Regression

Lecture 4

LSM3257

AY22/23; Sem 2 | Ian Z.W. Chan

Summary (Learning Objectives)

Advanced analyses: when to use and Decision tree

Regression

- What is it?
- Important concepts: Maximum Likelihood, slope (b), coefficient of determination (r^2)
- Types of Regression:
    - Linear (OLS) Regression: Assumptions, Power analysis, Fit, Check, Predict
    - Robust Regression
    - Polynomial Regression
    - Multiple Linear Regression: Model simplification, Model comparison, Multicollinearity

When to use more advanced analyses

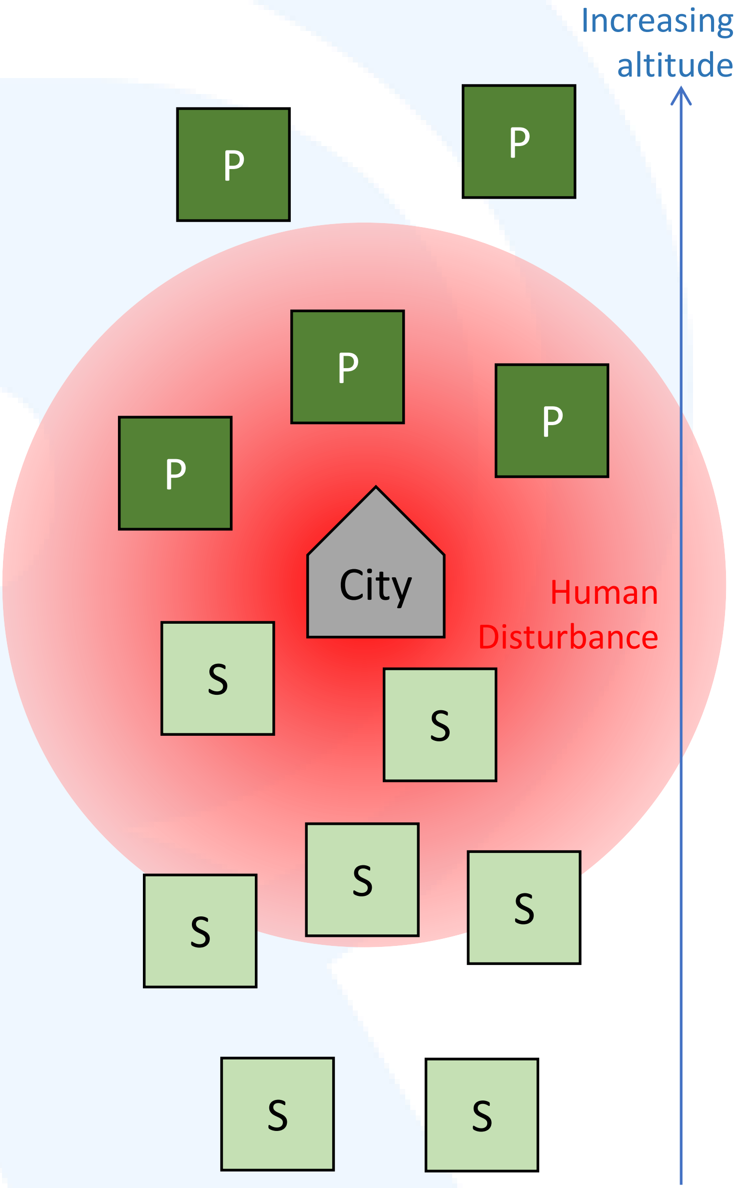
Observational studies with many confounders

More complex relationships (e.g. non-linear)

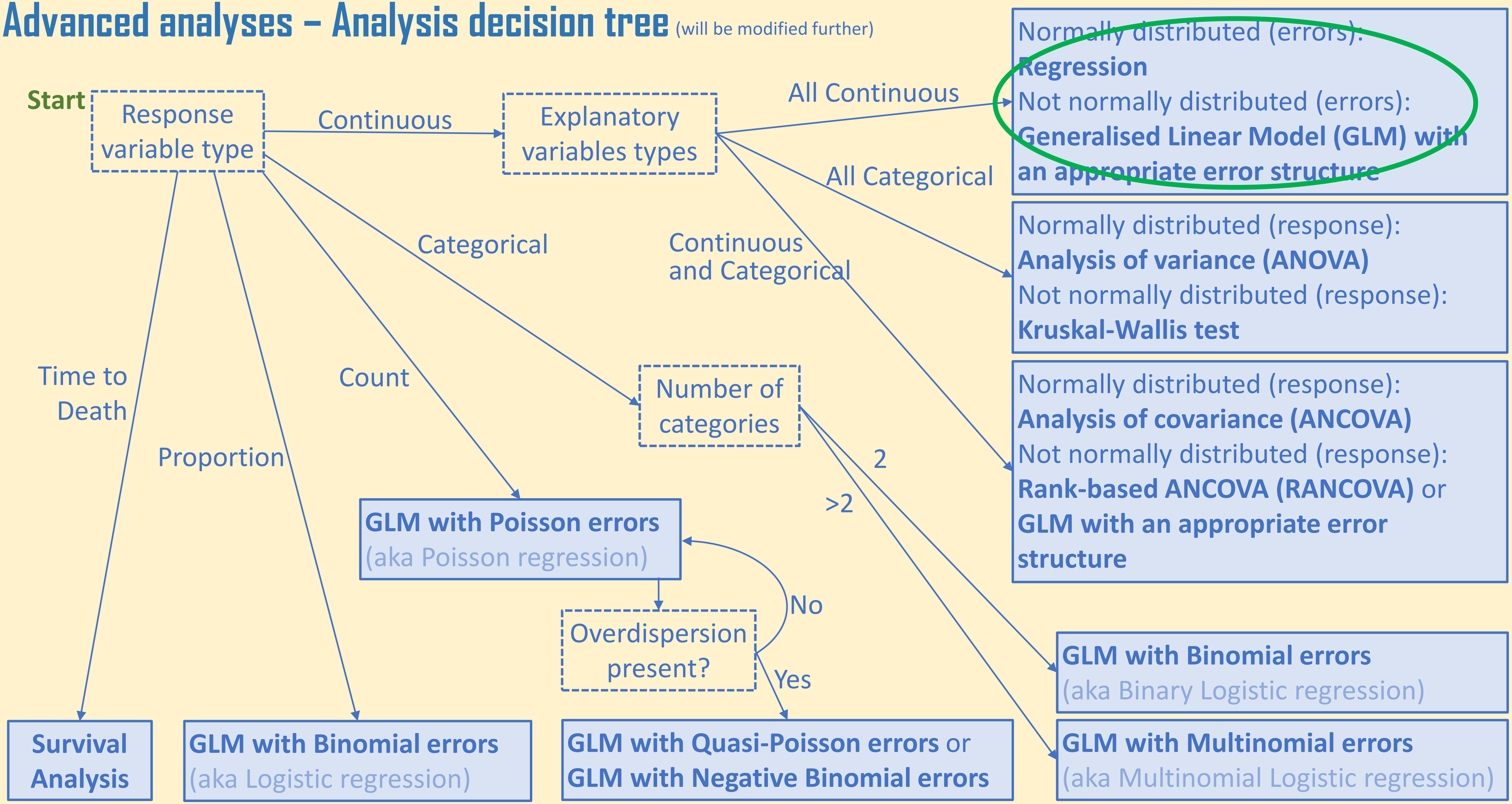
Multiple explanatory variables (still one response variable)

Investigating cause-and-effect rather than correlation

- E.g. Pearson correlation vs. Regression



Advanced analyses – Analysis decision tree (will be modified further)



What is Regression?

Used when your **ONE response variable is continuous** and **all your explanatory variables are continuous**.

From your data, you fit a model to describe a relationship between your explanatory and response variables.

For example, in ordinary least squares (OLS) linear regression (the simplest):

$$y_i = a + bx_i + \varepsilon_i \qquad \text{, where } \varepsilon_i \sim N(0, \sigma^2)$$

In English: the value of the response variable, y_i, can be predicted from the explanatory variable, x_i, by using a linear relationship with an intercept, a; a slope, b; and a **residual (aka errors), ε_i, which follow a normal distribution** of mean 0 and variance σ^2.

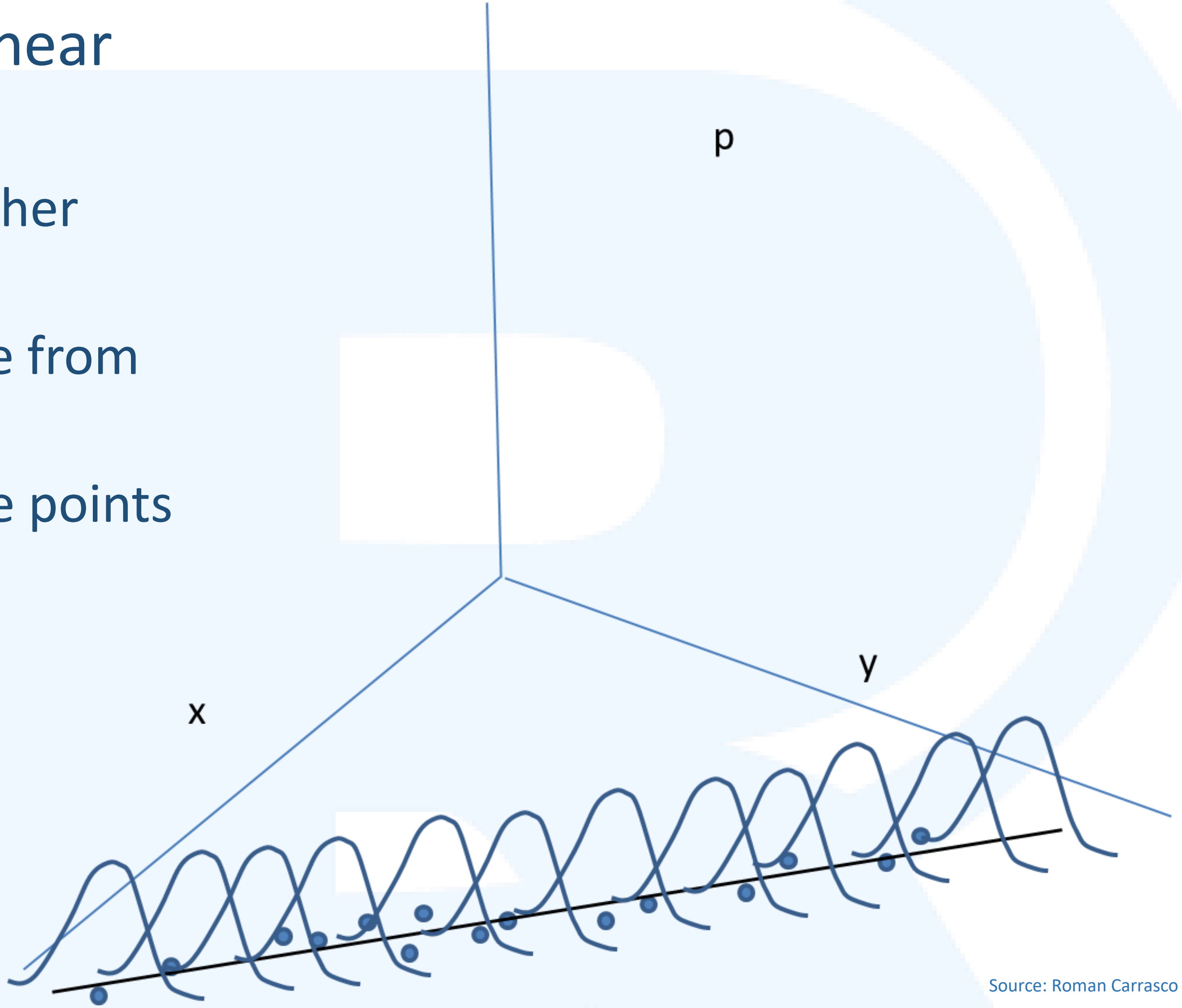
What is Regression?: A conceptual understanding

Diagrammatic representation of a linear regression:

p is the probability of observing other realisations of the data

Regression assumes the data come from these distributions

In reality, we only observe the blue points



Source: Roman Carrasco

Maximum Likelihood

When we fit a model, it will not be perfect, there will always be residuals. We quantify this using the sum of squared residuals (SSE) (see previous lecture): this is the variation in the datapoints NOT explained by our model.

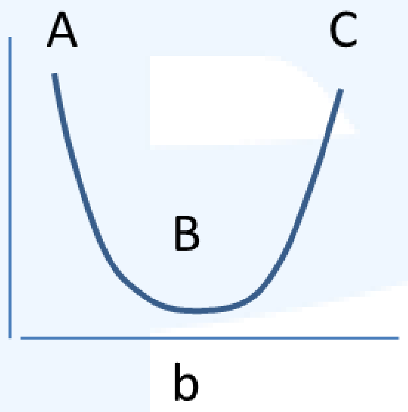
We adjust the intercept (a) and slope (b) of our model to minimise SSE: this is why it's known as a "Least Squares" method.

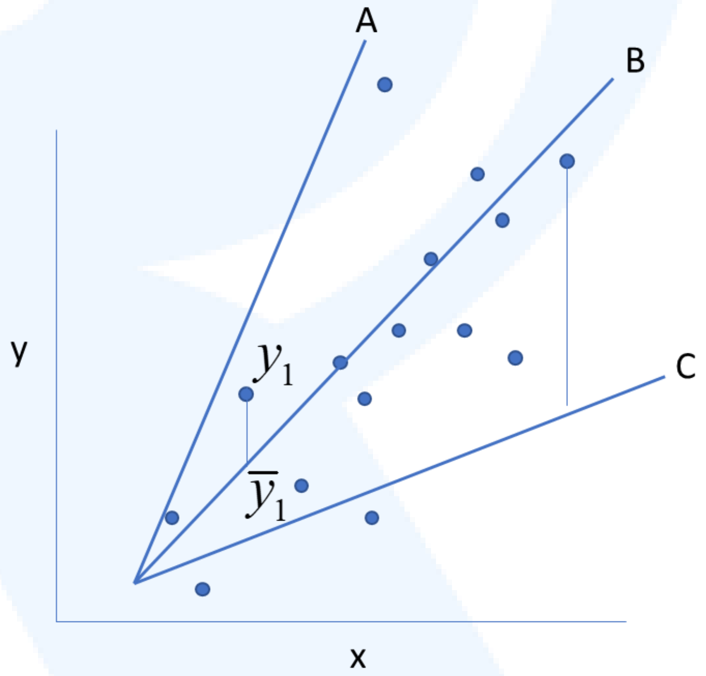
When SSE is minimised, these are the most likely values for a and b given the data (Maximum Likelihood).

If the model is too steep (A) or too gradual (C), the sum of squared residuals (SSE) is not minimised.

SSE is minimised with trend line B.

$$SSE = \sum_{i=1}^{N} \varepsilon^2$$





The slope: b

From the model, the most interesting parameter is b (the slope) which tells us:

a) Whether there is a relationship between x and y (p-value), and

b) The strength/nature of this relationship (the steepness of the slope).

The intercept can potentially be interesting in some cases.

Example: the length of a particular bone (y) at birth when age (x) = 0.

Coefficient of determination (r^2)

The Coefficient of Determination (r^2) is another statistic we are interested in.

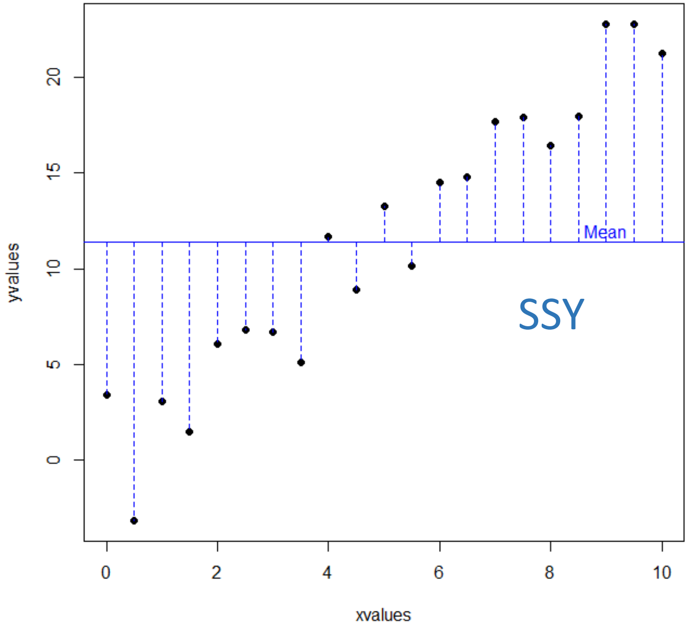
SSY (or SST) is the original variation (total sum of squares).

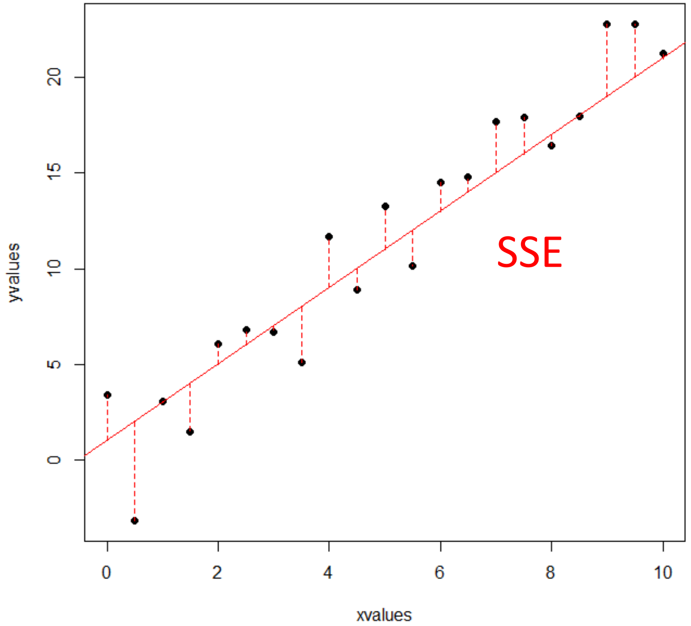
SSE is the residual variation (not explained by the model).

SSR (the variation explained by the model) = SSY – SSE.

$r^2 = \frac{SSR}{SSY}$, is the proportion of variation explained by the model (how well your explanatory variable predicts the response variable). Aka the goodness of fit.

r is the correlation coefficient.





Coefficient of determination (r^2)

Be careful: r^2 is useful but it's only part of the story.

A low r^2 value means we haven't included important variables that would help us explain a larger share of the remaining variance.

- However, the variables we have studied could still explain a share of the variance and that can still be quite interesting.

A very high r^2 (e.g. a saturated model with $r^2 = 1$) fits all the datapoints perfectly but may have no explanatory power!

- r^2 will increase with every variable we add. We therefore do not decide whether to include a variable into a model based on increases in r^2. If we do, we will end up with a saturated model.
- Rather, we need to capture a balance between complexity and goodness of fit (e.g. using AIC or adjusted r^2 or an equivalent criteria).

Types of Regression

<u>Linear regressions</u>

Linear regression aka **Ordinary Least Squares (OLS) regression**: the most simple and frequently used.

Robust regression: this is a more modern (somewhat less established) technique that makes the fit less sensitive to outliers.

Polynomial regression: not so frequent, used to test for simple non-linearities in the relationship between variables.

Multiple regression: similar to linear regression but with multiple explanatory variables.

Note: these (together with ANOVA and ANCOVA) are all called “linear models” and can all be run using lm() in R (except for Robust regression).

<u>More complex types not covered in this lecture</u>

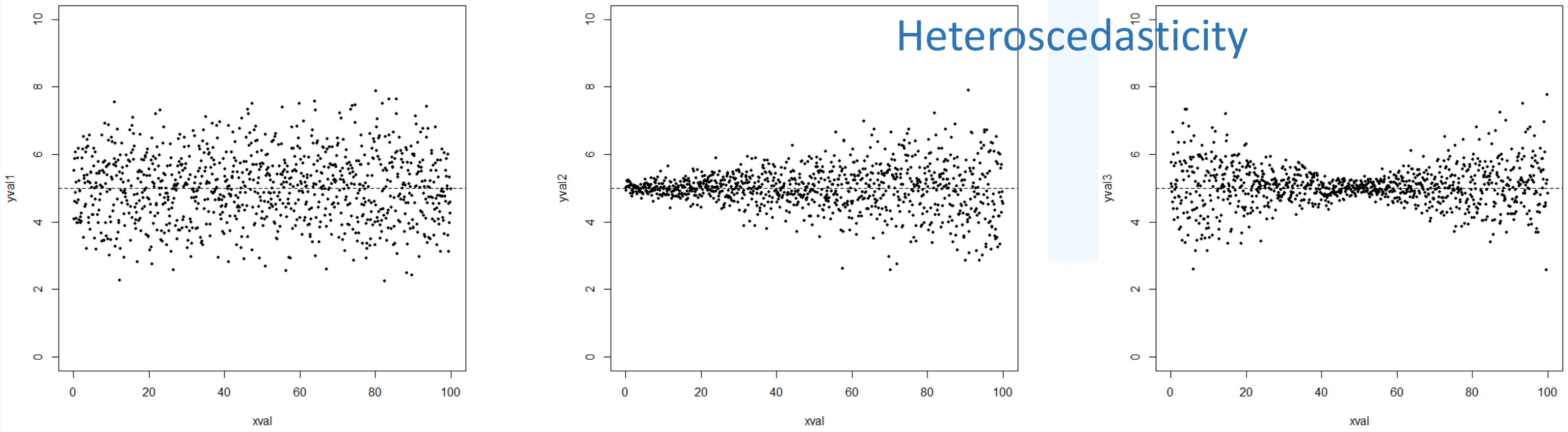
Piecewise regression: fitting 2 or more adjacent lines as opposed to 1 line throughout.

Non-linear regression: fitting complex curves to the data.

Linear (OLS) Regression

Assumptions

1) Homoscedasticity: variance of y is constant through all values of x or y.



2) For each x value, the values of y come from a normal distribution: i.e. the errors aka residuals (in SSE) are normally distributed.

3) The samples are independent from one another.

4) The relationship between y and x is linear.

Note: Biological data are very messy and tend to violate these assumptions and therefore more sophisticated analysis techniques are needed (covered in later lectures)—but regression is the foundation!

Power analysis (also applicable to the other types of regression covered)

#Install and load pwr package

#Code expected: pwr.f2.test(u = ?, f2 = ?, sig.level = 0.05, power=0.8)

u = number of coefficients in the model minus one; for $y_i = a + bx_i$, there are 2 coefficients (a and b), so u = 1

f2 = $r^2/(1-r^2)$; you have to decide your r^2 based on pilot studies or "approximation" (e.g. 0.5 means your model will explain 50% of all variation; a good range is 0.4-0.7); for this example, I choose 0.6

```
pwr.f2.test(u=1,f2=0.6/(1-0.6), sig.level = 0.05, power=0.8)
```

```
> pwr.f2.test(u=1,f2=0.6/(1-0.6), sig.level = 0.05, power=0.8)

     Multiple regression power calculation

              u = 1
              v = 5.716346
             f2 = 1.5
      sig.level = 0.05
          power = 0.8
```

#Calculate sample size needed, n = u + 1 + v (from the output) = 1 + 1 + 5.7 ≈ 8

Fitting a model

```
> str(swiss)
'data.frame':	47 obs. of  6 variables:
 $ Fertility       : num  80.2 83.1 92.5 85.8 76.9 76.1 83.8 92.4 82.
 $ Agriculture     : num  17 45.1 39.7 36.5 43.5 35.3 70.2 67.8 53.3
 $ Examination     : int  15 6 5 12 17 9 16 14 12 16 ...
 $ Education       : int  12 9 5 7 15 7 7 8 7 13 ...
 $ Catholic        : num  9.96 84.84 93.4 33.77 5.16 ...
 $ Infant.Mortality: num  22.2 22.2 20.2 20.3 20.6 26.6 23.6 24.9 21
```

#Inspect our dataset to see variable names

```
str(swiss)
```

Research Question: do fertility rates explain infant mortality?

#Model formula

```
mod1=lm(Infant.Mortality~Fertility,data=swiss)
```

This also works to fit the model:

mod1=lm(swiss$Infant.Mortality~ swiss$Fertility)

But it is not recommended because predict() will not work.

Save your model to this object

The formula

Your dataset

```
mod1=lm(Infant.Mortality~Fertility, data=swiss)
```

Function to fit a linear model (including a linear regression)

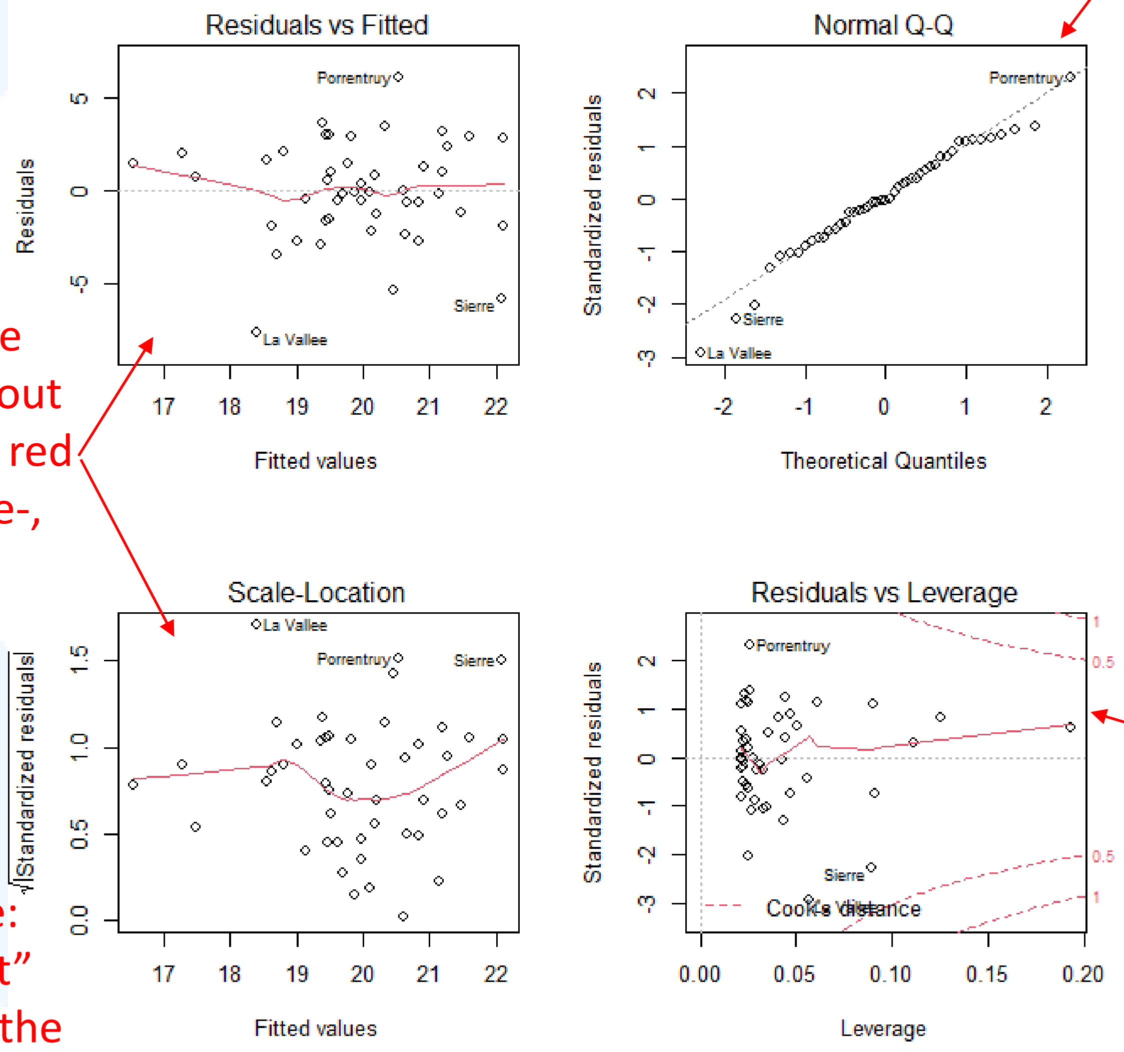
Response variable (the *y* in the equation)

Explanatory variable (the *x* in the equation)

Checking assumptions

#Diagnostic plots

```
par(mfrow=c(2,2))
plot(mod1)
```



<u>Inspecting for homoscedasticity</u>

- Both should look like "stars in the sky" (i.e. no patterns): red line about horizontal and spread around the red line stays relatively constant. Cone-, banana- or s-shape is bad.
- The Scale-Location plot is better when there are more values on one side of the x-axis.
- Top looks good; bottom looks marginal but still OK.
- There are some tests we can use: (a) bptest(mod1) from the "lmtest" package; (b) ncvTest(mod1) from the "car" package. But they tend to be problematic! So I use these plots.

<u>Inspecting for normality of errors</u>

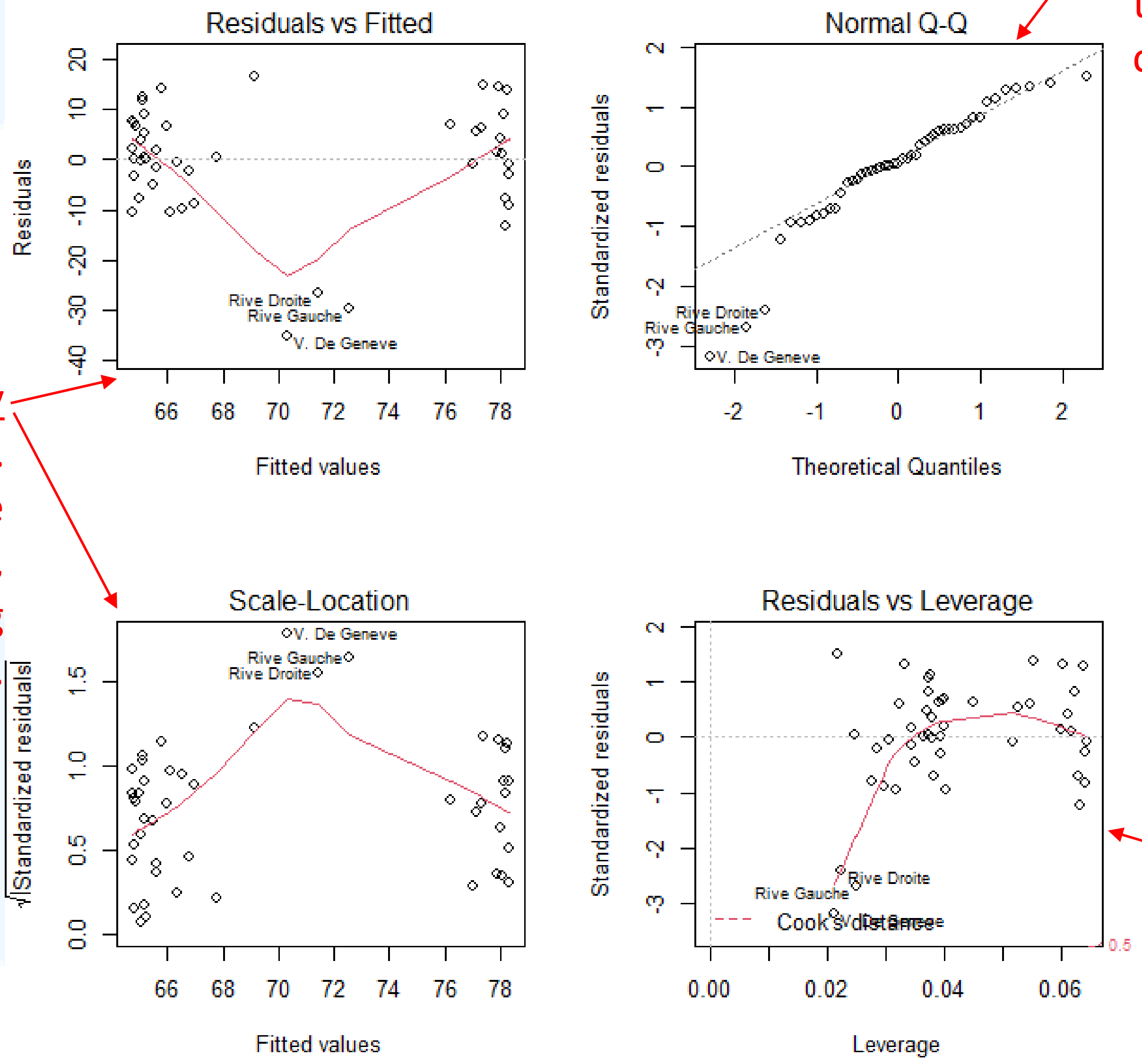
Should be along the diagonal dotted line.
This looks reasonable.
Any banana or s-shape is bad.
Can also test using: shapiro.test(resid(mod1)).

<u>Inspecting for outliers</u>

Should be no points outside the 0.5 or 1 dotted line.
This looks good.
Points outside the lines represent extreme datapoints against a regression line (they have a large effect on the regression results).

Checking assumptions – negative example

Inspecting for homoscedasticity
Bad—there is a clear pattern. There may be something else that needs to be accounted for, perhaps another interacting variable.



Inspecting for normality of errors
Values at bottom left are a little too far below the diagonal for comfort.

Inspecting for outliers
Looks good—no outliers.

Interpreting the model

#Call a summary()

`summary(mod1)`

```
> summary(mod1)

Call:
lm(formula = Infant.Mortality ~ Fertility, data = swiss)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.6038 -1.5673 -0.0607  1.8367  6.0788 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 13.12970    2.25063   5.834 5.51e-07 ***
Fertility    0.09713    0.03160   3.074  0.00359 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.677 on 45 degrees of freedom
Multiple R-squared:  0.1735,	Adjusted R-squared:  0.1552 
F-statistic: 9.448 on 1 and 45 DF,  p-value: 0.003585
```

Model that was run

Distribution of residuals

Whether the relationship is significant: **Yes—Fertility predicts Infant Mortality!**

The intercept is not so interesting for our purposes this time

Coefficient (*b*) value for the Fertility variable (with uncertainty estimate): for every increase of Fertility by 1, Infant.Mortality increases by 0.97 ± 0.032

Our variable accounts for only 17% of the variance but its **effect is still significant**

RSE, a measure of how much variation remains unexplained: square root of (SSE divided by df)

From an ANOVA table (next slide)

Adjusted R^2 tries to account for the number of variables in your model. Use “Multiple R-squared” if you have one explanatory variable and “Adjusted R-squared if you have more than one.

Interpreting the model

#Or call an ANOVA table (don't be confused by the name this is NOT doing an ANOVA)

`anova(mod1)`

Amount of variation explained by your model

F statistic (from tables) used to calculate the p-value

Whether your model explains significantly more variation than a null model

```
> anova(mod1)
Analysis of Variance Table

Response: Infant.Mortality
          Df Sum Sq Mean Sq F value   Pr(>F)
Fertility  1  67.72  67.717  9.4477 0.003585 **
Residuals 45 322.54   7.168
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Amount of unexplained variation remaining (in the residuals)

RSE^2 (RSE from the previous slide)

Note: the "summary" and "anova" output show similar data, but "summary" is formatted to identify relationships between the variables; whereas "anova" is formatted to show how much variation is explained by each variable.

Interpreting the model

Remember our research Question: do fertility rates explain infant mortality?

```
> summary(modl)

Call:
lm(formula = Infant.Mortality ~ Fertility, data = swiss)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.6038 -1.5673 -0.0607  1.8367  6.0788 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 13.12970    2.25063   5.834 5.51e-07 ***
Fertility    0.09713    0.03160   3.074  0.00359 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.677 on 45 degrees of freedom
Multiple R-squared:  0.1735,	Adjusted R-squared:  0.1552 
F-statistic: 9.448 on 1 and 45 DF,  p-value: 0.003585
```

Reporting that <Fertility> rates predict <Infant.mortality>:

"Fertility rates have a significant effect on infant mortality (*P* = 0.004). An increase of 1 in the fertility rate results in an increase in infant mortality of 0.097 ± 0.031 (mean ± SE)."

Note: 67.72/(67.72+322.54) = 0.1735

```
> anova(modl)
Analysis of Variance Table

Response: Infant.Mortality
          Df Sum Sq Mean Sq F value   Pr(>F)   
Fertility  1  67.72  67.717  9.4477 0.003585 **
Residuals 45 322.54   7.168                    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Reporting on how important a predictor <Fertility> is:

"Fertility rates account for 17.4% of the variation in infant mortality."

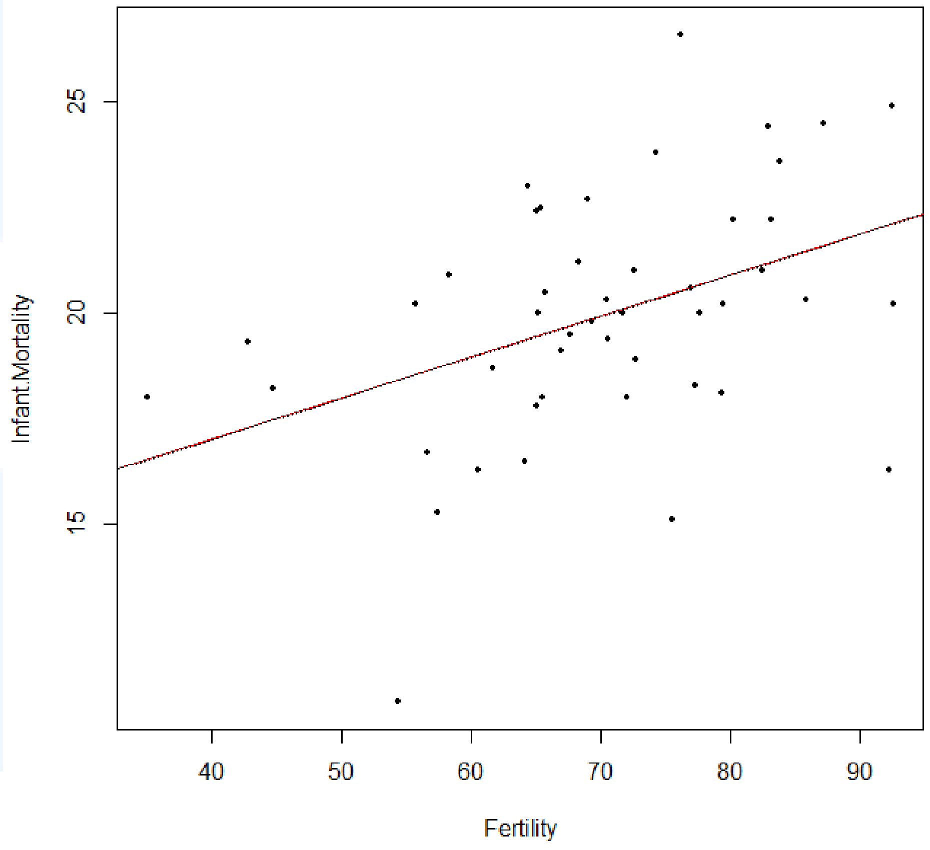
Visualising and Predicting using the model

#Scatterplot with linear trendline from model

```
plot(Infant.Mortality~Fertility,data=swiss,
pch=16,cex=0.5)
```

```
abline(lm(Infant.Mortality~Fertility,data=
swiss),col="red")
```

This plots the typical linear model (for Base R)



#Predict y-values for any x-value (remember it assumes a LINEAR relationship)

```
predict(mod1,list(Fertility=c(2,65,140)))
```

#Can be used to plot the line manually

```
lines(seq(30,100,0.01),predict(mod1,
list(Fertility=seq(30,100,0.01))),lty=9)
```

What if my response variable shows heteroscedasticity?

Option 1: Transform your variables

Try to apply one of these functions to your y-variable and try again:
log(), sqrt(), or cube root “x^(1/3)”

If it works, great! But be careful how you write up your report…

If it doesn’t work …

Option 2: Use GLS (later lecture)

What if my response variable has influential observations/outliers?

Next section: Robust Regression!

Robust Regression

Dealing with outliers/influential observations

#Create dataset and model

```
d2=read.table("diminish.txt",header=T)
plot(yv~xv,data=d2)
mod2=lm(yv~xv,data=d2) #check plots
```

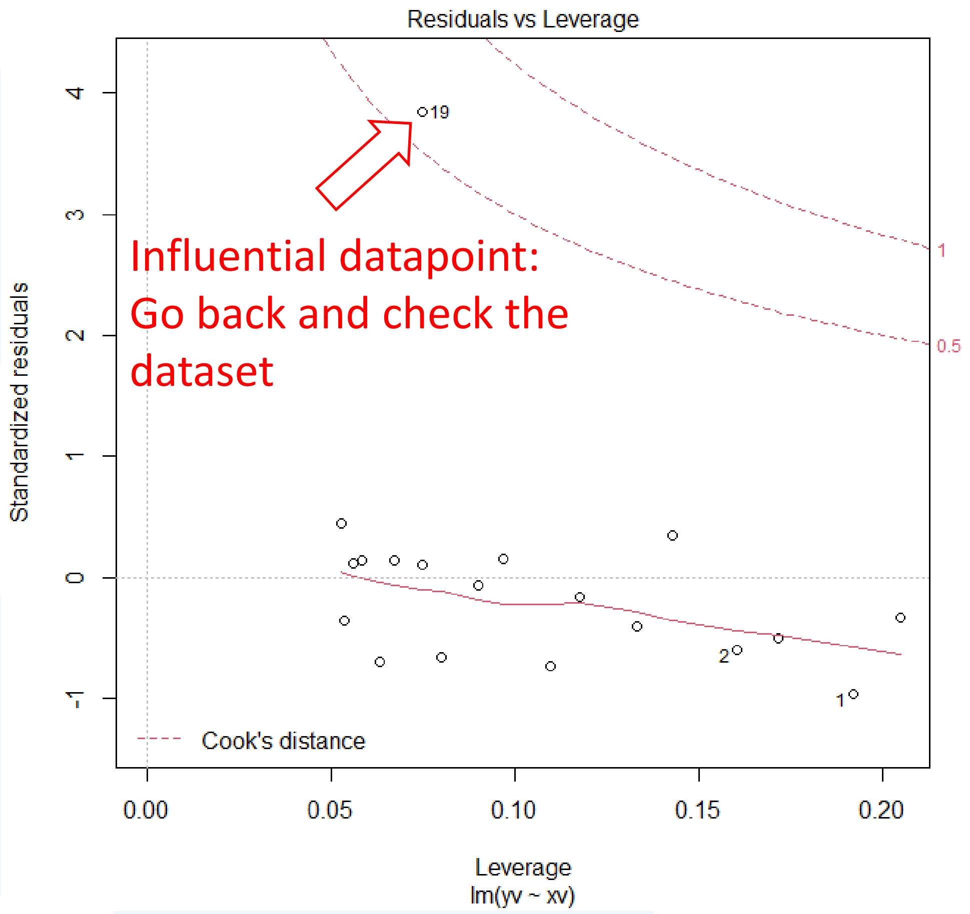
Option 1: Identify and **remove the outlier(s) manually** (typical cut-off value is Cook's Distance > 4/n).

```
which(cooks.distance(mod2)>4/nrow(d2)) #19
d3=d2[-19,] #remove row manually
```

Option 2: Run **Robust Regression** (if you cannot remove the point; decreases weightage of outliers in the regression).

```
#Install MASS package
mod2r=rlm(yv~xv,data=d2)
summary(mod2r) #t-value=9.1505, df=17
#have to calculate p-value manually
2*pt(9.1505,17,lower=F)
```

Diagnostic Plot 4 (Bottom Right)



What if my residuals are not normally distributed?

Option 1: Transform your response variable

Try to apply one of these functions to your y-variable and try again: log(), sqrt(), or cube root “x^(1/3)”

If it works, great! But be careful how you write up your report…

If it doesn’t work …

Option 2: Use a GLM (later lectures)

What if the relationship looks slightly non-linear?

Next section: Polynomial Regression!

Polynomial Regression

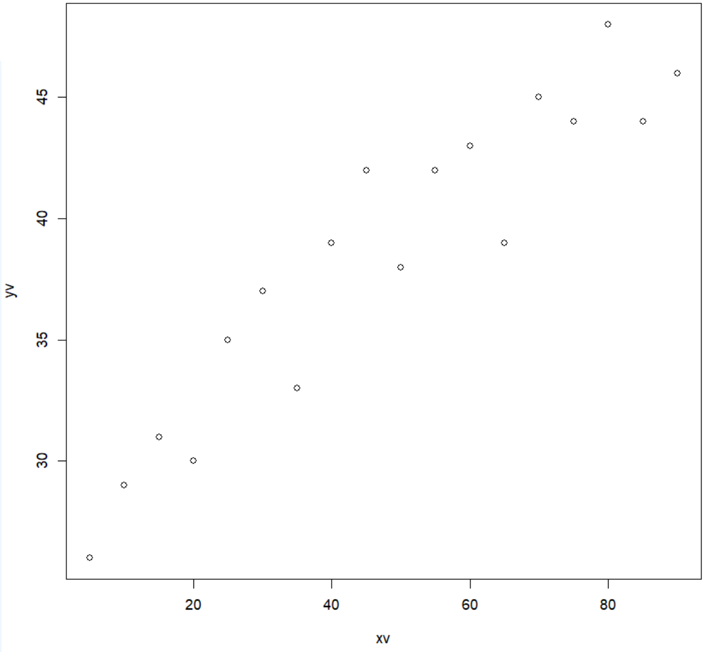
Polynomial but linear!

Used to model very simple curved relationships, e.g. square (second order power). It is rare (and frowned upon by reviewers) to use higher order powers.

Even though we square the explanatory variable, the parameters are still linear, i.e. the form is still a linear equation: $y = a + b(x^2)$.

#Plot <d3> from the previous section

```
plot(yv~xv,data=d3)
#What do you think: linear or curved?
```



#Fit a linear and a quadratic (2^{nd} order) model to compare

```
linMod3=lm(yv~xv,data=d3)
quadMod3=lm(yv~I(xv^2)+xv,data=d3)
```

Also include the first order variable

Add a “^2” to your x-variable, and put an “I( )” around it

Linear vs Quadratic model

#Check both models

```
par(mfrow=c(2,2))
plot(linMod3)
plot(quadMod3)
```

#What do you notice?

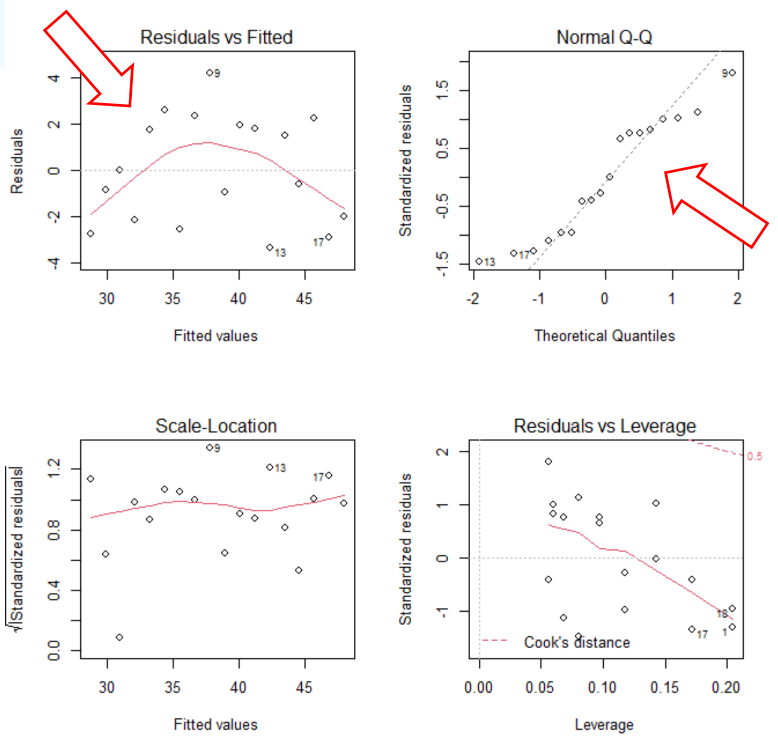
#Get results

```
summary(linMod3)
summary(quadMod3)
```

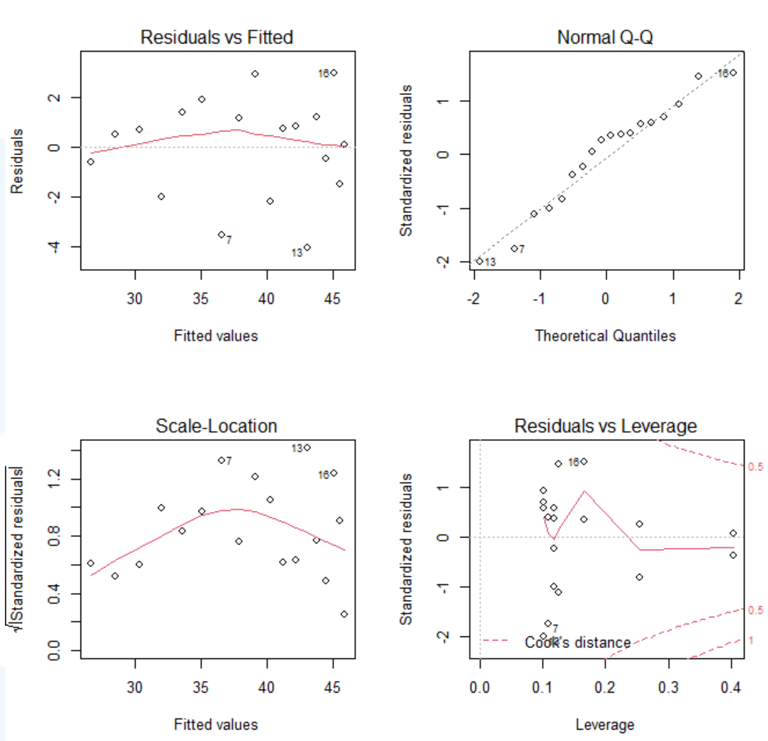
#All coefficients in both models are significant!

#How to decide?

Linear



Quadratic




```
> summary(linMod3)

Call:
lm(formula = yv ~ xv, data = d3)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.3584 -2.1206 -0.3218  1.8763  4.1782 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 27.61438    1.17315   23.54 7.66e-14 ***
xv           0.22683    0.02168   10.46 1.45e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.386 on 16 degrees of freedom
Multiple R-squared:  0.8725,	Adjusted R-squared:  0.8646 
F-statistic: 109.5 on 1 and 16 DF,  p-value: 1.455e-08
```

```
> summary(quadMod3)

Call:
lm(formula = yv ~ I(xv^2) + xv, data = d3)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.0490 -1.2890  0.6018  1.1829  2.9610 

Coefficients:
              Estimate  Std. Error t value Pr(>|t|)    
(Intercept) 24.6323529   1.6916139  14.561 2.95e-10 ***
I(xv^2)     -0.0018834   0.0008386  -2.246 0.040203 *  
xv           0.4057534   0.0819860   4.949 0.000175 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.131 on 15 degrees of freedom
Multiple R-squared:  0.9046,	Adjusted R-squared:  0.8919 
F-statistic: 71.12 on 2 and 15 DF,  p-value: 2.222e-08
```

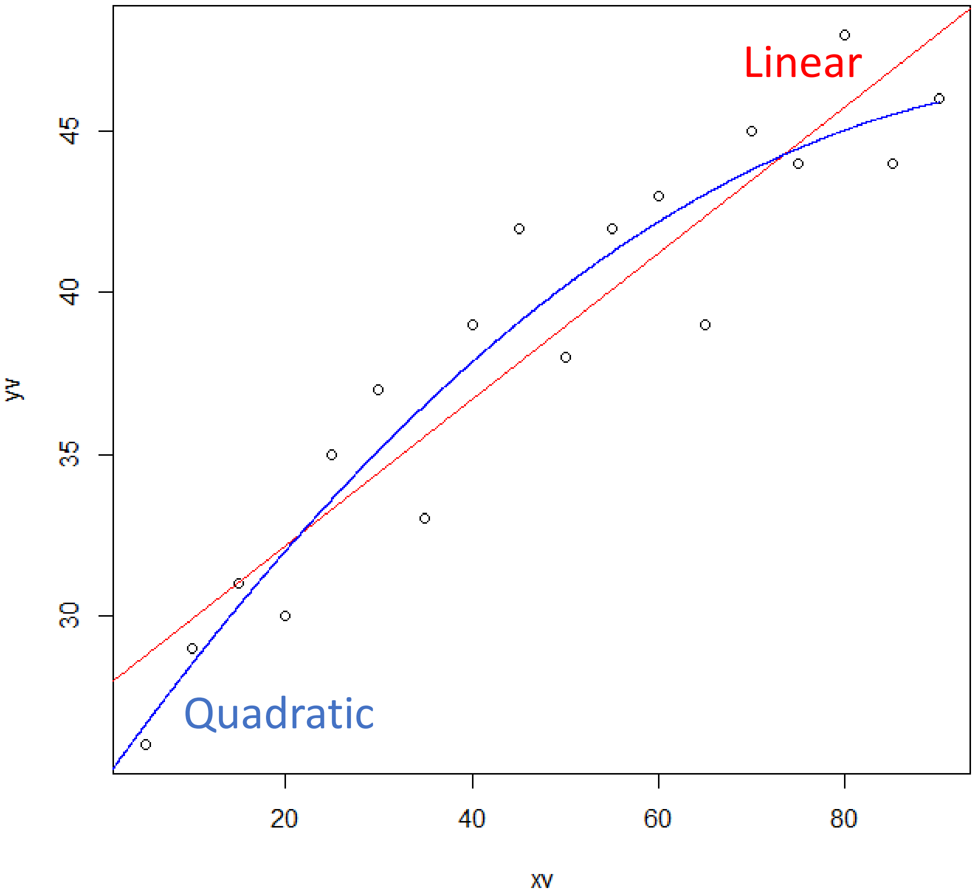
Quadratic model explains more variation (smaller residual) BUT uses up more df

Can compare using Adjusted R-squared

Linear vs Quadratic model

#Also can compare visually

```
plot(yv~xv,data=d3)
abline(lm(yv~xv,data=d3),col="red")
lines(seq(0,90,0.01),predict(quadMod3,
list(xv=seq(0,90,0.01))),col="blue")
```



#Or compare directly using an F test via the anova() function

```
anova(linMod3,quadMod3)
```

```
> anova(linMod3,quadMod3)
Analysis of Variance Table

Model 1: yv ~ xv
Model 2: yv ~ I(xv^2) + xv
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     16 91.057
2     15 68.143  1    22.915 5.0441 0.0402 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The quadratic model has significantly lower RSS (SSE) and is thus better

Note: Based on whether you input one or two models into the anova() function, it is smart enough to know what you want it to do.

Interpreting the model

Can also plot using ggplot:

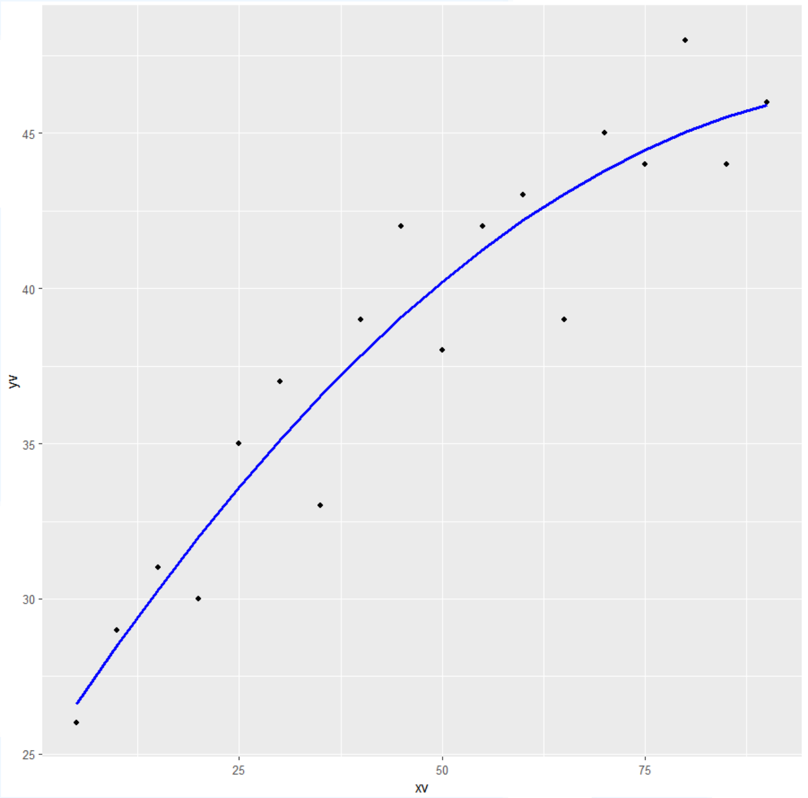
```
d3$y2v=predict(quadMod3)
ggplot(d3,aes(x=xv,y=yv))+geom_point()+
geom_line(aes(x=xv,y=y2v),col="blue",size=1)
```

At xv=0, the slope is 0.41: i.e. a 1 unit increase in xv results in a 0.41 units increase in yv. This increase slows by 0.0019 units for each unit of xv. At xv=1, the slope becomes 0.41-0.0019=0.4081.

"Within the range of 0 to 100 units, an increase in xv results in an increase in yv (p-value < 0.001) which slows by 0.0019 units for each unit of xv (p-value = 0.04)."

In real life don't just say "units", use the unit of the actual variable, e.g. "km", "ppm", "m/s"

If effect size is not important, you can just report the pattern and leave out the numbers: "an increase in yv that tapers off".




```
> summary(quadMod3)

Call:
lm(formula = yv ~ I(xv^2) + xv, data = d3)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.0490 -1.2890  0.6018  1.1829  2.9610 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 24.6323529  1.6916139  14.561 2.95e-10 ***
I(xv^2)     -0.0018834  0.0008386  -2.246 0.040203 *  
xv           0.4057534  0.0819860   4.949 0.000175 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.131 on 15 degrees of freedom
Multiple R-squared:  0.9046,	Adjusted R-squared:  0.8919 
F-statistic: 71.12 on 2 and 15 DF,  p-value: 2.222e-08
```

What if I have more than one explanatory variable?

Multiple regression!!

Multiple Linear Regression

Fitting a multiple regression

We use this when we have one continuous response variable (with normally distributed errors) and >1 continuous explanatory variables (do not have to be normally distributed).

$$y_i = \underbrace{a + b_1 x_{1i} + b_2 x_{2i} + \ldots + b_n x_{ni}}_{\text{Linear predictor}} + \varepsilon_i \qquad \text{, where} \quad \varepsilon_i \sim N(0, \sigma^2)$$

Things to consider:

Assumptions are the same as those under linear regression (earlier slide).

What **variables to include**?: do data exploration/visualisation

Is there any **non-linearity** in any variable?: include this in the model

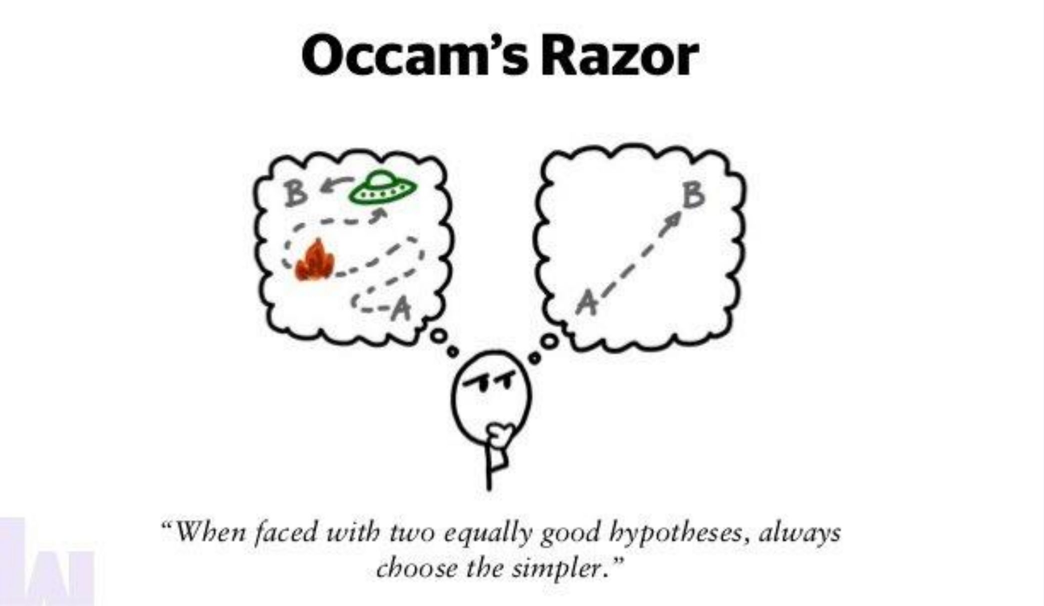
What variables may **interact**?: include this in the model

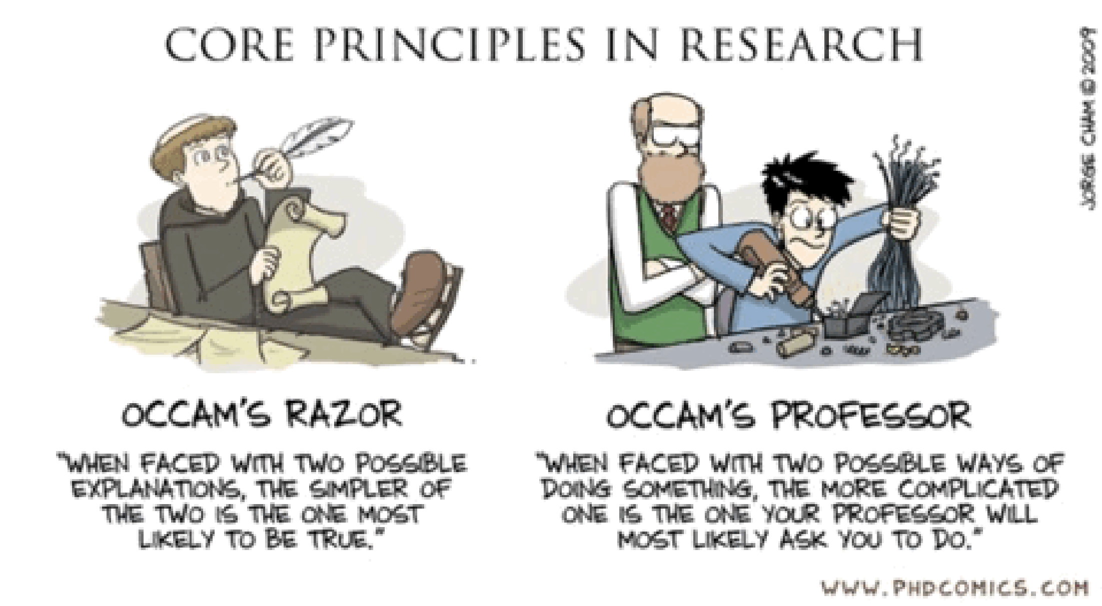
Are any variables **collinear**?: calculate the VIF of your variables

Simplifying and Choosing the right model

We **start with many different variables**, then we try to **remove non-significant variables** (i.e. those that do not help the model explain significantly more variance) and finally **choose the simplest model possible** (Occam's razor).

Source: https://tomasvotruba.com/blog/2020/03/09/art-of-letting-go/





We aim to achieve the **minimum adequate model**.

Simplifying and Choosing the right model

Two approaches:

1) Stepwise deletion approach (Traditional approach and still widely used)**.**

Start with a maximal model (with many variables and interactions, within reason) and simplify to a minimum adequate model (principle of parsimony).

We remove the most complicated elements first, one by one, in the following order:

1) Non-significant interaction terms*.

2) Non-significant quadratic/non-linear terms*.

3) Non-significant explanatory variables*.
(note: if a non-significant variable has a significant interaction, you CANNOT remove it).

*Highest p-value first

2) Information-theoretic approach.

Fit biologically-sound candidate models (based on existing knowledge) and choose the best model or set of models (and average them if more than one model is chosen).

Stepwise deletion approach - data exploration/visualisation

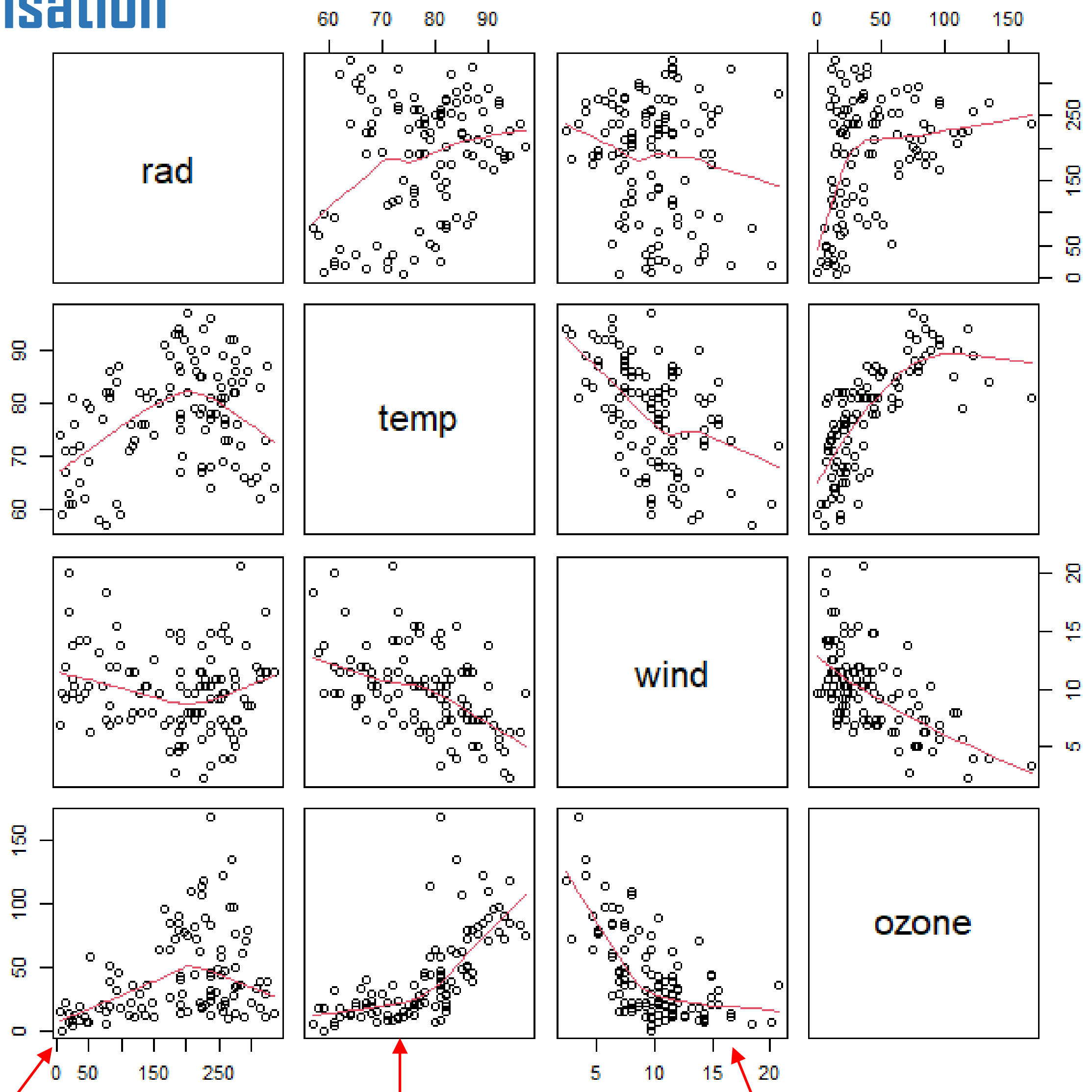
Example using the ozone dataset

We want to look at the effects of solar radiation (rad), temperature (temp) and wind speed (wind) on ozone concentration (ozone).

#Read in the dataset and visualise for relationships

```
d5=read.table("ozone.data.txt",header=T)
pairs(d5,panel=panel.smooth)
```

Adds the red lines in the plot



Unclear, maybe humped (curved) relationship?

Positive correlation, may be curved

Negative correlation, may be curved

Looks like all 3 may affect <ozone> and there may be non-linearity

Stepwise deletion approach - data exploration/visualisation

#Use coplot() to look for interactions between explanatory variables

```
coplot(ozone~wind|rad*temp,data=d5)
```

Main variables

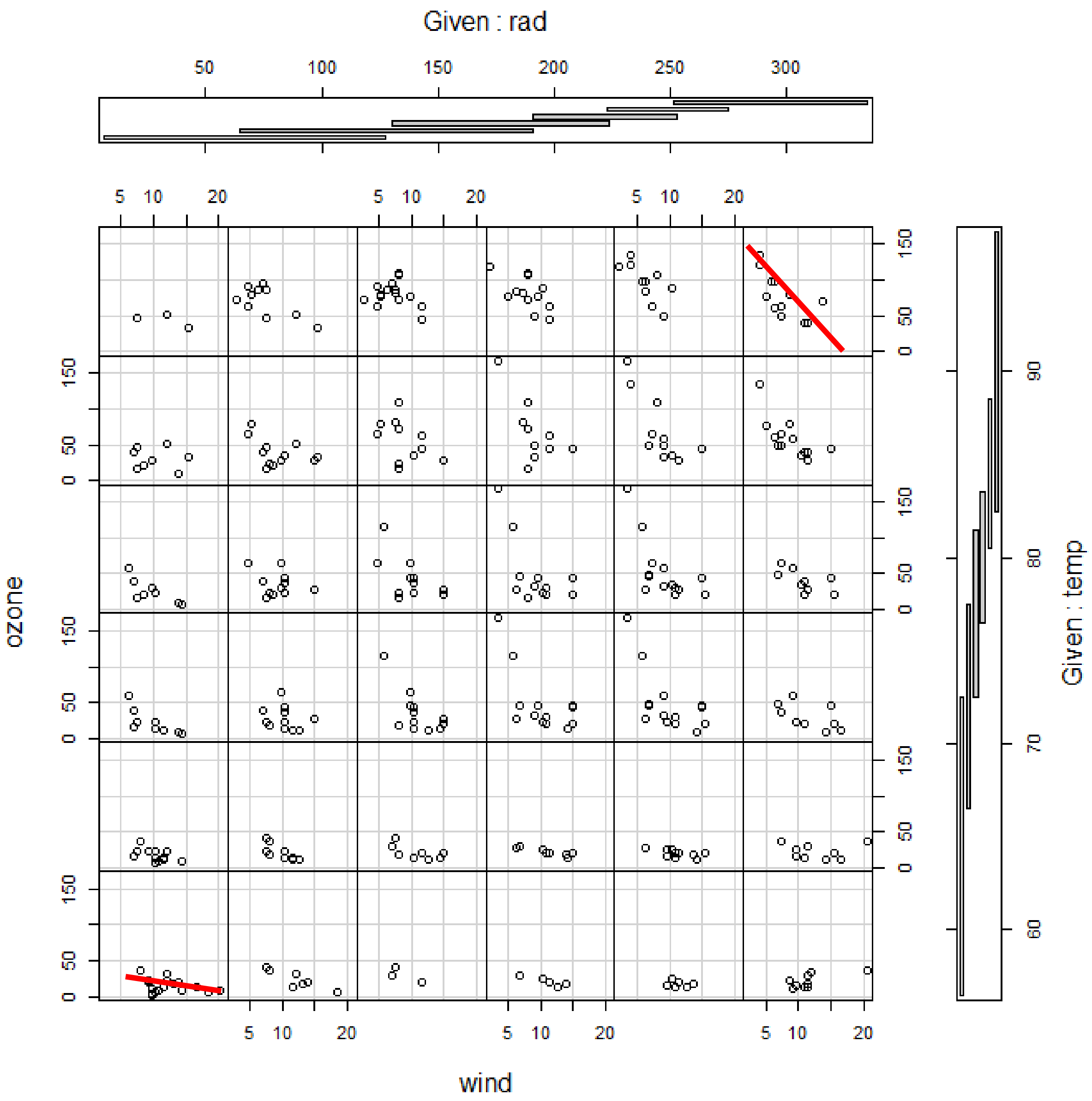
Split into different levels on the horizontal axis

Split into different levels on the vertical axis

There may be an interaction: the relationship between ozone and wind seems to become more negative when both rad and temp increase (see red lines)

#If only splitting one variable on the horizontal axis, add a row=1

```
coplot(ozone~wind|rad,data=d5,row=1)
```



Stepwise deletion approach – fitting the maximal model

Based on our plot, we decide to fit a model testing…

(this is a very complicated model, try not to do this in real life)

a) all the variables,

b) the interactions between all of them, and

c) for possible curvature.

Fitting the model:

```
mod5.1<-lm(ozone~temp*wind*rad+I(rad^2)+I(temp^2)+I(wind^2),data=d5)
```

The first order (i.e. non-quadratic) terms are already included in here

Note: Model formulae

y~x1+x2 (no interaction, only the individual effects of x1 and x2 are tested)

y~x1:x2 (interaction ONLY, the individual effects are not tested)

y~x1*x2 (individual effects AND interaction are tested, i.e. x1+x2+x1:x2)

Stepwise deletion approach – simplification

#Let's inspect the results

```
summary(mod5.1)
```

#Look at the most complicated term first, the 3-way interaction (temp:wind:rad). It is clearly not significant, so we remove it

```
mod5.2=update(mod5.1,~.-temp:wind:rad)
summary(mod5.2)
```

This period means everything in <mod5.1>

Minus only this 3-way interaction (the individual variables are kept)

#Next we look at the 2-way interactions and remove temp:rad first

```
mod5.3=update(mod5.2,~.-temp:rad)
summary(mod5.3)
```

mod5.1

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.683e+02  2.073e+02   2.741  0.00725 **
temp         -1.076e+01  4.303e+00  -2.501  0.01401 *
wind         -3.237e+01  1.173e+01  -2.760  0.00687 **
rad          -3.117e-01  5.585e-01  -0.558  0.57799
I(rad^2)     -3.619e-04  2.573e-04  -1.407  0.16265
I(temp^2)     5.833e-02  2.396e-02   2.435  0.01668 *
I(wind^2)     6.106e-01  1.469e-01   4.157 6.81e-05 ***
temp:wind     2.377e-01  1.367e-01   1.739  0.08519 .
temp:rad      8.403e-03  7.512e-03   1.119  0.26602
wind:rad      2.054e-02  4.892e-02   0.420  0.67552
temp:wind:rad -4.324e-04 6.595e-04  -0.656  0.51358
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

mod5.2

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.245e+02  1.957e+02   2.680   0.0086 **
temp        -1.021e+01  4.209e+00  -2.427   0.0170 *
wind        -2.802e+01  9.645e+00  -2.906   0.0045 **
rad          2.628e-02  2.142e-01   0.123   0.9026
I(rad^2)    -3.388e-04  2.541e-04  -1.333   0.1855
I(temp^2)    5.953e-02  2.382e-02   2.499   0.0141 *
I(wind^2)    6.173e-01  1.461e-01   4.225 5.25e-05 ***
temp:wind    1.734e-01  9.497e-02   1.825   0.0709 .
temp:rad     3.750e-03  2.459e-03   1.525   0.1303
wind:rad    -1.127e-02  6.277e-03  -1.795   0.0756 .
---
```

mod5.3

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.488e+02  1.963e+02   2.796  0.00619 **
temp        -1.144e+01  4.158e+00  -2.752  0.00702 **
wind        -2.876e+01  9.695e+00  -2.967  0.00375 **
rad          3.061e-01  1.113e-01   2.751  0.00704 **
I(rad^2)    -2.690e-04  2.516e-04  -1.069  0.28755
I(temp^2)    7.145e-02  2.265e-02   3.154  0.00211 **
I(wind^2)    6.363e-01  1.465e-01   4.343 3.33e-05 ***
temp:wind    1.840e-01  9.533e-02   1.930  0.05644 .
wind:rad    -1.381e-02  6.090e-03  -2.268  0.02541 *
```

Stepwise deletion approach – simplification

#Then we remove temp:wind (note here that wind:rad looks significant)

```
mod5.4=update(mod5.3,~.-temp:wind)
summary(mod5.4)
```

mod5.4

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.310e+02  1.082e+02   2.135  0.03514 *
temp        -5.442e+00  2.797e+00  -1.946  0.05440 .
wind        -1.080e+01  2.742e+00  -3.938  0.00015 ***
rad          2.405e-01  1.073e-01   2.241  0.02720 *
I(rad^2)    -2.010e-04  2.524e-04  -0.796  0.42770
I(temp^2)    4.484e-02  1.821e-02   2.463  0.01543 *
I(wind^2)    4.308e-01  1.020e-01   4.225 5.16e-05 ***
wind:rad    -9.774e-03  5.794e-03  -1.687  0.09463 .
```

#With temp:wind gone, wind:rad is no longer significant so we remove it too

```
mod5.5=update(mod5.4,~.-wind:rad)
summary(mod5.5)
```

mod5.5

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.985e+02  1.014e+02   2.942  0.00402 **
temp        -6.584e+00  2.738e+00  -2.405  0.01794 *
wind        -1.337e+01  2.300e+00  -5.810 6.89e-08 ***
rad          1.349e-01  8.795e-02   1.533  0.12820
I(rad^2)    -2.052e-04  2.546e-04  -0.806  0.42213
I(temp^2)    5.221e-02  1.783e-02   2.928  0.00419 **
I(wind^2)    4.652e-01  1.008e-01   4.617 1.12e-05 ***
```

#Next, the non-linear terms; remove rad^2

```
mod5.6=update(mod5.5,~.-I(rad^2))
summary(mod5.6)
```

mod5.6

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 291.16758  100.87723   2.886  0.00473 **
temp         -6.33955    2.71627  -2.334  0.02150 *
wind        -13.39674    2.29623  -5.834 6.05e-08 ***
rad           0.06586    0.02005   3.285  0.00139 **
I(temp^2)     0.05102    0.01774   2.876  0.00488 **
I(wind^2)     0.46464    0.10060   4.619 1.10e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

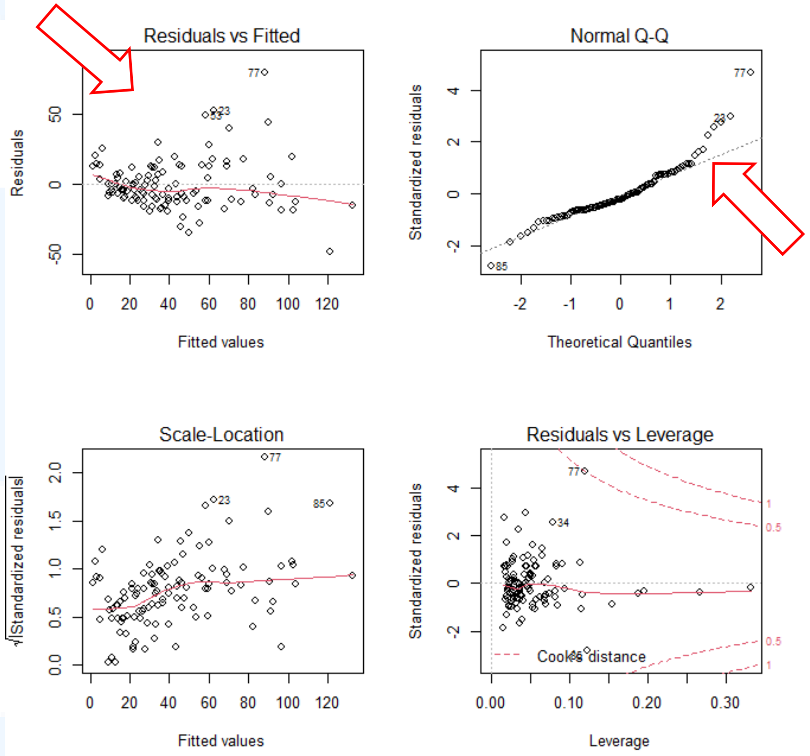
#Everything is significant; nothing can be removed: we have reached our ”minimum adequate model”

Stepwise deletion approach – model checking

#But don't forget to check the model!: Diagnostic plots

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

plot(mod5.6)
```



#Problems with heteroscedasticity and normality, so we cannot trust the results. Let's try a log transform:

```
mod5.7=lm(log(ozone)~temp+wind+rad+I(temp^2)+
I(wind^2),data = d5)

summary(mod5.7)
```

mod5.7

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	2.5538486	2.7359735	0.933	0.35274	
temp	-0.0041416	0.0736703	-0.056	0.95528	
wind	-0.2087025	0.0622778	-3.351	0.00112	**
rad	0.0025617	0.0005437	4.711	7.58e-06	***
I(temp^2)	0.0003313	0.0004811	0.689	0.49255	
I(wind^2)	0.0067378	0.0027284	2.469	0.01514	*

#After the log transform, it looks like temp^2 is no longer significant, so we remove it

Stepwise deletion approach – model checking

#Remove temp^2

```
mod5.8=update(mod5.7,~.-I(temp^2))
summary(mod5.8) #Everything is significant
```

mod5.8

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.7231644  0.6457316   1.120  0.26528
temp         0.0464240  0.0059918   7.748 5.94e-12 ***
wind        -0.2203843  0.0597744  -3.687  0.00036 ***
rad          0.0025295  0.0005404   4.681 8.49e-06 ***
I(wind^2)    0.0072233  0.0026292   2.747  0.00706 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

mod5.8 explains less variation, but not significantly less, so we therefore prefer the simpler model: mod5.8.

#Compare the last 2 models

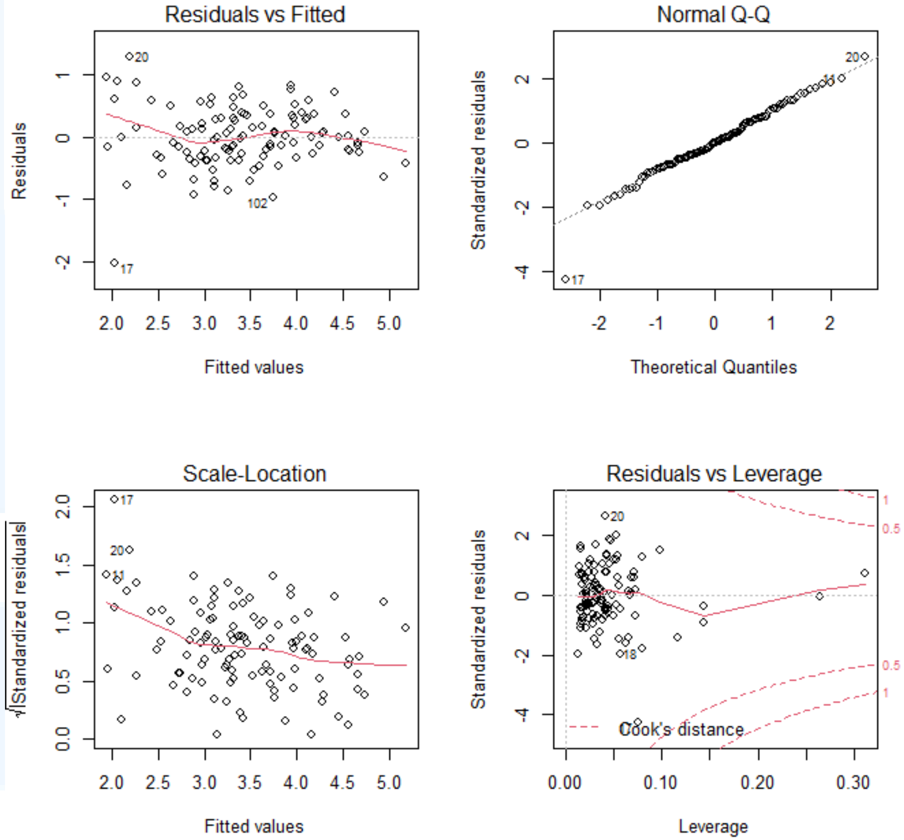
```
anova(mod5.7,mod5.8)
```

```
> anova(mod5.7,mod5.8)
Analysis of Variance Table

Model 1: log(ozone) ~ temp + wind + rad + I(temp^2) + I(wind^2)
Model 2: log(ozone) ~ temp + wind + rad + I(wind^2)
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    105 25.712
2    106 25.828 -1  -0.11614 0.4743 0.4925
```

#This is a minimum adequate model, let's check the assumptions again

```
par(mfrow=c(2,2))
plot(mod5.8)
```



#Scedasticity and normality look better but still a bit iffy: but notice that in both cases

Answer is the problem!

Stepwise deletion approach – model checking

#Remove row 17 and refit the model

```
d5=d5[-17,]
mod5.9=update(mod5.8,~.)
plot(mod5.9)
```

No minus term: we don't want to remove anything, we just want to refit the model with the updated dataset

#Everything looks great!

```
summary(mod5.9)
```

We can now interpret these results:

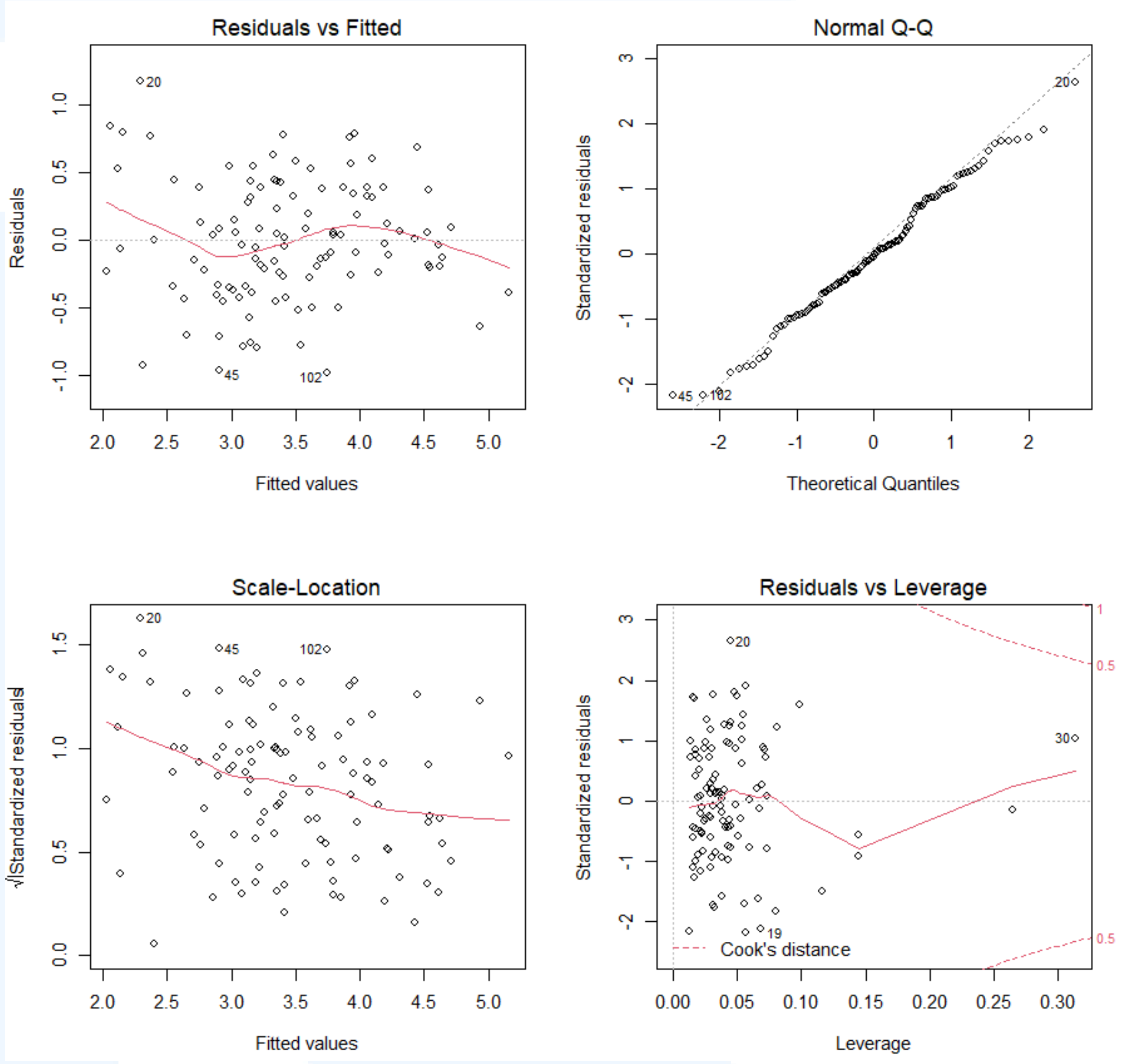
mod5.9

```
Call:
lm(formula = log(ozone) ~ temp + wind + rad + I(wind^2), data = d5)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.97682 -0.27335 -0.01435  0.36283  1.16883 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.1932358  0.5990022   1.992 0.048963 *  
temp         0.0419157  0.0055635   7.534 1.81e-11 ***
wind        -0.2208189  0.0546589  -4.040 0.000102 ***
rad          0.0022097  0.0004989   4.429 2.33e-05 ***
I(wind^2)    0.0068982  0.0024052   2.868 0.004993 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4514 on 105 degrees of freedom
Multiple R-squared:  0.6974,	Adjusted R-squared:  0.6859 
F-statistic:  60.5 on 4 and 105 DF,  p-value: < 2.2e-16
```



Stepwise deletion approach – interpreting results

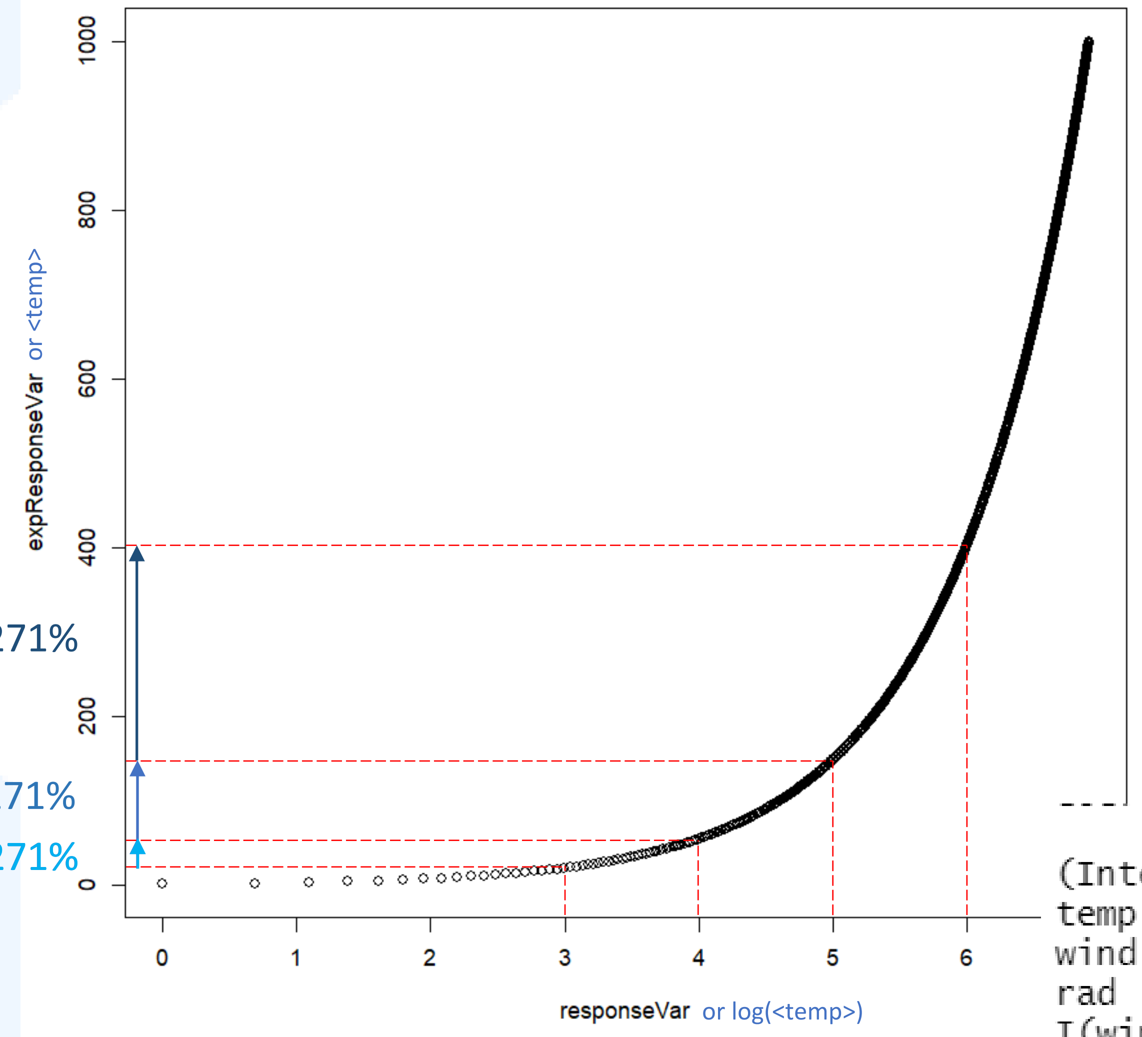
General interpretation:

A change in the log-transformed variable represents a PERCENTAGE CHANGE in the original variable: we call this kind of relationship “exponential”.

exp(6) = exp(5) x 271%

exp(5) = exp(4) x 271%

exp(4) = exp(3) x 271%



	Estimate	Std. Error
(Intercept)	1.1932358	0.5990022
temp	0.0419157	0.0055635
wind	-0.2208189	0.0546589
rad	0.0022097	0.0004989
I(wind^2)	0.0068982	0.0024052

“Ozone levels increase exponentially with temperature and radiance, and decrease exponentially with wind speed.”

Stepwise deletion approach – interpreting results

Reporting effect sizes of specific variables:

We have to calculate back from these values. The opposite of log() is exp():

Example with <temp>:

`exp(0.0419)=1.042`
`1.042-1=0.042`

"There was a 4.2% increase in ozone levels per unit increase in temperature."

Example with <wind>:

`exp(-0.221)-1=-0.197`

"There was a 19.7% decrease in ozone levels per unit increase in wind."

For <wind>^{2}: The decrease tapers off. "The negative relationship with wind tapers off at higher wind speeds."

mod5.9

```
Call:
lm(formula = log(ozone) ~ temp + wind + rad + I(wind^2), data = d5)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.97682 -0.27335 -0.01435  0.36283  1.16883 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.1932358  0.5990022   1.992 0.048963 *  
temp         0.0419157  0.0055635   7.534 1.81e-11 ***
wind        -0.2208189  0.0546589  -4.040 0.000102 ***
rad          0.0022097  0.0004989   4.429 2.33e-05 ***
I(wind^2)    0.0068982  0.0024052   2.868 0.004993 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4514 on 105 degrees of freedom
Multiple R-squared:  0.6974,	Adjusted R-squared:  0.6859 
F-statistic:  60.5 on 4 and 105 DF,  p-value: < 2.2e-16
```

How to interpret log-transformed variables

For more info, this explains it very well: https://data.library.virginia.edu/interpreting-log-transformations-in-a-linear-model/

Case A—only the response variable is log-transformed

Step 1: Take the exponential, exp(), of the coefficient and minus 1

Step 2: This is the proportional change in the original response variable for every unit change in the explanatory variable

Case B—only the explanatory variable is log-transformed

Step 1: Divide the coefficient by 100

Step 2: A 1% change in the explanatory variable changes the response variable by this amount

Case C—both variables are log-transformed

Step 1: The coefficient is the percentage change in the response variable for every 1% increase in the explanatory variable

Stepwise deletion approach – a small shortcut

#Can use step() to automate the first few steps of the deletion process

```
mod5.1s=step(mod5.1)

summary(mod5.1s)
```

mod5.1s

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 514.401470 193.783580   2.655  0.00920 ** 
temp        -10.654041   4.094889  -2.602  0.01064 *  
wind        -27.391965   9.616998  -2.848  0.00531 ** 
rad           0.212945   0.069283   3.074  0.00271 ** 
I(temp^2)     0.067805   0.022408   3.026  0.00313 ** 
I(wind^2)     0.619396   0.145773   4.249 4.72e-05 ***
temp:wind     0.169674   0.094458   1.796  0.07538 .  
wind:rad     -0.013561   0.006089  -2.227  0.02813 *  
```

Note: step() has removed all the clearly non-significant variable but kept some marginal ones. It is more lenient than us (which is what we want from an automated procedure). We will then have to continue manually from here.

Stepwise deletion approach – drawbacks

Potential bias in parameter estimates, especially for variables that are close to significant (to remove or not to remove?)

Potential inconsistencies with model selection algorithms when using automated functions, leading to different results (e.g. different order of parameter deletion)

Multiple hypothesis testing (do we need to correct for it?)

Over-reliance on a single best model (many models may fit the data nearly as well and this uncertainty is not represented).

Potential over-fitting if you start with high-order interaction terms.

Alternative: Information-theoretic approach!

Information-theoretic approach – fitting candidate models

#We first select models that **make sense biologically** (e.g. based on existing knowledge). Start simple (from the null model) and slowly increase the complexity of the models but base them on theory as much as possible

```
mod5.11=lm(log(ozone)~1,data=d5) #the Null model
mod5.12=lm(log(ozone)~temp,data=d5)
mod5.13=lm(log(ozone)~wind,data=d5)
mod5.14=lm(log(ozone)~rad,data=d5)
mod5.15=lm(log(ozone)~temp+wind,data=d5)
mod5.16=lm(log(ozone)~wind+rad,data=d5)
mod5.17=lm(log(ozone)~temp+rad,data=d5)
mod5.18=lm(log(ozone)~temp+wind+rad,data=d5)
mod5.19=lm(log(ozone)~temp+wind+rad+I(temp^2),data=d5)
mod5.110=lm(log(ozone)~temp+wind+rad+I(wind^2),data=d5)
mod5.111=lm(log(ozone)~temp+wind+rad+I(temp^2)+I(wind^2),data=d5)
```

Information-theoretic approach – choosing the best

We then **rank the models** using a Model Accuracy Metric which measures how well a model can predict data: a **LOWER VALUE IS BETTER**.

AIC penalises a model for having more predictive variables. Most common.

AICc is a version of the AIC for small sample sizes (n/K < 40; where n is sample size and K is number of parameters).

BIC (Bayesian Information Criterion) has a stronger penalty.

IMPORTANT: you can only use these to compare models when one is a subset of the other.

#We can use the “MuMIn” package to help us do this more quickly

```
require(MuMIn)
```

#Create the model.selection object – AIC, AICc or BIC will work

```
modsAIC=model.sel(mod5.11,mod5.12,mod5.13,mod5.14,mod5.15,mod5.16,mod5.17,mod
5.18,mod5.19,mod5.110,mod5.111,rank="AIC")
```

Information-theoretic approach – averaging

#View the results

```
modsAIC
```

#If there is one model that is clearly the best then use that model

		family	df	logLik	AIC	delta	weight
BEST	mod5.110	gaussian(identity)	6	-76.580	165.2	0.00	0.630
	mod5.111	gaussian(identity)	7	-76.330	166.7	1.50	0.297
	mod5.18	gaussian(identity)	5	-80.397	170.8	5.64	0.038
	mod5.19	gaussian(identity)	6	-79.463	170.9	5.77	0.035
	mod5.17	gaussian(identity)	4	-87.853	183.7	18.55	0.000
	mod5.15	gaussian(identity)	4	-90.086	188.2	23.01	0.000
	mod5.12	gaussian(identity)	3	-96.106	198.2	33.05	0.000
	mod5.16	gaussian(identity)	4	-106.821	221.6	56.48	0.000
	mod5.13	gaussian(identity)	3	-120.497	247.0	81.83	0.000
	mod5.14	gaussian(identity)	3	-128.069	262.1	96.98	0.000
WORST	mod5.11	gaussian(identity)	2	-141.013	286.0	120.87	0.000

Models ranked by AIC(x)

#Here, we have 2 models that are very close ($\Delta AIC < 2$ is usually used as a cut-off value) so we average them; the package will use their AIC scores as weights

```
modsAvg=model.avg(modsAIC,subset=delta<2)
```

#View the results

```
summary(modsAvg)
```

Information-theoretic approach vs. Stepwise deletion approach

The results are very similar.

Information-theoretic (modsAvg)

```
Model-averaged coefficients:
(full average)
             Estimate Std. Error Adjusted SE z value Pr(>|z|)
(Intercept)  1.3105344  1.8480233   1.8649659   0.703 0.482235
temp         0.0302002  0.0481962   0.0486239   0.621 0.534535
wind        -0.2166362  0.0608338   0.0615329   3.521 0.000430 ***
rad          0.0025398  0.0005417   0.0005479   4.635  3.6e-06 ***
I(wind^2)    0.0070675  0.0026711   0.0027018   2.616 0.008900 **
I(temp^2)    0.0001063  0.0003134   0.0003161   0.336 0.736655

(conditional average)
             Estimate Std. Error Adjusted SE z value Pr(>|z|)
(Intercept)  1.3105344  1.8480233   1.8649659   0.703 0.482235
temp         0.0302002  0.0481962   0.0486239   0.621 0.534535
wind        -0.2166362  0.0608338   0.0615329   3.521 0.000430 ***
rad          0.0025398  0.0005417   0.0005479   4.635  3.6e-06 ***
I(wind^2)    0.0070675  0.0026711   0.0027018   2.616 0.008900 **
I(temp^2)    0.0003313  0.0004811   0.0004867   0.681 0.496040
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Stepwise deletion (mod5.9)

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.1932358  0.5990022   1.992 0.048963 *
temp         0.0419157  0.0055635   7.534 1.81e-11 ***
wind        -0.2208189  0.0546589  -4.040 0.000102 ***
rad          0.0022097  0.0004989   4.429 2.33e-05 ***
I(wind^2)    0.0068982  0.0024052   2.868 0.004993 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note: Full vs. Conditional Average (Information-theoretic approach)

Full average will input a value of zero for parameters that were not selected in each of the models being averaged. I prefer this.

Conditional will not input any value. Conditional averaging introduces a bias (lower p-value) towards less selected variables.

With the Information-theoretic approach, technically you do not need to simplify further as this has already been taken into account by AIC. However, it's up to you to make the final decision based on biological intuition.

Multicollinearity – Variance Inflation Factor

Multicollinearity is when two or more explanatory variables are highly correlated.

- This will lead to unstable estimation of model parameters

- Happens quite frequently (e.g. when you measure height and weight) and we sometimes create it ourselves (e.g. creating a new variable by adding 2 existing variables).

To test for it, we measure the **Variance Inflation Factor (VIF)**: if a variable has a **VIF value > 3, we remove it**.

#Install the "car" package and fit a model with all your explanatory variables modelled individually (i.e. no interactions)

```
modTest=lm(ozone~temp+wind+rad,data=d5)
vif(modTest) #in this case, everything is OK
```

```
> vif(modTest)
    temp     wind      rad 
1.426701 1.346967 1.077688 
```

If there are variables with VIF > 3, update the model by removing the variable with the highest VIF—either manually fit a new model or use update()—then test this new model using vif(). Repeat this until no more variables have VIF > 3.

What results to present?

A good standard to follow, report at least these 3 things…

1) **Sample size** (N or n; but usage varies!!)—we usually report this at the start of the Results.

Usage 1: N is population size, n is sample size.

Usage 2: N is total sample size, n is group/level size (this is more useful for ecology).

2) **p-value** or *P* (note the italics):

If $P \geq 0.06$: report value to 2 decimals (e.g. $P = 0.82$, $P = 0.07$).

Reported as the analysis results

If $0.001 \leq P < 0.06$: report value to 2 significant figures (e.g. $P = 0.0013$, 0.053).

If $P < 0.001$: report "$P < 0.001$".

3) **Effect size ± confidence intervals**:

Mean ± SE

Slope ± 95% Confidence Intervals

Example: "In our experiment (N = 92), the treatment group had noses which were longer than the control group by 2 ± 0.56 cm (mean ± standard error) ($P = 0.0043$)."

Note: Different journals/companies have different practices.

E.g. Nature also wants the statistic used to calculate the P-value (e.g. t-statistic, F-statistics).

Summary (Learning Objectives)

Advanced analyses: when to use and Decision tree

Regression

- What is it?
- Important concepts: Maximum Likelihood, slope (b), coefficient of determination (r^2)
- Types of Regression:
    - Linear (OLS) Regression: Assumptions, Power analysis, Fit, Check, Predict
    - Robust Regression
    - Polynomial Regression
    - Multiple Linear Regression: Model simplification, Model comparison, Multicollinearity