STATISTICS: AN INTRODUCTION USING R

By M.J. Crawley

Exercises

4. REGRESSION

Regression is the statistical model we use when the explanatory variable is continuous. If the explanatory variables were categorical we would use Analysis of Variance (Exercises 5). If you are in any doubt about whether to use regression or analysis of variance, ask yourself whether your graphical investigation of the data involved producing scatter plots or bar charts. If you produced scatter plots, then the statistics you need to use are regression. If you produced bar charts, then you need Analysis of Variance.

To understand how the statistical part of R works, we shall work through a series of simple examples in sufficient detail that the calculations carried out within R become apparent. It is most important when learning how to use a new statistical package to understand exactly what the output means. What the numbers are, and where they come from. More experienced readers can skip this introductory material.

In all the examples that follow, we make the following assumptions:

- errors are normally distributed
- variances are constant
- the explanatory variable is measured without error
- all of the unexplained variation is confined to the response variable.

Later on, we shall deal with cases that have non-normal errors and unequal variances.

The Model

We begin with the simplest possible linear model; the straight line

$$y = a + bx$$

where a **response variable** y is hypothesised as being a linear function of the **explanatory variable** x, and the two **parameters** a and b. In the case of simple linear regression, the parameter a is called the *intercept* (the value of y when x = 0), and b is the *slope* of the line (or the gradient, measured as the change in y in response to unit change in x). The aims of the analysis are as follows:

- to estimate the values of the parameters a and b
- to estimate their standard errors
- to use the standard errors to assess which terms are necessary within the model (i.e. whether the parameter values are significantly different from zero)

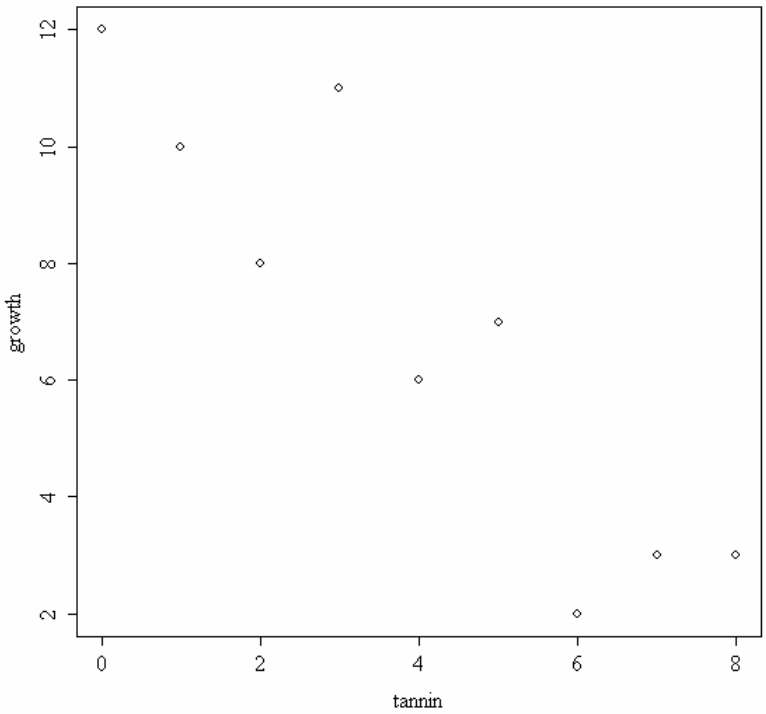
- to determine what fraction of the variation in *y* is explained by the model and how much remains unexplained

Data inspection.

The first step is to look carefully at the data. Is there an upward or downward trend, or could a horizontal straight line be fit through the data? If there is a trend, does it look linear or curvilinear? Is the scatter of the data around the line more or less uniform, or does the scatter change systematically as x changes?

Consider the data in the data frame called tannin. They show how weight gain (mg) of individual caterpillars declines as the tannin content of their diet (%) increases.

```
regression<-read.table("c:\\temp\\regression.txt",header=T)
attach(regression)
names(regression)
[1] "growth" "tannin"
plot(tannin,growth)
```



It looks as if there is a downward trend in *y* as *x* increases, and that the trend is roughly linear. There is no evidence of any systematic change in the scatter as *x* changes. This cursory inspection leads to several expectations:

- the intercept *a* is greater than zero
- the slope *b* is negative
- the variance in *y* is constant

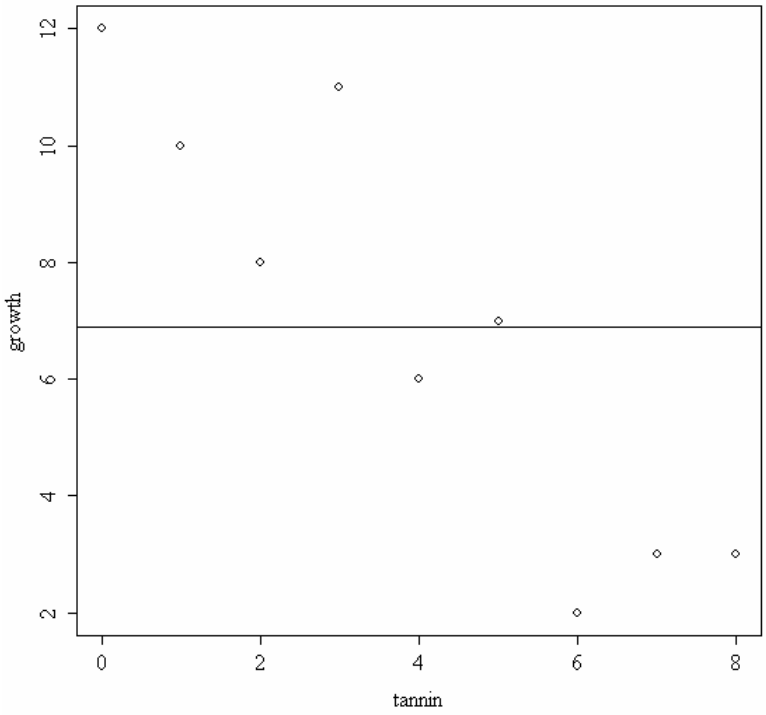
- the scatter about the straight line is relatively slight

It now remains to carry out a thorough statistical analysis to substantiate or refute these initial impressions, and to provide accurate estimates of the slope and intercept, their standard errors and the degree of fit.

Least squares

The technique of least squares linear regression defines the *best fit* straight line as the line which minimises *the sum of the squares of the departures of the y values from the line*. We can see what this means in graphical terms. The first step is to fit a horizontal line though the data, using **abline**, showing the average value of *y* (the first parameter is the intercept, the second is the slope):

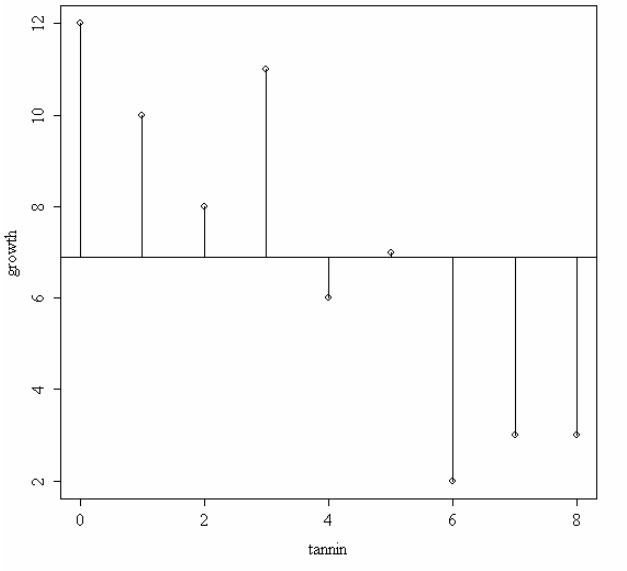
```
mean(growth)
[1] 6.888889
abline(6.889,0)
```



The scatter around the line defined by $\bar{y}$ is said to be the total variation in y. Each point on the graph lies a vertical distance $d = y - \bar{y}$ from the horizontal line, and we define the total variation in y as being the sum of the squares of these departures:

$$SST = \sum \left(y - \bar{y} \right)^2$$

where SST stands for *total sum of squares*. It is the sum of the squares of the vertical distances shown here:

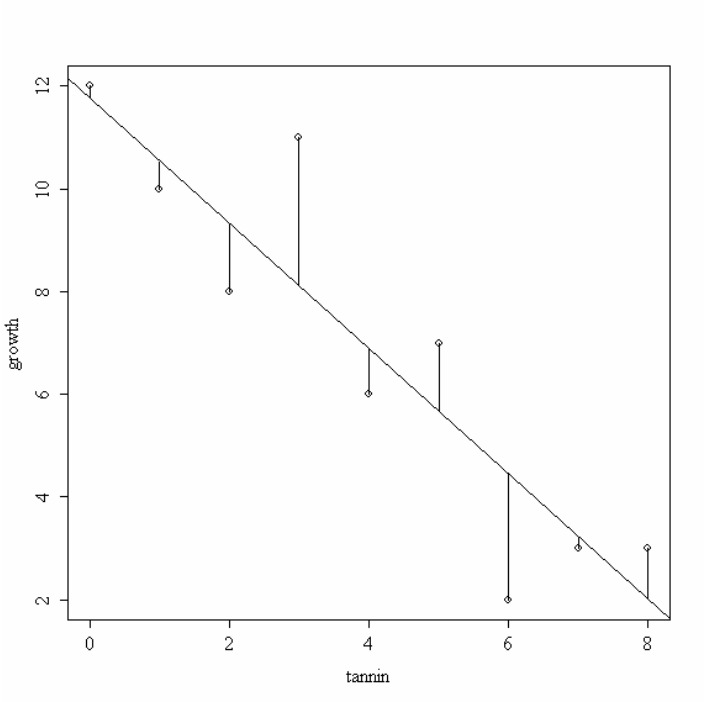


Now we want to fit a straight line through the data. There are two decisions to be made about such a best fit line:

- where should the line be located?
- what slope should it have?

Location of the line is straightforward, because a best fit line should clearly pass through the point defined by the average values of x and y . The line can then be pivoted at the point $(\bar{x}, \bar{y})$, and rotated until the best fit is achieved. The process is formalised as follows. Each point on the graph lies a distance $e = y - \hat{y}$ from the fitted line, where the predicted value $\hat{y}$ is found by evaluating the equation of the straight line at the appropriate value of x:

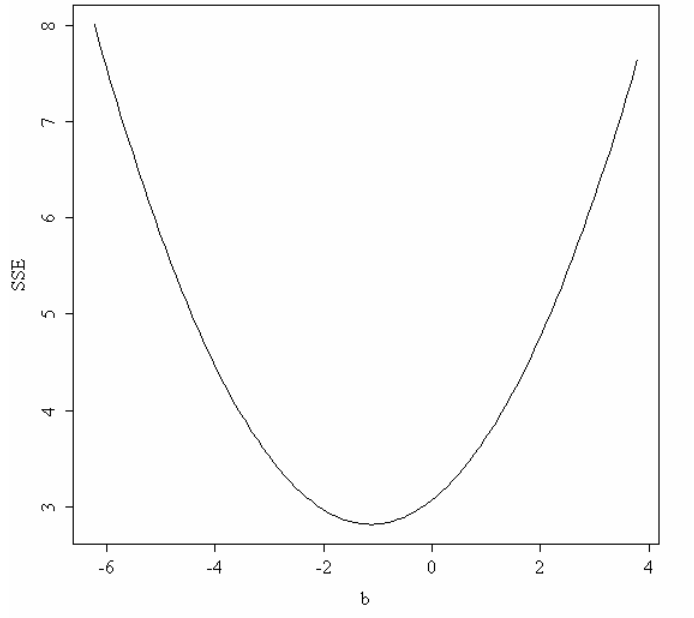
$$\hat{y} = a + bx$$



Let us define the error sum of squares as the sum of the squares of the e's:

$$SSE = \sum (y - \hat{y})^2$$

Now imagine rotating the line around the point $(\bar{x}, \bar{y})$. SSE will be large when the line is too steep. It will decline as the slope gets closer to the best-fit line, then it will increase again as the line becomes too shallow. We can draw a graph of SSE against the slope b, like this:



We define *the best fit line as the one that minimises SSE*. The maximum likelihood estimate of the slope is obviously somewhere around –1.2. To find this value analytically, we find the derivative of SSE with respect to *b*, set it to zero, and solve for b.

Maximum likelihood estimate of the slope

The location of the line is fixed by assuming that it passes through the point $(\bar{x}.\bar{y})$, so that we can rearrange the equation to obtain an estimate of the intercept *a* in terms of the best fit slope, *b*.

$$a = \bar{y} - b\bar{x}$$

We begin by replacing the average values of y and x by $\sum y / n$ and $\sum x / n$ so

$$a = \frac{\sum y}{n} - b\frac{\sum x}{n}$$

The *best fit* slope is found by rotating the line until the *error sum of squares*, SSE, is minimised. The error sum of squares is the sum of squares of the individual departures, $e = y - \hat{y}$, shown above:

$$SSE = \text{ minimum } \sum (y - a - bx)^2$$

noting the change in sign of *bx*. This is how the best fit is defined.

Next, we find the derivative of SSE with respect to *b*

$$\frac{dSSE}{db} = -2\sum x(y - a - bx)$$

because the derivative with respect to b of the bracketed term is -*x*, and the derivative of the squared term is 2 times the squared term. The constant -2 can be taken outside the summation. Now, multiplying through the bracketed term by *x* gives

$$\frac{dSSE}{db} = -2\sum xy - ax - bx^2$$

Now take the summation of each term separately, set the derivative to zero, and divide both sides by -2 to remove the unnecessary constant:

$$\sum xy - \sum ax - \sum bx^2 = 0$$

We can not solve the equation as it stands because there are two unknowns, *a* and *b*. However, we already know the value of a in terms of b. Also, note that $\sum ax$ can be

written as $a\sum x$, so, replacing *a* and taking both *a* and *b* outside their summations gives:

$$\sum xy - \left[\frac{\sum y}{n} - b\frac{\sum x}{n}\right]\sum x - b\sum x^2 = 0$$

Now multiply out the central bracketed term by $\sum x$ to get

$$\sum xy - \frac{\sum x \sum y}{n} + b\frac{(\sum x)^2}{n} - b\sum x^2 = 0$$

Finally, take the 2 terms containing *b* to the other side, and note their change of sign:

$$\sum xy - \frac{\sum x \sum y}{n} = b\sum x^2 - b\frac{(\sum x)^2}{n}$$

and then divide both sides by $\sum x^2 - (\sum x)^2 / n$ to obtain the required estimate b:

$$b = \frac{\sum xy - \frac{\sum x \sum y}{n}}{\sum x^2 - \frac{(\sum x)^2}{n}}$$

Thus, the value of b that minimises the sum of squares of the departures is given simply by

$$b = \frac{SSXY}{SSX}$$

where SSXY stands for the corrected sum of products (*x* times *y*; the measure of how *x* and y co-vary), and SSX is the corrected sum of squares for x, calculated in exactly the same manner as the total sum of squares SST, which we met earlier.
For comparison, these 3 important formulas are presented together:

$$SST = \sum y^2 - \frac{(\sum y)^2}{n}$$

$$SSX = \sum x^2 - \frac{(\sum x)^2}{n}$$

$$SSXY = \sum xy - \frac{\sum x \sum y}{n}$$

Note the similarity of their structures. SST is calculated by adding up *y* times *y* then subtracting the total of the *y*'s times the total of the *y*'s divided by *n* (the number of points on the graph). Likewise, SSX is calculated by adding up *x* times *x* then subtracting the total of the *x*'s times the total of the *x*'s divided by *n*. Finally, SSXY is calculated by adding up *x* times *y* then subtracting the total of the *x*'s times the total of the *y*'s divided by *n*.

It is worth reiterating what these three quantities represent. SST measures the total variation in the y values about their mean (it is the sum of the squares of the *d*'s in the earlier plot). SSX represents the total variation in x (expressed as the sum of squares of the departures from the mean value of x), and is a measure of the range of x values over which the graph has been constructed. SSXY measures the correlation between *y* and *x* in terms of the corrected sum of products. Note that SSXY is negative when *y* declines with increasing *x*, positive when *y* increases with *x*, and zero when *y* and *x* are uncorrelated.

Analysis of variance in regression

The idea is to partition the total variation in the response, SST, into 2 components: the component that is explained by the regression line (which we shall call SSR, the regression sum of squares) and an unexplained component (the residual variation which we shall call SSE, the error sum of squares).

Look at the relative sizes of the departures d and e in the figures we drew earlier. Now, ask yourself what would be the relative size of $\sum d^2$ and $\sum e^2$ if the slope of the fitted line were *not significantly different from zero*? A moment's thought should convince you that if the slope of the best fit line were zero, then the two sums of squares would be exactly the same. If the slope were zero, then the two lines would lie in exactly the same place, and the sums of squares would be identical. Thus, when the slope of the line is not significantly different from zero, we would find that SST = SSE. Similarly, if the slope was significantly different from zero (i.e. significantly positive or negative), then SSE would be substantially *less* than SST. In the limit, if all the points fell exactly on the fitted line, then SSE would be zero.

Now we calculate a third quantity, SSR, called the *regression sum of squares*:

$$SSR = SST - SSE$$

This definition means that SSR will be large when the fitted line accounts for much of the variation in *y*, and small when there is little or no linear trend in the data. In the limit, SSR would be equal to SST if the fit was perfect (because SSE would then equal zero), and SSR would equal zero if y was independent of x (because, in this case, SSE would equal SST).

These three quantities form the basis for drawing up the ANOVA table:

Source	SS	df	MS	F	F tables (5%)
Regression	SSR	1	SSR	$F = \frac{SSR}{s^2}$	qf(0.95,1,n-2)
Error	SSE	n-2	$s^2 = \frac{SSE}{n-2}$		
Total	SST	n-1			

The sums of squares are entered in the first column. SST is the corrected sum of squares of *y*, as given above. We have defined SSE already, but this formula is inconvenient for calculation, since it involves n different estimates of $\hat{y}$ and *n* subtractions. Instead, it is easier to calculate SSR and then to estimate SSE by difference. The regression sum of squares is simply:

$$SSR = b.SSXY$$

so that

$$SSE = SST - SSR$$

The second column of the anova table contains the degrees of freedom. The estimation of the total sum of squares required that one parameter $\bar{y}$, the mean value of *y*, be estimated from the data prior to calculation. Thus, the total sum of squares has *n*-1 degrees of freedom when there are *n* points on the graph. The error sum of squares could not be estimated until the regression line had been drawn through the data. This required that two parameters, the mean value of *y* and the slope of the line were estimated from the data. Thus, the error sum of squares has *n*-2 degrees of freedom. The regression degrees of freedom represents *the number of extra parameters* involved in going from the null model $y = \bar{y}$ to the full model $y = a + bx$. (i.e. 2 –1 = 1 degree of freedom) because we have added the slope, *b* to the model. As always, the component d.f. add up to the total d.f.

At this stage it is worth recalling that variance is ***always*** calculated as

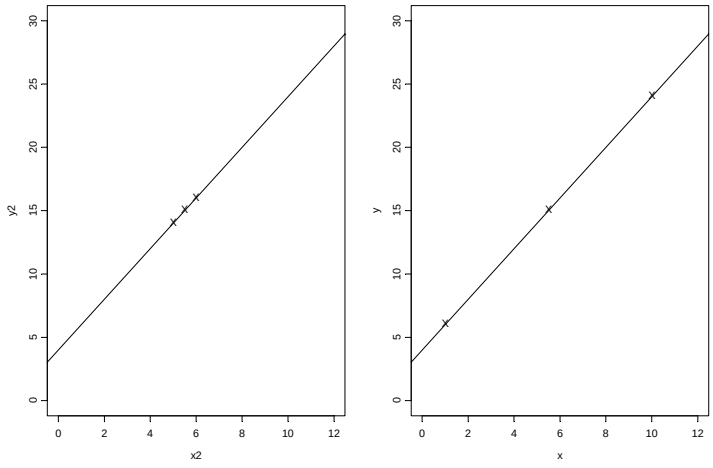
$$\text{variance} = \frac{\text{sum of squares}}{\text{degrees of freedom}}$$

The anova table is structured to make the calculation of variances as simple and clear as possible. For each row in an anova table, you simply divide the sum of squares by the adjacent degrees of freedom. The third column therefore contains two variances; the first row shows the regression variance and the second row shows the all-important error variance (s^2). One of the oddities of analysis of variance is that the variances are referred to as *mean squares* (this is because a variance is defined as a mean squared deviation). The error variance (also known as the *error mean square*, MSE) is the quantity used in calculating standard errors and confidence intervals for the parameters, and in carrying out hypothesis testing.

The **standard error of the regression slope**, *b*, is given by:

$$SE_b = \sqrt{\frac{s^2}{SSX}}$$

Recall that standard errors are *unreliability estimates*. Unreliability increases with the error variance so it makes sense to have s^2 in the numerator (on top of the division). It is less obvious why unreliability should depend on the range of x values. Look at these two graphs that have exactly the same slopes and intercepts. The difference is that the left hand graph has all of its x values close to the mean value of x while the graph on the right has a broad span of x values. Which of these would you think would give the most reliable estimate of the slope?



It is pretty clear that it is the graph on the right, with the wider range of x values. Increasing the spread of the x values reduces unreliability and hence appears in the denominator (on the bottom of the equation).

What is the purpose of the big square root term? This is there to make sure that the units of the unreliability estimate are the same as the units of the parameter whose unreliability is being assessed. The error variance is in units of *y squared*, but the slope is in units of *y* per unit change in x.

A 95% confidence interval for *b* is now given in the usual way:

$$CI_b = \text{t (from tables, } \alpha = 0.025, d.f. = n - 2).SE_b$$

where t is obtained using **qt**, the quantile of Student's t distribution (2-tailed, α = 0.025) with *n*-2 degrees of freedom.

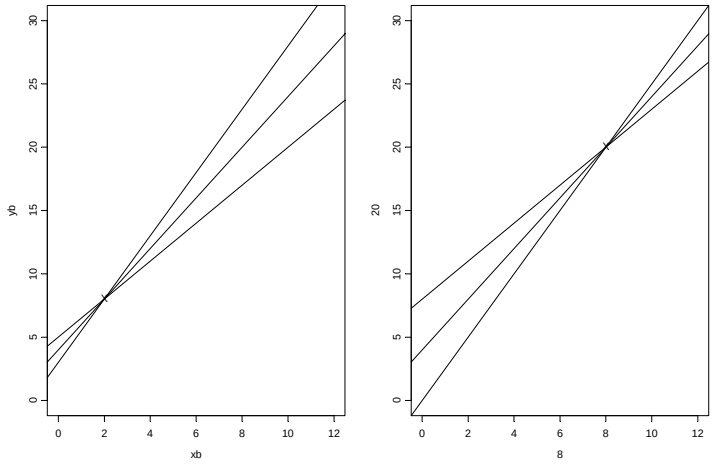
```
qt(.975,7)
```

```
[1] 2.364624
```

a little larger than the rule of thumb $(t \approx 2)$ because the degrees of freedom is so small. The **standard error of the intercept**, *a*, is given by:

$$SE_a = \sqrt{\frac{s^2 \sum x^2}{n.SSX}}$$

which is like the formula for the standard error of the slope, but with two additional terms. Uncertainly declines with increasing sample size n. It is less clear why uncertainty should increase with $\sum x^2$. The reason for this is that uncertainly in the estimate of the intercept increases, the further away from the intercept it is that the mean value of *x* lies. You can see this from the following graphs. On the left is a graph with a low value of $\bar{x}$ and on the right an identical graph (same slope and intercept) but estimated from a data set with a higher value of $\bar{x}$. In both cases there is a 25% variation in the slope. Compare the difference in the prediction of the intercept in the two cases.



Confidence in predictions made with linear regression declines with the square of the distance between the mean value of x and the value at which the prediction is to be made (i.e. with $(x - \bar{x})^2$). Thus, when the origin of the graph is a long way from the mean value of x, the standard error of the intercept will be large, and vice versa. A 95% confidence interval for the intercept, therefore, is

$$CI_a = \text{t (fromtables, } \alpha = 0.025, d.f. = n - 2).SE_a$$

with the same 2-tailed t-value as before. In general, the **standard error for a predicted value** $\hat{y}$ is given by:

$$SE_{\hat{y}} = \sqrt{s^2 \left[\frac{1}{n} + \frac{(x - \bar{x})^2}{SSX} \right]}$$

Note that the formula for the standard error of the intercept is just the special case of this for x = 0 (you should check the algebra of this result as an exercise). An important significance test concerns the question of whether or not the slope of the best fit line is significantly different from zero. If it is not, then the principal of

parsimony suggests that the line should be assumed to be horizontal. If this were the case, then y would be independent of x (it would be a constant, y = a).

Calculations involved in linear regression

The data are in a data frame called regression: the data frame contains the response variable, growth, and the continuous explanatory variable, tannin. You see the 9 values of each by typing the name of the data frame:

```
regression
```

```
  growth tannin
1     12      0
2     10      1
3      8      2
4     11      3
5      6      4
6      7      5
7      2      6
8      3      7
9      3      8
```

The numbers are mean weight gain (mg) of individual caterpillars and tannin content of their diet (%) . You will find it useful to work through the calculations long-hand as we go. The first step is to compute the "famous 5": $\sum x = 36$, $\sum x^2 = 204$, $\sum y = 62$, $\sum y^2 = 536$ and $\sum xy = 175$. Note that $\sum xy$ is 0x12 + 1x10 + 2x8 ... +8x3 = 175. Note the use of the semicolon **;** to separate two directives

```
sum(tannin);sum(tannin^2)
```

```
[1] 36
[1] 204
```

```
sum(growth);sum(growth^2)
```

```
[1] 62
[1] 536
```

```
sum(tannin*growth)
```

```
[1] 175
```

Now calculate the three corrected sums of squares, using the formulas on p.92:

$$SST = 536 - \frac{62^2}{9} = 108.889$$

$$SSX = 204 - \frac{36^2}{9} = 60$$

$$SSXY = 175 - \frac{36 \times 62}{9} = -73$$

then the slope of the best fit line is simply

$$b = \frac{SSXY}{SSX} = \frac{-73}{60} = -1.21666$$

and the intercept is

$$a = \bar{y} - b\bar{x} = \frac{62}{9} + 1.21666\frac{36}{9} = 11.755$$

The regression sum of squares is

$$SSR = b.SSXY = -1.21666 \times -73 = 88.82$$

so that the error sum of squares can be found by subtraction

$$SSE = SST - SSR = 108.89 - 88.82 = 20.07$$

Now we can complete the ANOVA table:

Source	SS	df	MS	F	Probability
Regression	88.82	1	88.82	30.98	0.0008
Error	20.07	7	2.867		
Total	108.89	8			

The calculated F ratio of 30.98 is much larger than the 5% value in tables with 1 degree of freedom in the numerator and 7 degrees of freedom in the denominator. To find the expected F value from tables we use **qf** (quantiles of the F distribution)

```
qf(0.95,1,7)
```

```
[1] 5.591448
```

We can ask the question the other way round. What is the probability of getting an F value of 30.98 if the null hypothesis of b = 0 is true ? We use 1 – **pf** for this

```
1-pf(30.98,1,7)
```

```
[1] 0.0008455934
```

so we can unequivocally reject the null hypothesis that the slope of the line is zero. Increasing dietary tannin leads to significantly reduced weight gain for these caterpillars. We can now calculate the standard errors of the slope and intercept

$$SE_b = \sqrt{\frac{2.867}{60}} = 0.2186$$

$$SE_a = \sqrt{\frac{2.867 \times 204}{9 \times 60}} = 1.041$$

To compute the 95% confidence intervals, we need the 2-tailed value of Student's t with 7 degrees of freedom

```
qt(.975,7)

[1] 2.364624
```

so the 95% confidence intervals for the slope and intercept are given by

$$a = 11.756 \pm 2.463$$

$$b = -1.21666 \pm 0.5169$$

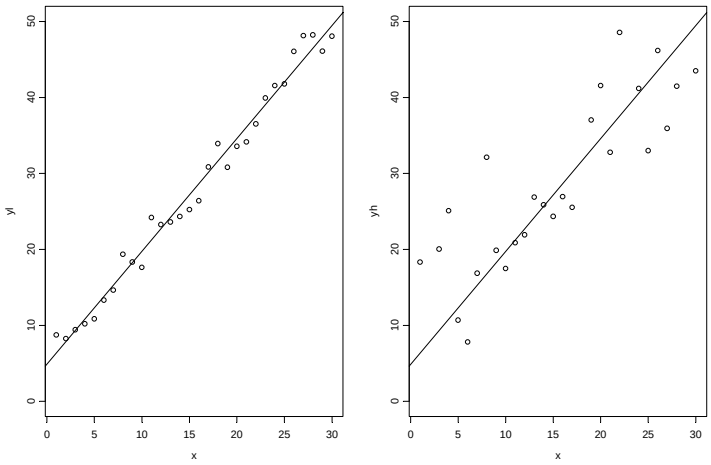
In written work we would put something like this:

"The intercept was 11.76 ± 2.46 (95% CI, n = 9)"

The reader needs to know that the interval is a 95% CI (and not say a 99% interval or a single standard error) and that the sample size was 9. Then they can form their own judgement about the parameter estimate and its unreliability.

Degree of scatter

Knowing the slope and the intercept tells only part of the story. The two graphs below have exactly the same slope and intercept, but they are completely different from one another in their degree of scatter.



It is obvious that we need to be able to quantify the degree of scatter, so that we can distinguish cases like this. In the limit, the fit could perfect, in which case all of the points fall exactly on the regression line. SSE would be zero and SSR = SST. In the opposite case, there is absolutely no relationship between y and x, so SSR = 0, SSE = SST, and the slope of the regression b = 0. Formulated in this way, it becomes clear that we could use some of the quantities already calculated to derive an estimate of scatter. We want our measure to vary from 1.0 when the fit is perfect, to zero when there is no fit at all. Now recall that the total variation in *y* is measured by SST. We can rephrase our question to ask: what fraction of SST is explained by the regression line? A measure of scatter with the properties we want is given by the ratio of SSR to SST. This ratio is called the *coefficient of determination* and is denoted by r^2:

$$r^2 = \frac{SSR}{SST}$$

Its square root, r, is the familiar *correlation coefficient*. Note that because r^2 is always positive, it is better to calculate the correlation coefficient from

$$r = \frac{SSXY}{\sqrt{SSX.SST}}$$

You should check that these definitions of r and r^2 are consistent. In statistical text books you will see the correlation coefficient defined in terms of covariance like this:

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

The two definitions are identical; our version using the sums of squares is simpler because the degrees of freedom in the numerator and denominator have cancelled out.

Using R for regression

The procedure so far should have been familiar. Let us now use R to carry out the same regression analysis. At this first introduction to the use of R each step will be

explained in detail. As we progress, we shall take progressively more of the procedures for granted. There are 6 steps involved in the analysis of a linear model:

- get to know the data;
- suggest a suitable model;
- fit the model to the data;
- subject the model to criticism;
- analyse for influential points;
- simplify the model to its bare essentials.

With regression data, the first step involves the visual inspection of plots of the response variable against the explanatory variable. We need answers to the following questions:

- is there a trend in the data?
- what is the slope of the trend (positive or negative)?
- is the trend linear or curved?
- is there any pattern to the scatter around the trend?

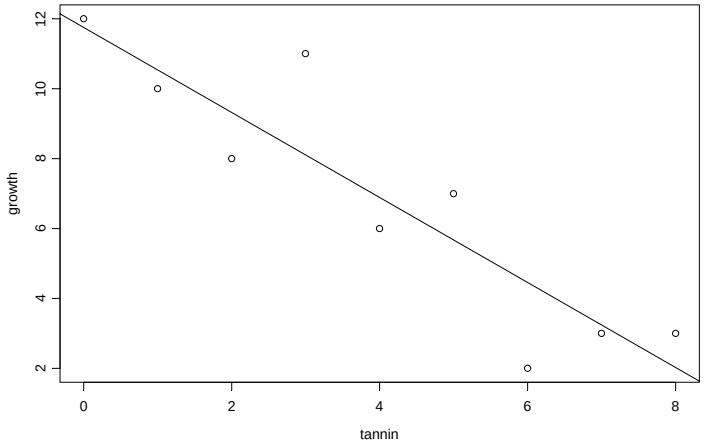
It appears that a linear model would be a sensible first approximation. There are no massive outliers in the data, and there is no obvious trend in the variance of *y* with increasing *x*. We have already attached the data frame called regression (p. 87). The scatterplot is obtained like this:

```
plot(tannin,growth)
```

Note that in the plot directive the arguments are in the order *x* then *y*. The fitted line added by **abline** like this:

```
abline(lm(growth~tannin))
```

where **lm** means linear model, **~** is pronounced tilde, and the order of the variables is *y* ~ *x* (in contrast to the plot directive, above).



Statistical modelling of regression proceeds as follows. We pick a name, say model, for the regression object, then choose a fitting procedure. We could use any one of several but let's keep it simple and use **lm**, the linear model that we have already used in **abline**, earlier. All we do it this:

```
model<-lm(growth~tannin)
```

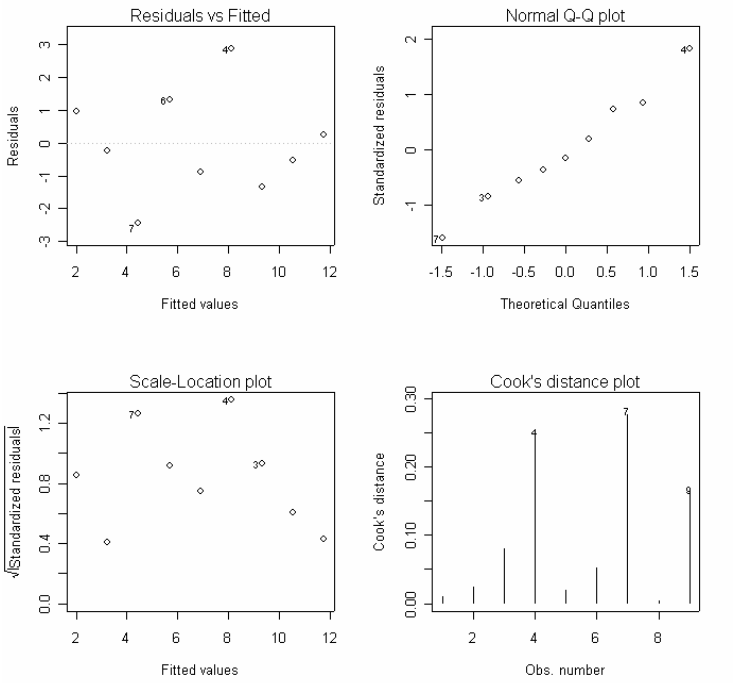
which is read: "model gets a linear model in which growth is modelled as a function of tannin". Now we can do lots of different things with the object called model. The first thing we might do is summarise it:

```
summary(model)
```

```
Call:
lm(formula = growth ~ tannin)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4556 -0.8889 -0.2389  0.9778  2.8944

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.7556     1.0408  11.295 9.54e-06 ***
tannin       -1.2167     0.2186  -5.565 0.000846 ***

Residual standard error: 1.693 on 7 degrees of freedom
Multiple R-Squared: 0.8157,     Adjusted R-squared: 0.7893
F-statistic: 30.97 on 1 and 7 DF,  p-value: 0.000846
```

The first part of the output shows the call (the model you fitted). This is useful as a label for the output when you come back to it weeks or months later. Next is a summary of the residuals, showing the largest and smallest as well as the 25% (1Q) and 75% (3Q) quantiles in which the central 50% of the residuals lie. Next the parameter estimates (labelled Coefficients) are printed, along with their standard errors just as we calculated them by hand. The t-values and *p*-values are for tests of the null hypotheses that the intercept and slope are equal to zero (against 2-tailed alternatives that they are not equal to zero). Residual standard error: 1.693 on 7 degrees of freedom is the square root of the error variance from our anova table. Multiple R-Squared: 0.8157 is the ratio SSR/SST we calculated earlier. F-statistic: 30.97 on 1 and 7 degrees of freedom, the *p*-value is 0.0008461 are the last two columns of out anova table. Next, we might plot the object called model. This produces a useful series of diagnostics.

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



Top left shows the residuals against the fitted values. It is good news because it shows no evidence if variance increasing for larger values of $\hat{y}$, and shows no evidence of curvature. Top right shows the ordered residuals plotted against the quantiles of the standard normal distribution. If, as we hope, our errors really are normal, then this plot should be linear. The data are very well behaved with only point number 4 having a larger residual than expected. The scale location plot is very like the first plot but shows the square root of the standardised residuals against the fitted values (useful for detecting non-constant variance). The last plot shows Cooks distance. This is an influence measure that shows which of the outlying values might be expected to have the largest effects on the estimated parameter values. It is often a good idea to repeat the modelling exercise with the most influential points omitted in order to assess the magnitude of their impact on the structure of the model.

Suppose we want to know the predicted value $\hat{y}$ at tannin = 5.5%. We could write out the equation in calculator mode, using the parameter estimates from the summary table, like this

```
11.7556-1.2167*5.5
```

```
[1] 5.06375
```

Alternatively, we could use the **predict** directive with the object called model. We provide the value for tannin = 5.5 in a **list** in order to protect the vector of x values (called tannin, of course) from being adulterated. The name of the x values to be used for prediction (tannin in this case) must be *exactly* the same as the name of the explanatory variable in the model.

```
predict(model,list(tannin=5.5))
```

```
 5.063889
```

For linear plots we use **abline** for superimposing the model on a scatterplot of the data points. For curved responses, it is sensible to use the predict function to generate the lines, by supplying it with a suitably fine-scale vector of x values (60 or more x points produce a smooth line in most cases: we shall do this later).

Summary

The steps in the regression analysis were:

- data inspection
- model specification
- model fitting
- model criticism.

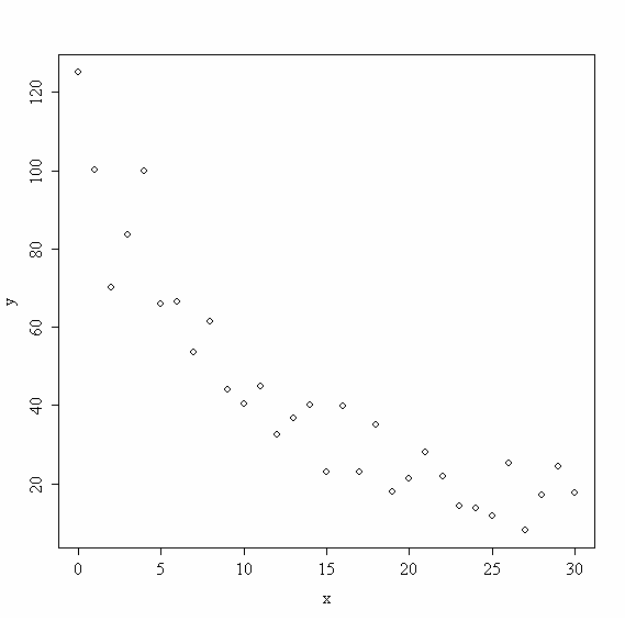
We conclude that the present example is well described by a linear model with normally distributed errors. There are some large residuals, but no obvious patterns in the residuals that might suggest any systematic inadequacy of the model. The slope of the regression line is highly significantly different from zero, and we can be confident that, for the caterpillars in question, increasing dietary tannin reduces weight gain and that, over the range of tannin concentrations considered, the relationship is reasonably linear. Whether the model could be used accurately for predicting what would happen with much higher concentrations of tannin, would need to be tested by further experimentation. It is extremely unlikely to remain linear, however, as this would soon begin to predict substantial weight losses (negative gains), and these could not be sustained in the medium term. We end by removing **rm** the current *x* and *y* vectors

```
par(mfrow=c(1,1))
rm(x,y)
```

Transformation of non-linear responses

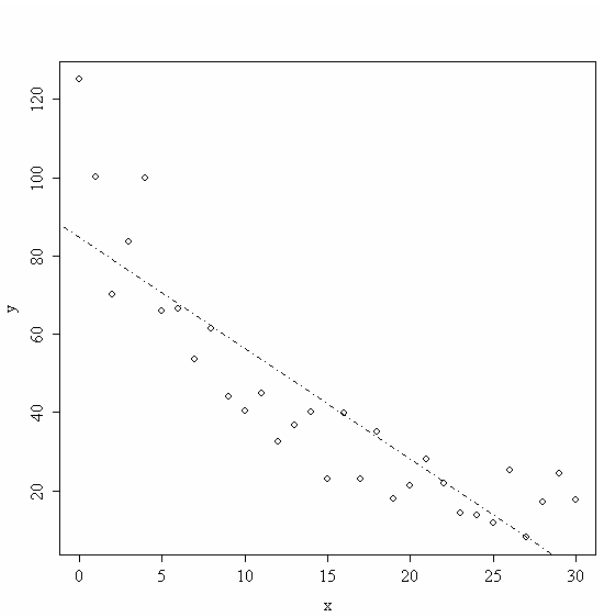
The next example involves data that can not be represented by a straight line. The response variable is dry mass of organic matter remaining after different periods of time in a decomposition experiment. The data are in a file called decay.txt

```
decay<-read.table("c:\\temp\\decay.txt",header=T)
attach(decay)
names(decay)
[1] "x" "y"
plot(x,y)
```



This does not look particularly like a straight-line relationship. Indeed, it makes no scientific sense to fit a linear relationship to these data, because it would begin to predict negative dry matter once time (x) got bigger than 30 or so. Let's put a straight line through the data using **abline** to get a better impression of the curvature.

abline(lm(y~x))



What we see are "groups of residuals". Below about x = 5 most of the residuals are positive, as is the case for the residuals for x bigger than about 25. In between, most of the residuals are negative. This is what we mean be "evidence of curvature". Or more forthrightly "systematic inadequacy of the model". Let's do a linear regression anyway, then look at what the model checking tells us. We call the model object "result" and fit the model as before:

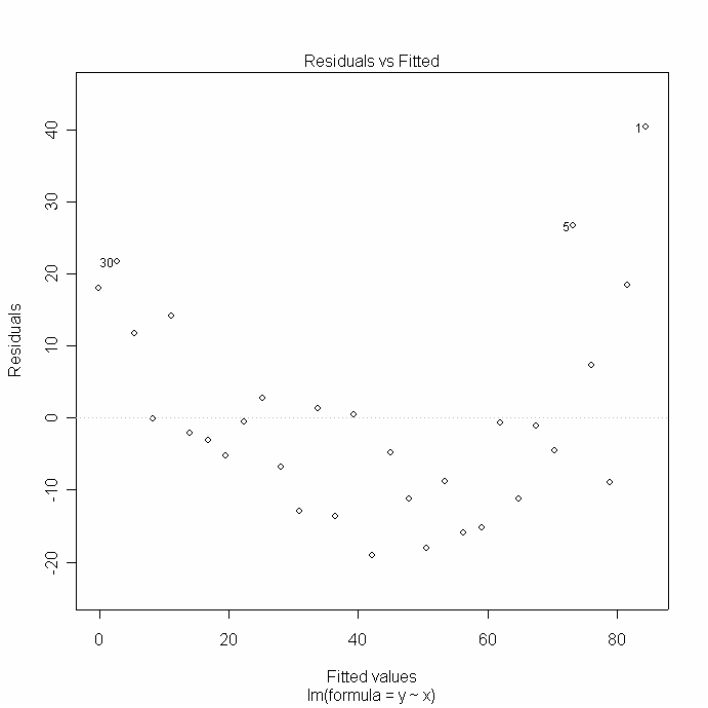
result<-lm(y~x)

summary(result)

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

Residuals:
    Min      1Q  Median      3Q     Max
-19.065 -10.029  -2.058   5.107  40.447

Coefficients:
            Estimate Std. Error t value  Pr(>|t|)
(Intercept)  84.5534     5.0277   16.82 2.22e-016 ***
x            -2.8272     0.2879   -9.82 9.94e-011 ***

Residual standard error: 14.34 on 29 degrees of freedom
Multiple R-Squared: 0.7688,     Adjusted R-squared: 0.7608
F-statistic:96.44 on1 and 29 degrees of freedom, p-value: 9.939e-011
```

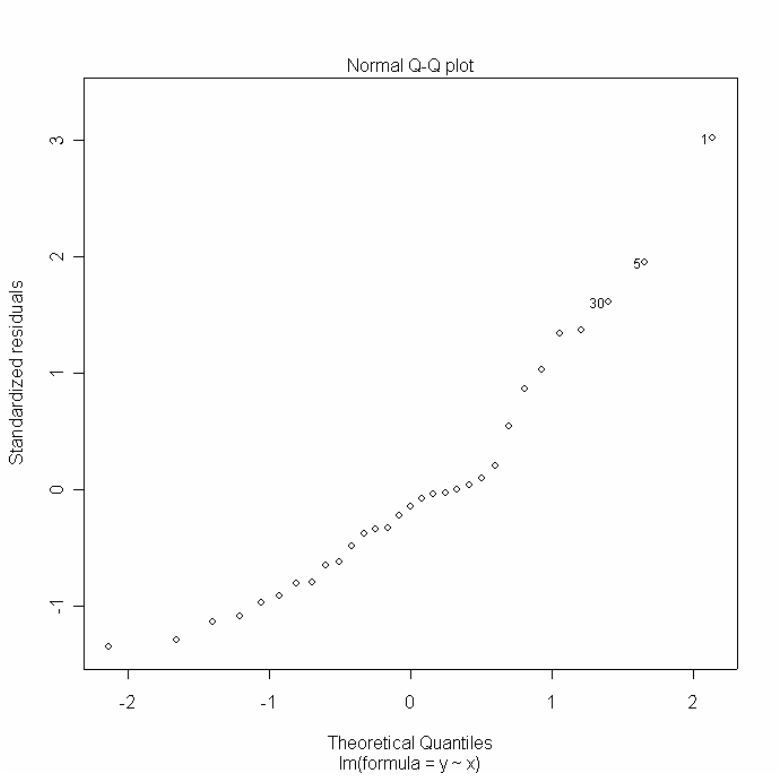
Everything is highly significant. But is the model any good? To test this we carry out the **diagnostic plots**.

plot(result)



The first plot (residuals against fitted values) shows the problem at once. This should look like the sky at night (i.e. no pattern) but in fact, it looks like a letter U. The positive residuals occur only for small and large fitted values of y (as we suspected from our earlier inspection of the abline plot). The message here is simple. A U-shaped plot of residuals against fitted values means that the linear model is inadequate. We need to account for curvature in the relationship between y and x.

The second diagnostic plot looks like this:



This is banana shaped, not straight as it should be. It gets much steeper above theoretical quantiles of about +0.5, further evidence that something is wrong with the model.

The first response to curvature is often transformation, and that is what we shall do here. When we have learned **glm** we shall have lots more options of things to do.

Looking back at the plot and thinking about the fact that it is a decay curve leads us first to try a model in which log(y) is a linear function of x.

You will recall that the equation for exponential decay looks like this:

$$y = ae^{-bx}$$

Taking logs of this, from left to right, we say "log y is ln(y), log a is ln(a), log of 'times' is 'plus', and log of exp(-bx) is –bx ". Remember that exp is the antilog function, so "log of antilog" cancels out (leaving –bx in this case).

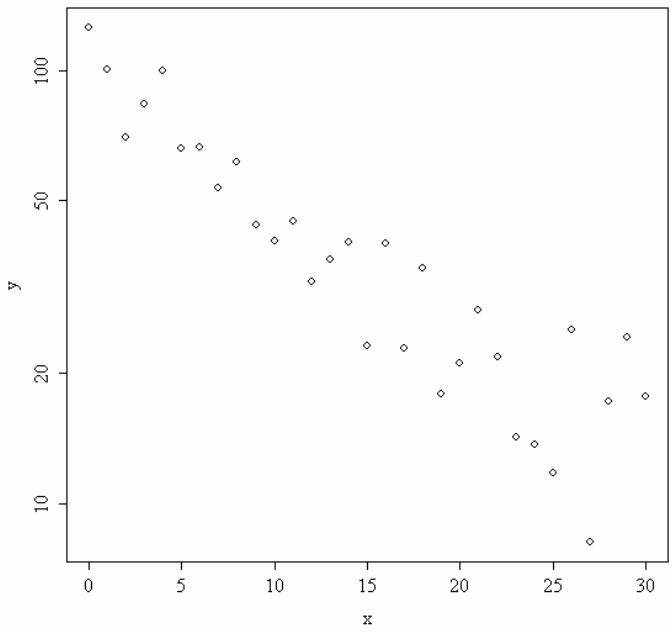
$$\ln(y) = \ln(a) - bx$$

Note that "+-b" becomes "-b" (when plus and minus occur together minus always wins). Now if you replace ln(y) with Y and ln(a) with A you get

$$Y = A - bx$$

which is a straight line: the intercept is A and the slope is –b. This suggests that if we plot of graph of ln(y) against x it should look much straighter:

```
plot(x,y,log="y")
```



That is, indeed, much straighter, but now a new problem has arisen: the variance in y increases as y gets smaller (the scatter of the data around the line is fan-shaped). We shall deal with that later.
For the moment, we fit a different linear model. Instead of

$$y \sim x$$

we shall fit

$$\log(y) \sim x$$

This is fantastically easy to do by replacing “y” in the model formula by “log(y)”. Let’s call the new model object “transformed” like this

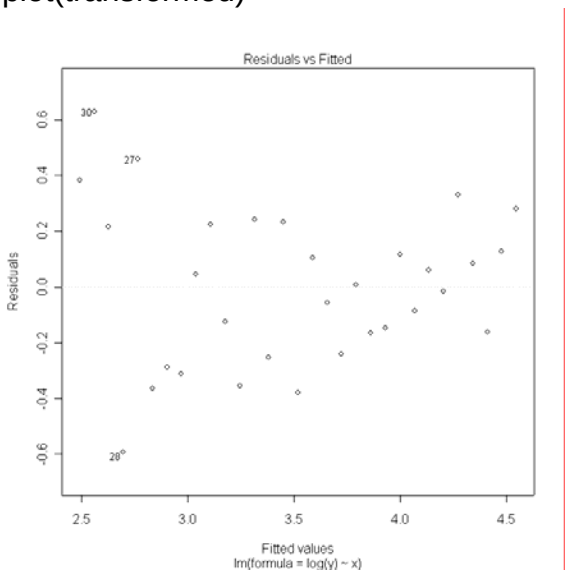
```
transformed <- lm(log(y)~x)
```

```
summary(transformed)
```

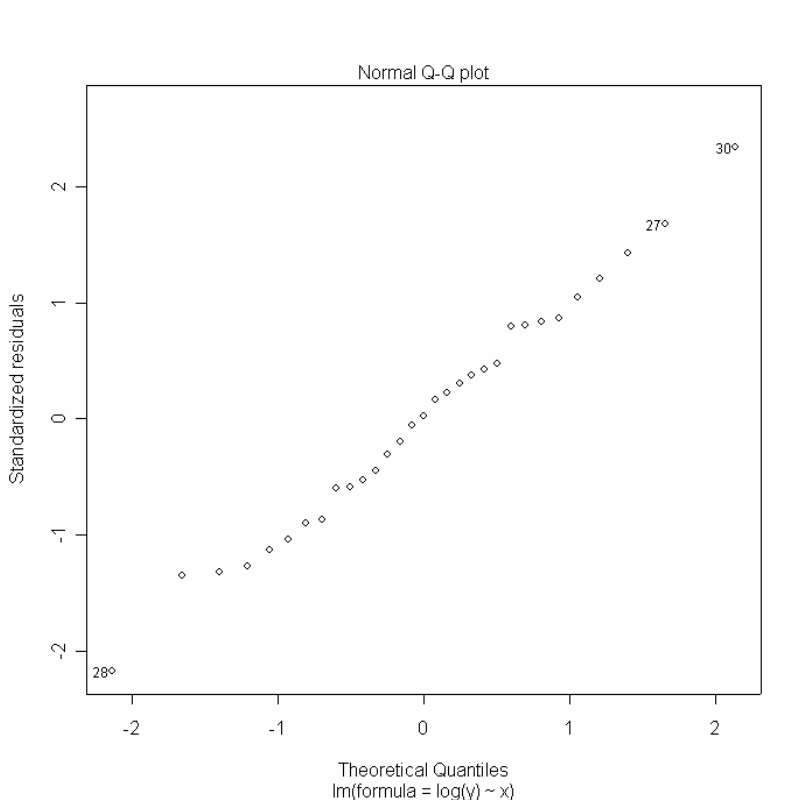
```
Call:
lm(formula = log(y) ~ x)

Residuals:
      Min        1Q    Median        3Q       Max
-0.593515 -0.204324  0.006701  0.219835  0.629730

Coefficients:
             Estimate Std. Error t value  Pr(>|t|)
(Intercept)  4.547386   0.100295   45.34  < 2e-016 ***
x           -0.068528   0.005743  -11.93 1.04e-012 ***
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05  `.'  0.1  ` '

Residual standard error: 0.286 on 29 degrees of freedom
Multiple R-Squared: 0.8308,     Adjusted R-squared: 0.825
F-statistic: 142.4 on 1 and 29 degrees of freedom,      p-value:
1.038e-012
```

As before, everything is highly significant. But does the model behave any better? Lets look at the diagnostic plots.

```
plot(transformed)
```



We have cured the U-shaped residuals, but at the price of introducing non-constancy of variance (in the jargon, this is the dreaded heteroscedasticity). What about the normality of the errors ? This is on the second plot (press the return key):



Not perfect, but very much better than before.

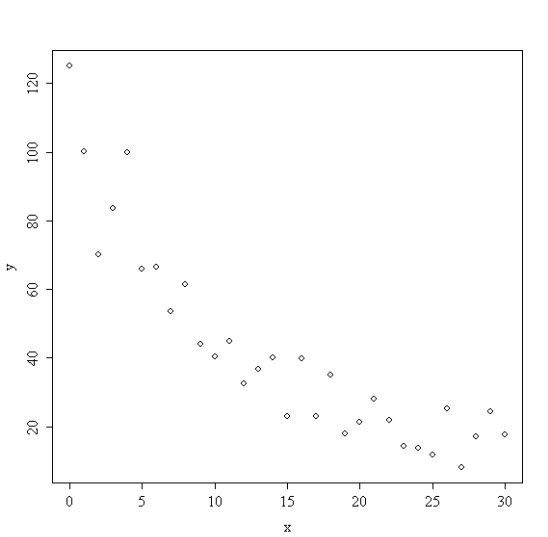
We finish at this point, leaving the problem of non-constant variance until we know about using **glm**s.

Generally, you would want to plot your model as a smooth line through the data. You use the **predict** function for this. There are 3 steps:

- use **seq** to generate a series of values for the x axis (they should have the same range as the data and about 100 increments to give the curve a nice smooth look; if you have too few increments, the curve will look as if it is made up from straight sections (which, of course, it is !))
- put these values into the **predict** function to get the matching y values of the fitted curve (this step can be tricky to understand at first)
- **back transform** the predicted values to get the smooth values for y
- use **lines** to overlay the smooth fitted curve on the data

Let's remind ourselves what the raw data look like

```
plot(x,y)
```



So we want our generated *x* values to go from 0 to 30 and ***they must be called x*** because that is the name of the explanatory variable in the model called 'transformed'. Now there's a slight problem here because *x* is the name of the vector containing our data points, and we don't want to mess with this. The solution is this. Generate the values for drawing the fitted curve under a ***different*** name, *smoothx*, like this:

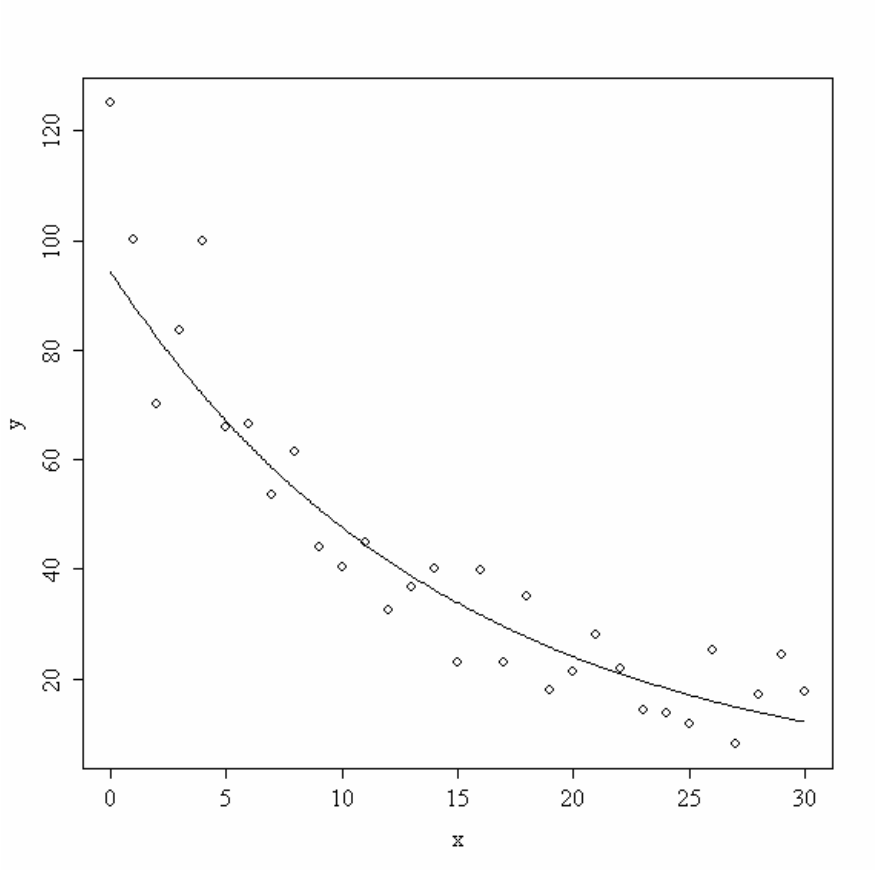
```
smoothx<-seq(0,30,0.1)
```

Fool the model into thinking that the data are called *x* rather than *smoothx* by using a **list** like this **list(x=smoothx)** inside the **predict** function. At the same time, we need to back transform the predicted values (which are in logs) using **exp** (the antilog function):

```
smoothy<-exp(predict(transformed,list(x=smoothx)))
```

It is important that you understand this. You will want to use predict a lot. We **predict** using the current model, which is called 'transformed'. This takes values of *x* because that was the name of the explanatory variable we used in fitting the model log(y)~x We use **list** to coerce the *smoothx* values into a vector that will be known internally as *x*, but which will not alter the values of our data called *x*. Finally, we back transform the predicted values to get our *smoothy* values on the untransformed scale. Since the model was log(y)~x, the appropriate back transformation is exp(predicted). Now (finally !) we can draw the fitted model through the data, using **lines**:

lines(smoothx,smoothy)



Summary

- regression involves estimating parameter values from data
- there are always lots of different possible models to describe a given data set
- model choice is a big deal (here we chose exponential in preference to linear)
- always use diagnostic plots to check the adequacy of your model after fitting
- the two big problems are **non-constant variance** and **non-normal errors**
- one solution for curvature in the relationship between y and x is transformation
- transformation can be specified simply in the model formula log(y) ~ x
- use predict to generate smooth curves for plotting through the scatter of data
- generate the values of x for calculating fitted values of y under a *different* name
- use list to coerce the different name into the original name: e.g. **list(x=smoothx)**
- don't forget to back transform before you draw the curve using **lines**