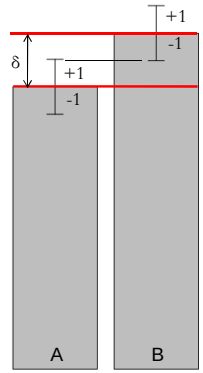
Session 7:
Error bars
Correlation
Regression (linear, non-linear)
Goodness of fit
Measurement scales: which analysis?

Error bars

A note on the use and abuse of error bars

- Everyone uses them, but not many understand what they mean
- We would like error bars to tell us this:
  - non-overlapping bars show significant difference of means
  - overlapping bars show non-significance
  - Remember from power of test that for the difference between means δ to be significant we require
  - $\delta = critical\text{-}t \cdot$ s.e.d.m.
  - δ is also known as LSD (Least Significant Difference)



A note on the use and abuse of error bars

- So one error bar 'arm' should be = 0.5 · *critical-t* · s.e.d.m.
- There are 3 different types of error bar in common use
  - 1. Error bar arm = 1 s.e.m.
    - This is too narrow because
      - s.e.m. < 0.5 · *t* · 1.4 · s.e.m.
      - (since t ≥ 2 and s.e.d.m. = 1.4 · s.e.m.)
  - 2. Error bar arm = 95% confidence interval = *t* · s.e.m.
    - This is too broad because
      - *t* · s.e.m. > 0.5 · *t* · 1.4 · s.e.m.
      - (since s.e.d.m. = 1.4 · s.e.m.)
  - 3. Error bar arm = LSD/2 = 0.5 · *critical-t* · s.e.d.m.
    - That's just right

In the simplest case
(same variance and sample size):
$s.e.d.m. = \sqrt{2}\ s.e.m. \approx 1.4\ s.e.m.$

Covariance & Correlation

Covariance and Correlation

- Covariance is the way two variables vary together, if covariance is 0 they are independent
  - We have already noted that positive covariance reduces the standard error of the difference of means s.e.d.m.

$$\text{variance of difference} = \text{var}(A) + \text{var}(B) - 2\,\text{covariance}(A, B)$$

  - In R, you can calculate the covariance of x and y with var() or cov()

```
> var(x,y)#normal var() command, just supply 2 variables
> cov(x,y)
```


Covariance and Correlation

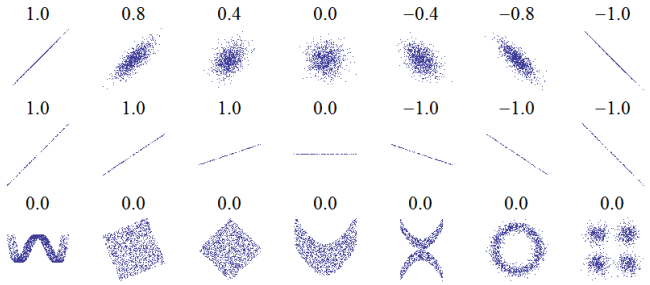
- The correlation coefficient *r* is a standardized measure of covariance
  - It is essentially the same as covariance but
  - On a scale from -1 to 1
  - Dimensionless (no units)

$$r = \frac{\text{cov}(x, y)}{\sqrt{\text{var}(x)\,\text{var}(y)}}$$

- That's why it is more useful
- In R, you can
  - calculate correlation with cor()
  - test for significance of correlation with cor.test()

Covariance and Correlation

- Correlation is a measure of *linear* dependence of variables
  - -1 Perfect negative correlation (all data on a straight line)
  - 0 No correlation (no linear dependence)
  - +1 Perfect positive correlation (all data on a straight line)



Covariance and Correlation

- Example data from Crawley's website: depth of water table in winter and summer in 9 different locations (m below the surface)

```
> water <- read.table("water_table.txt",header=TRUE)
> attach(water)
> names(water)
[1] "Location" "Summer"   "Winter"
> cor(Summer, Winter)
[1] 0.8820102
```

  - Conclusion: water levels in summer and winter are positively correlated
  - But is this correlation significant?

Covariance and Correlation

  - Let's test for significance

```
> cor.test(Summer,Winter)
        Pearson's product-moment correlation
data:  Summer and Winter
t = 4.9521, df = 7, p-value = 0.001652
alternative hypothesis: true correlation is not equal to
 0
95 percent confidence interval:
 0.5259984 0.9750087
sample estimates:
      cor
0.8820102
```

  - Conclusion: the positive correlation is significant

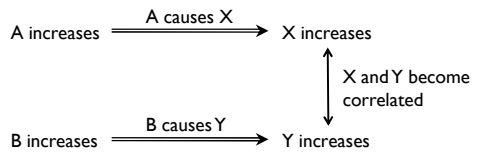
Covariance and Correlation

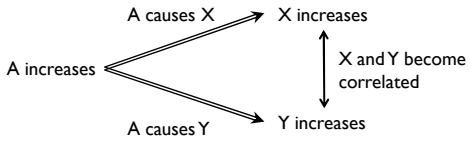
- There are different correlation coefficients for different scales of measurement
  - Parametric measure (can only be used for quantitative data) and assumes normally distributed data
    - Pearson's product moment correlation coefficient
      - Pearson's r
  - Non-parametric (can be used for ordinal but of course also for 'higher' scale data)
    - Spearman's rank correlation coefficient ϱ (rho) or r_s
      - Spearman's ρ
      - Special case of Pearson's r where the calculations are applied to rankings
    - Kendall's tau rank correlation coefficient
      - Kendall's τ
      - Measures the degree of correspondence between two rankings
  - You can choose these in cov(), cor() and cor.test() by setting the argument method="pearson" or "kendall" or "spearman"

Correlation does not imply causation

- Number of churches within a city's limits and number of bars
- Number of storks and number of births in villages in rural Holland
- Milk intake during childhood and subsequent heroin addiction
- Childhood vaccines and occurrence of autism
  - MMR vaccine is given around 12-18 months
  - First signs of autism are observed around 12-18 months
- Correlation is a necessary but not sufficient condition for causation

Correlation does not imply causation





- Therefore, if X and Y are correlated, that doesn't imply that X causes Y or Y causes X. They may have a common cause or the correlation is coincidental because two things increase at the same time by chance

Exercise

- Compute the correlation coefficient between y and x as defined below

```
> x <- seq(0,4*pi,pi/10)
> y <- cos(x)
```

  - How large is the correlation?
  - Are you surprised by the results?
    - If you don't understand the results, plot y versus x and you will see
  - Which correlation coefficients can you use?

Regression

Regression

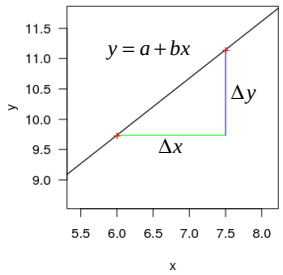
- Regression is the method of choice if all the explanatory variables and the response variable are quantitative measurements (interval or ratio scale)
  - Such quantitative data can be plotted on a scatter plot with continuous x and y axis
- We fit a model (e.g. a straight line or other statistical model) to our sample data in an optimal way and
  - estimate the parameters of the model (e.g. slope and intercept of the straight line)
  - estimate standard errors of the parameters

Regression

- What do I mean by fitting the model to the data optimally?
  - First, we fit the model to the data and not the other way round!
  - Second, optimal means that we calculate the maximum likelihood estimate of the parameter values. Given a model (say a linear model) we find those parameter values that make the data that we have observed the most likely
    - The maximum likelihood model is the model that minimizes the sum of the squared residuals (error sum of squares)
    - These are known as least squares methods

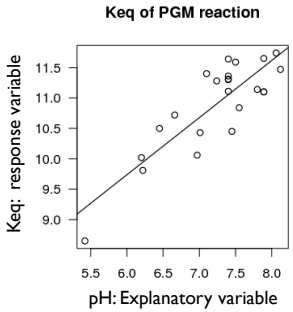
Linear Regression

- In linear regression we fit a straight line to the data
  - A straight line or linear equation is fixed when we have only two data points as we need to estimate two parameters
    - slope $b = \frac{\Delta y}{\Delta x}$
    - intercept $a = y - bx$



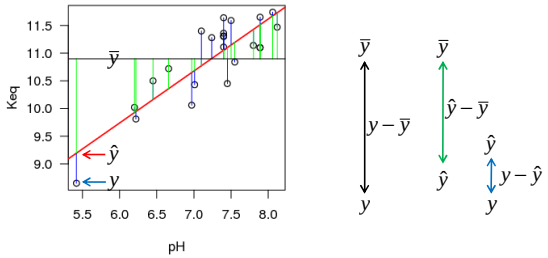
Linear Regression

- Let's fit a linear model to quantitative data
  - The equilibrium constant K_{eq} of the phosphoglycerate mutase (PGM) reaction depends on pH



Linear Regression: Partitioning Variance

- Key idea is to partition the total variance SS_{tot} into 2 parts
  - SS_{reg}: variance explained by the regression model (regression SS)
    - distance of model predicted value $\hat{y}$ from mean $\bar{y}$ (the mean is the null model): $\hat{y} - \bar{y}$
  - SS_{err}: unexplained variance (error SS)
    - distance of data point y to model prediction, the residual: $y - \hat{y}$
    - that's what least-squares minimizes!



Linear Regression

- Read in dataframe and plot data: pH as explanatory and K as response variable

```
k <- read.table("Keq_PGM.txt",header=TRUE)
attach(k)
names(k)
[1] "K" "T"  "pH"  "buffer"  "buffer_concn" "Mg" "Ic"
plot(pH,K,type="p",las=1,ylab="Keq",main="Keq of PGM
  reaction")
```


Linear Regression

- Now we can model K as a function of pH

```
model <- lm(K ~ pH)
```

  - lm() is the name of the function for linear models
  - The output of the lm() function is a list object with many parts and we save it in a variable named "model"
  - Read the "~" as "y is modelled as a function of x"
  - Instead of entering the full linear equation with all parameters (intercept a, slope b), we drop all parameters and replace the "=" sign with the "~" sign

$$y = a + bx \quad \rightarrow \quad y \sim x$$

- This is statistical modelling (not mathematical modelling)

Linear Regression

- All the information is now stored in the variable "model"
- The most important functions to access this information in "model" are the following

```
summary(model) # params, standard errors, F, p-value,
  etc.
confint(model) # confidence intervals for the parameters
coef(model) # short for coefficients(model), i.e.
  parameters
predict(model) # short for fitted(model) or
  fitted.values(model)
resid(model) # short for residuals(model)
influence.measures(model) # influence of different data
  points
plot(model, which=1:6) # 6 model checking diagnostics
abline(model) # quick way of plotting the regression
  line
```


Linear Regression

- Let's have a look at some of the output

This is not "pH" but the slope b, the coefficient with which the explanatory variable (pH) is multiplied

```
> summary(model)
Call:
lm(formula = K ~ pH)
Residuals:
     Min       1Q   Median       3Q      Max
-0.65000 -0.32131  0.06712  0.32305  0.62870
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.1033     0.9217   4.452 0.000221 ***
pH            0.9392     0.1269   7.403 2.79e-07 ***
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.3993 on 21 degrees of freedom
Multiple R-squared: 0.723, Adjusted R-squared: 0.7098
F-statistic:  54.8 on 1 and 21 DF,  p-value: 2.792e-07
```

Linear Regression

- summary(model) gives a wealth of information, but one thing is missing: confidence intervals for the parameters, for this you can use

```
> confint(model)
                2.5 %   97.5 %
(Intercept) 2.1864536 6.020159
pH          0.6753186 1.202989
```

- Retrieve parameter (coefficient) values

```
> coef(model) # short for coefficients(model)
(Intercept)          pH
  4.1033064   0.9391537
```

- To get directly at the parameters in a list is somewhat tricky, note the double brackets (for a vector etc. object you need only single [ ])

```
> p <- coef(model)
> intercept <- p[[1]]
> slope <- p[[2]]
```


Linear Regression: Model checking

- Model fitting by least squares is based on assumptions of
  - Normally distributed errors
  - Constant variance
- Linear models also assume that there is no curvature in the response
- In practice, we also want the data
  - evenly spread out on the x axis
  - lacking outliers
- The best way to check these assumptions is by looking at the residuals and at Cook's distance

Linear Regression: Model checking

- Model diagnostics are a must which is made easy using `plot(model)`
  - Looking at the residuals (observed – predicted) is very useful
    - Normality of errors: are the residuals randomly scattered?
      - A trend in the residuals suggests that a linear model is not adequate
    - Constant variance: the spread of the residuals should not increase with increasing x values
    - Normality of errors: check also the normal quantile-quantile plot
  - Cook's distance identifies data points with a strong influence on the parameters of the model
    - either because these are outliers
    - or because they have a high leverage (are far from the mean, think of a balance where the mean is the pivot point)

Exercise: linear regression

- Example: Bradford protein assay (you measure the extinction of protein standards (BSA) with known concentration to produce a calibration 'curve' and you measure the extinction of your samples)
  - Download text file "protein_assay.txt" from WebCT, read the text file in WordPad to check if there is a header, and read into R
  - Plot the data as points
  - Make a linear regression and plot the fitted model
    - This is our calibration 'curve' for a protein assay
    - Check the model: are there any problems with this fit???
  - Use the fitted equation to calculate the unknown protein concentrations from the following means of extinction measurements of 3 samples (E = 0.311, 0.728, 1.53)

Non-linear regression

Non-linear regression

- Non-linear regression allows you to fit any function to your data that you fancy
- Using the method of least-squares (minimizing the error sum of squares, i.e. the unexplained variance)
- R function nls() for non-linear least squares
  - Contrary to the linear model function lm(), you have to specify the complete equation with all parameters
  - Another difference is that you have to supply initial values for all parameters because the algorithm iteratively improves the fit (judged by least-squares) by varying the parameter values until no further improvement can be obtained (called convergence)
    - Initial values can often be guessed from a plot of the data, and such guesses are good enough most of the time
    - But there are cases where you only get convergence if you start with good guesses or where the model just won't fit (i.e. the model is inappropriate)

Non-linear regression

- For enzyme kinetics, we have theoretical models that predict the dependence of enzyme activity (reaction rate or velocity v) on substrate concentration s
  - First order kinetics (no saturation): rate proportional to substrate concentration $v \propto s$ (proportionality is a special case of linearity where the straight line goes through the origin) $v = k\,s$
  - Michaelis-Menten Kinetics (saturation) $v = \frac{V_{max}\,s}{K_s + s}$
  - Substrate inhibition kinetics (maximum) $v = \frac{V_{max}\,s}{K_s + s + s^2/K_{ss}}$
  - Degrees of freedom = # of data points - # of model params

Non-linear regression

- Let's fit these 3 models to our data and select the best in terms of AIC (good fit without having too many parameters)
  - Read in data, assign variable names s and a, and plot data

```
act <- read.table("KMINDOL.txt",header=TRUE)
attach(act)
names(act)
[1] "substrate_mM"      "activity_U_per_mg"
s <- substrate_mM # abbreviate variable names
a <- activity_U_per_mg
plot(s,a,las=1,type="p",xlim=c(0,10),ylim=c(0,6))
```

Non-linear regression

$v = k s$

- Model 1: velocity proportional to substrate concentration
  - We can use lm() for this linear model, but we want it to go through the origin (zero activity at zero substrate concn)

```
model <- lm(a ~ 0 + s)
plot(model, which=1:6)
summary(model)
confint(model)
AIC(model)
```

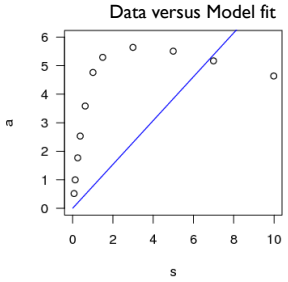
Look at model diagnostics and results

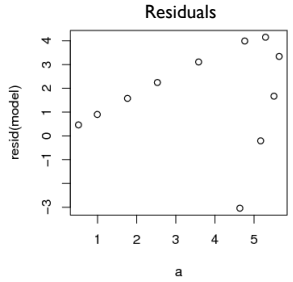
```
p <- coef(model)
k <- p[[1]] # only 1 parameter: first order rate const
x <- seq(0,10,length=200) # make vectors for plotting
y <- k*x
lines(x,y,col="blue") # plot the linear model
```


Non-linear regression

$v = k s$

- Model 1: velocity proportional to substrate concentration





Non-linear regression

$v = \frac{V_{max}\, s}{K_s + s}$

- Model 2: Michaelis-Menten kinetics
  - Now we use nls(), specify equation and initial parameter values

```
model <- nls(a ~ (vm*s)/(ks+s), start=list(vm=6,ks=1))
plot(a,resid(model))
summary(model)
confint(model)
AIC(model)
```

Look at model diagnostics and results

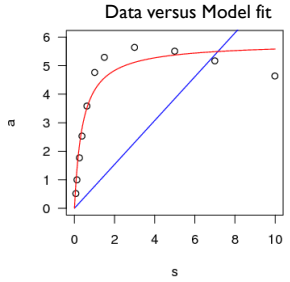
```
p <- coef(model)
vm <- p[[1]]
ks <- p[[2]]
y <- (vm*x)/(ks+x)
lines(x,y,col="red")
```

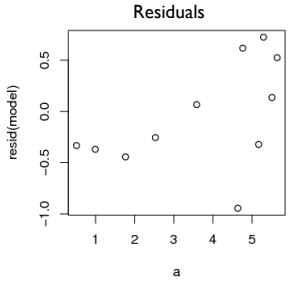
Plot fitted curve

Non-linear regression

$v = \frac{V_{max}\, s}{K_s + s}$

- Model 2: Michaelis-Menten kinetics





Non-linear regression

$$v = \frac{V_{max}\, s}{K_s + s + s^2/K_{ss}}$$

- Model 3: Substrate inhibition kinetics
  - Use nls() again, 1 more term and parameter

```
model <- nls(a ~ (vm*s)/(ks+s+s^2/kss),
  start=list(vm=6,ks=1,kss=10) )
plot(a,resid(model))
summary(model)
confint(model)
AIC(model)
```

Look at model diagnostics and results

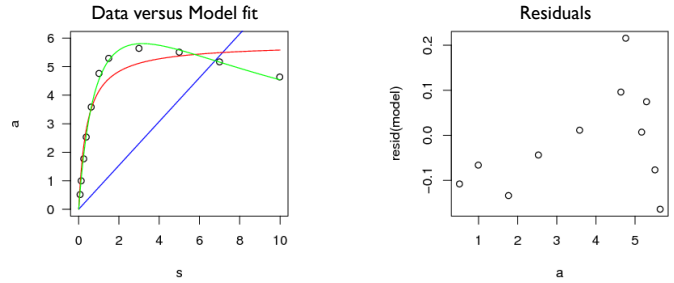
```
p <- coef(model)
vm <- p[[1]]
ks <- p[[2]]
kss <- p[[3]]
y <- (vm*x)/(ks+x+x^2/kss)
lines(x,y,col="green")
```

Plot fitted curve

Non-linear regression

$$v = \frac{V_{max}\, s}{K_s + s + s^2/K_{ss}}$$

- Model 3: Substrate inhibition kinetics



Goodness of fit

Goodness of fit measures

- The Coefficient of Determination r^2 is a measure of how much of the total variation our statistical model can explain
  - In other words: the fraction explained $r^2 = \frac{SS_{tot} - SS_{err}}{SS_{tot}} \approx \frac{SS_{reg}}{SS_{tot}}$
  - In our K ~ pH example, summary(model) told us that

```
Multiple R-squared: 0.723, Adjusted R-squared: 0.7098
```

  - Often referred to as "*r squared*"
  - The correlation coefficient r is the square root of r^2
  - The above is somewhat simplistic and holds only for linear regression where SS_{reg} and SS_{err} add up to SS_{tot}
    - In general, the coefficient of determination is not necessarily the square of the correlation coefficient and the second term in the definition of the coefficient of determination may differ from the first, more correct term

Goodness of fit measures

- In practice, the problem is not so much testing whether our model is significantly better than the null model (the mean), (the coefficient of determination, the fraction of the explained variance, is almost always > 0) but to
  - select the best model from several candidate models
  - determine the significance of the model parameters, are they all necessary to reduce the unexplained variance?
  - estimate the standard error/confidence intervals of the model parameters

Akaike's Information Criterion AIC

- The best model should
  - Fit the data well
    - Note that the more parameters a model has, the better it can fit the data, e.g. a straight line can fit 2 data points exactly, a quadratic polynomial can fit 3 data points exactly, and so on
    - The likelihood of obtaining the actually observed data given a particular model with its set of parameters measures the goodness of fit
  - But have as few parameters as possible and only contain significant parameters that can be estimated well from the data (there is enough information in the data to estimate them)
    - Penalize models with more parameters
  - Akaike's Information Criterion encompasses this trade-off
  - AIC = -2 log(likelihood) + 2 (number of parameters)
    - The lower the AIC, the better. Note AIC can become negative
    - The function in R to calculate AIC is called AIC(), note CAPITAL letters

Goodness of fit example: non-linear regression

- Summary of results
  - Model 1: first order kinetics, residuals deviate systematically, AIC = 56.3

$$v = k\,s$$

  - Model 2: MM kinetics, already much better but residuals deviate systematically, AIC = 21.8

$$v = \frac{V_{max}\,s}{K_s + s}$$

  - Model 3: substrate inhibition kinetics, much better fit with only one parameter more, AIC = -9.6

$$v = \frac{V_{max}\,s}{K_s + s + s^2/K_{ss}}$$

  - The best model has the lowest AIC

Exercise: non-linear regression

- Bacterial growth rate, maximum specific growth rate (h^{-1}), as a function of salt concentration (mol NaCl l^{-1})
  - Download text file "growth_salt.txt" and read into R
  - Here are two potentially suitable kinetic models (g is growth rate, N is salt concentration, g_{max} and the K's are parameters)
    - Edwards: $g = \frac{g_{max}\,N}{K_N + N}\exp(-N/K_I)$
    - Yano-Koga: $g = \frac{g_{max}\,N}{K_N + N + N^3/K_2^2}$
  - Fit both models to the data and plot data and fitted models in the same plot
  - Identify the best model(s) (giving the lowest AIC)

Measurement scales

Measurement scales

Quantitative or continuous variables	Ratio	Zero value defined so can calculate proportions	Concentrations, energy, lengths, absolute temperature Makes sense to say 'twice as much'
	Interval	Distances defined so can calculate differences	Temperature in Celsius, calendar dates
Categorical variables	Ordinal	Rank order defined	Low, medium, high (you can either rate or rank)
	Nominal	Distinctions defined	Bird, Mammal; Female, Male (you can classify but not rank)

Don't calculate sums, differences, means, etc.!

Example data: what scale?

A measurement scale guide to suitable modelling methods

- Decision steps to suitable statistical modelling methods
- Explanatory variables
  - All quantitative — Regression
  - All categorical — Analysis of variance (Anova)
  - Mixed — Analysis of covariance (Ancova)
- Response variables
  - Quantitative — Regression, Anova, Ancova (see above)
  - Proportion — Logistic regression
  - Count — Log linear models
  - Binary — Binary logistic analysis
  - Time-at-death — Survival analysis

Wrap up

- Use Excel for compiling data, but not statistical analysis beyond averaging
  - CI in Excel wrong
  - No histograms, no non-parametric, multivariate, ...
  - Lack of choice and lack of control, unclear how Excel calculates stats (does not fulfil requirements of a methods section)
  - Choice of analysis must be based on science, not what Excel can or can't do
- Learning programming
  - Learning to describe an experimental protocol precisely and completely, step by step, is like programming
  - Learning to proofread
  - Learning how a computer works
  - Automate tasks, combine methods, tailor
- Ability to read equations in a paper/book and use/implement