.62275  0.62775 -1.28225 -1.07725 -1.70425
## [29] -0.04725 -0.44725  0.35275 -0.43725
```

```
(mtcars$wt-mean(mtcars$wt))^2
```

```
##  [1] 0.3567075625 0.1171350625 0.8050575625 0.0000050625 0.0496175625
##  [6] 0.0589275625 0.1244325625 0.0007425625 0.0045225625 0.0496175625
## [11] 0.0496175625 0.7271825625 0.2629125625 0.3166875625 4.1320725625
## [16] 4.8697455625 4.5273200625 1.0347975625 2.5672050625 1.9106150625
## [21] 0.5658800625 0.0916575625 0.0474150625 0.3878175625 0.3940700625
## [26] 1.6441650625 1.1604675625 2.9044680625 0.0022325625 0.2000325625
## [31] 0.1244325625 0.1911875625
```

```
sum((mtcars$wt-mean(mtcars$wt))^2)
```

```
## [1] 29.67875
```

```
sqrt(sum((mtcars$wt-mean(mtcars$wt))^2))
```

```
## [1] 5.44782
```

```
sqrt(sum((mtcars$mpg-fitted(fit))^2)/(length(mtcars$mpg)-2)) / sqrt(sum((mtcars$wt-mean(mtcars$wt))^2))
```

```
## [1] 0.559101
```

```
SESlope <- sqrt(sum((mtcars$mpg-fitted(fit))^2)/(length(mtcars$mpg)-2)) / sqrt(sum((mtcars$wt-mean(mtca
```

As before, for the confidence interval of a mean we need to find the critical value. The critical value is a factor used to compute the margin of error. With simple linear regression, to compute a confidence interval for the slope, the critical value is a t score with degrees of freedom equal to n - 2. To find the critical value, we take these steps.

```
alpha<-1-95/100; alpha
```

```
## [1] 0.05
```

```
criticalProbability<-1-alpha/2;criticalProbability
```

```
## [1] 0.975
```

```
degreesOfFreedom<-length(mtcars$mpg)-2; degreesOfFreedom
```

```
## [1] 30
```

```
#The critical value is the t statistic having 30 degrees of freedom and a cumulative probability equal
critcalValue<-qt(.975,df=30);critcalValue
```

```
## [1] 2.042272
```

```
# Compute margin of error (ME): ME = critical value * standard error =
ME<-critcalValue * SESlope; ME
```

```
## [1] 1.141837
```

```
#Specify the confidence interval. The range of the confidence interval is defined by the sample statist

slope + ME
```

```
##        wt
## -4.202635
```

```
slope - ME
```

```
##        wt
## -6.486308
```

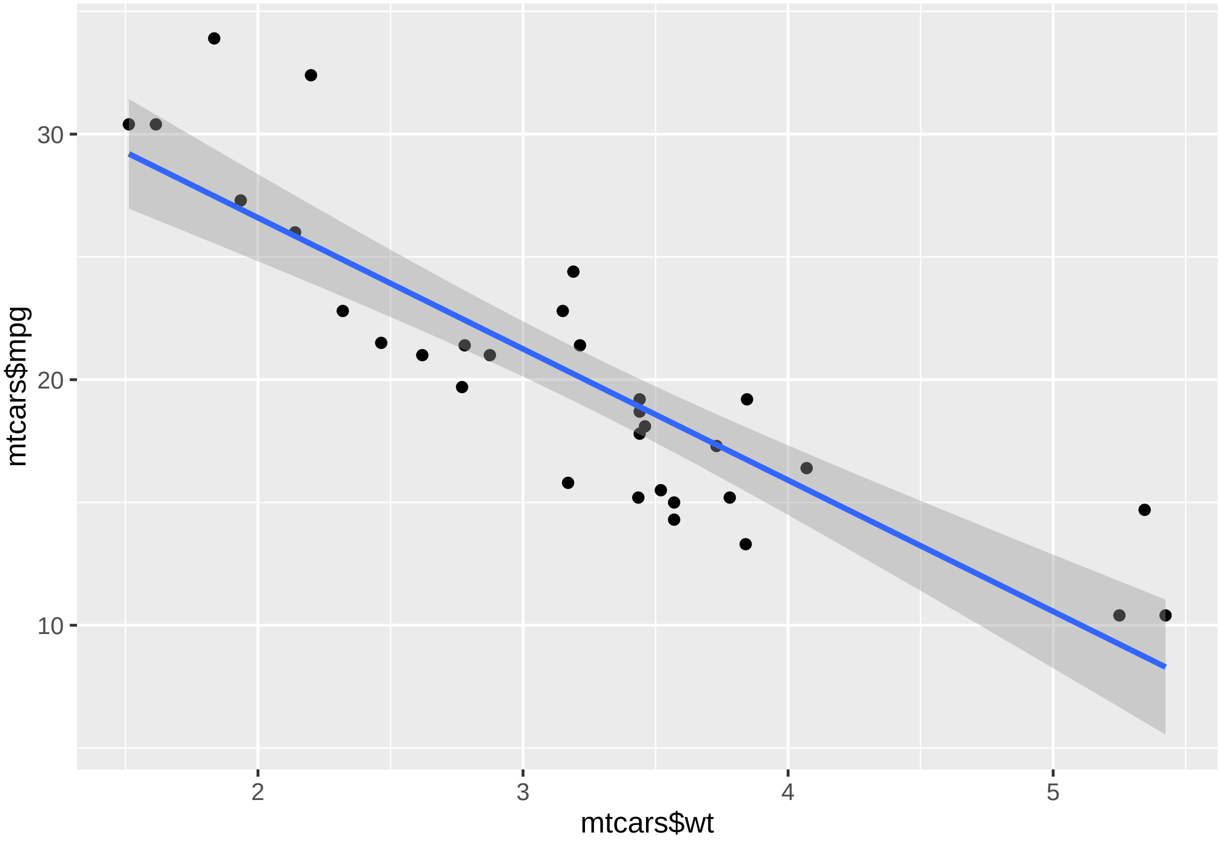
```
# Therefore, the 95% confidence interval is -6.486308 to -4.202635. That is, we are 95% confident that

# we can again check this against the inbuilt function
confint(fit, "wt", level = 0.95)
```

```
##        2.5 %    97.5 %
## wt -6.486308 -4.202635
```

In general, if the confidence intervals of the slope do not overlap with zero then we can infer a significant p-value for the slope.

```r
library(ggplot2)
plotCI = ggplot(mtcars,aes(x=mtcars$wt,y=mtcars$mpg)) + geom_point() + geom_smooth(method="lm");plotCI
```



Multiple Regression

In a multiple regression there are multiple predictors (X1,X2,X3…Xn), in contrast to a simple regression where there's only one predictor.

$\hat{Y} = B_0 + B_1X_1 + B_2X_2 + ...B_nX_n$

$\hat{Y}$ = predicted value on the outcome variable Y

B_0 = predicted value on Y when all X = 0

X_k = predictor variables

B_k = unstandardised regression coefficients

$Y - \hat{Y}$ = residual (prediction error)

k = number of predictor variables

Model R and R^2

R = multiple correlation coefficient which is the correlation between the predicted scores and the observed scores.

R = $r_{\hat{y}y}$

R^2 is the percentage of variance in Y explained by the model

```
setwd("C:\\Users\\akane\\Desktop\\Science\\Teaching\\R-Course-UCC")
data<-read.csv("Salaries.csv",header=T,sep=",")
head(data)
```

```
##   X      rank discipline yrs.since.phd yrs.service  sex salary
## 1 1      Prof          B            19          18 Male 139750
## 2 2      Prof          B            20          16 Male 173200
## 3 3  AsstProf          B             4           3 Male  79750
## 4 4      Prof          B            45          39 Male 115000
## 5 5      Prof          B            40          41 Male 141500
## 6 6 AssocProf          B             6           6 Male  97000
```

```
# recode the nominal variable of sex into a new dummy variable where female gets a 1 and male gets a 0
data$sex <- factor(with(data,ifelse((sex == "Female"),1,0)))
head(data)
```

```
##   X      rank discipline yrs.since.phd yrs.service sex salary
## 1 1      Prof          B            19          18   0 139750
## 2 2      Prof          B            20          16   0 173200
## 3 3  AsstProf          B             4           3   0  79750
## 4 4      Prof          B            45          39   0 115000
## 5 5      Prof          B            40          41   0 141500
## 6 6 AssocProf          B             6           6   0  97000
```

```
# we can invent a new variable for the number of publications
set.seed(100)
pubs <- round(rnorm(data$yrs.since.phd, mean= 30,sd=8))
head(pubs)
```

```
## [1] 26 31 29 37 31 33
```

```
data["pubs"] <- pubs
summary(data)
```

```
##        X               rank     discipline yrs.since.phd    yrs.service
##  Min.   :  1   AssocProf: 64   A:181      Min.   : 1.00   Min.   : 0.00
##  1st Qu.:100   AsstProf : 67   B:216      1st Qu.:12.00   1st Qu.: 7.00
##  Median :199   Prof     :266              Median :21.00   Median :16.00
##  Mean   :199                              Mean   :22.31   Mean   :17.61
##  3rd Qu.:298                              3rd Qu.:32.00   3rd Qu.:27.00
##  Max.   :397                              Max.   :56.00   Max.   :60.00
##  sex         salary            pubs
##  0:358   Min.   : 57800   Min.   : 6.00
##  1: 39   1st Qu.: 91000   1st Qu.:26.00
##          Median :107300   Median :30.00
##          Mean   :113706   Mean   :29.94
##          3rd Qu.:134185   3rd Qu.:35.00
##          Max.   :231545   Max.   :56.00
```

```r
m1<-glm(data$salary~data$pubs+data$yrs.since.phd+data$sex)
summary(m1)
```

```
##
## Call:
## glm(formula = data$salary ~ data$pubs + data$yrs.since.phd +
##     data$sex)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -85928  -19482   -3170   15585  101192
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          88632.5     6005.3  14.759   <2e-16 ***
## data$pubs              150.4      177.2   0.849   0.3965
## data$yrs.since.phd     959.4      108.4   8.853   <2e-16 ***
## data$sex1            -8536.2     4741.0  -1.801   0.0725 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 755078149)
##
##     Null deviance: 3.6330e+11  on 396  degrees of freedom
## Residual deviance: 2.9675e+11  on 393  degrees of freedom
## AIC: 9248.2
##
## Number of Fisher Scoring iterations: 2
```

$\hat{y} = 88632.5 - 150.4(pubs) + 959.4(PhD) - 8536.2(sex)$

But, what do these values actually mean?

88632.5 is the intercept of the model and is the predicted salary for a male professor who has no publications and has just graduated from his PhD (predicted score when all X=0).

959.4 is the predicted change in salary associated with an increase in one year since your PhD, for professors who have an average number of publications and averaged across men and women. In a multiple regression you have to take into account the other variables. It's not simply the predicted change in salary for a unit change in time since PhD.

Who makes more money, men or women? This can't be answered based on the output. -8536.2 is the predicted difference between men and women in the amount of money made for an average number of years since PhD and an average number of years service. The other values on the other predictors hide this effect.

According to this model, is the gender difference statistically significant? No because p > 0.05.

Matrix Algebra for Parameter Estimation in Multiple Regression

A matrix is a rectangular table of known or unknown numbers. The size, or order of a matrix is given by identifying the number of rows and columns (Really Cool). The order of the following matrix is 4 x 2.

```r
M<-matrix(c(1,5,3,4,2,1,4,2),nrow=4,byrow = FALSE);M
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    5    1
## [3,]    3    4
## [4,]    4    2
```

```
dim(M)
```

```
## [1] 4 2
```

The transpose M^T of a matrix (M) is formed by rewriting its rows as columns.

```
M<-matrix(c(1,5,3,4,2,1,4,2),nrow=4,byrow = FALSE);M
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    5    1
## [3,]    3    4
## [4,]    4    2
```

```
Mt<-t(M);Mt
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    3    4
## [2,]    2    1    4    2
```

Two matrices may be added or subtracted only if they are of the same order.

```
M<-matrix(c(1,5,3,4,2,1,4,2),nrow=4,byrow = FALSE);M
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    5    1
## [3,]    3    4
## [4,]    4    2
```

```
N<-matrix(c(2,4,1,3,3,5,2,1),nrow=4,byrow = FALSE);N
```

```
##      [,1] [,2]
## [1,]    2    3
## [2,]    4    5
## [3,]    1    2
## [4,]    3    1
```

```
M+N
```

```
##      [,1] [,2]
## [1,]    3    5
## [2,]    9    6
## [3,]    4    6
## [4,]    7    3
```

```
M-N
```

```
##      [,1] [,2]
## [1,]   -1   -1
## [2,]    1   -4
## [3,]    2    2
## [4,]    1    1
```

Two matrices may be multiplied when the number of columns in the first matrix is equal to the number of rows in the second matrix. If so, then we say they are conformable for matrix multiplication.

In our case we have to take the transpose of matrix M to make it conformable.

$R = M^T * N$

$R_{ij} = \Sigma(M^T_{ik} * N_{kj})$

The result is a matrix that has the number of rows of the first matrix and the number of columns in the second matrix.

```
M<-matrix(c(1,5,3,4,2,1,4,2),nrow=4,byrow = FALSE);M
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    5    1
## [3,]    3    4
## [4,]    4    2
```

```
N<-matrix(c(2,4,1,3,3,5,2,1),nrow=4,byrow = FALSE);N
```

```
##      [,1] [,2]
## [1,]    2    3
## [2,]    4    5
## [3,]    1    2
## [4,]    3    1
```

```
Mt<-t(M)
Mt%*%N # 2 x 2 matrix results
```

```
##      [,1] [,2]
## [1,]   37   38
## [2,]   18   21
```

Example of matrix algebra

Let's move from a raw dataframe to a correlation matrix. Remember for regular algebra you move from deviation scores to mean squares.

```
#create vectors -- these will be our columns
a <- c(3,3,2,4,4,5,2,3,5,3)
b <- c(2,2,4,3,4,4,5,3,3,5)
c <- c(3,3,4,4,3,3,4,2,4,4)

#create matrix from vectors
M <- cbind(a,b,c)
k <- ncol(M) #number of variables
n <- nrow(M) #number of subjects

#create means for each column
M_mean <- matrix(data=1, nrow=n) %*% cbind(mean(a),mean(b),mean(c)); M_mean
```

```
##       [,1] [,2] [,3]
##  [1,]  3.4  3.5  3.4
##  [2,]  3.4  3.5  3.4
##  [3,]  3.4  3.5  3.4
##  [4,]  3.4  3.5  3.4
##  [5,]  3.4  3.5  3.4
##  [6,]  3.4  3.5  3.4
##  [7,]  3.4  3.5  3.4
##  [8,]  3.4  3.5  3.4
##  [9,]  3.4  3.5  3.4
## [10,]  3.4  3.5  3.4
```

```
#creates a difference matrix which gives deviation scores
D <- M - M_mean; D
```

```
##          a    b    c
##  [1,] -0.4 -1.5 -0.4
##  [2,] -0.4 -1.5 -0.4
##  [3,] -1.4  0.5  0.6
##  [4,]  0.6 -0.5  0.6
##  [5,]  0.6  0.5 -0.4
##  [6,]  1.6  0.5 -0.4
##  [7,] -1.4  1.5  0.6
##  [8,] -0.4 -0.5 -1.4
##  [9,]  1.6 -0.5  0.6
## [10,] -0.4  1.5  0.6
```

```
#creates the covariance matrix, the sum of squares are in the diagonal and the sum of cross products ar
C <-  t(D) %*% D; C
```

```
##      a    b    c
## a 10.4 -2.0 -0.6
## b -2.0 10.5  3.0
## c -0.6  3.0  4.4
```

```
#pulls out the standard deviations from the covariance matrix, remember an exponent of 1/2 is the same
S <- diag(diag(C)^(-1/2)); S
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.3100868 0.0000000 0.0000000
## [2,] 0.0000000 0.3086067 0.0000000
## [3,] 0.0000000 0.0000000 0.4767313
```

```
#constructs the correlation matrix. In the diagonals are ones because each variable correlates with its
round (S %*% C %*% S, 2)
```

```
##       [,1]  [,2]  [,3]
## [1,]  1.00 -0.19 -0.09
## [2,] -0.19  1.00  0.44
## [3,] -0.09  0.44  1.00
```

```
# check with normal correlation code
cor(a,a)
```

```
## [1] 1
```

```
cor(a,b)
```

```
## [1] -0.1913898
```

```
cor(a,c)
```

```
## [1] -0.08869686
```

```
cor(b,c)
```

```
## [1] 0.4413674
```

Estimation of coefficients for multiple regression

The values of the coefficients (B) are estimated such that the model yields optimal predictions. As with simple regression we try to minimise the residuals.

The sum of the squared (SS) residuals is the value that is minimised where SS.Residual = $\Sigma(\hat{Y} - Y)^2$

How do we do that when we have multiple predictors in the model? We have to solve for multiple regression coefficients.

Standardised (where the regression constant is 0 i.e. the predicted score on Y when X = 0 is itself 0, this means we can drop it from the equation) and in matrix form, the regression equation is:

$\hat{Y} = B(X)$ where $\hat{Y}$ is a [Nx1] vector, with N = to the number of cases. B is a [kx1] vector, with k = to the number of predictors and X is a [Nxk] matrix.

For $\hat{Y} = B(X)$ we have to solve for B. To do this we replace $\hat{Y}$ with Y and conform for matrix muliplication to get Y=X(B). Now, let's make X square and symmetric. To do this, pre-multiply both sides of the equation by the transpose of X, X^T. If we do that Y=X(B) becomes $X^T(Y) = X^T(XB)$. Now, to solve for B, eliminate X^TX. To do this, pre-multiply by the inverse $(X^TX)^-1$. So, $X^TY = X^T(XB)$ becomes $(X^TX)^-1(X^TY) - (X^TX)^-1(X^TXB)$. Note that $(X^TX)^-1(X^TX) = l$ which is the identity matrix. And that lB = B. Therefore, $(X^TX)^-1(X^TY) = B$

The solution is therefore, $B = (X^TX)^-1(X^TY)$. We know the X and Y values, these are our variables. Let's use this formula to calculate B's from the raw data matrix used in the previous segment.

```r
#create vectors -- these will be our columns
y <- c(3,3,2,4,4,5,2,3,5,3)
x1 <- c(2,2,4,3,4,4,5,3,3,5)
x2 <- c(3,3,4,4,3,3,4,2,4,4)

#create matrix from vectors
M <- cbind(y,x1,x2)
k <- ncol(M) #number of variables
n <- nrow(M) #number of subjects

#create means for each column
M_mean <- matrix(data=1, nrow=n) %*% cbind(mean(y),mean(x1),mean(x2)); M_mean
```

```
##       [,1] [,2] [,3]
##  [1,]  3.4  3.5  3.4
##  [2,]  3.4  3.5  3.4
##  [3,]  3.4  3.5  3.4
##  [4,]  3.4  3.5  3.4
##  [5,]  3.4  3.5  3.4
##  [6,]  3.4  3.5  3.4
##  [7,]  3.4  3.5  3.4
##  [8,]  3.4  3.5  3.4
##  [9,]  3.4  3.5  3.4
## [10,]  3.4  3.5  3.4
```

```r
#creates a difference matrix which gives deviation scores
D <- M - M_mean; D
```

```
##          y   x1   x2
##  [1,] -0.4 -1.5 -0.4
##  [2,] -0.4 -1.5 -0.4
##  [3,] -1.4  0.5  0.6
##  [4,]  0.6 -0.5  0.6
##  [5,]  0.6  0.5 -0.4
##  [6,]  1.6  0.5 -0.4
##  [7,] -1.4  1.5  0.6
##  [8,] -0.4 -0.5 -1.4
##  [9,]  1.6 -0.5  0.6
## [10,] -0.4  1.5  0.6
```

```
#creates the covariance matrix, the sum of squares are in the diagonal and the sum of cross products ar
C <-  t(D) %*% D; C
```

```
##       y   x1   x2
## y  10.4 -2.0 -0.6
## x1 -2.0 10.5  3.0
## x2 -0.6  3.0  4.4
```

Since we used deviation scores we can substitute S_{xx} for X^TX and substitute S_{xy} for X^TY. Therefore, $B = (S_{xx})^-1S_{xy}$

```
E<-matrix(c(10.5,3,3,4.4),nrow=2,ncol=2);E
```

```
##      [,1] [,2]
## [1,] 10.5  3.0
## [2,]  3.0  4.4
```

```
F<-matrix(c(-2,-.6),nrow=2,ncol=1);F
```

```
##      [,1]
## [1,] -2.0
## [2,] -0.6
```

```
solve(E)%*%F
```

```
##              [,1]
## [1,] -0.188172043
## [2,] -0.008064516
```

An alternative formulation can be done as follows

```
y <- c(3,3,2,4,4,5,2,3,5,3)
x1 <- c(2,2,4,3,4,4,5,3,3,5)
x2 <- c(3,3,4,4,3,3,4,2,4,4)
Y <- as.matrix(y);
X <- as.matrix(cbind(1,x1,x2));
beta = solve(t(X) %*% X) %*% (t(X) %*% Y) ; beta
```

```
##             [,1]
##      4.086021505
## x1 -0.188172043
## x2 -0.008064516
```

```
model <- lm(y~1+x1+x2) ; model$coefficients
```

```
##  (Intercept)           x1           x2
##  4.086021505 -0.188172043 -0.008064516
```

The General Linear Model

The mathematical framework used in many common statistical analyses, including multiple regression. ANOVA is typically presented as distinct from multiple regression but it IS a multiple regression.

Characteristics of GLM

Linear: pairs of variables are assumed to have linear relations

Additive: if one set of variables predict another variable, the effects are thought to be additive.

But! This does not preclude testing non-linear or non-additive effects. We can trick the GLM into testing for these.

So GLM can accommodate such tests, for example, by transformation of variables so non-linear becomes linear or by moderation analysis where we trick the GLM into testing non-additive effects.

GLM Example with non-additive effects

$Y = B_o + B_1X_1 + B_2X_2 + B_3X_3 + e$

Y=faculty salary

X1=years since PhD

X2=number of publications

X3=(years x publications)

Here we're looking a a non-additive effect between number of publications and years since PhD. We might expect that number of publications matters more for early stage researchers. Note that the regression equation is still linear and additive, the non-additive part is hidden away in X3.

GLM Example with categorical predictors (ANOVA)

One-way ANOVA $Y = B_0 + B_1X_1 + e$

Y= faculty salary

X1=gender

Factorial ANOVA (factorial because we have multiple predictors) $Y = B_o + B_1X_1 + B_2X_2 + B_3X_3 + e$

Y= faculty salary

X1= gender

X2= race

X3= interaction (gender x race)

Here we have an interaction between two categorical predictors.

ANOVA

Appropriate when the predictors (IVs) are all categorical and the outcome (DV) is continuous

Most common application is to analyse data from randomised experiments.

More specifically, randomised experiements that generate more than 2 means. If there are only 2 means then we would use a t-test.

Dummy Coding

A system to code categorical predictors in a regression analysis.

For example, if we have a categorical predictor variable with 4 levels and we use it to predict the dependent variable.

For instance, the area of research is a psychology department has 4 levels, cognitive, clinical, developmental and social. The number of publications is the dependent variable.

We can dummy code the 4 levels so that we work with numbers instead of the category names.

Area	D1	D2	D3
cognitive	0	0	0
clinical	1	0	0
developmental	0	1	0
social	0	0	1

We're using cognitive as the reference group i.e. it gets zeroes across the board. We have 3 dummy codes because We have 4 levels for the grouping variable, the independent variable. It's the number of groups - 1. The regression constant i.e. the predicted score when all the Xs are zero, now has a clear meaning, it is the predicted score for the cognitive group.

Regression model

$\hat{Y} = B_0 + B_1(D1) + B_2(D2) + B_3(D3)$

```
setwd("C:\\Users\\akane\\Desktop\\Science\\Teaching\\R-Course-UCC")
hsb2 <- read.csv("hsb2.csv",header = T,sep = ",")
head(hsb2)
```

```
##    id female race ses schtyp prog read write math science socst
## 1  70      0    4   1      1    1   57    52   41      47    57
## 2 121      1    4   2      1    3   68    59   53      63    61
## 3  86      0    4   3      1    1   44    33   54      58    31
## 4 141      0    4   3      1    3   63    44   47      53    56
## 5 172      0    4   2      1    2   47    52   57      53    61
## 6 113      0    4   2      1    2   44    52   51      63    61
```

```
names(hsb2)[names(hsb2)=="race"] <- "field"
names(hsb2)[names(hsb2)=="write"] <- "publications"
head(hsb2)
```

```
##    id female field ses schtyp prog read publications math science socst
## 1  70      0     4   1      1    1   57           52   41      47    57
## 2 121      1     4   2      1    3   68            59   53      63    61
## 3  86      0     4   3      1    1   44            33   54      58    31
## 4 141      0     4   3      1    3   63            44   47      53    56
## 5 172      0     4   2      1    2   47            52   57      53    61
## 6 113      0     4   2      1    2   44            52   51      63    61
```

```
# creating the factor variable
hsb2$field.f <- factor(hsb2$field)
is.factor(hsb2$field.f)
```

```
## [1] TRUE
```

```
summary(lm(publications ~ field.f, data = hsb2))
```

```
##
## Call:
## lm(formula = publications ~ field.f, data = hsb2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.0552  -5.4583   0.9724   7.0000  18.8000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   46.458      1.842  25.218  < 2e-16 ***
## field.f2      11.542      3.286   3.512 0.000552 ***
## field.f3       1.742      2.732   0.637 0.524613
## field.f4       7.597      1.989   3.820 0.000179 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.025 on 196 degrees of freedom
## Multiple R-squared:  0.1071, Adjusted R-squared:  0.0934
## F-statistic: 7.833 on 3 and 196 DF,  p-value: 5.785e-05
```

```
mean(hsb2$publications[hsb2$field==1])
```

```
## [1] 46.45833
```

```
mean(hsb2$publications[hsb2$field==2])
```

```
## [1] 58
```

```
mean(hsb2$publications[hsb2$field==3])
```

```
## [1] 48.2
```

```
mean(hsb2$publications[hsb2$field==4])
```

```
## [1] 54.05517
```

The 46.458 value is the mean number of publications for our reference level i.e. that for cognition. The 11.542 represents the change in Y when we have a 1 unit increase in X. But think about what this means. A 1 unit increase in X moves us from cognitive to clinical. So this means that clinical has on average 11.542 more publications than cognitive. If we refer to the mean values this makes sense for cognitive has a mean of 46.458 and clinical has a value of 58. The developmental level differs from congitive by 1.742 following the same logic this makes sense.

We can refer to the p-values to test whether these differences are significant. For instance, the difference between cognitive and developmental is not significant.

If we wanted to look at the pairwise differences between the other levels we'd have to recode using a different level than cognitive as the reference group.

We can change the coding so that the intercept in the model is not the predicted score for cognitive but rather the predicted score for all professors across all groups. That kind of coding is called effects coding. Here is an example of unweighted effects coding:

Area	C1	C2	C3
cognitive	-1	-1	-1
clinical	1	0	0
developmental	0	1	0
social	0	0	1

The predicted score on Y when X = 0 now represents the average across all these groups.

```
tapply(hsb2$publication, hsb2$field, mean)
```

```
##        1        2        3        4
## 46.45833 58.00000 48.20000 54.05517
```

```
mean(tapply(hsb2$publication, hsb2$field, mean))
```

```
## [1] 51.67838
```

```
lm(publications ~ field.f, data = hsb2, contrasts = list(b = contr.sum))
```

```
## Warning in model.matrix.default(mt, mf, contrasts): variable 'b' is absent,
## its contrast will be ignored

##
## Call:
## lm(formula = publications ~ field.f, data = hsb2, contrasts = list(b = contr.sum))
##
## Coefficients:
## (Intercept)     field.f2     field.f3     field.f4
##      46.458       11.542        1.742        7.597
```

The intercept in unweighted effects coding won't exactly match the mean for the whole group because it doesn't take into account the different sizes for the levels within the groups. Here we have a different number of professors in each group. The coefficients now tell us the difference between the overall mean and each of the levels. Again, if we wanted to see the difference between cognitive and the overall mean we'd have to recode using some other level as the reference group.

```
length(hsb2$publications[hsb2$field==1])
```

```
## [1] 24
```

```
length(hsb2$publications[hsb2$field==2])
```

```
## [1] 11
```

We can use weighted effects coding to get the regression constant to be exactly equal to the grand mean of all of the data. And as it suggests we just weight the codings by the number of professors in each group.

Area	C1	C2	C3
cognitive	-Nclin/Ncog	-Ndev/Ncog	-Nsoc/Ncog

Area	C1	C2	C3
clinical	Nclin/Ncog	0	0
developmental	0	Ndev/Ncog	0
social	0	0	Nsoc/Ncog

Moderation analysis

A moderator has influence over the other effects/ relationships. It is synonymous with an interaction.

The way to think about what an interaction is, is that if you were to explain your findings to someone you would use the word 'depends'. Lets say someone asks you, "if people research a product, do they purchase it?" You might respond, "Well, it depends. For men, if they research a product, they typically end up buying one, but women enjoy looking at and thinking about products for its own sake; often, a woman will research a product, but have no intention of buying it. So, the relationship between researching a product and buying that product depends on sex." In this story, there is an interaction between product research and sex, or sex moderates the relationship between research and purchasing.

Example

Stereotype threat is a situational predicament in which people are or feel themselves to be at risk of conforming to stereotypes about their social group. For instance, women will perform worse in a test if they are told their group tends to perform worse beforehand.

X = experimental manipulation that is stereotype threat.

Y= Behavioural outcome in the form of an IQ test.

Z= Moderator, in the form of working memory capacity (WMC).

That is to say if you have a strong WMC you will be able to buffer the effect of the stereotype threat such that you're not as affected by it.

In general, a moderator variable (Z) will enhance a regression model if the relationship between X and Y varies as a function of Z. So in our example, maybe the buffering effect of WMC works for people on the lower end of the WMC but not for people on the higher end.

Moderation can be used in experimental research. The manipulation of an IV (X) causes change in a DV (Y). A moderator variable (Z) implies that the effect of the IV on the DV (X on Y) is NOT consistent across the distribtuion of Z.

It can also be used in correlational research where we assume a correlation between X and Y. A moderator variable (Z) implies that the correlation between X and Y is NOT consistent across the distribution of Z.

There's always the possiblity that there's a moderator varialbe that the researchers didn't test. So if X and Y are correlated, then we can use regression to predict Y from x:

$Y = B_0 + B_1X + e$

If there is a moderator, Z, then B_1 will NOT be representative across all Z. That is the relationship between X and Y is different at different levels of Z.

Moderation Model

If both X and Z are continuous:

$Y = B_0 + B_1X + B_2Z + B_3(X * Z) + e$

The GLM is being tricked into testing for a non-additive effect when it analyses B_3.

If X is categorical (3 levels in this case) and Z is continuous:

$Y = B_0 + B_1(D1) + B_2(D2) + B_3Z + B_4(D1 * Z) + B_5(D2 * Z) + e$

Here we will need to use dummy coding. We have 3 levels which means we need to use N-1 or 2 codes, D1 and D2. Then to represent moderation we will need two product components ($B_4(D1 * Z) + B_5(D2 * Z)$).

How to test for moderation

If both X and Z are continuous:

Model 1: No moderation

$Y = B_0 + B_1X + B_2Z + e$

Model 2: Moderation

$Y = B_0 + B_1X + B_2Z + B_3(X * Z) + e$

Then compare both models and see if model 2 is a better fit. If it is then you have a moderator effect. You can also look at the significance of the B_3 coefficient using the p-value.

If X is categorical and Z is continuous:

Model 1: No moderation

$Y = B_0 + B_1(D1) + B_2(D2) + B_3Z + e$

Model 2: Moderation

$Y = B_0 + B_1(D1) + B_2(D2) + B_3Z + B_4(D1 * Z) + B_5(D2 * Z) + e$

Again compare both models and see if model 2 is a better fit. If it is then you have a moderator effect. But here we can't look at the one coefficient for significance because there are two elements that carry the potential moderator effect, thus we have to use model selection.

We can compare Model and Model 2 in terms of overall variance explained, that is, R^2. And NHST is available for this comparison which allows us to make that statement that there is a significantly different change when we add the moderator (or not).

As we noted earlier, we can evaluate the B values for predictors associated with the moderation effect.

$(X * Z)$ for the continuous case

$(D1 * Z)$ and $(D2 * Z)$ for the categorical case

Back to the example

X = experimental manipulation that is stereotype threat.

Y= Behavioural outcome in the form of an IQ test.

Z= Moderator, in the form of working memory capacity (WMC).

Students completed a working memory task. Students were then assigned at random to one of 3 experimental conditions.

Explicit threat (n=50)

Implicit threat (n=50)

Control (n=50)

Students then completed an IQ test.

Since the experimental condition is categorical dummy coding is required.

Let the control group be the referent

Let D1 be the Explicit threat

Let D2 be the Implicit threat

```
setwd("C:\\Users\\akane\\Desktop\\Science\\Teaching\\R-Course-UCC")
data<-read.table("lab7.txt")
describeBy(data,data$condition)
```

```
## $control
##             vars  n   mean    sd median trimmed   mad    min    max range
## subject        1 50  25.50 14.58  25.50   25.50 18.53   1.00  50.00    49
## condition*     2 50   1.00  0.00   1.00    1.00  0.00   1.00   1.00     0
## IQ             3 50  97.88 20.93  99.50   97.47 25.20  46.00 141.00    95
## WM             4 50 102.18 18.79  99.50  100.55 20.02  71.00 159.00    88
## WM.centered    5 50   3.10 18.79   0.42    1.47 20.02 -28.08  59.92    88
## D1             6 50   0.00  0.00   0.00    0.00  0.00   0.00   0.00     0
## D2             7 50   0.00  0.00   0.00    0.00  0.00   0.00   0.00     0
##             skew kurtosis   se
## subject     0.00    -1.27 2.06
## condition*   NaN      NaN 0.00
## IQ          0.04    -0.50 2.96
## WM          0.73     0.41 2.66
## WM.centered 0.73     0.41 2.66
## D1           NaN      NaN 0.00
## D2           NaN      NaN 0.00
## 
## $threat1
##             vars  n   mean    sd median trimmed   mad    min    max range
## subject        1 50  75.50 14.58  75.50   75.50 18.53  51.00 100.00    49
## condition*     2 50   2.00  0.00   2.00    2.00  0.00   2.00   2.00     0
## IQ             3 50  52.16 13.79  52.00   51.75 11.86  21.00  83.00    62
## WM             4 50 100.80 16.85  97.50   99.90 16.31  72.00 138.00    66
## WM.centered    5 50   1.72 16.85  -1.58    0.82 16.31 -27.08  38.92    66
## D1             6 50   1.00  0.00   1.00    1.00  0.00   1.00   1.00     0
## D2             7 50   0.00  0.00   0.00    0.00  0.00   0.00   0.00     0
##             skew kurtosis   se
## subject     0.00    -1.27 2.06
## condition*   NaN      NaN 0.00
## IQ          0.16    -0.25 1.95
## WM          0.48    -0.56 2.38
## WM.centered 0.48    -0.56 2.38
## D1           NaN      NaN 0.00
## D2           NaN      NaN 0.00
## 
## $threat2
##             vars  n   mean    sd median trimmed   mad    min    max range
## subject        1 50 125.50 14.58 125.50  125.50 18.53 101.00 150.00    49
## condition*     2 50   3.00  0.00   3.00    3.00  0.00   3.00   3.00     0
## IQ             3 50  48.02 12.45  47.00   47.40 14.83  28.00  79.00    51
## WM             4 50  94.26 18.77  92.00   93.97 17.79  55.00 133.00    78
## WM.centered    5 50  -4.82 18.77  -7.08   -5.10 17.79 -44.08  33.92    78
## D1             6 50   0.00  0.00   0.00    0.00  0.00   0.00   0.00     0
## D2             7 50   1.00  0.00   1.00    1.00  0.00   1.00   1.00     0
##              skew kurtosis   se
## subject     0.00    -1.27 2.06
## condition*   NaN      NaN 0.00
## IQ          0.38    -0.56 1.76
## WM          0.16    -0.71 2.65
## WM.centered 0.16    -0.71 2.65
## D1           NaN      NaN 0.00
## D2           NaN      NaN 0.00
##
## attr(,"call")
## by.data.frame(data = x, INDICES = group, FUN = describe, type = type)
```

```
# look at the row variable IQ to see if there has been an effect of the threat on IQ.
#We can see there is.

# we can look at the correlation of IQ on working memory by condition
library(plyr)
ddply(data, .(condition), summarise, "corr" = cor(IQ, WM, method = "spearman"))
```

```
##   condition       corr
## 1   control 0.07403312
## 2   threat1 0.74498651
## 3   threat2 0.61362271
```

```
#this indicates that the condition is having a strong effect
```

Here is the result of applying model 1 without moderation terms

```
model1<-lm(IQ~WM+D1+D2,data=data)
summary(model1)
```

```
##
## Call:
## lm(formula = IQ ~ WM + D1 + D2, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.339  -7.294   0.744   7.608  42.424
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  59.78635    7.14360   8.369 4.30e-14 ***
## WM            0.37281    0.06688   5.575 1.16e-07 ***
## D1          -45.20552    2.94638 -15.343  < 2e-16 ***
## D2          -46.90735    2.99218 -15.677  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.72 on 146 degrees of freedom
## Multiple R-squared:  0.7246, Adjusted R-squared:  0.719
## F-statistic: 128.1 on 3 and 146 DF,  p-value: < 2.2e-16
```

59.78635 is the predicted score on Y when all Xs are 0. That is the predicted IQ for someone who had a 0 on the working memory task and is in the control condition.

The coefficients for D1 and D2 show that both threat conditions are having a strong negative effect.

Here is the result of applying model 2 with the moderation terms

```r
model2<-lm(IQ~WM+D1+D2+(WM*D1)+(WM*D2),data=data)
summary(model2)
```

```
##
## Call:
## lm(formula = IQ ~ WM + D1 + D2 + (WM * D1) + (WM * D2), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.414  -7.181   0.420   8.196  40.864
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  85.5851    11.3576   7.535 4.95e-12 ***
## WM            0.1203     0.1094   1.100  0.27303
## D1          -93.0952    16.8573  -5.523 1.52e-07 ***
## D2          -79.8970    15.4772  -5.162 7.96e-07 ***
## WM:D1         0.4716     0.1638   2.880  0.00459 **
## WM:D2         0.3288     0.1547   2.125  0.03529 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.38 on 144 degrees of freedom
## Multiple R-squared:  0.7409, Adjusted R-squared:  0.7319
## F-statistic: 82.35 on 5 and 144 DF,  p-value: < 2.2e-16
```

0.1203 is the coefficent for working memory. That's the slope relating working memory to IQ for the control condition.

0.4716 is the difference in slope between the control condition and the threat 1 condition. Before, for our dummy code example, this unit change in X showed the predicted difference in means. Now because we have moderation worked in, it is the predicted difference in slopes. And that is we want to test. Indeed, there are significant moderation effects (p = 0.00459 for $WM * D1$ and p = 0.03529 for $WM * D2$).

We can also perform a model comparison to test for the significance of the moderation terms in the model 2 verus model 1.

```r
anova(model1,model2)
```

```
## Analysis of Variance Table
##
## Model 1: IQ ~ WM + D1 + D2
## Model 2: IQ ~ WM + D1 + D2 + (WM * D1) + (WM * D2)
##   Res.Df   RSS Df Sum of Sq      F  Pr(>F)
## 1    146 31655
## 2    144 29784  2    1871.3 4.5238 0.01243 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value here shows that there is a signifcant difference in the amount of variation explained by the moderation effect.

By plotting the slopes of these moderation effects we can see if there is evidence of the effects. If the slopes are not parallel we have evidence of moderation because it shows that the effect (i.e. WM) is not consistent across groups.

The three lines are similar at high values of WM which is what we initially predicted. High values of WM are meant to buffer the effect of the threat conditions.

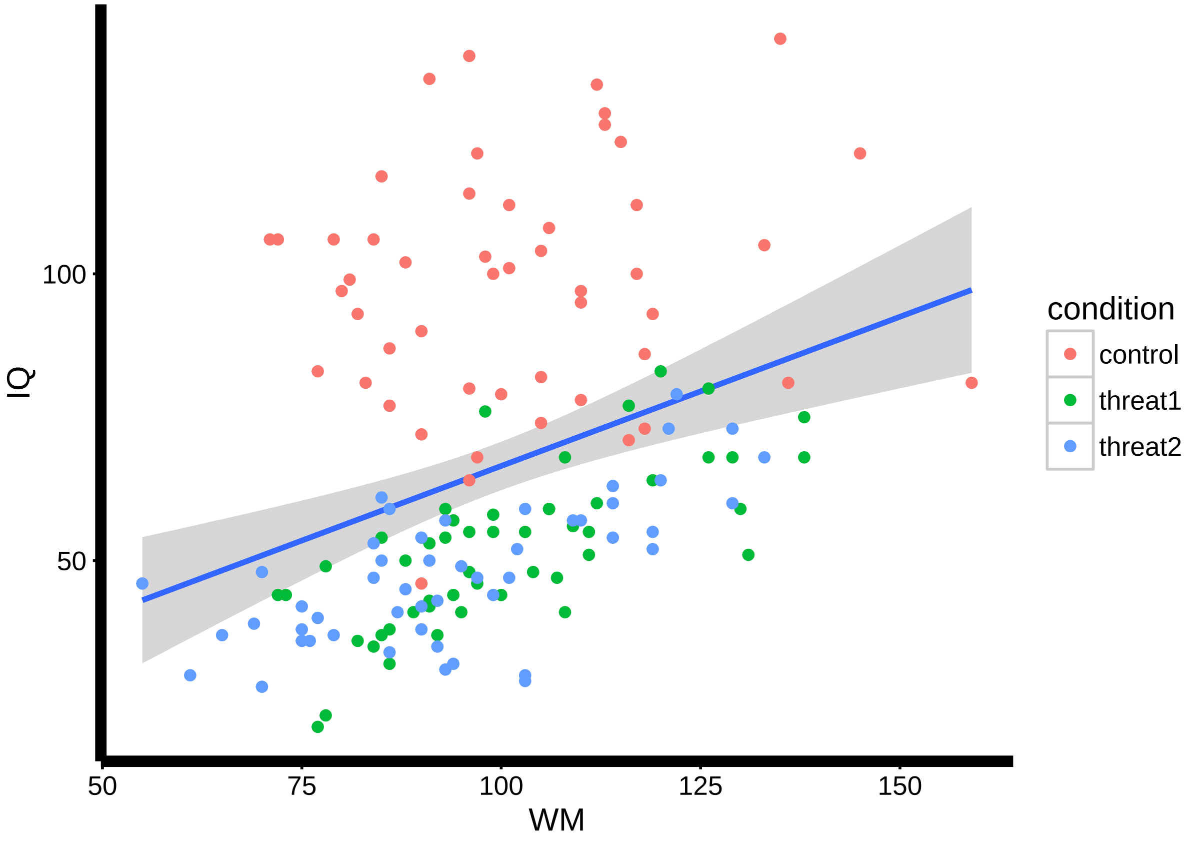
```
color <- c("red","green","blue")
plot1<-ggplot(data, aes(x = WM, y = IQ)) + stat_smooth(method="lm", se=F) +
  geom_point(aes(color=condition))

plot1+theme_bw() +
  theme(plot.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank() )+
  theme(panel.border= element_blank())+
  theme(axis.line.x = element_line(color="black", size = 2),
       axis.line.y = element_line(color="black", size = 2))
```

```
## Warning in if (se) "confidence" else "none": the condition has length > 1
## and only the first element will be used
```

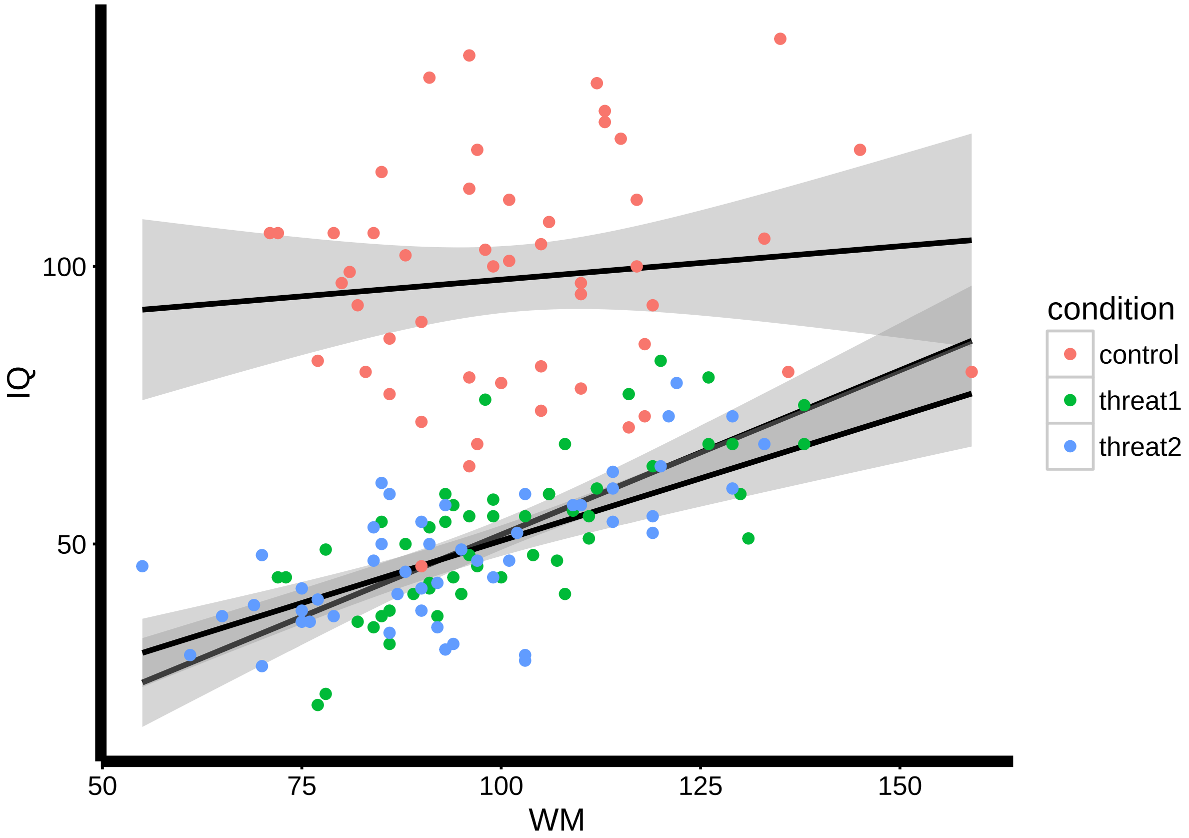
```
## Warning in if (se.fit) list(fit = predictor, se.fit = se, df = df,
## residual.scale = sqrt(res.var)) else predictor: the condition has length >
## 1 and only the first element will be used
```

```
## Warning in if (se) {: the condition has length > 1 and only the first
## element will be used
```



```
plot2<-ggplot(data, aes(x = WM, y = IQ)) +
  geom_smooth(aes(group=condition), method="lm", se=T, color="black", fullrange=T) +
  geom_point(aes(color=condition))


plot2+theme_bw() +
  theme(plot.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank() )+
  theme(panel.border= element_blank())+
  theme(axis.line.x = element_line(color="black", size = 2),
        axis.line.y = element_line(color="black", size = 2))
```



In sum, all that is implied by "interaction" (or "moderator") is that the way one predictor relates to the outcome depends on the level of the other predictor.

Moderation Centering Predictors

How to centre? You simply create deviation scores $X_c = X - M$ After centering the mean of the predictors will be 0.

Conceptual reason for centering

There are conceptual reasons for doing so. To take one example:

Y=child's verbal ability

X=mother's vocabulary

Z=child's age

In this case the intercept B_0 is the predicted score on Y when all predictors (X,Z) are zero

If X = 0 or z = 0 is meaningless, or impossible (as would be the case here for a mother's vocab or child's age being 0) then B_0 will be difficult to interpret

In contrast, if X = 0 and Z = 0 are the average then B_0 is easy to interpret. We'll then have our intercept to mean the predicted score on Y for a child with an average age who has a mother with an average vocab.

The regression coefficient B_1 is the slope for X assuming an average score on Z.

No moderation effect implies that B_1 is consistent across the entire distribution of Z.

In contrast, a moderation effect implies that B_1 is NOT consistent across the entire distributin of Z.

Where in the distribution of Z is B_1 most representative of the relationship between X and Y?

For an additive regression model if I centre my predictors, the regression constant will change but the slopes will not.

For a regression model with a moderation effect, the regression constant will change as before when centering. But now, so too will the individual slopes of X and Z. However, the higher order term, the one that denotes the interaction, will NOT change. The point is that the lower order slopes change because of the interaction, so we should focus on the interaction term itself.

Centering can also give us the best estimate for the slope if we did not include the moderation effect. That's because we're getting the average effect by centering.

Statistical reason for centering

The predictors, X and Z, can become highly correlated with the product (X*Z). This is an instance of multicolinearity: when two predictor variables in a GLM are so highly correlated that they are essentially redundant and it becomes difficult to estimate B values associated with each predictor. It's very difficult for the regression to pick out the unique variance explained by each of the predictors in Y when it comes to the computation of the matrix algebra.

Moderation Example 2

Example as before:

X = experimental manipulation of stereotype threat

Y= behavioural outcome of IQ score

Z=moderator of working memory capacity (WM)

Now we have to centre our continuous predictor. So if X is categorical and Z is continuous we compare 2 models:

Model 1 with no moderation

$Y = B_0 + B_1(D1) + B_2(D2) + B_3 Z.centre + e$

Model 2 with moderation

$Y = B_0 + B_1(D1) + B_2(D2) + B_3 Z.centre + B_4(D1 * Z.centre) + B_5(D2 * Z.centre) + e$

In our data the WMS is now in deviation form.

The setup is that students took a working memory task and were assigned at random to one of 3 experimental conditions. The students then performed an IQ test.

Here's model 1 uncentred and centred:

```
setwd("C:\\Users\\akane\\Desktop\\Science\\Teaching\\R-Course-UCC")
dir()
```

```
##  [1] "centre.txt"                "hsb2.csv"
##  [3] "lab7.txt"                  "mediation.txt"
##  [5] "NHST.jpg"                  "R course.Rmd"
##  [7] "Salaries.csv"              "Statistics Teaching.Rmd"
##  [9] "Statistics_Teaching.html"  "Statistics_Teaching.pdf"
## [11] "Statistics_Teaching.Rmd"   "Statistics_Teaching_files"
```

```
data<-read.table("centre.txt")
head(data)
```

```
##   subject condition  IQ  WM WM.centered D1 D2
## 1       1   control 134  91       -8.08  0  0
## 2       2   control 121 145       45.92  0  0
## 3       3   control  86 118       18.92  0  0
## 4       4   control  74 105        5.92  0  0
## 5       5   control  80  96       -3.08  0  0
## 6       6   control 105 133       33.92  0  0
```

```
model1<-lm(IQ~WM+D1+D2,data=data)
summary(model1)
```

```
##
## Call:
## lm(formula = IQ ~ WM + D1 + D2, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.339  -7.294   0.744   7.608  42.424
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  59.78635    7.14360   8.369 4.30e-14 ***
## WM            0.37281    0.06688   5.575 1.16e-07 ***
## D1          -45.20552    2.94638 -15.343  < 2e-16 ***
## D2          -46.90735    2.99218 -15.677  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.72 on 146 degrees of freedom
## Multiple R-squared:  0.7246, Adjusted R-squared:  0.719
## F-statistic: 128.1 on 3 and 146 DF,  p-value: < 2.2e-16
```

```
model1.centred<-lm(IQ~WM.centered+D1+D2,data=data)
summary(model1.centred)
```

```
##
## Call:
## lm(formula = IQ ~ WM.centered + D1 + D2, data = data)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max 
## -47.339  -7.294   0.744   7.608  42.424 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  96.72429    2.09267  46.220  < 2e-16 ***
## WM.centered   0.37281    0.06688   5.575 1.16e-07 ***
## D1          -45.20552    2.94638 -15.343  < 2e-16 ***
## D2          -46.90735    2.99218 -15.677  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 14.72 on 146 degrees of freedom
## Multiple R-squared:  0.7246, Adjusted R-squared:  0.719 
## F-statistic: 128.1 on 3 and 146 DF,  p-value: < 2.2e-16
```

Compared to before the only value that has changed is the intercept because now we have centred the data. Now the intercept is the mean of the IQ scores.

Here's model 2 uncentred and centred:

```
model2<-lm(IQ~WM+D1+D2+(WM*D1)+(WM*D2),data=data)
summary(model2)
```

```
## 
## Call:
## lm(formula = IQ ~ WM + D1 + D2 + (WM * D1) + (WM * D2), data = data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -50.414  -7.181   0.420   8.196  40.864 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  85.5851    11.3576   7.535 4.95e-12 ***
## WM            0.1203     0.1094   1.100  0.27303    
## D1          -93.0952    16.8573  -5.523 1.52e-07 ***
## D2          -79.8970    15.4772  -5.162 7.96e-07 ***
## WM:D1         0.4716     0.1638   2.880  0.00459 ** 
## WM:D2         0.3288     0.1547   2.125  0.03529 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 14.38 on 144 degrees of freedom
## Multiple R-squared:  0.7409, Adjusted R-squared:  0.7319 
## F-statistic: 82.35 on 5 and 144 DF,  p-value: < 2.2e-16
```

```
model2.centred<-lm(IQ~WM.centered+D1+D2+(WM.centered*D1)+(WM.centered*D2),data=data)
summary(model2.centred)
```

```
## 
## Call:
## lm(formula = IQ ~ WM.centered + D1 + D2 + (WM.centered * D1) + 
##     (WM.centered * D2), data = data)
## 
## Residuals:
```

```
##      Min       1Q  Median       3Q      Max
## -50.414  -7.181   0.420   8.196  40.864
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      97.5070     2.0619  47.289  < 2e-16 ***
## WM.centered       0.1203     0.1094   1.100  0.27303
## D1              -46.3652     2.9038 -15.967  < 2e-16 ***
## D2              -47.3223     2.9439 -16.075  < 2e-16 ***
## WM.centered:D1    0.4716     0.1638   2.880  0.00459 **
## WM.centered:D2    0.3288     0.1547   2.125  0.03529 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.38 on 144 degrees of freedom
## Multiple R-squared:  0.7409, Adjusted R-squared:  0.7319
## F-statistic: 82.35 on 5 and 144 DF,  p-value: < 2.2e-16
```

Now the intercept will change as well as anything below the highest order coefficients.

We can now compared the 2 centred models to see which is the best fit. ALl of the numbers will be the same because all we did was change the scale.

```
anova(model1.centred,model2.centred)
```

```
## Analysis of Variance Table
##
## Model 1: IQ ~ WM.centered + D1 + D2
## Model 2: IQ ~ WM.centered + D1 + D2 + (WM.centered * D1) + (WM.centered *
##     D2)
##   Res.Df   RSS Df Sum of Sq      F  Pr(>F)
## 1    146 31655
## 2    144 29784  2    1871.3 4.5238 0.01243 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Mediation

We'll use our previous example:

X = experimental manipulation of stereotype threat

Y= behavioural outcome which is the IQ score

M=mediator(mechanism) which is working memory capactiy.

In this case, we're asking why does stereotype threat manipulate IQ, perhaps it's working memory capacity?

A mediation analysis is typically conducted to better understand an observed effect of an IV on a DV or a correlation between X and Y.

In our example, why and how, does stereotype threat influence IQ test performance?

Mediation helps get around the problem of 'correlation does not equal causation" because it allows us to infer a mechanism.

If X and Y are correlated BECAUSE of the mediator M, then (X->M->Y):

$Y = B_0 + B_1 M + e$

&

$M = B_0 + B_1X + e$

X should predict M and M should predict Y.

If X and Y are correlated BECAUSE of the mediator M, and:

$Y = B_0 + B_1M + B_2X + e$

What will happen to the predictive value of X? In other words will B_2 be significant? If when we add in the mediator the main effect goes away (e.g. the effect of stereotype threat on IQ) then we have evidence for a mediating effect.

A mediator variable (M) accounts for some or all of the relationship between X and Y. When it's some it's a partial mediation (here the regression coefficient will drop but may still be significant), when it's all it's full mediation (here the regression coefficient will drop to 0).

We still have to remember that correlation does not imply causation. In other words, there is a BIG difference between statistical mediation and true causal mediation.

How to test for mediation

Run 3 regression models

lm(Y~X)

Regression coefficient for X should be signifcant

lm(M~X)

Regression coefficient for X should be signifcant

lm(Y~X+M)

Regression coefficient for M should be signifcant. Here's where we look to see what the effect is on the regression coefficient for X.

To try this on our example it will look like this:

Students are randomly assigned to 1 of the 2 experimental conditions: threat or control.

They complete a working memory task and then complete an IQ test. The difference to before is that they complete the working memory test after the threat condition. It is assumed that the threat will affect both IQ and working memory.

```
library(multilevel)
setwd("C:\\Users\\akane\\Desktop\\Science\\Teaching\\R-Course-UCC")
MED<-read.table("mediation.txt")
head(MED)
```

```
##   subject condition  IQ WM
## 1       1   control  73 37
## 2       2   control 128 77
## 3       3   control  83 32
## 4       4   control  83 33
## 5       5   control  64 53
## 6       6   control  95 46
```

```
model.ALL <- sobel(MED$condition, MED$WM, MED$IQ)
model.ALL
```

```
## $`Mod1: Y~X`
##             Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)    97.32   2.070678 46.999106 4.965625e-69
## predthreat    -11.00   2.928380 -3.756342 2.927777e-04
##
## $`Mod2: Y~X+M`
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 55.997696 4.64403367 12.057987 5.303590e-21
## predthreat  -2.407489 2.31641713 -1.039316 3.012416e-01
## med          0.752409 0.07999527  9.405669 2.576515e-15
##
## $`Mod3: M~X`
##             Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)    54.92   1.900708 28.894501 9.487423e-50
## predthreat    -11.42   2.688007 -4.248501 4.906391e-05
##
## $Indirect.Effect
## [1] -8.592511
##
## $SE
## [1] 2.219233
##
## $z.value
## [1] -3.871839
##
## $N
## [1] 100
```

The predicted score on IQ in control conditions is 97.32. The slope for the Y~X is -11 because control is 0 and threat is 1. Therefore as we move from control to threat we get an -11 drop in IQ.

For working memory as a funcion of threat threat there is a similar drop becuase the slope is -11.42.

In the model where we have Y~X+M the effect of threat on IQ is no longer significant (0.3012). That is because we added working memory into the model.

The interpreation of this analysis is that there is full mediation. The direct effect is no longer significant after adding the mediator into the regression equation. The Sobel test is also significant. Is the effect of the path from X through M to Y where the null is that this is 0. Similiar to other tests, it's a ratio.

Mediation through Structural Equation Modelling path analysis

Mediation analyses are typically illustrated using "path models"

Path models are a form of structural equation modeling that have their own sort of notation.

Rectangles: observed variables (X,Y,M)

Circles: unobserved variables (e)

Triangles: constants

Arrows: associations

T-tests

Let's assume a simple experimental design An independent variable with either vaccine or placebo. And a dependent variable which is the rate of polio.

This will give us two means that can be compared using a t-test. NHST can be conducted, yielding a p-value. Effect size can also be calculated. And confidence intervals around the sample means can also be reported.

$z = (observed - expected)/SE$

$t = (observed - expected)/SE$

Where SE is standard error

When to use z and t?

A z-test is used when comparing a sample mean to a population mean when the standard deviation of the population is known.

A single sample t-test is used when comparing a sample mean to a population mean when the standard deviation of the population is unknown. A dependent t-test is used when evaluating the difference between two related samples (e.g. when the same people are measured twice).

An independent t-test is used when evaluating the difference between two independent samples.

test	observed	expected	SE
z	sample mean	population mean	SE of the mean
t (single sample)	sample mean	population mean	SE of the mean
t (dependent)	sample mean of difference scores	population mean of difference scores	SE of the difference
t (independent)	difference between two sample means	difference between two population means	SE of the difference between means

For dependent we get difference scores at the level of the individual and for the independent we get difference scores at the level of the group.

p-values for z and t

Exact p-value depends on:

Directional or non-directional test

Degrees of freedom (df): different t-distributions for different sample sizes

degrees of freedom for z and t tests

test	df
z	NA
t (single sample)	N-1
t (dependent)	N-1
t (independent)	(N1-1)+(N2-1)

Dependent t-tests

Also known as paried samples t-test. This is appropriate when the same subjects are being compared e.g. pre/post design. Or when two samples are matched a the level of individual subjects, allowing for a difference score to be calculated.

A thorough analysis will include a t-value, a p-value, Cohen's d (effect size which is M/SD) and a confidence interval (interval estimate)

The t-value: t= (observed - expected)/SE t=(M-0)/SE=M/SE

Calculate a difference score for each individual and then average them. The expected value under NHST is 0. So it comes down to what did you observe relative to what you would expect due to chance.

The effect size here, Cohen's d, is calculated as the mean of the difference scores divided by the standard deviation of the difference scores. We divide by SD instead of SE because SE is biased by N (recall that SE=SD/SQRT(N)). A Cohen's d of 1 means that you want up by a standard deviation.

Confidence interval:

upper bound = M+t(SE); lower bound = M-t(SE)

t-value depends on the level of confidence and t-distribution.

Let's use the example of the wine ratings to see if there was a difference in ratings for white vs red. We'll use the Australian data because it was the only country that provided a normal distribution for both red and white which is an assumption of the t-test.

```r
library(lsr)
x<-rnorm(100,80,1)
y<-rnorm(100,82,1)
t.test(x,y, paired =T)
```

```
##
##  Paired t-test
##
## data:  x and y
## t = -16.363, df = 99, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.467454 -1.933749
## sample estimates:
## mean of the differences
##               -2.200602
```

```r
cohensD(x,y,method="paired")
```

```
## [1] 1.636286
```

Negative value for t is because of the order. It's arbitrary to the order we chose our x and y.

The 95% confidence intervals are around the mean difference. What's important about the interval estimate is that it does not cover 0 so it should be signifcant.

Independent t-tests

Compares two independent samples e.g. males and females, control and experimental etc.

Let's use the working memory training example with more detail. Here four independent groups trained for different amounts of time (8,12,17 or 19 days)

t-value t=(Observed - Expected)/SE

t=(M1-M2)/SE

SE=(SE1+SE2)/2

Cohen's d

$d = (M1 - M2)/SD_pooled$

$SD_pooled = (SD1 + SD2)/2$

We need to pool the SDs of the two groups to get their average variance in the case of independent t.

Extra step for indepenent t

Homogeneity of variance is assumed. The pooled SD is appropriate only if the variances in the two groups are equivalent.

If not, then the homogeneity of variance assumption is violated. Simulations indicates this results in an increased probability of tpe 1 error.

We can detect a violation of this assumption using Levene's test. If this is significant then the homogeneity of variance is violated.

If it is violated we can adjust the df and p-value using Welch's procedure, which can protect against type 1 error, or e can use a non-parametric test.

```
# example with equal variance
library(car)
sample1<-rnorm(100,50,1)
sample2<-rnorm(100,50,1)
y <- c(sample1, sample2)
group <- as.factor(c(rep(1, length(sample1)), rep(2, length(sample2))))
leveneTest(y,group)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1  0.2201 0.6395
##       198
```

```
# independent t test
t.test(y~group,var.equal = T)
```

```
##
##  Two Sample t-test
##
## data:  y by group
## t = -1.2389, df = 198, p-value = 0.2169
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4793083  0.1094419
## sample estimates:
## mean in group 1 mean in group 2
##        49.96254        50.14747
```

```
cohensD(y~group,method = "pooled")
```

```
## [1] 0.1752022
```

```
#example with unequal variance
sample1<-rnorm(100,50,1)
sample3<-rnorm(100,50,4)
y1 <- c(sample1, sample3)
group <- as.factor(c(rep(1, length(sample1)), rep(2, length(sample3))))
leveneTest(y1,group)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)    
## group   1  110.38 < 2.2e-16 ***
##       198                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Conducting multiple t-tests is tedious and increases the probability of Type 1 error. Instead, when there are more than two group means to compare, conduct analysis of variance (ANOVA).

Analysis of Variance (ANOVA)

Appropriate when the predictos (IVs) are all categorical and the outcome (DV) is continuous. Most common application is to analyse data from randomised controlled experiments.

More specifically, randomised controlled experiments that generate more than two group means. If only two group means use a dependent or indenpent t-test.

If you have more than two group means and they're all independent then you use a between groups ANOVA. If you have more than two group means but they're dependent then you use a repeated measures ANOVA.

The null hypothesis for an ANOVA is that all groups are equal.

The ANOVA will tell us if there is an overall effect somewhere. That's what the F-test does.

ANOVA typically involves NHST. The test statistic is the F-test (F-ratio). F = (variance between groups)/(variance within groups)

The variance within groups is unsystematic and what we would expect due to chance.

F-tests

Like the t-test and family of t-distributions the F-test has a family of F-distributions. The distribution to assume depends on the number of subjects per group and the number of groups.

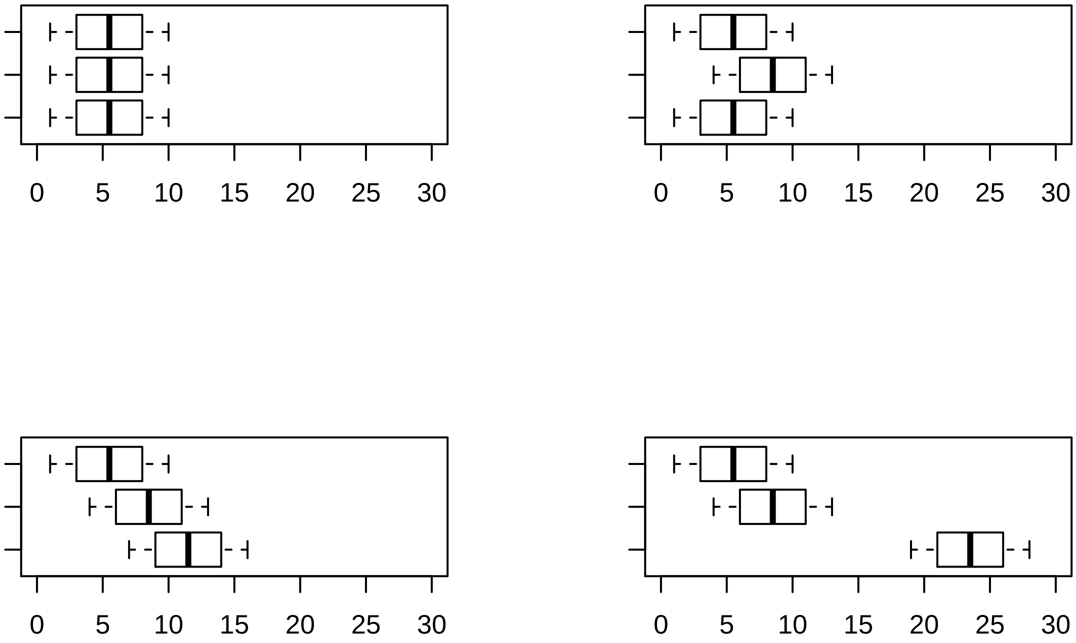
F tests are most commonly used for two purposes:

1. in ANOVA, for testing equality of means (and various similar analyses); and
2. in testing equality of variances

```
##        value        numdf        dendf 
## 1.592065e-30 2.000000e+00 2.700000e+01
```

```
##     value     numdf     dendf 
##  3.272727  2.000000 27.000000
##     value     numdf     dendf
##  9.818182  2.000000 27.000000
```

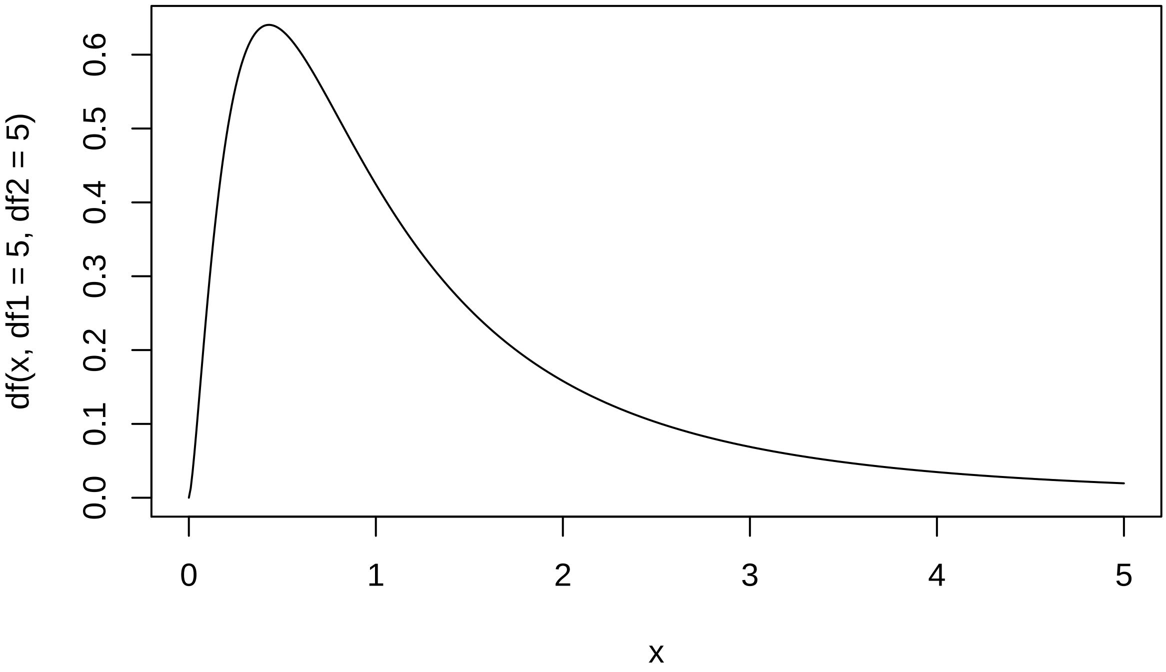
```
##    value    numdf    dendf
## 101.4545   2.0000  27.0000
```



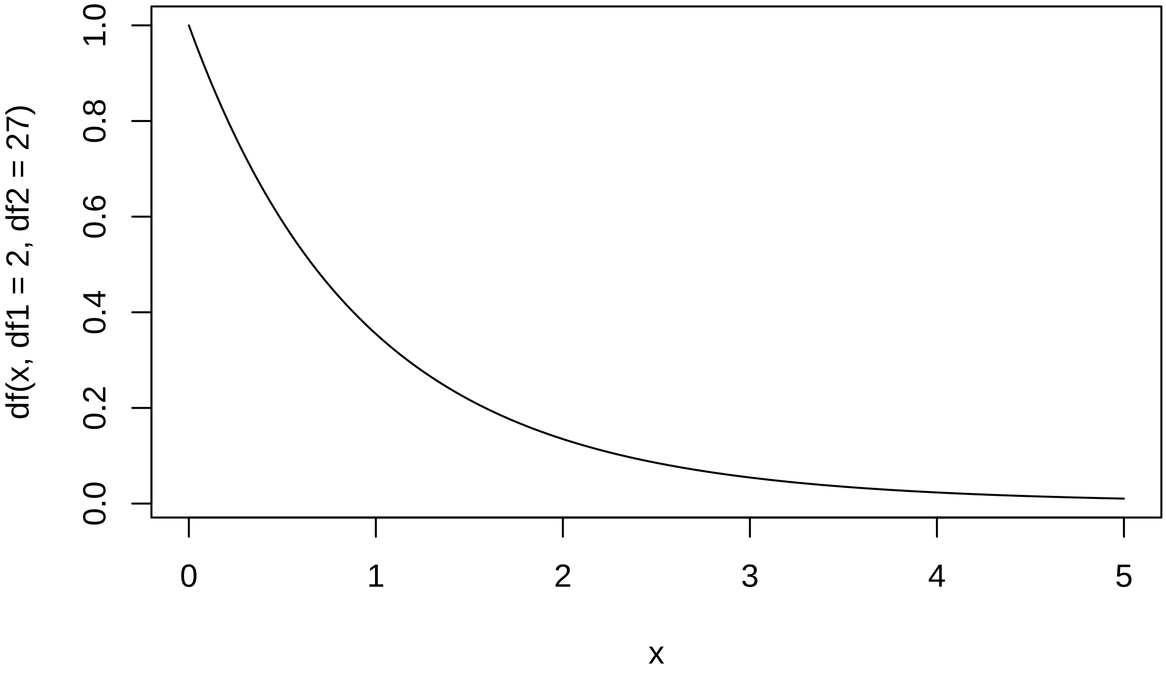
If the null hypothesis (equality of population means) were true, you'd expect some variation in sample means, and would typically expect to see F ratios roughly around 1. Smaller F statistics result from samples that are closer together than you'd typically expect, so you aren't going to conclude the population means differ.

That is, for ANOVA, you'll reject the hypothesis of equality of means when you get unusually large F-values and you won't reject the hypothesis of equality of means when you get unusually small values (it may indicate something, but not that the population means differ).

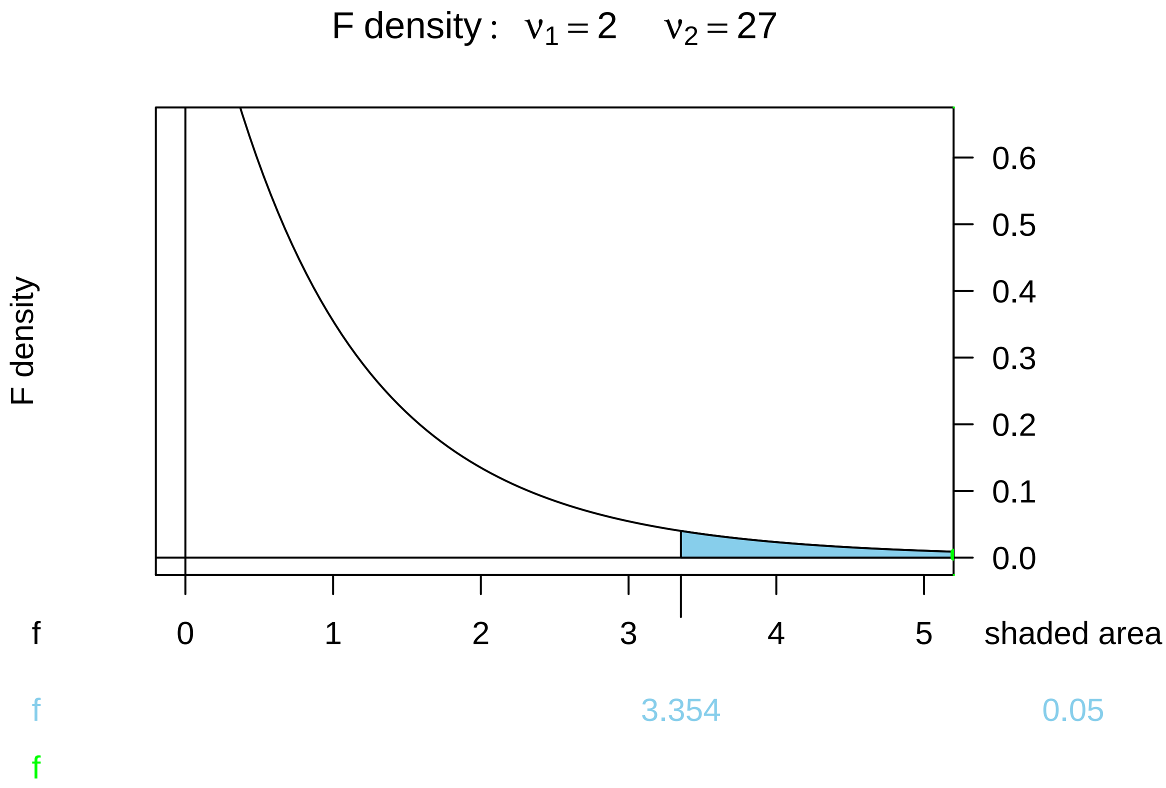
```
x<-seq(0,5,0.01); plot(x,df(x,df1=5,df2=5), type="l")
```



```
x<-seq(0,5,0.01); plot(x,df(x,df1=2,df2=27), type="l")
```



This illustration shows that we only want to reject when F is in its upper tail.



2) F tests for equality of variance* (based on variance ratios). Here, the ratio of two sample variance estimates will be large if the numerator sample variance is much larger than the variance in the denominator, and the ratio will be small if the denominator sample variance is much larger than variance in the numerator.

That is, for testing whether the ratio of population variances differs from 1, you'll want to reject the null for both large and small values of F.

- (Leaving aside the issue of the high sensitivity to the distributional assumption of this test (there are better alternatives) and also the issue that if you're interested in suitability of ANOVA equal-variance assumptions, your best strategy probably isn't a formal test.)

F-ratio can be written in a number of ways:

F = between groups variance/ within groups variance

F = $MS_between$ / MS_within

F = MS_A / $MS_{S/A}$

MS = mean squares = variance

If we take the last line to describe F

then $MS_A = SS_A$ / df_A

and $MS_{S/A} = SS_{S/A}/df_{S/A}$

We compare each group mean to the grand mean to get variance across groups.

$SS_A = n\Sigma(Y_j - Y_T)^2$ where

Y_j are the group means and

Y_T is the grand mean

For within groups we take an individual's score and take away the group mean to get the within group sums of squares.

$SS_{S/A} = \Sigma(Y_{ij} - Y_j)^2$ where

Y_{ij} are individual scores and

Y_j are the group means

$df_A = a - 1$

$df_{S/A} = a(n - 1)$

$df_TOTAL = N - 1$

ANOVA can also be biased by sample size so we can derive an effect size. In this case it's called eta-squared (analagous to R^2), and is the proportion of variance in the dependent variable exaplained by the independent variable.

$\eta = SS_A/SS_Total$

Assumptions of ANOVA

DV is continuous (interval or ratio variable)

DV is normally distributed.

Homogeneity of variance: within-groups variance is equivalent for all groups. To test this we can use Levene's test.

As with the independent t test we pool the standard deviations across groups. If Levene's test is significant then homogeneity of variance assumption has been violated and we instead conduct pairwise comparisons using a restricted error term.

```
y1<-rnorm(20,1,1)
y2<-rnorm(20,1.5,1)
y3<-rnorm(20,2,1)
y4<-rnorm(20,2.5,1)

y = c(y1, y2, y3, y4)
n = rep(20, 4)
n
```

```
## [1] 20 20 20 20
```

```
group = rep(1:4, n)

data = data.frame(y = y, group = factor(group))
fit = aov(y ~ group, data)
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## group      3 31.267 10.4225  11.574 2.492e-06 ***
## Residuals 76 68.440  0.9005
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(fit)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = y ~ group, data = data)
##
## $group
##          diff         lwr      upr     p adj
## 2-1 0.2207022 -0.56756581 1.008970 0.8825074
## 3-1 1.0153012  0.22703317 1.803569 0.0061179
## 4-1 1.5626890  0.77442099 2.350957 0.0000093
## 3-2 0.7945990  0.00633096 1.582867 0.0474469
## 4-2 1.3419868  0.55371877 2.130255 0.0001545
## 4-3 0.5473878 -0.24088021 1.335656 0.2701415
```

Our F-value tell us that we have that much times between groups variance as within. It is the ratio of the two mean squares. And the mean squares come from the sum of squares divided by the degrees of freedom.

The TukeyHSD is a post-hoc that does a pair-wise comparison and takes into account the fact we're doing multiple comparisons. It thereby protects from overinflating the chance of Type 1 error.

Post-hoc tests

Post-hoc tests in general, allow for multiple pairwise comparisons without an increase in the probability of a Type 1 error.

Many procedures are available, the degree to which p-values are adjusted varies according to procedure. The most liberal involves no adjustment, the most conservative involves the Bonferoni procedure.

Because p = 0.05, if we do the same experiment over and over again, say 100 times to a population where there is no effect, I will get a significant value 5 times. This is what happens when we do multiple comparisons and is the reason we conduct post-hoc tests.

```
# say for 6 possible pairwise comparisons
p.adjust(0.05,method = "bonferroni",6)
```

```
## [1] 0.3
```

Repeated Measures ANOVA

Previously we dealt with between samples ANOVA which is analagous to independent t tests. Repeated measures ANOVA is analagous to dependent t tests.

Pros and cons of repeated measures ANOVA

Pros

Less cost (fewer subjects required)

More statistical power. Subjects may reveal consistent individual differences across experiment i.e. the variance across subjects may be systematic. If so, it will not contribute to the error term.

In a between groups design we have two areas where variance can occur 1. the systematic / between groups variance or 2. unsystematic/ within groups variance.

In a repeated measures design we also have the subject variance. Thus, unsystematic/ within groups variance is reduced as a function of stable subject variance.

Error in a repeated measures design is the inconsistency of subjects from one condition to another.

$F_A = MS_A/MS_{A*S}$

where A is the variance attributable to your independent variable.

How much error did we create due to our manipulation is covered by the numerator term MS_A.

The error term in the denominator of the F ratio is mean squares for the interaction between your independent variable and subject variability.

Mean Squares and F Ratio

$MS_A = SS_A/df_A$

$MS_{AxS} = SS_{AxS}/df_{AxS}$

$F = MS_A/MS_{AxS}$

Cons

Counterbalancing

Consider a simple design with just two conditions, A1 and A2 and we're worried about the order to apply these conditions.

One approach is a Blocked Design. Subjects are randomly assigned to one of two "order" conditions e.g. half do the order A1,A2 and the other half do the order A2,A1.

Another approach is a randomised design such that the conditions are presented randomly in a mixed fashion e.g. A2,A1,A2,A2,A2,A1,A2…

Now suppose a = 3 i.e. 3 levels to the independent variable and a blocked design. There are 6 possible orders (3!). This could spiral out of control with ever more levels.

A1,A2,A3

A1,A3,A2

A2,A1,A3

A2,A3,A1

A3,A1,A2

A3,A2,A1

To get around this we implement a "Latin Squares" design. Latin Squares aren't completely counterbalanced but every condition appears at every position at least once. For example, if a=3, then,

A1,A2,A3

A2,A3,A1

A3,A1,A2

Missing Data

More problematic here because the subjects are compared across conditions. Two issues to consider, the relative amount of missing data and the pattern of missing data.

We can test if the missing data has a pattern. For any variable of interest (X) create a new variable (XM). Set XM=0 if X is missing and XM=1 if X is not missing.

Conduct a t-test with XM as the independent variable. If this is significant then the pattern of missing data may be lawful.

Say we measured age, and we want to measure if the missing data is lawful as a function of age. Young people tend not to answer the question 'what is your ethnicity' and this approach could be used to tease out such a pattern.

Remedies to missing data

Drop all the cases without perfect profile. Though this is very drastic, use only if you can afford it.

Alternatively, keep all cases and estimate the values of the missindg data points. There are several options for how to estimate values e.g. multiple regression.

Sphericity assumption

Homogeneity of variance

AND

Homogeneity of covariance

If we have an independent variable with 3 levels, the standardised pairwise covariance (correlation) between each should be approximately the same.

You can test for this using Mauchly's test. If it's significant, then report an adjusted p-value using Greenhouse-Geisser or Huyn-Feldt.

Repeated Measures ANOVA Example

```
#Compare prices in local shops. List of ten representative grocery items and then went to four local st

groceries = read.table(header=T, row.names=1, text="
 subject            storeA  storeB  storeC  storeD     ####
 lettuce              1.17    1.78    1.29    1.29        #
 potatoes             1.77    1.98    1.99    1.99        #
 milk                 1.49    1.69    1.79    1.59        #
 eggs                 0.65    0.99    0.69    1.09        #
 bread                1.58    1.70    1.89    1.89        ### you can copy and paste this
 cereal               3.13    3.15    2.99    3.09        #   part from the table above
 ground.beef          2.09    1.88    2.09    2.49        #
 tomato.soup          0.62    0.65    0.65    0.69        #
 laundry.detergent    5.89    5.99    5.99    6.99        #
 aspirin              4.46    4.84    4.99    5.15     ####
 ")

gr2 = stack(groceries)                                # tidy up data
gr2$subject = rep(rownames(groceries), 4)             # create the "subject" variable
```

```
gr2$subject = factor(gr2$subject)                   # define the subject as factors
colnames(gr2) = c("price", "store", "subject")      # rename the columns
gr2                                                 # take a look
```

```
##    price  store           subject
## 1   1.17 storeA           lettuce
## 2   1.77 storeA          potatoes
## 3   1.49 storeA              milk
## 4   0.65 storeA              eggs
## 5   1.58 storeA             bread
## 6   3.13 storeA            cereal
## 7   2.09 storeA       ground.beef
## 8   0.62 storeA       tomato.soup
## 9   5.89 storeA laundry.detergent
## 10  4.46 storeA           aspirin
## 11  1.78 storeB           lettuce
## 12  1.98 storeB          potatoes
## 13  1.69 storeB              milk
## 14  0.99 storeB              eggs
## 15  1.70 storeB             bread
## 16  3.15 storeB            cereal
## 17  1.88 storeB       ground.beef
## 18  0.65 storeB       tomato.soup
## 19  5.99 storeB laundry.detergent
## 20  4.84 storeB           aspirin
## 21  1.29 storeC           lettuce
## 22  1.99 storeC          potatoes
## 23  1.79 storeC              milk
## 24  0.69 storeC              eggs
## 25  1.89 storeC             bread
## 26  2.99 storeC            cereal
## 27  2.09 storeC       ground.beef
## 28  0.65 storeC       tomato.soup
## 29  5.99 storeC laundry.detergent
## 30  4.99 storeC           aspirin
## 31  1.29 storeD           lettuce
## 32  1.99 storeD          potatoes
## 33  1.59 storeD              milk
## 34  1.09 storeD              eggs
## 35  1.89 storeD             bread
## 36  3.09 storeD            cereal
## 37  2.49 storeD       ground.beef
## 38  0.69 storeD       tomato.soup
## 39  6.99 storeD laundry.detergent
## 40  5.15 storeD           aspirin
```

```
aov.out = aov(price ~ store + Error(subject/store), data=gr2)
summary(aov.out)
```

```
##
## Error: subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  9  115.2    12.8
##
```

```
## Error: subject:store
##           Df Sum Sq Mean Sq F value Pr(>F)  
## store      3 0.5859 0.19529   4.344 0.0127 *
## Residuals 27 1.2137 0.04495                 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The first part of the summary is the subject variance.

The second part is the effect of our condition. The df for our condition is 3 because we have 4 shops.

We can then carry out our post hoc tests. In this case it's the Holm method.

```
with(gr2, pairwise.t.test(price,store, paired = T))
```

```
## 
##  Pairwise comparisons using paired t tests 
## 
## data:  price and store 
## 
##        storeA storeB storeC
## storeB 0.17   -      -     
## storeC 0.17   0.69   -     
## storeD 0.07   0.49   0.33  
## 
## P value adjustment method: holm
```

Here is the Bonferroni method:

```
with(gr2, pairwise.t.test(price,store, paired = T,p.adjust.method = "bonferroni"))
```

```
## 
##  Pairwise comparisons using paired t tests 
## 
## data:  price and store 
## 
##        storeA storeB storeC
## storeB 0.20   -      -     
## storeC 0.21   1.00   -     
## storeD 0.07   1.00   0.66  
## 
## P value adjustment method: bonferroni
```

Chi Square Tests

How to deal with situations where we have a categorical outcome variable e.g. a nominal variable like diagnosis (positive or negative), verdict(guily or innocent) etc. as well as categorical predictors.

The Chi-square goodness of fit statistic determines how well a distribution of proportions "fits" an expected distribution.

In election polls, is there a statistically significant difference in voter preference among candidates.

The Chi-square test of independence determines whether there is a relationship between two categorical variables.

In election polls, is there a relationship between thegender of the voter and the candidate they prefer i.e. is there a contingency.

Chi-square goodness of fit

City Mayoral election

Assume a small poll was conducted (N=60) where people were asked do you intend to vote for CQ, JL or Other?

Let's assume our results were 23 for CQ, 12 for JL and 25 for Other.

Our Chi-square goodness of fit tests the null hypothesis that there are equal proportions across those categories.

The alternative hypothesis is that there are unequal proportions.

$\chi^2 = \sum \frac{(O-E)^2}{E}$

O = observed

E = expected

df = # of categories - 1

p-value depends on χ^2 and df

Similar to t distributions there are a famil of chi-square distributions whose shape is determined by df.

To estimate effect size we use Cramer's V (or Phi)

$\phi_c = sqrt(\frac{\chi^2}{N(k-1)})$

N = sample size

k = # of categories

CQ	JL	Other
20 (E) 23(O)	20 (E) 12(O)	20 (E) 25(O)

Subject	O	E	(O-E)	(O-E)^2	(O-E)^2 / E
CQ	23	20	3	9	0.45
JL	12	20	-8	64	3.2
Other	25	20	5	25	1.25
Total	60	60	0	98	4.9

χ^2 = 4.90, df=2, p = 0.09

Therefore, retain the null hypothesis and conclude that the slight preferences observed here are not statistically significant.

For the effect size,

$\phi_c = sqrt(\frac{\chi^2}{N(k-1)})$

$\phi_c = sqrt(\frac{4.90}{60(3-1)}) = 0.20$

This can be interprted like a correlation coefficient or a standardised regression coefficient.

Chi square test of independence

The Chi-square test of independence determines whether there is a relationship between two categorical variables.

In election polls, is there a relationship between thegender of the voter and the candidate they prefer i.e. is there a contingency.

Again, use the example of a mayoral election in a city.

Assume a small poll was conducted of 200 people. More males than females in the poll (n=140, n=60)

Do you intend to vote for CQ, JL or other?

Our null is that there is no relationship between gender and voter preference.

Our alternative is that there is a relationship between gender and voter preference.

$\chi^2 = \sum \frac{(O-E)^2}{E}$

the same as before

df is different and is now (# of rows - 1) * (# of columns -1)

p-value depends on χ^2 and df

To estimate effect size we again use Cramer's V (or Phi)

$\phi_c = sqrt(\frac{\chi^2}{N(k-1)})$

N = sample size

k = # of categories or # of rows (whichever is less)

We have to compute the expected frequencies. The proportion of male and female voters for each candidates should be the same as the overall voter preference rates.

To compute the expected frequencies

E=(R/N)*C

E= Expected frequency

R= #of entries in the cell's row

N= total # of entries

C= # of entries in the cell's column

Here is the table for the observed frequencies

	CQ	JL	Other	Row Sums
Female	40	10	10	60
Male	90	40	10	140
Column Sums	150	50	20	200

Here is the table for the expected frequencies

	CQ	JL	Other	Row Sums
Female	(60/200) x 130 = 39	(60/200) x 50 = 15	(60/200) x 20 = 6	60
Male	(140/200) x 130 = 91	(140/200) x 50 = 35	(140/200) x 20 = 14	140
Column Sums	130	50	20	200

Calculate the Chi square score

Subject	O	E	(O-E)	(O-E)^2	(O-E)^2 / E
F/CQ	40	39	1	1	0.03
F/JL	10	15	-5	25	1.67
F/Other	10	6	4	16	2.67
M/CQ	90	91	1	1	0.01
M/JL	40	35	5	25	0.71
M/Other	10	14	-4	16	1.14
Sum	200	200	0	84	6.23

χ^2 = 6.23, df = 2, p=0.04

Reject the null and conclude that there is a significant relationship between gender of the voter and the candidate.

For th effect size,

$\phi_c = sqrt(\frac{\chi^2}{N(k-1)})$

$\phi_c = sqrt(\frac{6.23}{200(2-1)}) = 0.18$

```
Observed <- matrix(c(40,10,10,90,40,10),nrow = 2,ncol = 3, byrow = T)
chisq.test(Observed)
```

```
##
##  Pearson's Chi-squared test
##
## data:  Observed
## X-squared = 6.2271, df = 2, p-value = 0.04444
```

Assumptions

Adequate expected cell counts. A common rule is 5 or more in al cells of a 2x2 table, and 5 or more in 80% of cells in larger tables, and no cells with zero.

When this assumption is not met, Fisher's exact test, a non-parametric test, is recommended.

Independence The observations are assumed to be independent of each other.

This means chi-square cannot be used to test correlated data (like matched pairs of panel data)

In such cases, McNemar's test of dependent proportions is recommended.

Binary Logistic Regression

Outcome variable is a binary categorical but predictors are categorical or continuous.

Aside from that it is the same logic as multiple regression.

When the outcome has 2 levels it's a binary logistic regression.

When the outcome has > 2 levels it's a multinomial regression.

Formula

$ln(\hat{Y}/(1-\hat{Y})) = B_0 + \Sigma(B_k X_k)$

$\hat{Y}$=predicted value on the outcome variable Y

B_0= predicted value on Y when all X = 0

X_k= predictor variables

B_k= unstandardised regression coefficients

$(Y - \hat{Y})$= residual (prediction error)

k = the number of predictor variables

Why $ln(\hat{Y}/(1-\hat{Y}))$?

The predicted score must fall between 0 and 1. The logistic function allows us to do this.

Why not calculate the probability of the outcome = $B_0 + \Sigma(B_k X_k)$???

Because there is no guarantee that the linear combination of predictors will produce a score between 0 and 1

A transformation is therefore applied.

Odds ratio is the probability of the outcome over 1 minus the probability of the outcome.

Odds = P(outcome)/(1-P(outcome))

For example, what are the odds a flipped coin will land heads? Odds = 0.5/0.5 = 1

The take the natural log of the odds, which is called the log-odds or logit

Logit = ln(P(outcome))/(1-P(outcome)) Logit = $ln(\hat{Y}/(1-\hat{Y}))$

P(outcome) = odds/(1+odds)

Odds = P(outcome)/P(~outcome)

e.g. if P=0.5 then Odds=1 and Logit=0.

Example

Outcome variable = faculty promotion to tenure

Predictor variable = publications (pubs)

Logit(promotion)=$B_0 + B_1(pubs)$

Logit(promotion)=$0 + 0.39(pubs)$

For every one unit increase in pubs, the logit increases by 0.39.

Logit = ln(P(outcome))/(1-P(outcome))

Odds = P(outcome)/P(~outcome)

Logit = 0.39 translates to an odds ratio of 1.48. This means that the odds of promotion are multiplied by 1.48 for each increment in publications.

Thus, if the odds or promotion with 16 publications is 1.27 then the odds of promotion with 17 publications is 1.27/1.48 = 1.88

This can also be presented in terms of probability.

Pubs = 17 mean P(promotion) = 0.65, because P(promotion) = Odds/(1+Odds) = 1.88/2.88 = 0.65

We can have a few hypothesis tests with binary logistic regression.

Is an individual predictor variable significant?

Is the overall model significant?

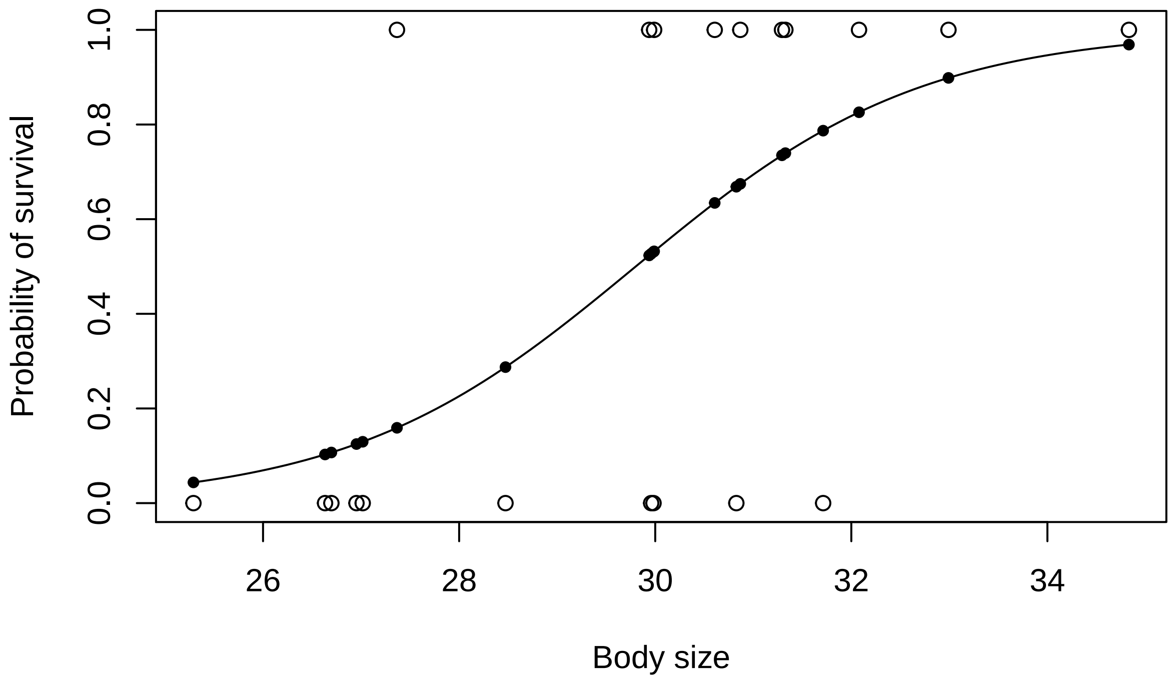
Is Model A significantly better than Model B?

To test each predictor variable, we will look at the regression coefficient, the odds ratio and the Wald test which tests the model Vs the model without the predictor.

To test the overall mode, compare the chi-square for the model to the chi-square of a model with no predictors (the null model) and/or compare models.

Also, does the model classify cases correctly?

```
##    bodysize survive
## 1  25.28988       0
## 2  26.63232       0
## 3  26.69711       0
## 4  26.95281       0
## 5  27.01651       0
## 6  27.36608       1
## 7  28.47262       0
## 8  29.93836       1
## 9  29.95836       0
## 10 29.98275       0
## 11 29.98879       1
## 12 30.60686       1
## 13 30.82722       0
## 14 30.86752       1
## 15 31.29265       1
## 16 31.32744       1
## 17 31.71269       0
## 18 32.07859       1
## 19 32.99109       1
## 20 34.83154       1
```



Binary Logistic Regression Example

Example is based on mock jury research (Diamond & Casper 1992)

People (mock jurors) watched a video of the sentencing of a murder trial in which the defendant had already been found guilty.

The issue for the jurors to decide was whether the defendant deserved the death penalty.

Assume the data were collected "pre-deliberation" which means that each juror was asked to provide his or her vote on the death penalty verdict before the jurors met as a group to decide the overall jury verdict.

Outcome variable (Y)

Verdict is a 1 for voted in favour of the death penalty and is a 0 for voted against the death penalty

Predictors (Xs) which are attitudes the jurors had themselves.

danger (dangerousness) individual's beliefs as to the future dangerousness of the defendant

rehab (rehabilitation) individual's beliefs as to the importance of rehabilitation as a goal of criminal sentencing

punish (punishment) individual's beliefs as to the importance of punishment as a goal of criminal sentencing.

gendet (general deterrence) individual's beliefs as to the importance of general deterrence as a goal of criminal sentencing.

specdet (specific deterrence) individual's beliefs as to the importance of specific deterrence as a goal of criminal sentencing.

incap (incapacitation) individual's beliefs as to the importance of incapacitation as a goal of criminal sentencing.

all measured on a scale of 0-10

The General Linear Model will not guarantee a predicted outcome score between 0 and 1.

The Logit transformation is a feature of an even more "general" mathematical framework in regression than the general linear model.

This is known as the Generalised Linear Model and allows for non-linear relationships between predictors and the outcome variable.

```
##    bodysize survive
## 1  25.08226       0
## 2  25.38019       0
## 3  26.28004       0
## 4  27.16521       0
## 5  27.61190       0
## 6  28.03798       1
## 7  28.61215       0
## 8  30.36651       1
## 9  30.42831       0
## 10 30.61430       0
## 11 30.66739       1
## 12 30.69262       1
## 13 31.06941       0
## 14 31.37493       1
## 15 31.50947       1
## 16 31.77770       1
## 17 32.19427       0
## 18 32.41963       1
## 19 32.81658       1
## 20 34.93535       1
```

```
##
## Call:
## glm(formula = survive ~ bodysize, family = binomial, data = dat)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.80165  -0.62690   0.01073   0.82101   1.81070
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -20.4922     9.6773  -2.118   0.0342 *
## bodysize      0.6801     0.3177   2.141   0.0323 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27.726  on 19  degrees of freedom
## Residual deviance: 19.485  on 18  degrees of freedom
## AIC: 23.485
##
## Number of Fisher Scoring iterations: 5
##  (Intercept)     bodysize
## 1.259906e-09 1.974081e+00
```

```
## [1] 8.240489
```

```
## [1] 1
```

```
## [1] 0.00409661
```

```
## $rawtab
##        resp
##         0 1
##   FALSE 6 1
##   TRUE  4 9
##
## $classtab
##        resp
##           0   1
##   FALSE 0.6 0.1
##   TRUE  0.4 0.9
##
## $overall
## [1] 0.75
##
## $mcFadden
## [1] 0.2972128
```

More than 2 categories on the outcome = multinomial logistic regression. A-1 logistic regression equations are formed where A = # of groups and one group serves as a reference group.

Assumptions

Two primary constraints

1. The normal distribution in Y.

2. Linear relationship between predictor variables and outcomes variables.

What do we do when we can't satisfy these assumptions?

For the first, i.e. a normal dist. in Y how do we test for it?

Histograms and summary stats

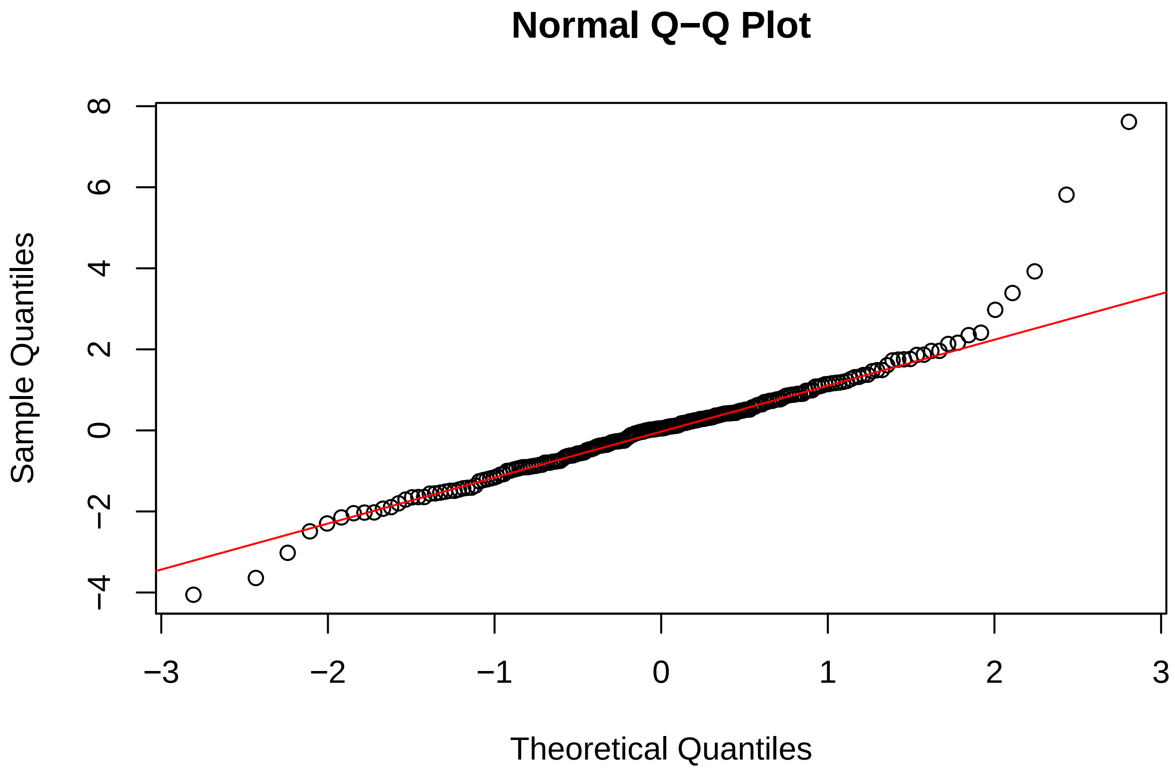
look for extreme skew (>3) and/or kurtosis (>10) and/or outliers (e.g. cases +/- 3 SDs from the mean)

Q-Q plots

A plot of the sorted values from the data set against the expected values of the corresponding quantiles from the standard normal distribution.

If the distribution is normal then the plotted points should approximately lie on a straight line.

```
y <- rt(200, df = 5)
qqnorm(y); qqline(y, col = 2)
```



Empirical tests, such as D'Agostino's Kˆ2 test

Neither the dependent nor independent variable needs to be normally distributed. In fact they can have all kinds of loopy distributions. The normality assumption applies to the distribution of the errors

Probability Density Functions

The "bell curve" shape is governed by the PDF. However, the actual "y"-value of this curve is itself more or less meaningless. The integral of the PDF $f(x)$ gives the probability that your random variable is less than some value:

$P(x < X) = \int_{\infty}^{X} f(x)dx.$

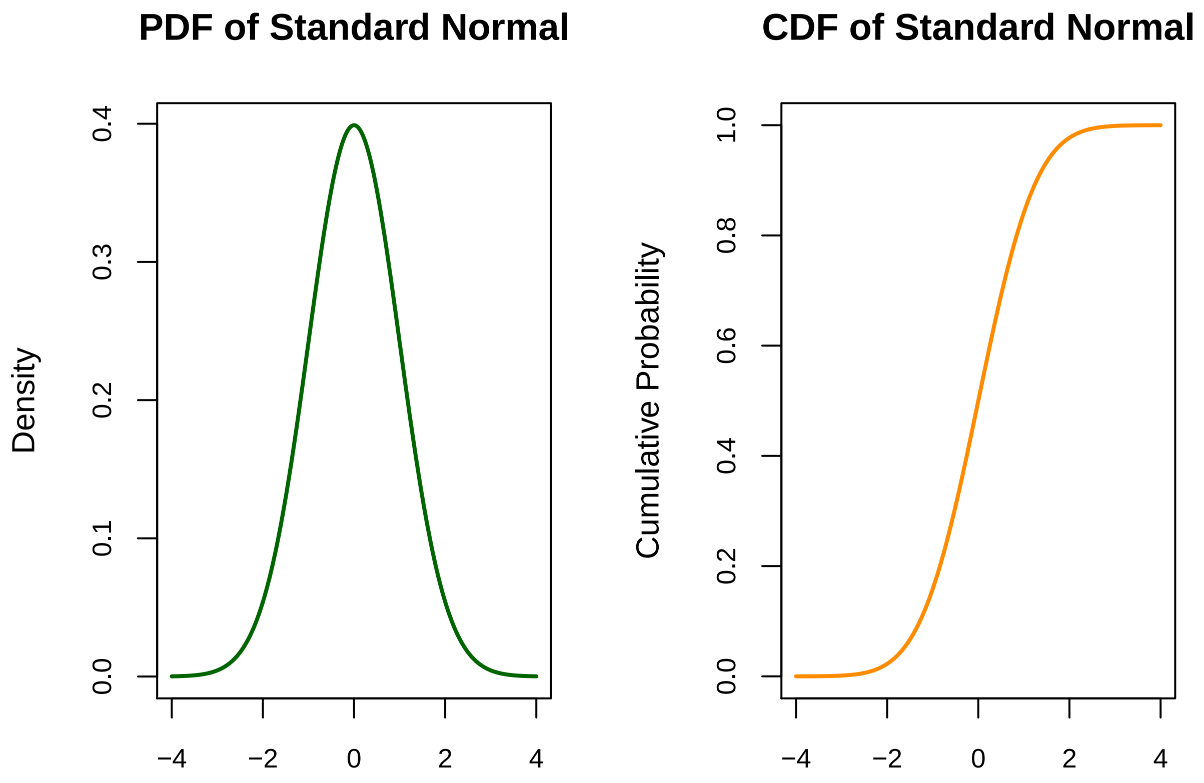
This is known as the CDF, or cumulative distribution function. By the fundamental theorem of calculus, the PDF is then the derivative of the CDF; that is, the PDF is the derivative of a function that returns a probability. So what is that intuitively? Honestly… it's not really anything. The "units" of the vertical axis in the PDF plot don't lead to anything intuitive; they are meaningful, but only in a derived, mathematical sense.

The area under the pdf equals 1. If the pdf value $f(x)$ exceeds 1 for some and indeed many values of x, that is perfectly fine: but $f(x)$ cannot exceed 1 for all x in an interval I of length exceeding 1. If the latter condition were to hold, then:

$\int_I f(x)dx =$ area under pdf in interval I > 1

in violation of the constraint that the total area is 1. The value of $f(x)$ is not a probability. The units of $f(x)$ are probability per unit length and you must multiply by length (more generally, find an area) to get a probability.

As a consequence, some people wish to think that $f(X)$ is the probability that $x = X$, but this is untrue for continuous distributions ($P(x = X) = 0$). However, for the PDF's discrete analog, the Probability Mass Function (PMF), this statement is quite true.



t-test

t-Test to compare the means of two groups under the assumption that both samples are random, independent, and come from normally distributed population with unknown but equal variances

To solve this problem we must use to a Student's t-test with two samples, assuming that the two samples are taken from populations that follow a Gaussian distribution (if we cannot assume that, we must solve this problem using the non-parametric test called Wilcoxon-Mann-Whitney test). Before proceeding with the t-test, it is necessary to evaluate the sample variances of the two groups, using a Fisher's F-test to verify the homoskedasticity (homogeneity of variances). In R you can do this in this way:

```r
a = c(175, 168, 168, 190, 156, 181, 182, 175, 174, 179)
b = c(185, 169, 173, 173, 188, 186, 175, 174, 179, 180)

var.test(a,b)
```

```
##
##  F test to compare two variances
##
## data:  a and b
## F = 2.1028, num df = 9, denom df = 9, p-value = 0.2834
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.5223017 8.4657950
## sample estimates:
## ratio of variances
##           2.102784
```

We obtained p-value greater than 0.05, then we can assume that the two variances are homogeneous. Indeed we can compare the value of F obtained with the tabulated value of F for alpha = 0.05, degrees of freedom of numerator = 9, and degrees of freedom of denominator = 9, using the function qf(p, df.num, df.den):

```
qf(0.95, 9, 9)
```

```
## [1] 3.178893
```

Note that the value of F computed is less than the tabulated value of F, which leads us to accept the null hypothesis of homogeneity of variances.NOTE: The F distribution has only one tail, so with a confidence level of 95%, p = 0.95. Conversely, the t-distribution has two tails, and in the R's function qt(p, df) we insert a value p = 0975 when you're testing a two-tailed alternative hypothesis. Then call the function t.test for homogeneous variances (var.equal = TRUE) and independent samples (paired = FALSE: you can omit this because the function works on independent samples by default) in this way:

```
t.test(a,b, var.equal=TRUE, paired=FALSE)
```

```
##
##  Two Sample t-test
##
## data:  a and b
## t = -0.94737, df = 18, p-value = 0.356
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -10.93994   4.13994
## sample estimates:
## mean of x mean of y
##     174.8     178.2
```

We obtained p-value greater than 0.05, then we can conclude that the averages of two groups are significantly similar. Indeed, the value of t-computed is less than the tabulated t-value for 18 degrees of freedom, which in R we can calculate:

```
qt(0.975, 18)
```

```
## [1] 2.100922
```

This confirms that we can accept the null hypothesis H_0 of equality of the means.

Interactions

Interactions allow us assess the extent to which the association between one predictor and the outcome depends on a second predictor.