Regression — Fitting Equations To Data

Jonathan Godfrey (Editor)

February 18, 2014

Acknowledgements

This manual was compiled from course notes for previous offerings of 143.222/228.222 Technological Mathematics A and 161.320 Fitting Regression Models. Staff who are therefore owed a vote of thanks include Wyatt Page, Doug Stirling, and Alasdair Noble.

Contents

1 Introduction

The problem of generating an equation to describe data is one that crops up in many scientific and technical fields. The aim of fitting an equation is usually to either derive or verify the relationship between a dependent (response) variable and one or more independent (predictor) variables. Examples include: predicting the yield of a process as input or operating conditions change, estimating heat transfer coefficients for an exchanger, and attempting to predict the inflation rate from various economic indicators. The method which is used involves finding an equation (mathematical model) that adequately describes the relationship (if there is one!) between the response and predictor variables, and estimating the coefficients (parameters) in this equation. The simple linear regression models of first year courses are rarely useful in practice. These notes cover a range of additional topics that will turn these simple models into useful and valid regression models that allow flexibility and applicability.

When we wish to fit an equation or curve to a set of experimental or tabulated data, there are two basic approaches:

1. Regression Analysis — Especially appropriate for experimental data with some 'scatter' or random variation. Some may use:

   (a) Linear models. For example

   $$y = a + bx_1 + cx_2 + dx_3 + \dots \tag{1}$$

   (b) Polynomial models. For example

   $$y = a + bx + cx^2 + dx^3 + \dots \tag{2}$$

   In 161.100 Principles of Statistics (and most first year statistics courses) we used a single independent variable, and the method of least-squares to minimize the error of fit for the chosen function. In this course we extend the ideas of simple regression to multiple regression, polynomial regression, piecewise functions, and model selection approaches.

2. Continuous or Piecewise Polynomial Fitting — More suited to tabulated data such as thermodynamic data containing little random error. Can use:

   (a) Finite differences — Single continuous polynomial

   (b) Splines — Piecewise polynomials

   Option a) gives a polynomial approximation that will fit some data points. While option b) gives a polynomial approximation that fits all data points and approximates the shape of the function. We will see how to create splines in a regression framework in Section 14 but will not discuss interpolation or smoothing techniques available in many mathematical (rather than statistical) software packages.

Why might we want to fit an equation or model to a set of data? The usual aim is to derive or verify some relationship between a dependent (response) variable and one or more independent (predictor) variables. We will use regression techniques to do this because they offer the opportunity to gauge a model's ability to describe the phenomenon under consideration, whether it be for description or prediction purposes. Typical applications include:

1. Predict the rate of inflation from economic indicators.

2. Predict the yield of some process as input conditions change.

3. Estimate an unknown parameter for a theoretical function applied to an observed set of data.

Mathematical models can be classified in several ways. The first distinction of interest is whether a model is mechanistic or empirical. A mechanistic model is one where the equation is derived from physical considerations — Ohm's Law is a simple example. The form of the model is fixed, and the main purpose of fitting it to the data is often to estimate the parameters rather than generate or verify predictions. To use the example of Ohm's Law

$$V = IR \tag{3}$$

we might collect a series of current readings over a range of voltages and fit the equation:

$$y = \beta x \tag{4}$$

where y is the voltage (V), x is the current (I), and β is the resistance (R) to be estimated in the equation-fitting process. Another area where mechanistic models are common is chemical kinetics - here the aim of fitting a rate equation to data is often to validate a kinetic mechanism and explain (justify, excuse or whatever) the values obtained for the estimated parameters in the fitted model.

In contrast an empirical model is simply a mathematical curve or set of 'French curves' used to describe the data. The exact parameter values generally have little physical significance, as the aim is to produce an equation which will serve as an interpolating function for future data gathered at similar operating conditions. Sometimes this approach may give insight into how to derive a suitable model. Instrument calibration curves are one commonly encountered example. Empirical models are often set up as polynomials because polynomials are a versatile and reasonably flexible way to describe bumps, wiggles and curves in data. The actual model form can often be obtained simply by inspecting a plot of the data, particularly if there is only one predictor variable and the data are reasonably 'well-behaved', that is, vary in a smooth and continuous fashion. Instrument calibration curves are often straight lines.

Models can also be classified from a mathematical view as linear or non-linear. A linear model is linear in the parameters, that is, the model is of the form that the predicted response (y) is a linear combination of the parameters (β). The following are examples of linear models:

1. $y = \beta_0 + \beta_1 x$
2. $y = \beta_0 + \beta_1 x1 + \beta_2 x_2$
3. $y = \beta_0 + \beta_1 x + \beta_2 x^2$
4. $y = \beta_1 \sin(x_1) + \beta_2 \ln(x_2/x_3) + \beta_3 x_4 x_5$

Note that some of these models are for non-linear functions. It is important to know the difference between a non-linear function that can be fitted using a linear model and one that actually requires a non-linear model. Non-linear models have no restriction on the form of the equation fitted. Examples are:

1. $y = (\beta_0 + \beta_1 x_1)/(\beta_2 x_2 + \beta_3 x_3)$
2. $y = \beta_0 e^{-\beta_1 x}$

Estimation of model parameters is much easier for linear than non-linear models. In each case the p estimated parameters are the solution of a set of p simultaneous equations (called the normal equations). A linear model generates linear equations, which are relatively straightforward to solve. For a non-linear model the normal equations are non-linear, and finding the solution is a difficult and time-consuming process with no guarantee of eventual success. Because of this, empirical models are almost always set up in a linear form (usually a low order polynomial).

On the other hand, mechanistic models are almost always non-linear in form (Ohm's Law is an exception). Sometimes it is possible to transform the equation into linear form by taking logs, reciprocals etc., or by some rearrangement of the model form. For example, the model

$$y = ax^b \tag{5}$$

can easily be converted to a linear form by taking logs, giving

$$\ln y = \ln a + b \ln x \tag{6}$$

In such circumstances we would fit a straight line to the transformed data for the x and y variables. Having found the parameters for the straight line, we would rearrange the terms to give an equation of the form given in Equation 5. In practice Technologists and Engineers often go to great lengths to linearize a model. The main aim of regression modelling is to estimate the model parameters, the β's for the linearized models. You might practice with the following models:

Exercise 1 *The Arrhenius equation*

$$k = Ae^{-E_a/RT} \tag{7}$$

is used to determine the reaction rate coefficient at a given temperature, where k is the reaction rate coefficient (measured per second), A is a constant, E_a is the activation energy (measured in J/mol), R is the universal gas constant (8.314 J/mol.K), and T is the temperature in degrees Kelvin. If you were given data for the reaction rates at certain temperatures, and wanted to estimate A and E_a, how would you rearrange the data and the model so that linear regression could be employed? (Answer on page 58)

2 Fitting linear models

As previously stated, a linear model is one that is linear in the parameters, such as

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon \tag{8}$$

where y is a measured value of the response variable, $x_1, \ldots, x_p$ are the corresponding values of the p predictor variables, $\beta_0, \ldots, \beta_p$ are the unknown parameters to be estimated and ϵ is a random error term. Sometimes there is no constant term in the equation, so the number of parameters reduces to p.

As a simple example of what is involved in fitting a linear model to data, we will work through an example from Draper and Smith (1998) in detail. The data are monthly operating records from a chemical plant giving the steam consumption in pounds used per month and the average operating temperature in °F (pardon the units). The question posed is whether there is any systematic association between these variables, or in more formal terms can we construct an adequate model relating them. The data, which appear in Data Set 1, are in a comma delimited file called `Steam.csv`. We read this file into R and identify the variables in the data set using the `read.csv` and `str` commands in the following way:

Data Set 1 — Part of Steam.csv from Draper and Smith (1998)

Steam	*Temp*	*Steam*	*Temp*	*Steam*	*Temp*
10.98	*35.3*	*9.14*	*57.5*	*8.11*	*70*
11.13	*29.7*	*8.24*	*47.4*	*7.83*	*70*
12.51	*30.8*	*12.19*	*28.9*	*8.88*	*74.5*
8.4	*58.8*	*11.88*	*28.1*	*7.68*	*72.1*
9.27	*61.4*	*9.57*	*39.1*	*8.47*	*58.1*
8.73	*71.3*	*10.94*	*57.8*	*8.86*	*44.6*
7.36	*74.4*	*9.58*	*48.5*	*10.36*	*33.4*
8.50	*77.7*	*10.09*	*59.3*	*11.08*	*28.6*
7.82	*70.7*				

```
> Steam <- read.csv("Steam.csv")

> str(Steam)

'data.frame':        25 obs. of  9 variables:
 $ SteamUse: num  10.98 11.13 12.51 8.4 9.27 ...
 $ Storage : num  5.2 5.12 6.19 3.89 6.28 5.76 3.45 6.57 5.69 6.14 ...
 $ Glycerin: num  0.61 0.64 0.78 0.49 0.84 0.74 0.42 0.87 0.75 0.76 ...
 $ Wind    : num  7.4 8 7.4 7.5 5.5 8.9 4.1 4.1 4.1 4.5 ...
 $ CalDays : int  31 29 31 30 31 30 31 31 30 31 ...
 $ OpDays  : int  20 20 23 20 21 22 11 23 21 20 ...
 $ ColdDays: int  22 25 17 22 0 0 0 0 0 0 ...
 $ Temp    : num  35.3 29.7 30.8 58.8 61.4 71.3 74.4 76.7 70.7 57.5 ...
 $ Startups: int  4 5 4 4 5 4 2 5 4 5 ...
```

Prior to formulating any tentative model (and all models are initially tentative) it is often useful to construct a plot of the data. For particular models under consideration the plot may confirm the expected behaviour or show any unexpected deviations. If the model form is in doubt, then such a plot can also assist in formulating a model to try. Exhibit 1 presents a scatter plot or the data of this example. From this plot, and in the absence of any mechanistic information it would seem reasonable to choose a straight line as a tentative model, that is

$$\text{Steam Use } (y) = \beta_0 + \beta_1 \text{ Average Temperature} + \epsilon \tag{9}$$

where β_0 and β_1 are parameters to be estimated from the data and ϵ is a random error term representing the discrepancy between observed and predicted response values.

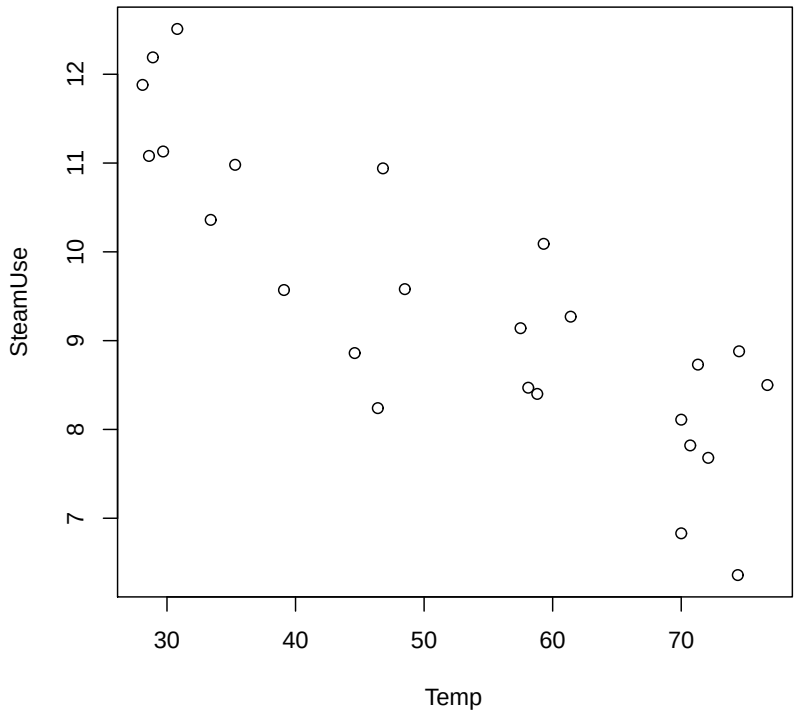
N.B. If the initial plot of the data shows a pattern that is not a linear relationship, we should not bother fitting a straight line to it. This is a crucial assumption of model fitting via least squares regression. There are other assumptions that will be discussed in Section 3.2.

To go back just a little, the model form is tentative (although there is little doubt in this simple case). The adequacy of the model can only be tested after it has been fitted to the data. If the model needs some modification or reformulation, this will be apparent from looking at the results of the fit. A suitably altered model can then be fitted, tested for adequacy, modified if necessary etc. In complex model-fitting tasks, it is unusual for the first possible model tried to be suitable with no further changes.

Exercise 2 *Investigate the data for the hardness of concrete based on the quantity of cement put in the mix, using the data found in cement.csv or on page 65. Would you fit a linear model to this data? (Answer on page 58)*

Exhibit 1 Plot of Steam Use versus Average Temperature.

```
> attach(Steam)
> plot(Temp,SteamUse)
> detach(Steam)
```



3 The least-squares concept

After we have obtained a tentative model form, there are still many possible straight lines that might be said to reasonably represent the data, characterized by different values for the estimated β's. There are a number of criteria that could be used to specify the 'best' estimates of these parameters, but the one thing they all have in common is that the 'best' estimates are those which minimize (in some fashion) the deviations between the observed and predicted values of the dependent variable y (these discrepancies are call *residuals*). For example we might look to minimize the sum of the absolute values of the residuals or to minimize the largest residual or to minimize (in some sense) the perpendicular distances between the observed y values and the predictor equation. All of these have some merit in various situations, but the most commonly used one that we will be discussing is the least-squares criterion, where we estimate values of β_0 and β_1 that minimize the sum of squared residuals, where a residual is defined as the discrepancy between observed and predicted y's, or

$$e_i = y_i - \left(\hat{\beta}_0 + \hat{\beta}_1 x_i\right) \tag{10}$$

Residuals are therefore observed as a result of fitting a model to data, while errors ϵ are theoretical in nature. We usually use the notation e for residuals instead of the Greek ϵ which is reserved for error.

This process of fitting models by least squares is also referred to as linear least squares, linear estimation and (multiple) linear regression. For the Steam Use example, with 25

Exhibit 2 Regression Analysis

```
> Steam.lm1 = lm(SteamUse~Temp, data=Steam)
> summary(Steam.lm1)

Call:
lm(formula = SteamUse ~ Temp, data = Steam)

Residuals:
   Min     1Q Median     3Q    Max
-1.679 -0.529 -0.122  0.799  1.346

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  13.6230     0.5815   23.43  < 2e-16 ***
Temp         -0.0798     0.0105   -7.59  1.1e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.89 on 23 degrees of freedom
Multiple R-squared:  0.714,        Adjusted R-squared:  0.702
F-statistic: 57.5 on 1 and 23 DF,  p-value: 1.05e-07
```

observations, the sum of squared residuals is

$$\text{SS}_{\text{Residuals}} = \sum_{i=1}^{25} e_i^2 = \sum_{i=1}^{25} \left[y_i - \left(\hat{\beta}_0 + \hat{\beta}_1 x_i \right) \right]^2 \tag{11}$$

The minimum value of $\text{SS}_{\text{Residuals}}$ will be when the first partial derivatives of Equation 11 with respect to the parameters are equal to zero, leading to two simultaneous equations for $\hat{\beta}_0$ and $\hat{\beta}_1$. We use the 'hat' symbol to indicate that these quantities are the least squares estimates of β_0 and β_1. The normal equations used to find these estimates for the Steam Use example are

$$\begin{aligned} 25\hat{\beta}_0 + \left(\sum_{i=1}^{25} x_i \right) \hat{\beta}_1 &= \sum_{i=1}^{25} y_i \\ \left(\sum_{i=1}^{25} x_i \right) \hat{\beta}_0 + \left(\sum_{i=1}^{25} x_i^2 \right) \hat{\beta}_1 &= \sum_{i=1}^{25} x_i y_i \end{aligned} \tag{12}$$

The solution for these equations is where $\hat{\beta}_0 = 13.623$ and $\hat{\beta}_1 = -0.07983$, giving a least squares straight line

$$\text{Steam Use}(y) = 13.623 - 0.07983 \times \text{ Average Temperature} \tag{13}$$

R and other statistical software packages can be used to find these parameters. Microsoft EXCEL could be used to get these parameters, but its capacity is limited and will not allow the completion of all the tasks described in these notes. A basic regression of Steam Use on Average Temperature would result in the R output presented in Exhibit 2. It's important that the values of parameters actually make sense to the real world context being modelled. For the Steam Use example, the amount of steam used decreases as the average temperature increases, because the sign of the coefficient is negative. Note that the output given by R is somewhat different to that which you saw if you've used Minitab or other statistical software. In fact we even had to ask for it using the `summary` command.

Exercise 3 *Use R to fit a linear model to the data for the hardness of cement batches found in cement.csv or on page 65. Interpret the coefficients in your model. (Answer on page 58)*

The model-fitting process does not stop with the production of parameter estimates. Logical questions arise about the ability of the model to predict future observed y's (That is, what is a typical figure for the discrepancy one might expect) and about the exact parameter values obtained. It has already been mentioned that there are many similar straight lines that might be considered reasonable. One useful way of looking at the uniqueness of the estimated parameters is to construct a sum-of-squares surface. This is just a contour diagram showing what the sum of squared residuals would be for various values of the β's. The least-squares estimates correspond to the bottom of the ellipsoidal bowl in the contour plot in Exhibit 3. If the bowl has steep sides, then the sum-of-squared residuals is more sensitive to changes in the values and conversely with shallow sides the sum of squared residuals is less sensitive to the exact values. Steep contours therefore correspond to a situation where the estimated values can be looked on as precise, and the converse holds for shallow contours. A mesh plot is also presented in Exhibit 3 so that you can see the three dimensionality of the surface being considered.

Another point worth noting is that the bowl in these plots is not circular, and that the axes of the elliptical contours are inclined. In general terms we can say the values that give the same sum of squared residuals have about the same probability of being the 'true' values, so by some trading off between the two coefficients we can come up with many pairs of $\hat{\beta}_0$ and $\hat{\beta}_1$, values that give quite a reasonable fit to the data. We will look at this question in more detail when we consider the precision of the estimated parameters.

3.1 Least-squares: The general case

For the general linear model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon \tag{14}$$

we can tabulate the observations used in fitting the data with a value for the y (response) and each of the x's (predictors). Since the model as written has $p + 1$ parameters, we obviously need a minimum of $p+1$ observations to fit it in the first place (think about how many points are needed to define a straight line, a parabola, ...). If we want to investigate the question of model adequacy (only the brave and foolish don't) then more than $p + 1$ observations are necessary. If the model has no constant term β_0, then of course we only need p observations as a minimum. The data should be tabulated with each observation (unit) using one row of a worksheet, and each variable having its own column. See any of the files associated with the data sets given in this manual for examples.

The general linear model can be written in matrix form as

$$\underline{Y} = \underline{X}\underline{\beta} + \underline{\epsilon} \tag{15}$$

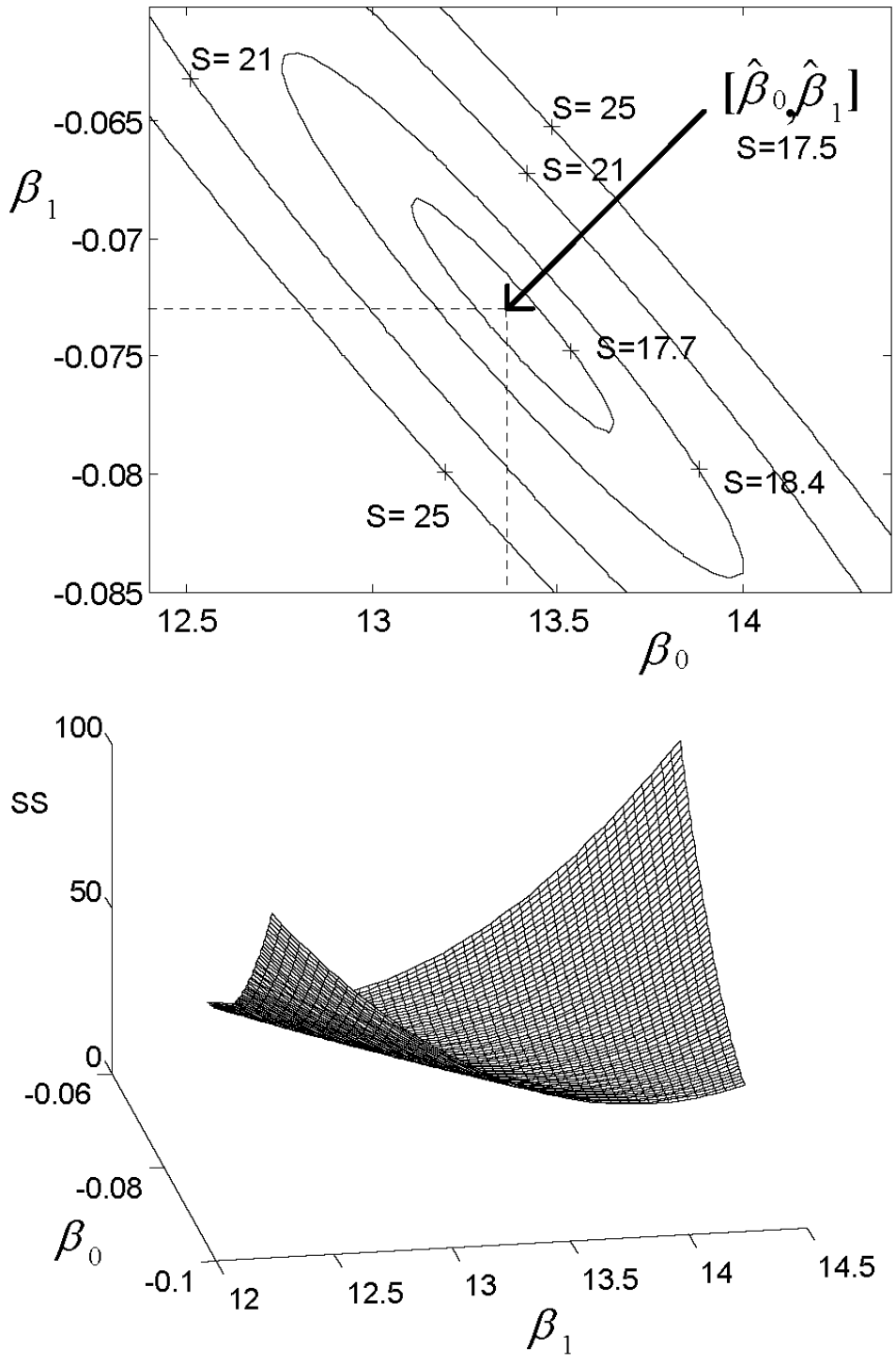
where the matrices are

$$\underline{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \underline{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} & \ldots & x_{1p} \\ 1 & x_{21} & x_{22} & x_{23} & \ldots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & x_{n3} & \ldots & x_{np} \end{bmatrix}, \quad \underline{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}, \quad \underline{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix} \tag{16}$$

The constant term can be thought of as multiplying a variable x_0 whose value is always 1. So the general form of the matrix is simply a table in which each column contains the values of one or the variables in the equation. So, for example, if one is fitting the quadratic expression

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 \tag{17}$$

Exhibit 3 Contour plot (top) and mesh plot (bottom) of the residual sum of squares versus values of β_1 and β_0.



then the first column of the matrix will be all 1's, the second will be the values of x and the third will be the squares of the x values. If the model has no constant term, then obviously there will not be an initial column of 1's. The $p+1$ normal equations for estimating the parameters are

$$\left[\underline{X}^T\underline{X}\right]\underline{\hat{\beta}} = \underline{X}^T\underline{Y} \tag{18}$$

where $\underline{X}^T$ is the transpose of $\underline{X}$, and $\underline{\hat{\beta}}$ is the vector of least squares parameter estimates. The solution to the normal equations is

$$\underline{\hat{\beta}} = \left[\underline{X}^T\underline{X}\right]^{-1}\underline{X}^T\underline{Y} \tag{19}$$

The method can be illustrated using the Steam Use data, where

$$\underline{X} = \begin{bmatrix} 1 & 35.3 \\ 1 & 29.7 \\ \vdots & \vdots \\ 1 & 28.6 \end{bmatrix}, \quad \underline{Y} = \begin{bmatrix} 10.98 \\ 11.13 \\ \vdots \\ 11.08 \end{bmatrix}, \tag{20}$$

$$\underline{X}^T\underline{X} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 35.3 & 29.7 & \dots & 28.6 \end{bmatrix} \begin{bmatrix} 1 & 35.3 \\ 1 & 29.7 \\ \vdots & \vdots \\ 1 & 28.6 \end{bmatrix} = \begin{bmatrix} 25 & 1328 \\ 1328 & 77722 \end{bmatrix}, \tag{21}$$

$$\underline{X}^T\underline{Y} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 35.3 & 29.7 & \dots & 28.6 \end{bmatrix} \begin{bmatrix} 10.98 \\ 11.13 \\ \vdots \\ 11.08 \end{bmatrix} = \begin{bmatrix} 235.6 \\ 11821.4 \end{bmatrix} \tag{22}$$

See how these matrices relate to the original data presented in Data Set 1. You will not be expected to perform the matrix algebra yourself, but you should at least know that it's possible just in case your computers are out of action for six months!

3.2 Least-squares assumptions

The procedure we have just gone through for fitting a straight line to the Steam Use data is a perfectly satisfactory way of fitting a line to any data (linear or otherwise) if the aim of the exercise is only to produce an equation that more or less follows the data. However, in most cases we wish to go further than this and make statistical statements about the adequacy or otherwise of the equation, and about the precision of the parameter values and the prediction from the model (That is, what sort of +/- range can we reasonably ascribe to them). The mathematics that enable us to do this is based on a number of assumptions about the model and the data. Statistical results related to significance levels, confidence intervals etc. are only valid if the assumptions are more or less satisfied. It is very easy to forget this when deluged by a large quantity of six-digit numbers from a computer with impeccable credentials! The relevant assumptions are:

1. The x values are fixed and known exactly — The x values are set at precisely known values, and only the consequent y's are subject to random error. This assumption is often strained when looking at historical data based on observation, rather than manipulation of the system producing the data. In many cases one is not even sure which variables to regard as predictors and which should be treated as response! It is worth noting in connection with this point that a reversal of x and y roles also gives a different equation in most cases. In practice, the requirements for this assumption can be relaxed somewhat, so long as the random error variance in x is appreciably less than that in y.

2. $E(\epsilon_i) = 0$, for all i — It is assumed that we have the correct (true) model form, so that there is no bias in the model. Fitting a straight line to data with quadratic curvature would be one way that this assumption would be violated. In such a case the model form is simply not appropriate. Detection of the problem for the case of only one x variable is not difficult, but where there are a number of predictors it may be less obvious.

3. $E\left(\epsilon_i^2\right) = \sigma^2$, for all i — The random errors which contaminate the y values all have a constant variance. This means a constant amount not a constant percentage. It is not hard to find situations where this assumption does not hold. For example, in instrument calibration curves, it is common to find that the percentage error is approximately constant. In such a case the least squares procedure will pay more attention to the points with greater variance. The correct approach when this problem exists is either to used weighted least squares (where each residual has an associated weighting factor) or to transform the y values into another metric (e.g. log, square root reciprocal etc.) where the assumption is satisfied. As a case in point, if the random error in y is a constant percentage of y, then the random error in $\ln y$ will be a constant amount.

4. $E(\epsilon_i \epsilon_j) = 0$, for all $i \neq j$ — The covariance or correlation between different random errors is zero. That is, the random errors are independent and not associated with each other in any systematic fashion. The assumption is of doubtful validity when the data are a time series (e.g. sequential points taken during a batch chemical reaction).

5. The random errors are normally distributed (also called *Gaussian*).

As was previously mentioned, the least squares criterion is only one of several that could be used for fitting equations to data. However, if the above assumptions are valid, least squares does have a number of advantages from mathematical and statistical points of view. The statistical statements about significance of the result, confidence intervals for parameter estimates and model predictions etc. are based on the above five assumptions. This last assumption is usually reasonable but there are many well-known cases where this does not hold.

4 Testing the adequacy of a fitted model

One of the last things to do with a fitted model is to believe it. A fitted model and the related statistical statements derived from the fitting process can only be accepted as valid if some effort has been made to confirm that the assumptions just discussed are reasonable for the data and model concerned.

Model adequacy can be tested in two ways. The first (qualitative) method is to create the plots for a *residual analysis*, and the second (more quantitative) method is to use numerical approaches like one-way ANOVA or correlations. ANOVA is the abbreviation statisticians use for *analysis of variance* and is covered in more detail in Section 4.2.

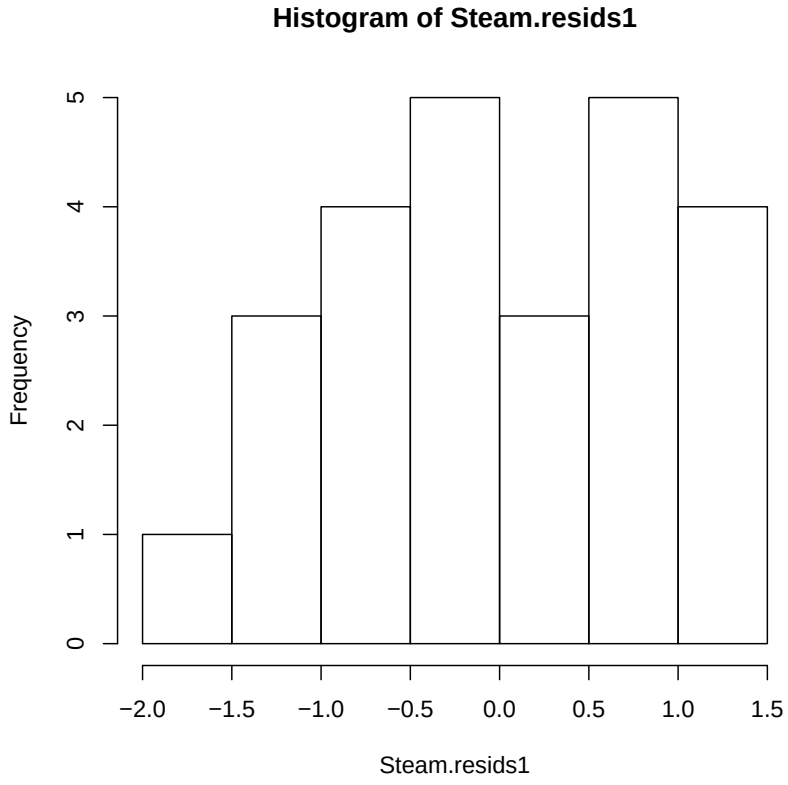
4.1 Residual plots

Residual plots should always be examined before a model is accepted as adequate. The basic idea is that the model should:

1. Satisfy all the necessary assumptions.

Exhibit 4 Histogram of the residuals left over after fitting a model for Steam Use based on the Average Temperature.

```
> hist(Steam.resids1)
```



2. Explain all the systematic variation in the data.

For an adequate model many residual plots should resemble a piece of paper that was held behind a letterbox slot when a shotgun was fired at it. Any deviation from this pattern indicates that there are problems with either the model or the data. The following examples illustrate some of the ways this can happen. Note that when we want to look at the residuals from a model when using R, we must first obtain the residuals using the `resid` command, for example:

```
> Steam.resids1 = resid(Steam.lm1)
```

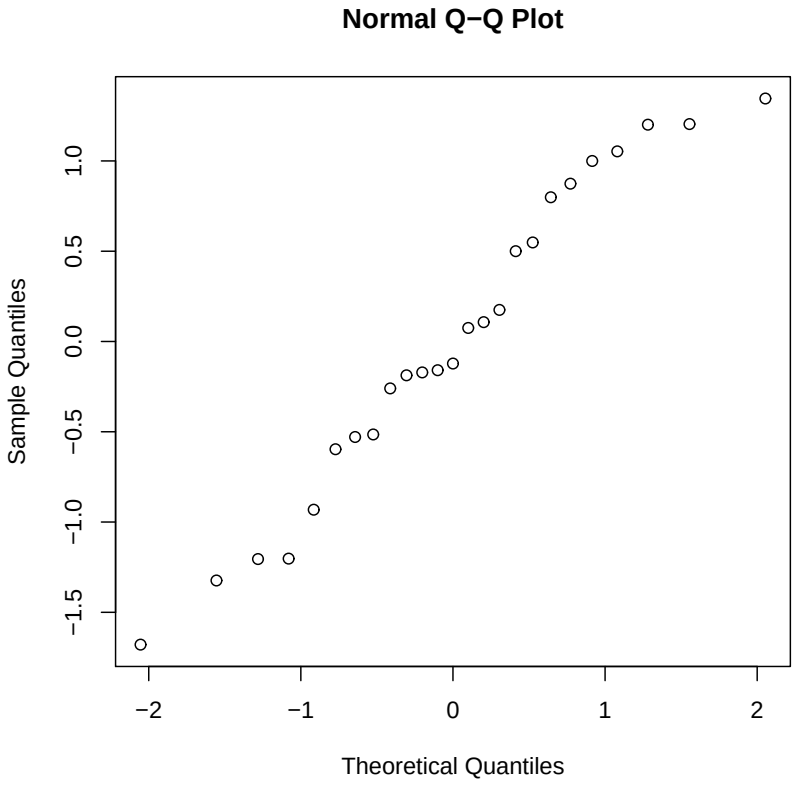
This will save time as we now just ask R to generate the appropriate plots for the stored set of residuals.

Non-normal residuals

Remember that we wish to have normally distributed and independent residuals from fitting any model. A histogram of the residuals may show skewness (lack of symmetry), but a normal probability plot can also be produced which will highlight problems, especially if there are outliers. We use the `hist` command to create the histogram and the `qqnorm` command to create the normal probability plot seen in Exhibits 4 and 5 respectively. If there is evidence that the residuals are not normally distributed, then we need to either transform our variables or look for further enhancements of our model.

Exhibit 5 Normal probability plot for the residuals left after fitting the model for Steam Use dependent on the Average Temperature.

```
> qqnorm(Steam.resids1)
```



Outliers

From Exhibits 4 and 5 we can see that no outliers have been exposed by just considering the residuals themselves. In practice, this is unusual to say the least because data have a strange habit of not being neat and tidy. The Steam Use data set is moderately sized. When working with realistic data sets you'll be confronted with small and large quantities of data that pose different problems. Obviously a small sample of results will not be suitably plotted on a histogram, while a large data set is likely to throw out some real points that look unusual.

On occasion, you'll find that a few residuals may be very large compared to the others, indicating that the predictions at these points are a long way from the observed values. When you used Minitab to fit a model to data in the past, the software highlighted any points that had unusual residuals. Recall that Minitab flags all points with residuals more than two standard deviations away from zero, and further that we actually expect around 5% of points to exceed this arbitrary criterion anyway. There is a (understandable) tendency to simply discard these points because the reading must have been wrong. The temptation to go this way should be resisted unless a concrete physical reason for the discrepancy can be identified.

One problem with *true* outliers that will often not show up on residual plots is that they may also be at *influential points* in the data set. For example, if a straight line is being fitted to data mainly in the range of 0-5 x-units and there is a single reading at $x = 10$ which happens to be in error then this y value will 'pull' the whole line in the

Exhibit 6 Creation and examination of the leverages for the first `Steam` example.

```
> Steam.lev1 = hatvalues(Steam.lm1)
> plot(Steam.lev1, Steam.resids1, xlab="Leverages", ylab="Residuals")

> 2*mean(Steam.lev1)

[1] 0.16
```

direction of the error and not show as an outlier. If you took the time to actually view a scatter plot of the data you are trying to model, this problem would have been noticed before you finished collecting data.

There are other (more advanced) ways of identifying the influence individual points are having on the model fitting process. R (and other good statistical software) can calculate these other measures including:

- h_{ii} — the *leverage* put on the model by individual observations.
- DFITS.
- Cooks D. — a measure of influence based on both leverage and the size of residuals.

You do not need to calculate these measures yourself. We will look at leverage and Cook's distance in particular because they form part of R's default analysis. Knowledge that other measures exist is useful so that you could look them up when/if required.

Leverage is based on the difference individual observations have from all other observations in the data set. An observation with high leverage forces the model to be closer to it at the expense of the overall fit. Such points are therefore likely to have smaller residuals as a consequence. A useful rule of thumb to say that observations are having high influence on the model is when an observation has more than twice the average influence. Exhibit 6 shows the use of the `hatvalues` command in the working for the `Steam` data example. Cook's distances are found using

$$D_i = \frac{r_I^2}{\sum_{i=1}^{I} h_i} \frac{h_{ii}}{1 - h_{ii}} \tag{23}$$

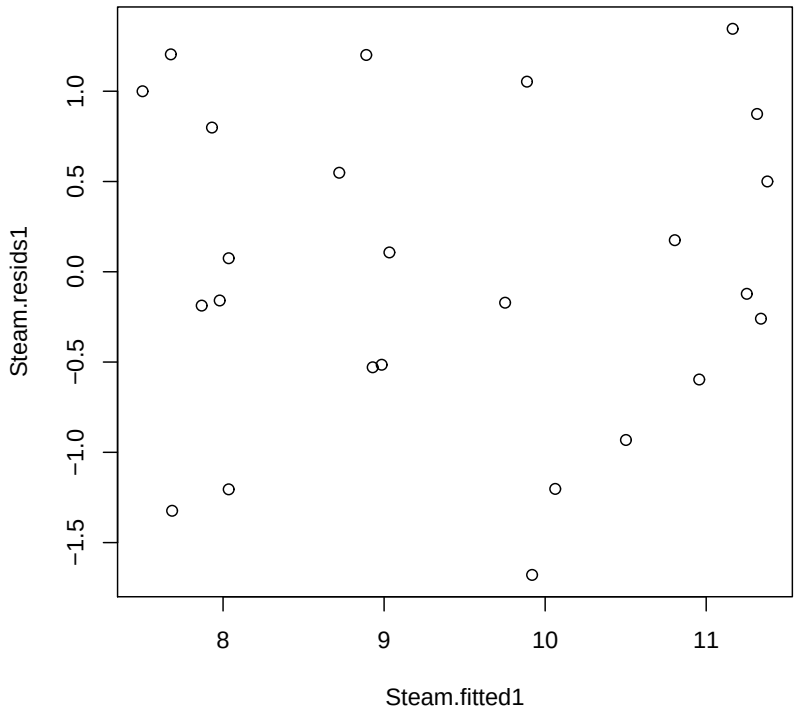
The summation in the denominator means that whenever we use Cook's distance as a gauge of influence, we can always use the same rule of thumb. A Cook's distance of greater than one is a sign that an observation has influenced the fit of the model. Cook's distances are often graphed but we will see a text based approach for identifying any influential observations. R uses the `cooks.distance` command to generate the scores and we then apply suitable subscripting to see if any observations have undue influence using:

```
> Steam.Cooks1 = cooks.distance(Steam.lm1)
> Steam[Steam.Cooks1>1, c(1,8)]

[1] SteamUse Temp
<0 rows> (or 0-length row.names)

> max(Steam.Cooks1)

[1] 0.1523
```

It seems that no observations have undue influence on the simple linear regression model we've been working with for the `Steam` data. In fact we can see that the highest Cook's distance was considerably far from one in the output above.

Exhibit 7 A plot of residuals versus fitted values found after fitting the linear model for Steam Use based on Average Temperature.

```
> Steam.fitted1 = fitted(Steam.lm1)
> plot(Steam.fitted1, Steam.resids1)
```



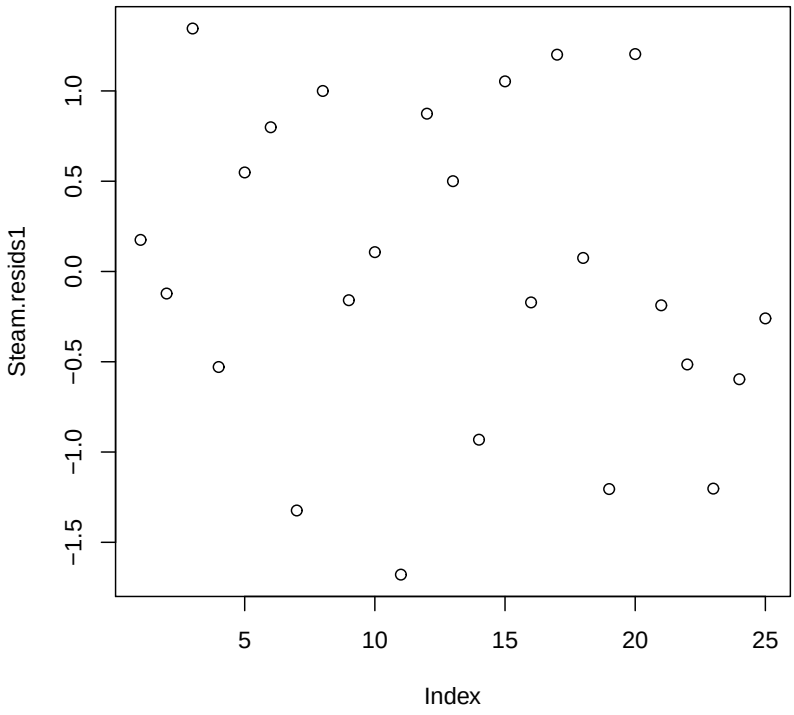
So what happens if we do find a point with undue influence on our model? Removing a point that does have undue influence on the model is a possible strategy. You should question this act, and the impact that it has though. Is it the model fitting process that has a problem, or the data collection; or is there something strange in the phenomenon you are attempting to model? Experimentation with these ideas might be the only way to make a decision in the end. There is no hard and fast rule to follow in this respect. If you find that the points at one end of your data are causing a problem you might reduce your focus and leave them out; of course your model is not appropriate for those points and you also know you can't make predictions for them either.

Non-constant error variance

A plot of the residuals against x or y will often show if the assumption of constant error variance looks to be reasonable for the data. If not, then we need to use weighted least squares (an advanced topic not covered in this course) or transform the y's (see Section 10). It is strongly advised however, that you use the `resid` and `fitted` commands to find the residuals and fitted values from the model, so that you can compare the constancy of residual variance. An example appears in Exhibit 7. Once you've determined that a set of residuals do have non-constant variance, then you would seek a cause for that non-constant variance. This *heteroscedasticity* may arise as a model is not fully specified (we need to add more terms into the model) or needs to be modelled differently.

Exhibit 8 Plot of the residuals found after fitting a model to explain Steam Use by a function of Average Temperature versus the order of the data.

```
> plot(Steam.resids1)
```



Time trend

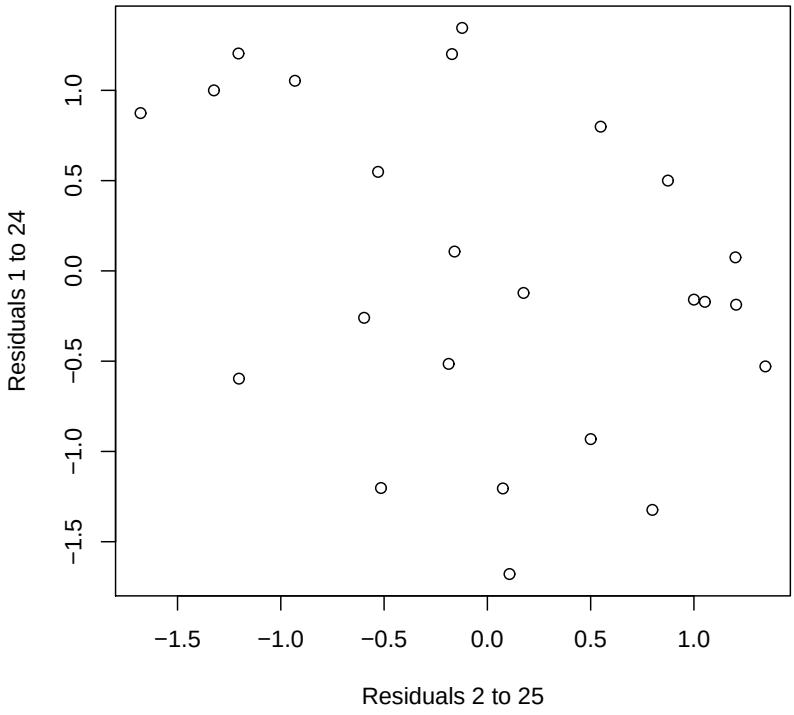
A serial plot of the residuals in the sequence that the data were gathered, such as presented in Exhibit 8, can sometimes reveal that some outside factor has been varying gradually over the period. If this is the case then it is necessary to find what that factor might be and include it in a revised model. If you know this sort of problem is likely to affect any future experiment, you should randomize the order of the runs in which you collect data. Finding a way of modelling any time trend can often be difficult.

Patterns in the residuals versus order plot may also indicate that errors are cumulatively affecting the predicted response. In some observational experiments where data is recorded at evenly spaced intervals that are too close together, the error from one observation may not have been 'washed out' of subsequent observations. See the notes on dealing with this situation in Section 11.

You can rest assured that a formal test for this is available — it's called a Durbin-Watson test. You don't need to know how to use it at this stage, but you should investigate further if you believe some serial correlation exists in your data. If you think serial correlation might be a problem, you should plot each residual against its preceding residual. This simple diagnostic graph is given for the `Steam` data in Exhibit 9. The moderately large negative correlation of –0.354, found using the `cor` command, between residuals and preceding residuals suggests that successive observed values oscillate around the fitted values throughout the twenty-five months of our data collection. This problem might be resolved with an improved model but is unfortunately common in many situations where

Exhibit 9 Plot and correlation of the residuals found after fitting a model to explain Steam Use by a function of Average Temperature versus the residual for the preceding observation.

```
> plot(Steam.resids1[-25], Steam.resids1[-1], xlab= "Residuals 2 to 25", ylab="Residuals 1 to 24")
```




```
> cor(Steam.resids1[-25], Steam.resids1[-1])

[1] -0.3542
```

data are collected over time.

Missing terms

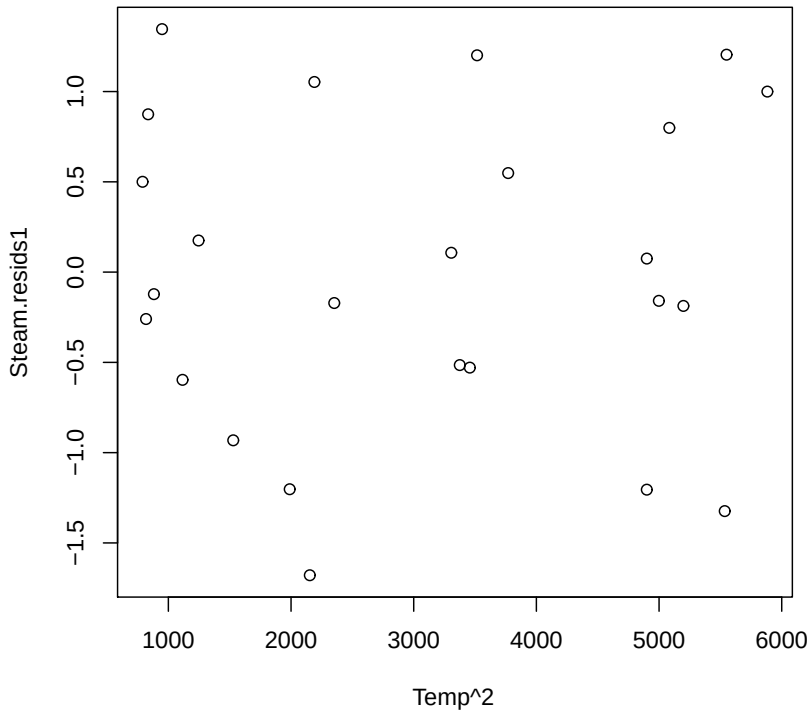
Sometimes a model with a term x_i also needs a term in x_i^2. In this case, a plot of the residuals against the affected x variable will show curvature, and a plot of the residuals against the squared term would show linearity. The residuals from the linear function of average temperature are plotted against the average temperature squared in Exhibit 10. Note that in creating this graph, we have employed the `attach` and `detach` commands. They give us direct access to the variables in a data set by name. Their use can save typing, but it is not necessary to keep attaching and detaching all the time. It is done here for completeness.

Missing variables

At times you'll fit a model that doesn't include all the variables that should be in the model. We call these *lurking variables* and attempt to discover their existence (if possible) and include them in the model.

Exhibit 10 A plot of the residuals found after fitting the linear model for Steam Use based on Average Temperature, plotted against the Average Temperatures squared.

```
> attach(Steam)
> plot(Temp^2, Steam.resids1)
> detach(Steam)
```



A plot of the residuals against any related variables not in the model should have no structure. The first model for Steam Use only used Average Temperature as a predictor, but other variables are actually available (see the full data set on page 64). Most of the other variables available are fairly discrete, so the variable `Wind` has been used for illustration in Exhibit 11. If this graph showed any systematic pattern you should consider adding the predictor to the model and starting the residual analysis over.

If you felt that your model was insufficiently explaining the response variable, you'd be well advised to look at these plots. It's easier to see what of the currently unexplained variation in the response variable can be explained by lurking variables than it would be to estimate the strength of a secondary relationship between lurking variables and the response.

Exercise 4 *Use R to investigate the assumptions of linear regression using the model fitted to the data for the hardness of cement batches (found in cement.csv or on page 65). What (if any) problems do you observe with your model as a result? (Answer on page 58)*

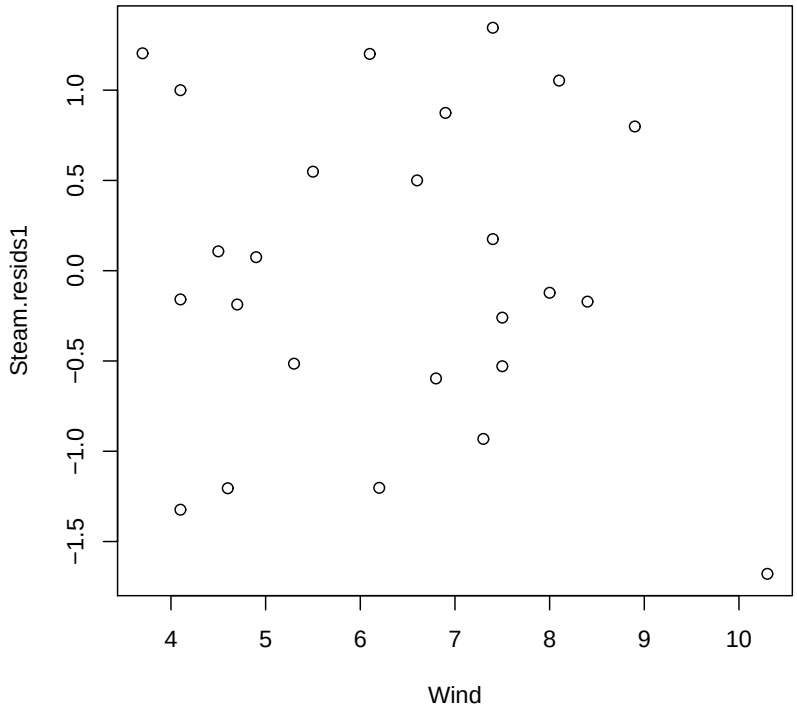
4.2 A formal test for lack-of-fit

The residual term ϵ in a fitted model originates from two sources:

1. The net effect of all inherent fluctuations and disturbances from 'unassignable causes' (otherwise known as random error or *pure error*).

Exhibit 11 A plot of the residuals found after fitting the linear model for Steam Use based on Average Temperature, plotted against another potential explanatory variable (Wind).

```
> attach(Steam)
> plot(Wind, Steam.resids1)
> detach(Steam)
```



2. The difference between the fitted model and the 'true' (but unknown) model form that really represents the process (otherwise known as *systematic error* or *lack-of-fit* error).

To do a formal lack of fit test we need independent estimates of these two errors. To estimate pure error we need some repeated points in the data — these are known as *replicates*. The differences between results from replicates are only caused by random variation and have nothing to do with whatever model was fitted. The formal test is just a comparison of these sources of error to see if the lack-of-fit error is appreciably greater than the pure error. The logic is illustrated in the following example.

Data Set 2 presents results from an experiment that was set up to examine the corrosion resistance of a copper-nickel alloy with varying iron content. The experiment involved exposing standard samples to a corrosive environment under standard conditions and measuring the weight loss.

It can be read into R using the `read.csv` command as useed previously.

```
> CuNiAlloy <- read.csv("CuNiAlloy.csv")

> str(CuNiAlloy)

'data.frame':        13 obs. of  2 variables:
 $ WtLoss: num  128 130 128 124 122 ...
 $ Fe    : num  0.01 0.01 0.01 0.48 0.48 0.71 0.71 0.95 1.19 1.44 ...
```

Data Set 2 — CuNiAlloy.csv from Draper and Smith (1998)

wtloss	*%Fe*	*wtloss*	*%Fe*	*wtloss*	*%Fe*
127.6	*0.01*	*110.8*	*0.71*	*92.3*	*1.44*
130.1	*0.01*	*113.1*	*0.71*	*91.4*	*1.44*
128	*0.01*	*103.9*	*0.95*	*83.7*	*1.96*
124	*0.48*	*101.5*	*1.19*	*87.2*	*1.96*
122	*0.48*				

Exhibit 12 Regression output for the weight loss of a copper nickel alloy against its iron content.

```
> CuNiAlloy.lm1 = lm(WtLoss~Fe, data=CuNiAlloy)
> summary(CuNiAlloy.lm1)

Call:
lm(formula = WtLoss ~ Fe, data = CuNiAlloy)

Residuals:
   Min     1Q Median     3Q    Max
-3.983 -1.978  0.160  0.709  5.741

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   129.70       1.45    89.2  < 2e-16 ***
Fe            -23.83       1.33   -18.0  1.7e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.17 on 11 degrees of freedom
Multiple R-squared:  0.967,        Adjusted R-squared:  0.964
F-statistic:  323 on 1 and 11 DF,  p-value: 1.69e-09
```

Fitting a straight line gives the R output in Exhibit 12. From this and the fitted line plot in Exhibit 13, we can say that the relationship between the variables looks reasonably linear. Note the use of the `abline` command to add a straight line after using `plot` to obtain the scatter plot.

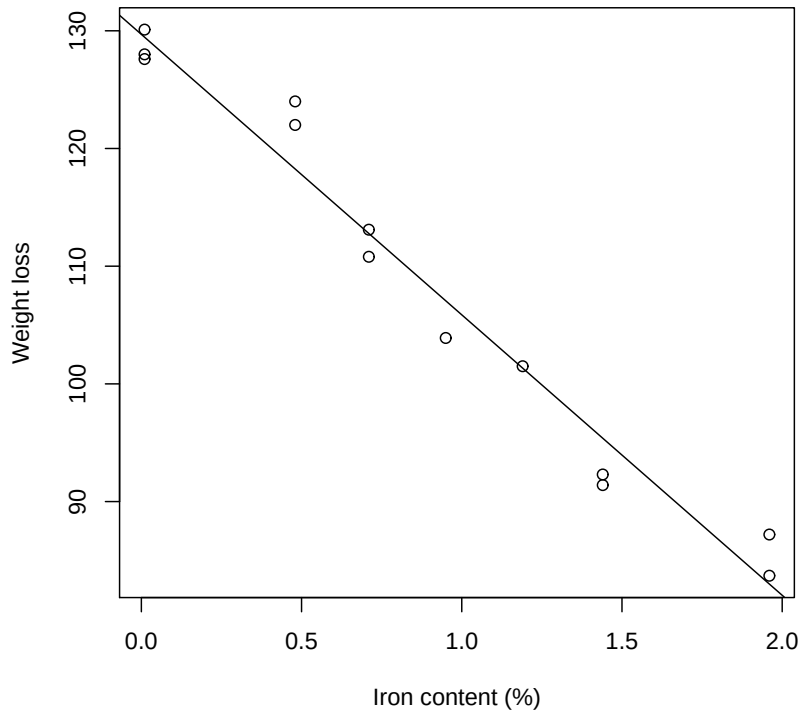
Closer examination of the data shows an anomaly. For most cases where there are replicated data values all the y's within the replicate sets fall on the same side of the line. Now one of the underlying assumptions about the fitted model is the variation of the observations about the line is due only to random error. Therefore we would expect an adequate model to go through the middle of replicate data most of the time.

The problem is more apparent when we look at a plot of the residuals from the model against the x's, given in Exhibit 14. The assumption that we have the 'right' model form means that we would expect residuals from replicate measurements to straddle zero in a majority of cases. Of the five sets of residuals from replicated experiments in these data, only one does this. In more formal terms, an adequate model should give residuals that have the same mean (zero) and a constant variance. For the purposes of a formal test we therefore ask what is the probability of getting the present sample if the original statement about mean zero and constant variance is true. To put it another way, suppose that we were sampling from a population that was supposed to be mean zero and constant variance and we got the following results:

1. Sample (i) — two values less than zero, one greater than zero.
2. Sample (ii) — two values considerably greater than zero and close together.

Exhibit 13 Fitted line plot for the weight loss of a copper nickel alloy against its iron content.

```
> attach(CuNiAlloy)
> plot(Fe, WtLoss, xlab="Iron content (%)", ylab="Weight loss")
> abline(CuNiAlloy.lm1)
> detach(CuNiAlloy)
```



3. Sample (iii) — one value zero, one less than zero.
4. Sample (iv) — two values considerably less than zero and close together.
5. Sample (v) — two values greater than zero.

We might begin to wonder whether all these samples really come from the same population. In terms of model fitting we would be doubtful that random error was the only cause of variation about the line.

The statistical tool for testing this type of hypothesis is the one-way analysis of variance (ANOVA). One-way ANOVA's are done when we have data in several groups corresponding to 'levels' of some external factor and we wish to test the hypothesis that the group means are all the same. It can therefore be considered an extension of a two-sample t-test. In this situation (testing lack of fit in a model) the levels are the replicate conditions in the data set, and the 'data values' are residuals from the fitted model. You can use the `aov` command to perform a one-way ANOVA on the residuals stored in one column and a tag to identify which are replicates in another column (The independent variable can be used, but only after converting to a factor using the `as.factor` command). The results from performing this one-way ANOVA are given in Exhibit 15.

What does all this actually mean? The residuals we were looking at fell into several groups, corresponding to the seven distinct experimental conditions used. Each group has

Exhibit 14 Plot of residuals versus the explanatory variable (iron content).

```
> CuNiAlloy.resids1 = resid(CuNiAlloy.lm1)
> plot(CuNiAlloy$Fe, CuNiAlloy.resids1, xlab="Iron content (%)", ylab="Residuals")
```

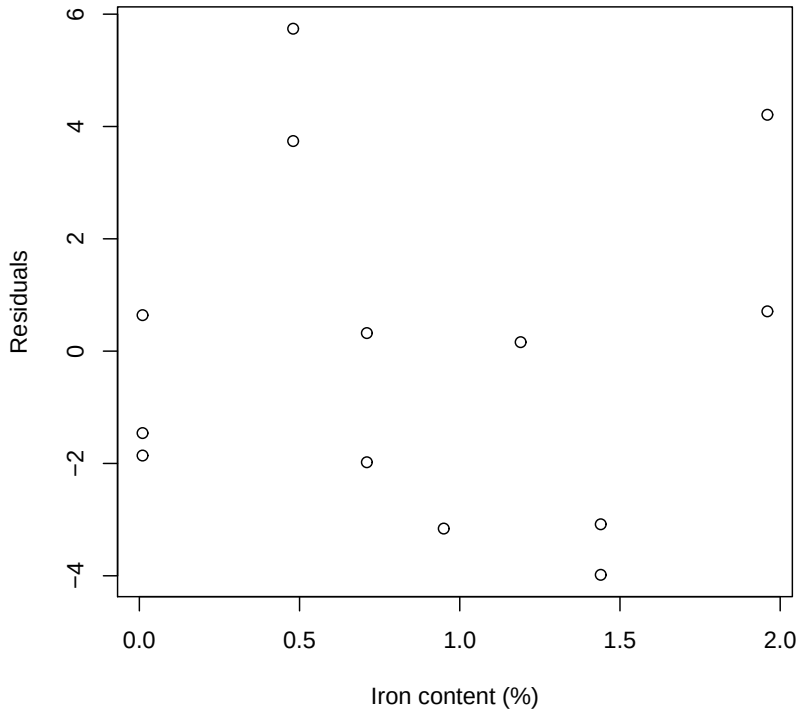


Exhibit 15 One-way Analysis of Variance

```
> CuNiAlloy.aov1 = aov(CuNiAlloy.resids1 ~ as.factor(CuNiAlloy$Fe))
> summary(CuNiAlloy.aov1)

                        Df Sum Sq Mean Sq F value Pr(>F)
as.factor(CuNiAlloy$Fe)  6   95.8   15.96    6.48  0.019 *
Residuals                6   14.8    2.46
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Exhibit 16 One-way Analysis of Variance to test the suitability of a linear predictor

	Source	DF	SS	MS	F	P
Analysis of Variance for residuals	Factor	5	95.77	19.15	7.77	0.013
	Error	6	14.78	2.46		
	Total	11	110.55			

a mean, a standard deviation and degrees of freedom. The *error mean square* is found by psumming the variances from each replicate group weighted by the number of degrees of freedom for each replicate group. The pooled standard deviation is just the square root of the error mean square. The factor mean square is another estimate of variance based on the means from each group, and the hypothesis that all groups have the same mean is tested by looking at the ratio of the two values,

$$\frac{\text{Factor Mean Square}}{\text{Error Mean Square}} \tag{24}$$

and comparing it with the appropriate F-value from tables. R calculates a p-value for this F-ratio for you, which indicates the probability that the null hypothesis of equal means is true.

When the ANOVA is being done with ordinary data, then the total degrees of freedom is $n-1$, that is one less than the number of data values. The degrees of freedom for error comes from the number of entries in each group, and the degrees of freedom for 'factor' is the number of levels minus one. The total degrees of freedom is the sum of the degrees of freedom for 'factor' and 'error' added together. This means we can often just use subtraction to get the correct degrees of freedom in each line of the ANOVA table. The reason for the total degrees of freedom being $n-1$ is that one item (the overall mean) has been estimated from the data in doing the analysis. When the data are residuals from a fitted model as in this case, the total degrees of freedom is $(n-(p+1))$, where $p+1$ is the number of parameters estimated in the fitted model. As before, the error degrees of freedom comes from the number of items in each replicate group, and the factor degrees of freedom is obtained by the difference. The sums of squares associated with each source of variation are unaffected, so the ANOVA table has to be modified slightly (by hand) to give the output in Exhibit 16.

The adjusted ratio of mean squares is 7.77, and we can use R to check the p-value using the `pf` command. We know that we have 5 numerator degrees of freedom and 6 denominator degrees of freedom.

```
> 1-pf(7.77, 5, 6)

[1] 0.01341
```

which when inserted into the adjusted analysis of variance table will be truncated considerably.

In practical terms, this means that there is significant lack-of-fit, the variation of the data about the line is significantly greater than the variation within replicate data sets. The initial feeling that having the replicates all fall on one side or the other of the line is confirmed by the formal significance test. This means that the fitted straight line does not adequately represent the data, and the regression information about standard deviations of parameters (in particular) has no meaning.

This process of using a one-way ANOVA is actually rather tedious and it is possible to use R's linear model functionality to investigate the problem of lack-of-fit instead. We will see that we use R's management of the degrees of freedom to our advantage here. First observe that we can see how many levels of iron are observed in our small data set using:

Exhibit 17 Regression Analysis

```
> CuNiAlloy.lm3 = lm(WtLoss~as.factor(Fe), data=CuNiAlloy)
> anova(CuNiAlloy.lm3)

Analysis of Variance Table

Response: WtLoss
              Df Sum Sq Mean Sq F value  Pr(>F)
as.factor(Fe)  6   3338     556     226 8.5e-07 ***
Residuals      6     15       2
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> CuNiAlloy.lm4 = lm(WtLoss~as.numeric(Fe) + as.factor(Fe), data=CuNiAlloy)
> anova(CuNiAlloy.lm4)

Analysis of Variance Table

Response: WtLoss
               Df Sum Sq Mean Sq F value  Pr(>F)
as.numeric(Fe)  1   3242    3242 1315.86 2.9e-08 ***
as.factor(Fe)   5     96      19    7.77   0.013 *
Residuals       6     15       2
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> levels(as.factor(CuNiAlloy$Fe))

[1] "0.01" "0.48" "0.71" "0.95" "1.19" "1.44" "1.96"
```

The `levels` function prints the different levels of a factor; in the analyses above we have assumed that our predictor is numeric. In what follows we will force R to see our predictor as we want it seen.

When we fit a predictor to a model while forcing it to be a factor, using the `as.factor` command, the model will be fitted as the highest order polynomial possible. For our example there are seven different iron levels to examine. Recall from your knowledge of one-way analysis of variance that this means that iron fitted as a factor will use up six degrees of freedom. We saw already that fitting iron as a linear predictor in a simple linear regression model uses up only one degree of freedom. The remaining five degrees of freedom are actually the nonlinear component we are after in this search for any lack-of-fit.

To illustrate the point, see the two sets of output presented in Exhibit 17. The first model uses iron content as a factor and includes both the linear and nonlinear components of the predictor. This corresponds with the output given in Exhibit 15 above, and is presented for illustration only. See how the sums of squares for the predictor of 3337.5 is split into two components in the second ANOVA table. The linear predictor explains a sum of squares of 3241.8, leaving 95.8 as the sum of squares for the nonlinear elements of the iron content; check that $3241.8 + 95.8 = 3337.5$ once rounding to one decimal place is taken into account. More importantly, notice that the p-value for the nonlinear component of the factor version of iron content is significant, indicating that the linear model is not sufficient. This value and its associated F-value are the same as for the manually adjusted ANOVA of Exhibit 16, although we haven't introduced any rounding errors this time. Also note that we used the `as.numeric` command to force the `FE` variable to be a linear predictor for clarification; it isn't needed as this variable was given a single degree of freedom in its original form.

Lack of fit generally comes about because the equation fitted to the data is not sufficient i.e. it needs more terms. These may take the form of additional terms in the x variables already used (e.g. addition of quadratic, cubic etc terms) or terms involving ex-

tra x variables. The latter is a more likely explanation of the lack of fit in this corrosion example.

It should be apparent from all this that (strictly speaking) a lack-of-fit test can only be done if the data contain replicate observations. This requirement can be relaxed somewhat for 'near-replicates' by arbitrarily grouping residuals into categories for the ANOVA approach. When designing an experiment you should of course include replicates wherever possible.

There is one other point to be made about replicated data, and that is that replication involves more than just taking the same reading twice. For example, if the replicate corrosion data in the above example were simply obtained by dividing one sample into two or three for analysis, then this would very likely give an estimate of random error variability that was too small. Replication of a data point in experimental studies involves doing the whole process of setting up a measurement from scratch each time. Too many supposed replicates in data sets are simply repeat readings.

5 Over-fitting and under-fitting

A model is said to underfit the data when a lack-of-fit test or residual plots show that additional terms are needed to give an adequate fit. In the case of the corrosion loss data, we probably need data on some of the other factors which were varying in the experiments e.g. temperature or something like that.

When there are a large number of possible x variables to put into a model there is a strong temptation to fit a model containing all (or at least many) of them so that every independent variable has a chance to influence the prediction.

The product of such an approach is often an overfitted model i.e. one where an adequate model could have been obtained with fewer independent variables. There are a number of penalties for overfitting which will be discussed in more detail later. Not the least of them is that the insertion of extra (unnecessary) terms in a model needlessly inflates the variance associated with predictions from that model.

The model using a factor as a predictor is another example of over-fitting. Our model is perfect for iron levels when only one observation was observed in the corrosion example. This is quite clearly flawed as we know that our observations are likely to be measured with error and this 'overfitted' model leads us to believe several of the residuals should be zero.

Since overfitting (like underfitting) has its problems, it is important that only the terms which are genuinely needed in a model should be included. This poses an interesting problem for the analyst confronted with historical data where there are a large number of x variables. The problem appears in a particularly acute form when analysing something like process plant log-sheet data (typically everything under the sun is written down) with the aim of building a model to predict yields or something similar from independent variables.

Similar problems are common in economics when trying to predict inflation, unemployment etc. from various leading indicators. Attempts to build a model by simply throwing in every variable that might be useful usually come unstuck. Unfortunately, there is no unique solution to the problem of selecting which variables belong in the model unless we try all possible subsets of x variables.

5.1 Not enough terms in the model

In a preliminary investigation, you would usually use scatter plots to find which variables of all those collected are related to the response variable of interest. Plots are often of

dubious use when there is more than one x variable because the data frequently contain no sets where the x variable of interest has been varied alone. It can be quite time-consuming to evaluate the usefulness of additional terms in the model by investigating the relationships between the residuals of your current model and all possible explanatory variables that are currently unused. There is however a procedure that can be used to help with this problem. We can test whether a particular term is needed in a model subject to the assumption that all the other terms are needed. Consider again the general linear model

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_p x_p \tag{25}$$

and denote the residual sum of squares from fitting this model as SS_{p+1}. The way to test whether a particular term is useful in the model is to see what happens to the sum of squares of residuals when it is deleted. Obviously if this sum of squares increases by a large amount then the term was necessary, otherwise it was not. Now it is an unfortunate fact of life that deleting any term from a model will always give an increase in the sum of squared residuals, so we need a formal test as to whether this increase was significant or not.

The first step therefore in deciding whether a term is needed or not is to fit a reduced form of the model without the term of interest. This model will have p parameters and a sum of squared residuals SS_p. The purpose of the significance test that follows is to determine whether the change in SS,

$$D = SS_{p+1} - SS_p \tag{26}$$

is 'large' or not. Obviously, the first problem is "what do we measure it against". As is so often the case in statistics, the yardstick to use is an estimate of the random error, here denoted s_{pe} as it is an estimate of the variance of the pure error. If this change is large relative to the random error then the term is needed, and conversely if it is small the term is not needed.

To formally test the significance of the term we form the ratio

$$Q = \frac{D}{s_{pe}^2} \tag{27}$$

and compare it with the tabulated F value for 1 numerator degree of freedom and v denominator degrees of freedom, where v is the number of degrees of freedom associated with the estimate of variance (found as the error df in the ANOVA).

If a variance estimate based on replicates is not available then it is customary to use

$$s_{pe}^2 = \frac{SS_{p+1}}{n - (p+1)} \tag{28}$$

which is the sum of squares from the model with more parameters divided by the number of data points minus the number of parameters in that model.

If we look on the simple exercise of calculating a standard deviation as using the residuals from a model

$$\hat{y} = \hat{\beta}_0 \tag{29}$$

then we can see this method for getting a pure error variance estimate is totally consistent with what was done in elementary statistics for estimating a variance. The quantity (number of data points minus the number of parameters estimated) is the degrees of freedom associated with the variance estimate.

If the Q-ratio is greater than the tabulated F-value, then the term under examination is required in the model; conversely if it is smaller, then the term can be left out. The

Data Set 3 — River.csv

Flow	*Depth*	*Flow*	*Depth*	*Flow*	*Depth*
0.636	*0.34*	*0.487*	*0.29*	*5.89*	*0.73*
0.319	*0.29*	*0.924*	*0.41*	*1.979*	*0.46*
0.734	*0.28*	*7.35*	*0.76*	*1.124*	*0.4*
1.327	*0.42*				

rationale for this test is quite simple. If the term under examination is x_k, then the change in sum of squares D is an estimate of the amount of variation in the y's explained by the term $\beta_k x_k$ in the model. We than measure this against the random (i.e. unexplained) error variation to see if it is significant. We need this formal test because in general, the addition of any extra term (e.g. the wholesale tea price on the 1933 shanghai market) will decrease the sum of squared residuals by a perceptible amount.

Two very important points should be noted about this procedure for testing the significance of individual terms in a fitted model. The first is that the judgement on whether the term $\beta_k x_k$ should be included in the model is based on the assumption that all the other p terms will be included. The results of testing the significance of the $\beta_k x_k$ term in the presence of a different subset of predictors may be quite different! The second point is a related one — although it is assumed that the other p predictors belong in the model, this test on one term (known as a partial F-test for those who collect such information) does not test whether they 'belong' or imply that they do!

Another way of stating these points is that there is no unique value for either the coefficients in the model or the 'variation explained' by each term in the model when one is looking at different subsets of predictors. In general, when a term is added to or deleted from the model, all the other coefficients will change (along with the corresponding D-values). This problem is the reason why regression applied to plant data can often give confusing and erroneous results and will be dealt with in more detail when we discuss stepwise regression later (see Section 15.1).

We will now look at an example that illustrates the methods outlined and points made. The data which follows (from the Minitab Student Handbook) are measurements of the depth (m) and stream flow (m^3/hour) in a river. It is rather tedious to measure the flow in a real river (it involves scanning the whole cross-section with a velocity meter and integrating the results) but since there must be a relationship between the depth and the flow it would be simpler to obtain the flow data once, deduce the flow verse depth relationship and base future flow figures simply on a depth measurement which is much simpler to make. The data are presented in Data Set 3.

After ensuring the file `River.csv` is in your working directory, the data can be read in and investigated using the following commands.

```
> River <- read.csv("River.csv")

> str(River)

'data.frame':        10 obs. of  2 variables:
 $ Flow : num  0.636 0.319 0.734 1.327 0.487 ...
 $ Depth: num  0.34 0.29 0.28 0.42 0.29 0.41 0.76 0.73 0.46 0.4
```

The scatter plot of the data given in Exhibit 18 shows curvature, but we will fit a straight line to demonstrate what is involved in testing model adequacy and the need to include further terms in the model. The regression output obtained from R is presented in Exhibit 19. As we might expect from the original plot of the raw data, the plot of the residuals against x (depth), given in Exhibit 20, shows that this straight line model is

Exhibit 18 A plot of river flow versus depth (x_i).

```
> plot(River$Depth, River$Flow, xlab = "Depth", ylab = "Flow")
```

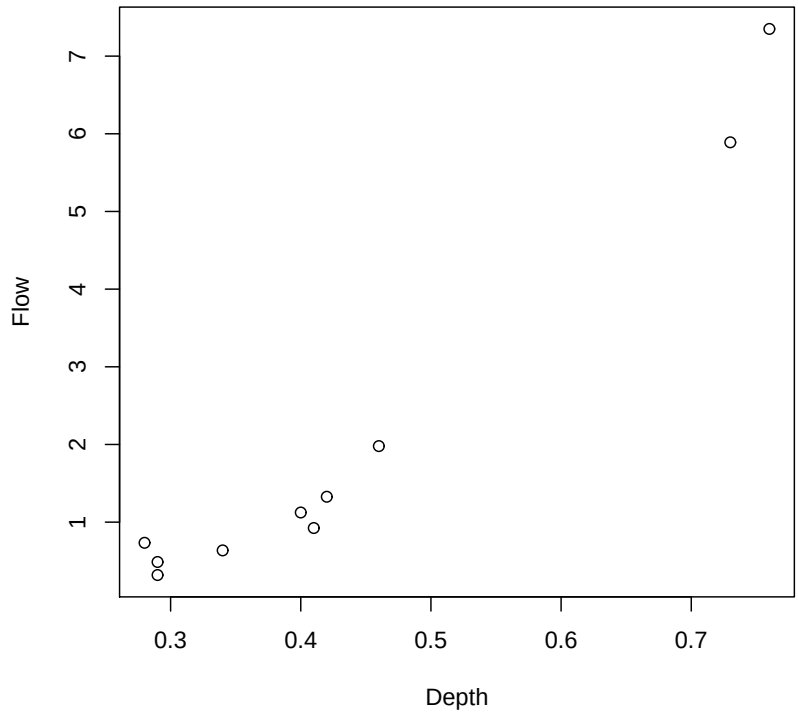
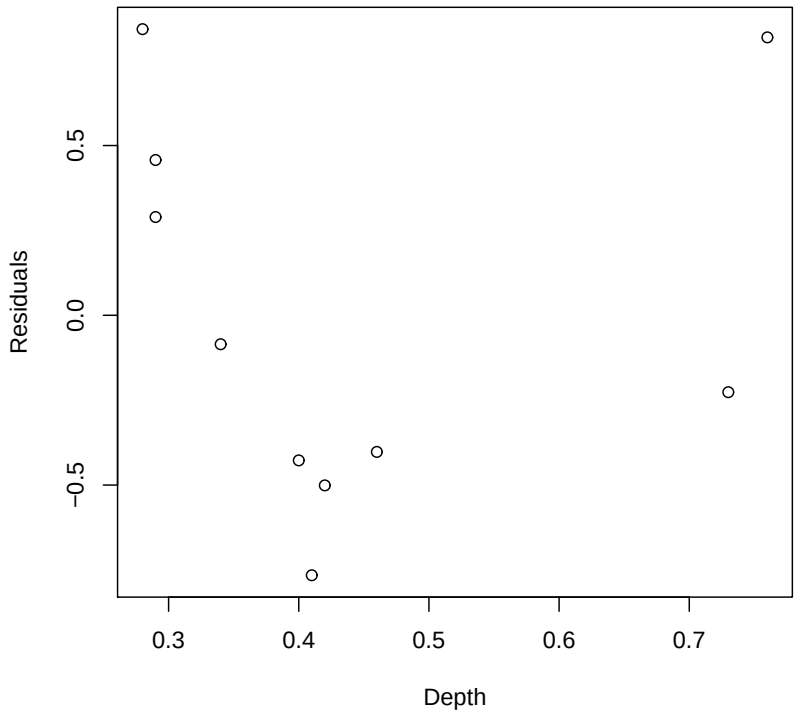


Exhibit 19 Regression Analysis for the `River` data

```
> River.lm1 = lm(Flow~Depth, data = River)
> summary(River.lm1)

Call:
lm(formula = Flow ~ Depth, data = River)

Residuals:
   Min     1Q Median     3Q    Max
-0.766 -0.421 -0.156  0.415  0.843

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -3.982      0.543   -7.33  8.1e-05 ***
Depth         13.834      1.161   11.92  2.3e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.603 on 8 degrees of freedom
Multiple R-squared:  0.947,        Adjusted R-squared:  0.94
F-statistic:  142 on 1 and 8 DF,  p-value: 2.26e-06
```

Exhibit 20 A plot of the residuals from fitting a linear function of river depth to explain the flow rate against the depth of the river.

```
> plot(River$Depth, resid(River.lm1), xlab="Depth", ylab="Residuals")
```



manifestly inadequate. The residuals are not random and the curvature suggests that we need a quadratic term in depth as well.

Fitting a model with a quadratic term included gives the output in Exhibit 21. Notice that all the coefficients in this model are considerably different from the ones in the straight-line model! The residual plot from this model, given in Exhibit 22, now looks quite tidy, so it is obvious from non-statistical considerations that the quadratic term was necessary.

We will now look at a partial F-test on the quadratic term to see if the result from formal statistics checks out with intuition. The reduction in the sum of squares from adding the quadratic term is 2.367, and the estimated variance of errors 0.078 is based on the residual sum of squares from the quadratic model.

The test now consists of comparing the Q-ratio with the tabulated F-value for 1 numerator degree of freedom and 7 denominator degrees of freedom. The appropriate table value is 5.59, therefore the additional term has clearly improved matters considerably i.e. the variation that it has 'taken up' is rather more then we might have expected by just putting a column of random numbers into the model as a possible predictor.

All this is rather painful so it is fortunate that R has a more convenient method for dealing with the comparison of two models like this. The `anova` command can compare two (or more) models that are *nested*; this is when one model is a subset of another model. Exhibit 23 shows how this is done by R. We will see how this is extended to more than two models in due course.

Exhibit 21 Regression Analysis

```
> River.lm2 = lm(Flow~poly(Depth,2, raw=TRUE), data = River)
> summary(River.lm2)

Call:
lm(formula = Flow ~ poly(Depth, 2, raw = TRUE), data = River)

Residuals:
    Min      1Q  Median      3Q     Max
-0.4061 -0.1637 -0.0026  0.1990  0.3277

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                    1.68       1.06    1.59   0.1561
poly(Depth, 2, raw = TRUE)1  -10.86       4.52   -2.40   0.0472 *
poly(Depth, 2, raw = TRUE)2   23.54       4.27    5.51   0.0009 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.279 on 7 degrees of freedom
Multiple R-squared:  0.99,        Adjusted R-squared:  0.987
F-statistic:  347 on 2 and 7 DF,  p-value: 1e-07
```

Exhibit 22 A plot of the residuals from fitting a quadratic form of a river's depth to explain the flow rate of the river against the river's depth.

```
> plot(River$Depth, resid(River.lm2), xlab="Depth", ylab="Residuals")
```

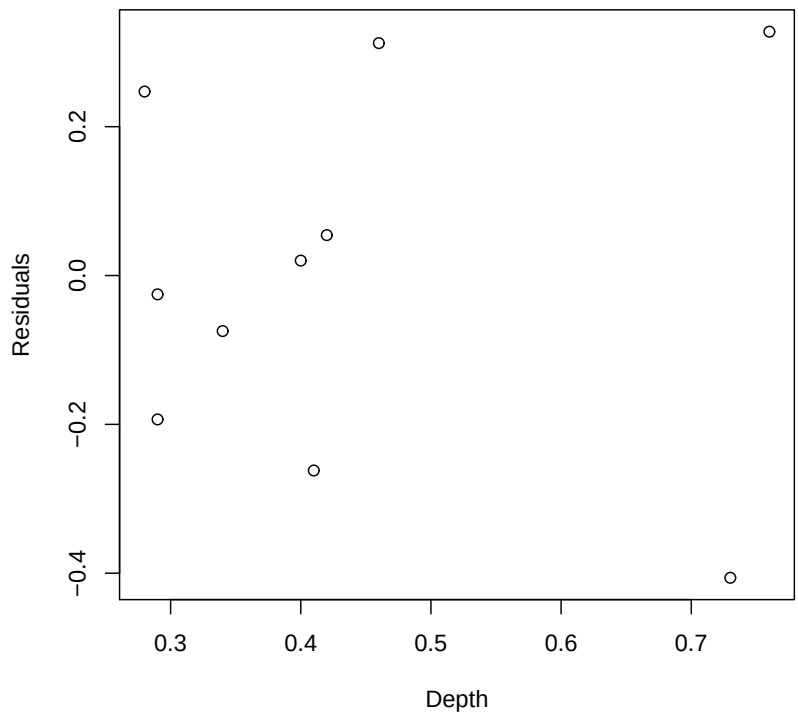


Exhibit 23 Comparison of the linear and quadratic models to explain `Flow` using `Depth` as a predictor.

```
> anova(River.lm1, River.lm2)

Analysis of Variance Table

Model 1: Flow ~ Depth
Model 2: Flow ~ poly(Depth, 2, raw = TRUE)
  Res.Df   RSS Df Sum of Sq    F Pr(>F)
1      8 2.913
2      7 0.547  1      2.37 30.3  9e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.2 Too many terms in the model

The first indication that a term is erroneously included in the model is that its p-value is too high. We usually want all p-values to be less than 0.05, so you'd start removing terms by dropping the worst term first (the one with the highest p-value). Once all p-values are at acceptable levels, you can then be sure that the collection of individual terms should be included in the model; it is possible that some of these terms are useful predictors only if the other terms are also included in the model. If you had any qualms about terms you could go through the process outlined above.

When too many terms have been included in a model, the estimates of variances of the other fitted parameters are overstated. We can calculate *variance inflation factors* or *VIFs* using an add-on package for R, to indicate variables that could be left out of our model. The output in Exhibit 24 shows what happens when we have too many variables in a model to explain Steam Use. Note that terms with high VIFs (> 10) can also have high p-values. This is not always the case (see Exhibit 24), and you could benefit from checking VIFs even when all terms seem significant. Note: Before issuing the `vif` command, you will need to have issued the `library` command sometime in the same R session. There are other reasons why this model is less than desirable but it has been presented for illustration even though it will not be recommended for use.

If you observe that variables have high VIF values then you might see what happens from dropping that term from your model. Pay special attention to the standard deviation of the residuals; this is important for the precision of all parameter estimates in your model.

6 Overall ANOVA for a fitted model

The partial F-test looked at in the previous section can be done for more than one term at a time in a model — we can test the significance of several terms as a group together. We shall see later that this can give the somewhat confusing result that a group of terms this way appear to be significant, but looked at individually some (or even all) of them do not. If we wish to test a group, say k of the $p+1$ parameters in a model then we look at the change in the residual sum of squares as before. We need to add the sequential sum of squares for each of the terms together and adjust the degrees of freedom in the F-test. We will then use k numerator degrees of freedom and v denominator degrees of freedom. This is of course just a generalization of the test we used before on a single term. One particular case of this test is done routinely by R (and most other statistical packages). This is a test of whether all terms in the model except $\hat{\beta}_0$ give a significant improvement over the model $\hat{y} = \hat{\beta}_0$. The material that follows here applies only to models with a constant term — for models without a constant this overall ANOVA is not possible.

The total sum of squares is simply the variation of the observed y values about their

Exhibit 24 Regression Analysis

```
> Steam.lm2 = lm(SteamUse~., data=Steam)
> anova(Steam.lm2)

Analysis of Variance Table

Response: SteamUse
          Df Sum Sq Mean Sq F value  Pr(>F)
Storage    1   9.37    9.37   26.05 0.00011 ***
Glycerin   1   1.83    1.83    5.09 0.03846 *
Wind       1  16.96   16.96   47.17 3.8e-06 ***
CalDays    1   0.73    0.73    2.02 0.17458
OpDays     1   5.57    5.57   15.49 0.00118 **
ColdDays   1  13.20   13.20   36.71 1.7e-05 ***
Temp       1   9.89    9.89   27.51 8.0e-05 ***
Startups   1   0.51    0.51    1.42 0.25157
Residuals 16   5.75    0.36
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> library(car)
> round(vif(Steam.lm2),2)

 Storage Glycerin     Wind  CalDays   OpDays ColdDays     Temp Startups
   14.78    16.90     2.20     1.83     3.99     4.69     5.27     2.22
```

Comments: Note that the use of the full stop on the right hand side of the model statement tells R to include all other variables in the model. Also note that the *car* package must be installed prior to using functions within the package.

overall mean, and we can partition this into two components via the regression sum of squares (the variation explained by the model) and the residual or error sum of squares representing the variation which has not been explained by the model. The use of a one-way ANOVA for testing lack of fit is a further partitioning of the residual sum of squares into pure error and lack-of-fit components. The fraction of the total variation explained by the model is often quoted as a single value called the *coefficient of determination* defined as

$$R^2 = \frac{SS_{\text{Regression}}}{SS_{\text{Total}}} \tag{30}$$

Look at any of the regression outputs in this manual to check that this is so. For those who are curious, R is the multiple correlation coefficient - the correlation between the observations y_i and predictions $\hat{y}_i$.

Returning to the question of an F-test for the model as a whole, the total sum of squares is simply what the sum of squared residuals would be from fitting the model $y = \beta_0$ ($\hat{\beta}_0$ would of course be the average, $\bar{y}$). The regression sum of squares is the decrease in the sum of squared residuals after all the terms have been put in, and our estimate of variance comes from the residual sum of squares.

We are testing a model here with three parameters (constant, `depth` and `depth`^{2} terms) against a model with only one (the constant). The residual sum of squares from the full model is 0.547, and this gives a variance estimate of $0.547/7 = 0.07807$ (given as the residual mean square in Exhibit 21). The change (decrease) in the residual sum of squares is 54.105 (the regression sum-of-squares) and since this involved two extra parameters the numerator of our Q-ratio is $54.105/2 = 27.053$ (the regression mean square). The conclusion about significance or otherwise of the model as a whole is therefore based on comparing the Q-ratio 27.05/0.078 (=346) with the tabulated F-value for 2 numerator DF and 7 denominator DF. The appropriate tabulated F-value for a 5% significance level is 4.73, so the variation explained by these 'extra' terms is highly significant. R prints the

p-value for you so that you don't have to work with the tables. You could also use R to find the critical value of the F-distribution instead of tables. In simple terms, we have explained enough of the data with the model to make the work of fitting it worthwhile. It should be stressed though that the test we have just done applies to the model as a whole; we cannot make any statements about individual terms from this result.

Exercise 5 *Re-examine the results of a straight line (linear) fit to the Steam Use data given in Exhibit 2, and compare it to the regression that also includes a variable for* `Temp`^{2}. *(Answer on page 60)*

7 Precision of the parameter estimates

One of the assumptions underlying the use of least squares for fitting models is that the measured y values are composed of two parts; a true value $E(y)$ and a random error ϵ. $E(y)$ is the expected value for y and can be thought of as the 'long term average' value that would be obtained from a large number of replicated measurements. A consequence of this assumption is that if the y values are subject to uncertainty or random error then the $\hat{\beta}$ values in the model must also be affected. In practical terms, if we were to go out and gather the same data several times then each set of results would yield slightly different $\hat{\beta}$ values. In statistical terms, there is a standard deviation associated with each estimated $\hat{\beta}$ which can of course also be translated into a confidence interval for the 'true' β value. Minitab prints this information out in every regression analysis so it is therefore possible to define a confidence interval for each of the 'true' values based on each of the estimated values.

Remember that for any statistic $\hat{\mu}$ with an estimated standard deviation s, a $100(1-\alpha)\%$ confidence interval is defined by

$$\hat{\mu} - t_{v,\alpha}s \leqslant \mu \leqslant \hat{\mu} + t_{v,\alpha}s \tag{31}$$

where $t_{v,\alpha}$ is the value in the t-tables for a significance level α and v degrees of freedom associated with the estimated standard deviation s. You learned this back in introductory statistics courses looking at confidence intervals for the mean of a distribution, but the result is perfectly general.

A $100(1-\alpha)\%$ confidence interval for the β values in a fitted model is therefore

$$\hat{\beta}_i - t_{v,\alpha}s(\hat{\beta}_i) \leqslant \beta_i \leqslant \hat{\beta}_i + t_{v,\alpha}s(\hat{\beta}_i) \tag{32}$$

The exact value of v in this formula is found as the total number of observations minus the number of parameters fitted in the model minus one. You would find this number in the 'df' column corresponding to the 'error/residual' line of the analysis of variance part of any regression output. For the Steam Use data we have 23 residual degrees of freedom (see the output in Exhibit 2). If we wish to create 95% confidence intervals for the two parameters we would use $t_{23,\ 0.05} = 2.07$, giving 95% confidence intervals for the parameters

$$\begin{aligned} \text{for } \beta_0 \quad & 13.6230 \pm 2.07 \times 0.5815, \qquad \text{and} \\ \text{for } \beta_1 \quad & -0.07983 \pm 2.07 \times 0.01052 \end{aligned} \tag{33}$$

Note that in each case the interval does not include zero. An alternative way to get the same result is to look at the t-test given by R. R evaluates the t-statistic for each parameter and compares it to the relevant tabulated t-value. For both these coefficients this ratio is greater in magnitude than 2.07, and is given a p-value of less than 0.0005, so the hypothesis that zero is a plausible 'true' value is rejected. Looking at the t-ratio in this fashion is

just a significance test of the Null Hypothesis that $\beta_i = 0$, with the alternative hypothesis that $\beta_i \neq 0$. i.e. an ordinary two-sided t-test. Testing the significance of a coefficient in this fashion is exactly the same as looking at the Q-ratio for that coefficient i.e. the partial F-test we have already examined in detail. In fact, the Q-ratio for a coefficient is the square of the t-ratio defined above. If you examine a table of F-values, then you will find that the column for 1 numerator degree of freedom is also the square of the corresponding value from the t-tables. All the assumptions about partial F-tests (e.g. all the other terms will stay in the model) also apply to significance tests on the coefficients done using the t-ratio. It also happens that the t-ratio can change as the model form is changed (just as the Q-ratio does). The advantage of using the t-ratio rather than the Q-ratio to test the significance of individual terms is that most regression packages (including R) routinely print these t-values in the analysis. If you prefer (for whatever reason) to use a Q-ratio and the F tables, then this can be done by simply squaring the printed t-ratios.

8 Precision of model predictions

The random error in measured y values that introduces uncertainty into the parameter estimates also has the flow-on effect of introducing error into model predictions based on these (contaminated) estimates.

R can be used to form prediction intervals for any combination of the x values of the p variables in the model. It would seem logical that predictions near the 'centre' of the data should be more precise than predictions that represent extrapolations. The further x is from the centre of the data the greater the variance of the predicted value ($\hat{y}$) and the wider any related interval estimates will be.

The graph in Exhibit 26 shows an 'envelope' of 95% confidence intervals for the mean predicted value, as well as the 95% prediction intervals from the Steam Use model. We can see that the precision of the results is highest (i.e. the narrowest intervals) near the centre of the data and that they get worse at the ends. We can also see how it is not a good idea to extrapolate a fitted model any great distance beyond the scope of the data on which it is based.

A point worth making about this graph is that eight (about a third) of the data points fall outside the 95% confidence envelope. This is because the interval calculated relates to the 'true' value of y, and not to the values observed experimentally. These are of course contaminated with random error.

The expression for the variance of a prediction gives an estimate (derived from the model) of the interval which is most likely to contain the 'true' y value to the specified confidence level. In this sense the 'true' y value can be thought of as being the long-term average that would be measured at a set of x conditions if many repeated measurements were taken. The interval represents our uncertainty about 'where the line is'. This interval is not the same as a 95% confidence interval for a single future measurement which will be called a *prediction interval*.

To obtain the interval for a single future measurement (or equivalently its variance) we have to allow for the fact that there are two sources of uncertainty in the value: the uncertainty about where the line is (as just discussed) and further random sampling error. The variance of a future measurement is therefore obtained by adding the variances associated with these two sources. These intervals have been produced (in numeric terms) using the `predict` command in Exhibit 25 for the average temperatures 30, 40, 50, and 60. Note that we must tell R which of the prediction or confidence intervals for the response we want, and that we create a new data object for these values using the `data.frame` command.

Exhibit 25 Regression analysis

```
> new <- data.frame(Temp = c(30, 40, 50, 60))
> predict(Steam.lm1, new, interval="prediction")

     fit   lwr   upr
1 11.228 9.287 13.17
2 10.430 8.532 12.33
3  9.632 7.753 11.51
4  8.833 6.949 10.72

> predict(Steam.lm1, new, interval="confidence")

     fit    lwr    upr
1 11.228 10.614 11.843
2 10.430  9.971 10.889
3  9.632  9.259 10.004
4  8.833  8.431  9.235
```

Note that the prediction interval is much wider than the confidence interval. Also observe the shape of the intervals and the way they get wider as the prediction is further away from the centre of the graphic in Exhibit 26.

In most cases with this type of calculation where the model is based on a reasonably large number of data values, the variance of a prediction makes up only a small part of the total uncertainty for future measurements and the variance of a future measurement is essentially equal to the pure error variance.

9 Bias in the estimated parameters

It has been mentioned several times that one of the least squares assumptions is that the chosen model form is the right one for the data i.e. there is no bias or systematic error in the predictions. If the model is correct, then the parameter estimates obtained are unbiased estimates of the coefficients in the model. If the model is not the correct one, then this is no longer true.

The extent of the bias depends not only on the nature of the true and postulated models, but also on the values of the x variables which enter the regression model.

If the empirical model includes the terms that would be included in the 'true' model the estimated parameters will be unbiased. An unbiased parameter estimate means that the expected value of the estimate is equal to the actual (true) value of the parameter, i.e. $E\left(\hat{\beta}_i\right) = \beta_i$.

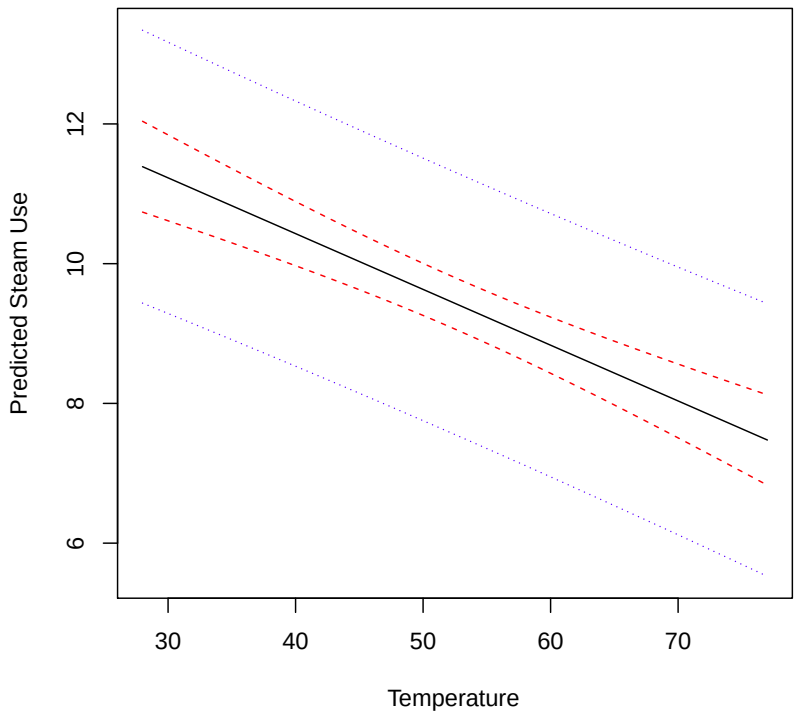
We need to be careful when including variables in the model as predictors are frequently partially or fully correlated with one another. This is especially true when data are from observational studies rather than designed experiments.

If we know that there is a chance that two variables affect the outcome of a third, and we wish to develop a model to describe the relationship, we would normally ensure that there is balance of the different combinations of the two predictors. If we could get these predictors to be completely uncorrelated (knowledge of one gives no indication of the other) we have what are known as orthogonal variables. The main strength of orthogonal variables is that the coefficient of one variable is not affected by insertion into the model of any variables that are orthogonal to the first variable. At this stage you might recall the term 'independent' from previous statistics courses. The estimation of the coefficients for orthogonal variables are independent of one another.

When explanatory variables are related to one another in some way, they will both explain a certain proportion of the variation in the response variable. Let's look at what

Exhibit 26 Prediction and confidence intervals for the `Steam` data, based on the output in Exhibit 25.

```
> new <- data.frame(Temp = seq(round(min(Steam$Temp),0), round(max(Steam$Temp),0)))
> pred.w.plim <- predict(Steam.lm1, new, interval="prediction")
> pred.w.clim <- predict(Steam.lm1, new, interval="confidence")
> matplot(new$Temp,cbind(pred.w.clim, pred.w.plim[,-1]), lty=c(1,2,2,3,3), type="l", xlab="Temperature",
```



happens in the Steam Use example when a second variable is added to the model. Exhibit 27 shows the output for modelling Steam Use using the average temperature and the variable that accounts for the amount of wind in the area. Compare this output to that created when the two predictors are entered into the model statement in the opposite order, given in Exhibit 28. Note that very little of the regression output actually changes in a material way. The coefficients for the model's equation are the same, the R^2 is the same, the details of the residuals are the same; so what's different? The only difference in the analyses is that the variable `Wind` explains a different sequential sum of squares when it is put into the model after `Temp`. The variable `Wind` is in fact rather insignificant when the variable `Temp` is available, as indicated by its p-value in Exhibits 27 and 28. To see the sequential sums of squares analyses we must use the `anova` command in R to generate the analyses in Exhibit 29. Using the sequential sum of squares explained by each variable shows that the `Wind` variable in its own right as a single predictor is useful, but its ability to explain much of the variation in `SteamUse` is negligible when `Temp` has already been used in the model. It's not so obvious when `Wind` has been inserted first. Some software, notably R (when you use the `summary` command not the `anova` command) and Minitab, shows you the p-value for the t-test in the output for each variable based on the adjusted sum of squares approach. We now see that this is effectively the same as repeating the sequential sum of squares based F-test assuming all variables to be the last to enter the model.

Exhibit 27 Regression Analysis for the `Steam` data, using `Temp` and `Wind` as predictors

```
> Steam.lm3 = lm(SteamUse~Temp+Wind, data=Steam)
> summary(Steam.lm3)

Call:
lm(formula = SteamUse ~ Temp + Wind, data = Steam)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4303 -0.5647 -0.0463  0.8083  1.3233

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  14.2978     1.4187   10.08  1.0e-09 ***
Temp         -0.0842     0.0136   -6.20  3.1e-06 ***
Wind         -0.0699     0.1337   -0.52     0.61
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.905 on 22 degrees of freedom
Multiple R-squared:  0.718,        Adjusted R-squared:  0.692
F-statistic:   28 on 2 and 22 DF,  p-value: 8.99e-07
```

Exhibit 28 Regression Analysis for the `Steam` data, using `Wind` and `Temp` as predictors

```
> Steam.lm4 = lm(SteamUse~Wind+Temp, data=Steam)
> summary(Steam.lm4)

Call:
lm(formula = SteamUse ~ Wind + Temp, data = Steam)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4303 -0.5647 -0.0463  0.8083  1.3233

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  14.2978     1.4187   10.08  1.0e-09 ***
Wind         -0.0699     0.1337   -0.52     0.61
Temp         -0.0842     0.0136   -6.20  3.1e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.905 on 22 degrees of freedom
Multiple R-squared:  0.718,        Adjusted R-squared:  0.692
F-statistic:   28 on 2 and 22 DF,  p-value: 8.99e-07
```

Exhibit 29 Analysis of variance tables for the `Steam` data with predictors entered in different orders

```
> anova(Steam.lm3)

Analysis of Variance Table

Response: SteamUse
          Df Sum Sq Mean Sq F value  Pr(>F)
Temp       1   45.6    45.6   55.73 1.8e-07 ***
Wind       1    0.2     0.2    0.27    0.61
Residuals 22   18.0     0.8
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(Steam.lm4)

Analysis of Variance Table

Response: SteamUse
          Df Sum Sq Mean Sq F value  Pr(>F)
Wind       1   14.4   14.36    17.6 0.00038 ***
Temp       1   31.5   31.46    38.5 3.1e-06 ***
Residuals 22   18.0    0.82
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When two (or more) explanatory variables are related to one another we say that *collinearity* exists in our data. We will often pick up the fact that a variable is not required in our model using the variance inflation factors (discussed in Section 5.2) as well as the fact that one of these variables will have a poor p-value. Removing one of the variables will improve the p-value of the other variable. Some judgement will be required to choose which variable is to be left out of the model.

10 Transformations can make modelling simpler

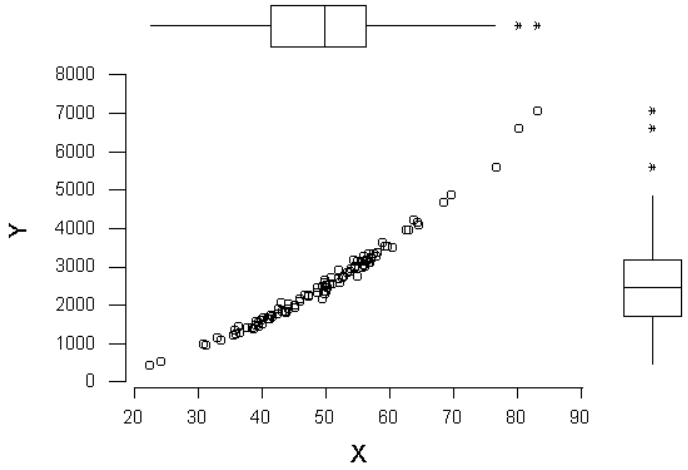
We saw earlier that a transformation could often be used to take a nonlinear function and make it linear to simplify the model fitted using regression. Transformations can also be used to remove the curvature in data to improve the fit of a model, and in some circumstances can remove the problem of *heteroscedasticity* — non-constant variance of residuals.

Remember that the residuals from any model we fit need to be normally distributed and have constant variance across the range of the explanatory variables, fitted values and the response variable. Look at Exhibit 30 which shows the marginal distributions of the two variables in the scatter plot. The lack of symmetry of the y variable means that it is unlikely that the residuals will have constant variance. Finding a suitable transformation of the y variable will allow a simple straight line model to be fitted.

Appropriate transformations depend on the situation that cause the data not to be symmetric. If data are positively skewed, a transformation that reduces the spread of the higher numbers is required. Square roots and other power transformations with exponents between zero and one reduce positive skewness. Conversely, powers greater than one expand the lower numbers more than the higher numbers and are useful when data are negatively skewed.

Sometimes another transformation is appropriate to the situation. When the shape of the relationship appears hyperbolic, taking the reciprocal of the skewed variable can straighten the relationship. Logarithmic transformations are useful for reducing positive skewness, and the following example shows that they can help when heteroscedasticity

Exhibit 30 A plot of points that generally conform to a model $y^2 = x + \epsilon$. The marginal distributions of both variables have been added to show the lack of symmetry in the y variable.



(non-constant variance) of residuals causes a problem.

Data from an experiment that tested the relationship between the number of defective gaskets and the speed of the machine used to manufacture them appear in Data Set 6 on Page 65 and are in a comma delimited file called `Gaskets.csv`. We read this file into R and identify the variables in the data set using the `read.csv` and `str` commands in the following way:

```
> Gaskets <- read.csv("Gaskets.csv")

> str(Gaskets)

'data.frame':        15 obs. of  2 variables:
 $ Speed     : int  100 100 100 100 100 150 150 150 150 150 ...
 $ Defectives: int  15 23 11 14 18 19 29 20 35 24 ...
```

A simple regression of `Defectives` on `Speed` is given in Exhibit 31, and includes a lack of fit test which indicates that the model is good. The plot of residuals from this model against the `Speed` of the machine, appearing in Exhibit 32, shows some evidence that the residuals do not have constant variance — indicating that the model is not appropriate for use. Re-fitting the model to the natural log of the number of defectives instead gives the output in Exhibit 33. The residuals from this model are much less unequally distributed as seen in Exhibit 34. The upshot of this is that the regression equation is now for a nonlinear function. We can backtransform

$$\ln(\text{Defectives}) = 1.98 + 0.00791 \times \text{Speed} \tag{34}$$

to give an equation that has `Defectives` as its subject. Unfortunately, this new equation

$$\text{Defectives} = 7.242e^{0.00791\text{Speed}} \tag{35}$$

cannot be used for prediction intervals. All such predictions using the model can, however, be made on the transformed scale (Exhibit 33) and subsequently backtransformed.

Exhibit 31 Regression analysis for the `Gaskets` data using the untransformed number of defectives as the response variable

```
> Gaskets.lm1 = lm(Defectives~Speed, data=Gaskets)
> summary(Gaskets.lm1)

Call:
lm(formula = Defectives ~ Speed, data = Gaskets)

Residuals:
   Min     1Q Median     3Q    Max
 -9.43  -5.38  -1.03   3.42  12.57

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -3.3667     6.6685   -0.50  0.62211
Speed         0.1940     0.0429    4.52  0.00057 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.78 on 13 degrees of freedom
Multiple R-squared:  0.611,        Adjusted R-squared:  0.582
F-statistic: 20.5 on 1 and 13 DF,  p-value: 0.000573
```

Exhibit 32 A plot of the residuals from the linear model for the number of defective gaskets against the `speed` of the machine.

```
> plot(Gaskets$Speed,  resid(Gaskets.lm1), xlab= "Speed", ylab = "Residuals")
```

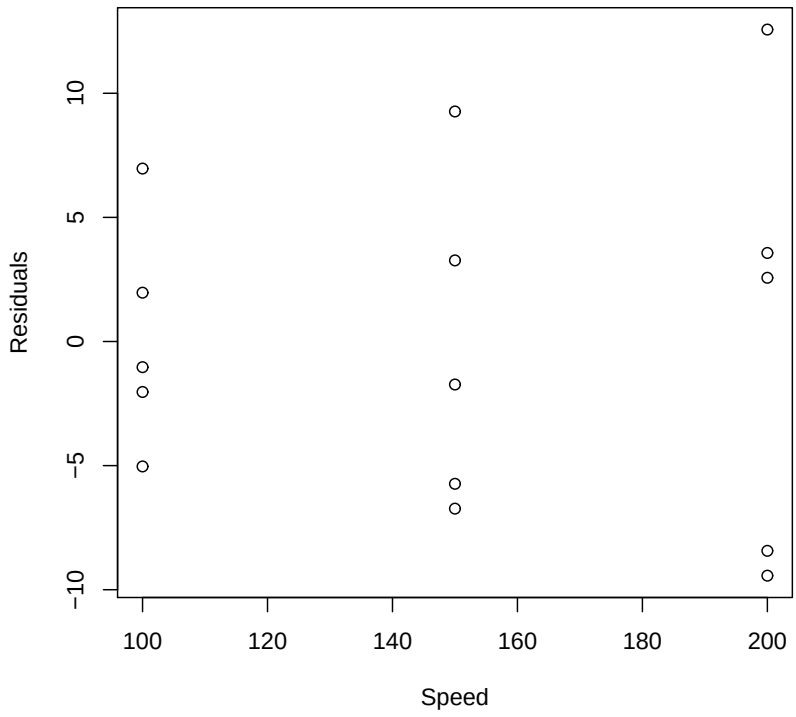
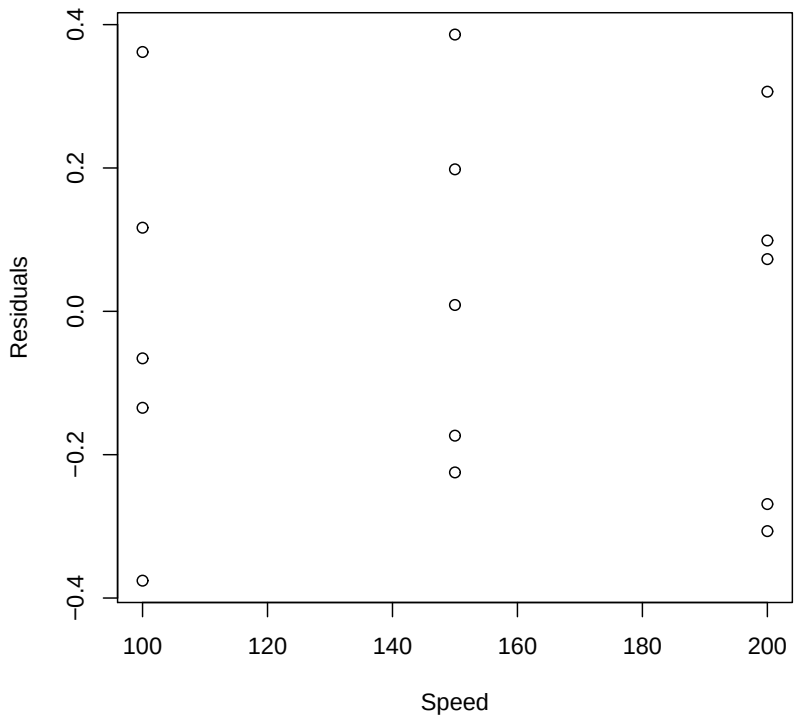


Exhibit 33 Regression analysis for the `Gaskets` data using the natural logarithm of the number of defectives as the response variable

```
> Gaskets.lm2 =  lm(log(Defectives)~Speed, data=Gaskets)
> summary(Gaskets.lm2)

Call:
lm(formula = log(Defectives) ~ Speed, data = Gaskets)

Residuals:
    Min      1Q  Median      3Q     Max
-0.3758 -0.1991  0.0089  0.1574  0.3861

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.98258    0.25178    7.87  2.7e-06 ***
Speed        0.00791    0.00162    4.88    3e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.256 on 13 degrees of freedom
Multiple R-squared:  0.647,        Adjusted R-squared:  0.62
F-statistic: 23.9 on 1 and 13 DF,  p-value: 0.000298
```

Exhibit 34 A plot of the residuals from the linear model for the natural log of the number of defective gaskets against the speed of the machine.

```
> plot(Gaskets$Speed,  resid(Gaskets.lm2), xlab= "Speed", ylab = "Residuals")
```



Transformations can be employed when we know how data have arisen. We know that data that are binomial in their distribution will have non-constant variance — remember that the variance of the binomial parameter is dependent on that parameter

$$Var\left(\hat{p}\right) = \frac{\hat{p}\left(1-\hat{p}\right)}{n} \tag{36}$$

but is also dependent on the sample size n. If n changes too much it will be difficult to overcome this sort of heteroscedasticity using a transformation.

When count data are examined, they often arise from a Poisson process. Remember a Poisson process has a rate parameter λ and both the mean and variance are related to this parameter. Mendenhall (1968) recommends using the square root of the response variable in such circumstances in order to stabilize its variance. Mendenhall (1968) provides a means of determining the appropriate transformation, but this is beyond the scope of this course. If your residuals vs fitted values plot shows heteroscedasticity, you might consider using one of the following transformations:

- If the variance of the expected response is linearly related to the expected response then use the square roots of the original response variable.
- If the variance of the expected response is linearly related to the square of the expected response then use the natural logarithm of the original response variable.
- If the expected response is binomially distributed, then use

  $$\sin^{-1}\sqrt{y} \tag{37}$$

  where the y values are entered in radians. Note that this transformation is most suitable when the response is expressed as a decimal fraction rather than as a percentage.

11 Dealing with an accumulation of errors

There are experiments that measure rates of change over time, such as chemical reactions. When a chemical compound is in a reactive state, the rate of reactivity will depend on the current mixture. If we wished to model the data arising from such an experiment we'd run into trouble modelling the real rate of reactions, rather than the observed rate. Let t be the label attached to observations taken at the tth time point, and assume that data are recorded at evenly spaced time intervals. If the reaction rate β is dependent on the current amount of reactive material y_t then we have a series of observations which can be written as

$$\begin{aligned} y_1 &= \beta y_0 + \epsilon_1 \\ y_2 &= \beta y_1 + \epsilon_2 \\ y_t &= \beta y_{t-1} + \epsilon_t \\ y_{t+1} &= \beta y_t + \epsilon_{t+1} \end{aligned} \tag{38}$$

and so on for all possible time points. The problem with modelling this data in its current form is that the y-values on the right-hand-side are not known with certainty. In fact we could re-express all observations in this series as

$$y_t = y_0 + \sum_{k=1}^{t} \epsilon_k \tag{39}$$

which shows that the errors are accumulating over time.

If we were to model the amount of the reactive substance remaining over time we'd use some formula like

$$y_t = f(t) + \epsilon \tag{40}$$

where $f(t)$ is a function of time. In spite of the change in the formulation to get rid of the y's from the right-hand-side, we still have an accumulation of errors over time. If the rate of change is constant over time we have

$$y_t = \beta t + \epsilon \tag{41}$$

Altering the way we look at this data helps us out here. Instead of looking at the actual quantity of reactive material, we could look at the change of reactive material from one time point to the next. If we let $\Delta y_t = y_t - y_{t-1}$ we can fit a model

$$\Delta y_t = \beta + \epsilon \tag{42}$$

There are some important things to note about taking these differences to get data for the new model. First of all, the ϵ's in Equation 41 and 42 are not the same.

The second problem poses more difficulty for us though. Fitting the model in Equation 42 is not all that easy using software like Minitab, as it doesn't like having a model with no predictors in it. R on the other hand does allow for a model like this using code of the form:

```
> lm(y~1)
```

How realistic is a constant rate of change though? We should question the constant rate of change by allowing at least a linear change in the reaction rate over time. We perform a simple regression using a specifically created time variable as the predictor.

12 Creation of new variables

When there are two groups of individuals in a population for which you are creating a model, whether it be subjects with different genders or two different products under examination, you can develop a model that is responsive to a difference between the groups.

First of all, you need to create a variable that reflects membership of a group. We prefer to use what we call an *indicator variable* in this instance. These variables are binary, and take on either a zero or a one depending on the situation. We might use a one to indicate a male subject and a zero to indicate a female subject. We could use a one to refer to the fact that that subject had some extra factor applied to it. Adding such a variable to the most basic model

$$y = \beta_0 + \beta_1 x_1 + \epsilon \tag{43}$$

gives

$$y = \beta_0 + \beta_1 x_1 + \beta_2 I_1 + \epsilon \tag{44}$$

This equation reduces to the simple model when I_1 is zero, but when I_1 is set to one a constant amount is added onto the response variable. The result is that the fitted values lay on a pair of parallel lines and the response value depends on the group to which the observation belongs.

If we have two groups, it's actually more likely that they have different responses to the predictor variable. We should test the significance of this difference. This is only possible

if we fit a model to all observations at the same time. Taking the model in Equation 44 and adding a term that is zero when I_1 is zero, and x_1 when $I_1 = 0$ gives

$$y = \beta_0 + \beta_1 x_1 + \beta_2 I_1 + \beta_3 I_1 x_1 + \epsilon \tag{45}$$

The term $I_1 x_1$ is an interaction term. It should only be fitted when both I_1 and x_1 variables have been include in the model. You can look at the p-value to see if the interaction term is necessary and this will tell you if the groups respond differently to the predictor x_1.

The problem with indicator variables is that we will need to generate many columns for them and any interactions that may need to be investigated. R has a much easier way to deal with these situations. A discrete-valued variable (e.g. gender or ethnicity) is referred to as a *factor*. Interactions between factors and continuous-valued variables can be inserted in your model without the need to actually create the necessary indicator variables or the additional interaction variables.

A common scenario for requiring indicator variables arises when data are collected in a pattern, such as one record per day of the week or certain hours each day. You can create indicator variables for each day, month, hour, etc. as your scenario requires. It's unlikely that this sort of scenario requires different slopes for each subgroup, but testing for this interaction is always a good idea.

You may be able to see the need for these indicator variables in your model by plotting the data effectively. Use different colours or symbols and consider using the residuals from your basic model rather than the actual data itself. Another good plot offered by R is an interaction plot which takes averages within groups and removes much of the noise in the data. This graph works best when you have replicate data and defined x values rather than many different x values to consider – this is usually a result of using data from a planned experiment rather than observational studies.

13 Polynomial regression

Often when your data shows curvature, a transformation of either the response or explanatory variable will not solve the problems found in a residual analysis. It is possible to create terms that are higher powers of the explanatory variable and fit a polynomial model. Whatever model you choose, you must ensure that the assumptions of regression are not violated.

As an example of modelling a relationship with a polynomial, we will consider how we should model the winning speeds for the Indianapolis 500 race for the years 1911 to 1971. This isn't exactly related to technology or engineering but the data set lends itself very nicely to the material of this and the following section.

The data are in a comma delimited file called `Steam.csv`. We read this file into R and identify the variables in the data set using the `read.csv` and `str` commands in the following way:

```
> Indy500 <- read.csv("indianapolis.csv")

> str(Indy500)

'data.frame':        55 obs. of  2 variables:
 $ Speed: num  74.6 78.7 75.9 82.5 89.8 ...
 $ Year : int  1911 1912 1913 1914 1915 1916 1919 1920 1921 1922 ...
```

This data set is displayed in the scatterplot in Exhibit 35. You can also see the data in Data Set 7 on page 65. Note that the race was not run during the years that the United States was involved in World Wars I and II, being 1917-1918 and 1942-1945.

Exhibit 35 A plot of the winning speeds (averaged over the entire race) in the Indianapolis 500 against the year in which the race was held.

```
> plot(Indy500$Year, Indy500$Speed, xlab = "Year", ylab = "Speed")
```

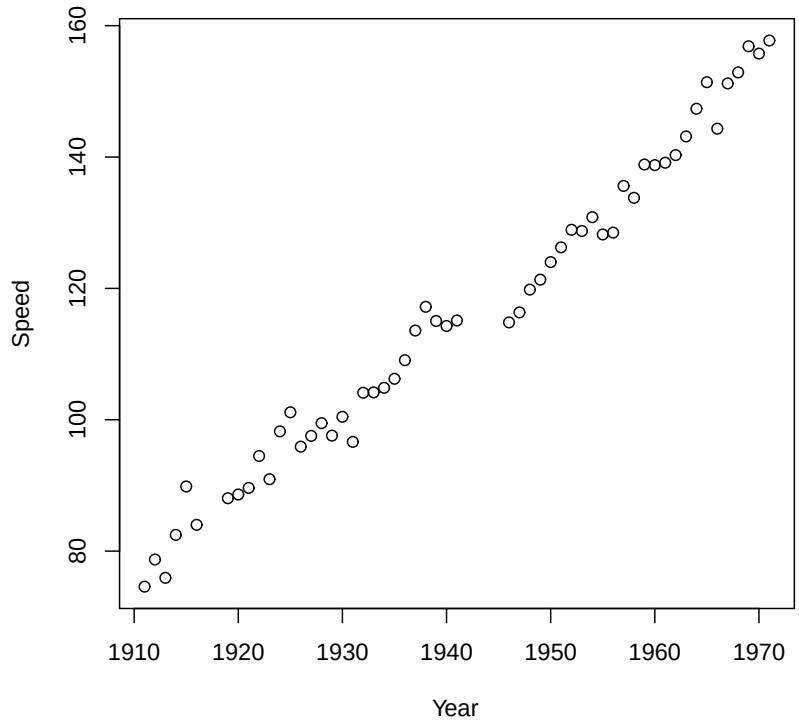
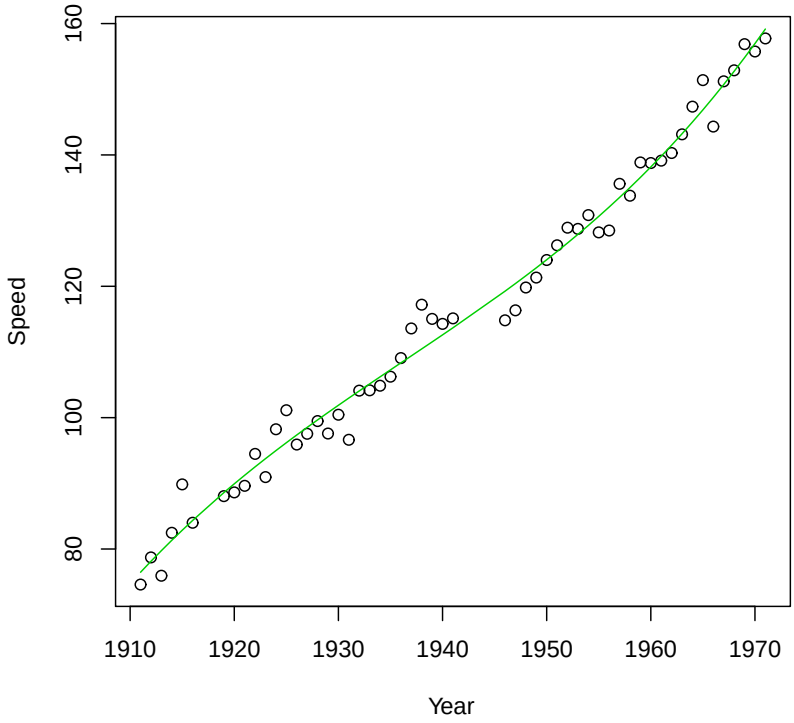


Exhibit 36 Regression Analysis

```
> Indy500$Yr = Indy500$Year - 1900
> Indy500.lm1  <- lm(Speed ~ Yr, data = Indy500)
> Indy500.lm2 <- lm(Speed ~ poly(Yr, 2, raw=TRUE), data = Indy500)
> Indy500.lm3 <- lm(Speed ~ poly(Yr, 3, raw=TRUE), data = Indy500)
> Indy500.lm4 <- lm(Speed ~ poly(Yr, 4, raw=TRUE), data = Indy500)
> anova(Indy500.lm1, Indy500.lm2, Indy500.lm3, Indy500.lm4)

Analysis of Variance Table

Model 1: Speed ~ Yr
Model 2: Speed ~ poly(Yr, 2, raw = TRUE)
Model 3: Speed ~ poly(Yr, 3, raw = TRUE)
Model 4: Speed ~ poly(Yr, 4, raw = TRUE)
  Res.Df RSS Df Sum of Sq     F  Pr(>F)
1     53 599
2     52 516  1      82.9 10.46 0.00216 **
3     51 397  1     119.2 15.03 0.00031 ***
4     50 396  1       0.6  0.07 0.79064
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Exhibit 37 A plot of the fitted values for the polynomial regression of winning speeds in the Indianapolis 500 against the year in which the race was held.

```
> plot(Indy500$Year, Indy500$Speed, xlab = "Year",  ylab = "Speed")
> lines(Indy500$Year, fitted(Indy500.lm3), col = 3)
```



Fitting a model explaining `Speed` with powers of `yr` up to 4, gives the output in Exhibit 36. Note that a coding is used in place of the original `Year` variable. We see that four models have been fitted using the `lm` command and the use of `poly` in the model formulation. The `anova` command is used to compare these nested models sequentially. The F-ratio for the quartic model (and therefore quartic term) is not significant, so we conclude that it will give little improvement over the cubic model. The cubic term is however significant, so we conclude that a cubic model is appropriate. The plot of this fitted curve (given in Exhibit 37) was obtained by plotting the predicted values for the cubic model against `year`. The output from the regression analysis of the (preferred) cubic model is given in Exhibit 38 for completeness.

Note however that although polynomials are a very flexible way to model 'local' curvature, they will rarely be of any use for extrapolation. For example, consider use of the cubic model to explain the speeds driven by winning drivers in the Indianapolis 500 over the last thirty years (beyond the time when data has been used). In the years 1990 to 2010, with five year intervals, the model would project the winning speeds to be 217, 237, 261, 288, and 318 miles per hour respectively. (That's pretty quick!) We obviously should not use this data to predict the winning speeds of races held recently.

Exhibit 38 Regression Analysis

```
> summary(Indy500.lm3)

Call:
lm(formula = Speed ~ poly(Yr, 3, raw = TRUE), data = Indy500)

Residuals:
   Min     1Q Median     3Q    Max
-6.353 -1.448 -0.175  1.627  6.981

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                 5.44e+01   4.37e+00   12.45  < 2e-16 ***
poly(Yr, 3, raw = TRUE)1  2.36e+00   3.90e-01    6.05  1.7e-07 ***
poly(Yr, 3, raw = TRUE)2 -3.58e-02   1.04e-02   -3.45  0.00114 **
poly(Yr, 3, raw = TRUE)3  3.28e-04   8.38e-05    3.91  0.00027 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.79 on 51 degrees of freedom
Multiple R-squared:  0.986,        Adjusted R-squared:  0.986
F-statistic: 1.23e+03 on 3 and 51 DF,  p-value: <2e-16
```

14 Piecewise functions

An alternative approach for flexible curve fitting is to specify a different straight line or curve for different ranges of x-values. For example, suppose that we want to use a different linear relationship in each of three ranges of x-values: $x < k_1$, $k_1 \leqslant x < k_2$, and $x \geqslant k_2$ The most general form of such a relationship is illustrated in Exhibit 39.

We would usually not consider such discontinuities at the changeover points, k_1 and k_2, which we generally call *knots*. If we insist that the three lines meet at k_1 and k_2, they can be parameterised as is illustrated in Exhibit 40.

This type of model is called a piecewise linear model, and it can be expressed as a single linear multiple regression. Note that we assume the knots k_1 and k_2 are known constants in this model and do not need to be estimated. Piecewise linear models are often used when the explanatory variable is time, and the knots are times when there was a change in the system being monitored (e.g. the 2nd World War, or a stock market crash). For the Indianapolis 500 race data that was analysed in the previous section, it has been suggested that there was a change in technology at each of the two World Wars, and that this might have resulted in discrete changes to the curves at these points. We will assume potential knots at years 1917.5 and 1943.5, midway between the non-racing years at the two wars. A piecewise linear model will be fitted, using new variables:

1. `PostWW1` — an indicator variable to show that the race was held after World War I.
2. `AfterWW1` — the increment of 'Year' over 1917.5.
3. `PostWW2` — an indicator variable to show that the race was held after World War II.
4. `AfterWW2` — the increment of 'Year' over 1943.5.

See how these variables are created in Exhibit 41. The base year in this example is again 1900, so the variable `Yr` replaces the year, and has been defined as 'Year'–1900. Regressing `Speed` against `Yr`, `AfterWW1`, and `AfterWW2` gives the output presented in Exhibit 41. You could go to the extremes of calculating the benefit of adding the two additional time variables to the model, but all terms in the model have low p-values suggesting that they should all be included.

Exhibit 39 An example of a piecewise linear function with discontinuities.

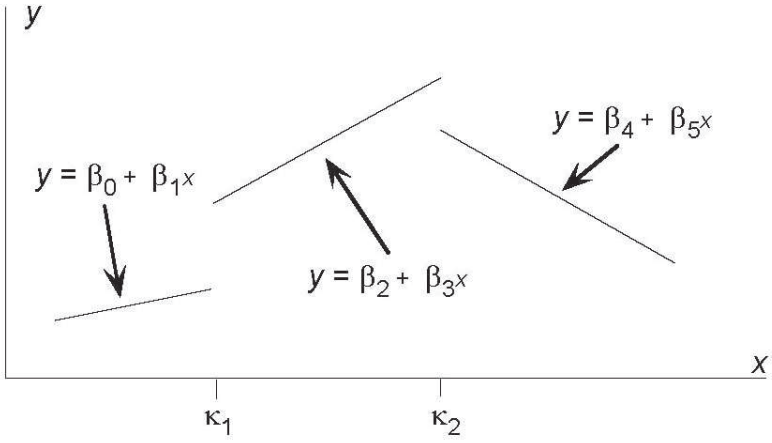


Exhibit 40 An example of a piecewise linear function with knots.

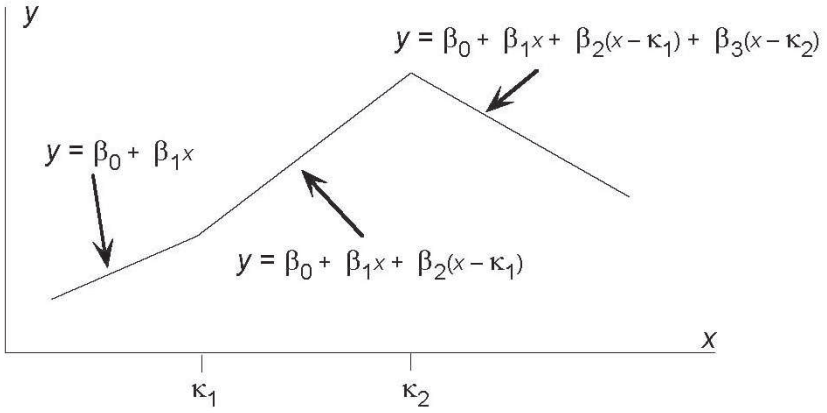


Exhibit 41 Regression analysis

```
> Indy500$PostWW1 = as.numeric((Indy500$Year>1917.5))
> Indy500$AfterWW1 = Indy500$PostWW1 * (Indy500$Year - 1917.5)
> Indy500$PostWW2 = as.numeric((Indy500$Year>1943.5))
> Indy500$AfterWW2 = Indy500$PostWW2 * (Indy500$Year - 1943.5)
> Indy500.lm5 = lm(Speed ~ Yr+AfterWW1+AfterWW2, data=Indy500)
> summary(Indy500.lm5)

Call:
lm(formula = Speed ~ Yr + AfterWW1 + AfterWW2, data = Indy500)

Residuals:
   Min     1Q Median     3Q    Max
-5.618 -1.961 -0.122  1.382  7.823

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   53.777      5.755    9.34  1.2e-12 ***
Yr             1.984      0.366    5.41  1.7e-06 ***
AfterWW1      -0.965      0.408   -2.37  0.02174 *
AfterWW2       0.473      0.117    4.03  0.00018 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.98 on 51 degrees of freedom
Multiple R-squared:  0.984,        Adjusted R-squared:  0.984
F-statistic: 1.08e+03 on 3 and 51 DF,  p-value: <2e-16
```

Exhibit 42 Regression Analysis

```
> Indy500.lm6 = lm(Speed~Yr+PostWW1+AfterWW1+PostWW2+AfterWW2, data=Indy500)
> summary(Indy500.lm6)

Call:
lm(formula = Speed ~ Yr + PostWW1 + AfterWW1 + PostWW2 + AfterWW2,
    data = Indy500)

Residuals:
   Min     1Q Median     3Q    Max
-6.474 -1.429 -0.125  1.440  5.603

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   47.388      8.428    5.62  8.9e-07 ***
Yr             2.484      0.619    4.01  0.00021 ***
PostWW1       -4.805      2.930   -1.64  0.10743
AfterWW1      -1.222      0.625   -1.96  0.05622 .
PostWW2       -7.036      1.671   -4.21  0.00011 ***
AfterWW2       0.396      0.106    3.73  0.00049 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.59 on 49 degrees of freedom
Multiple R-squared:  0.989,        Adjusted R-squared:  0.988
F-statistic:  861 on 5 and 49 DF,  p-value: <2e-16
```

There seems to have been a change in slope at the two wars, but was there a discontinuity? To examine this hypothesis, we must fit a separate least squares line in each of the three segments. This can be accomplished by adding the two extra indicator variables, `PostWW1` and `PostWW2` to the model to allow the intercepts, in addition to the slopes, to differ between the segments. Exhibit 42 shows the output from fitting all five terms into the one model. Look at the fitted values from this model (given in Exhibit 43) and compare them to those found for the polynomial regression model. The p-values for the additional terms are significant, but not as strongly as the slope variables had been. This suggests that there was some noticeable change in the rate of improvement in the technology used in racecars after the world wars, but that the actual winning speeds dropped as a result of the wars. It would be possible to reduce the number of terms in this model without sacrificing the explanatory power of the model. You might want to look into this further for yourself. Remember a simpler model might be the more useful model.

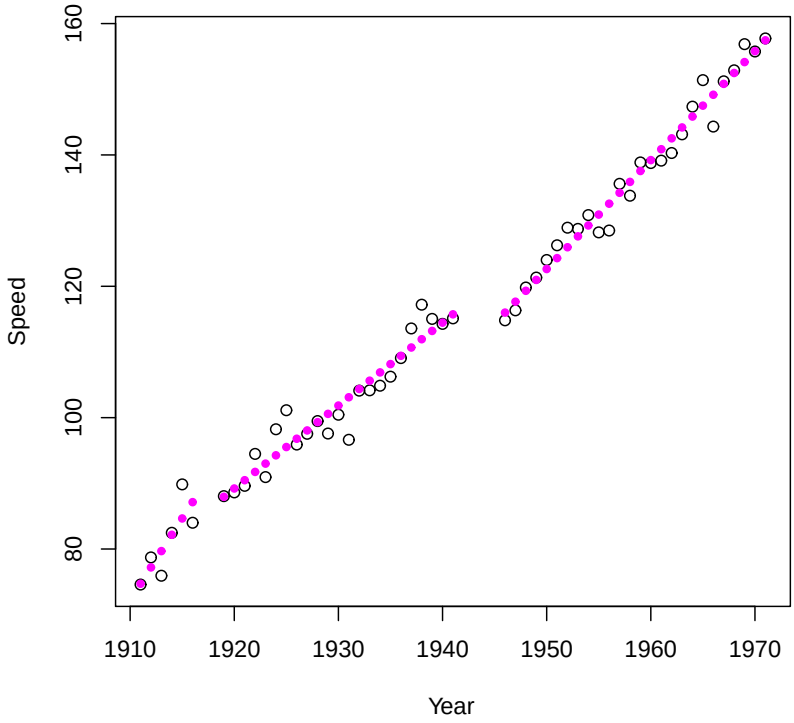
The piecewise linear model can be extended to allow successive segments to be quadratic (or even higher-order polynomial) curves. Such piecewise quadratic (or cubic, etc.) models are usually constrained to force the curves to meet at the knots and also to force the slopes of the two curves that meet at a knot to be the same. Such models are called quadratic (or cubic, etc.) splines.

You can test for discontinuities, changes of slope and changes of curvature at the knots of these piecewise models by testing the significance of the coefficient associated with the relevant term in the model. Remember that:

- An indicator variable that is 0 before a knot κ and 1 after it adds a discontinuity at the knot.
- A variable that is 0 before κ and is $x - \kappa$ after it causes a change of slope at the knot.
- A variable that is 0 before κ and is $(x - \kappa)^2$ after it causes a change of curvature at the knot.

Exhibit 43 A plot of the fitted values for the piecewise regression of winning speeds in the Indianapolis 500 against the year in which the race was held.

```
> plot(Indy500$Year, Indy500$Speed, xlab = "Year", ylab = "Speed")
> points(Indy500$Year, fitted(Indy500.lm6), pch=20, col = 6)
```



15 Variable selection techniques

In some circumstances when there are a large number of explanatory variables that could be included in a model, it will prove difficult to know where to start. Most statistical packages offer several approaches for building models that can be compared to one another. Each of these models can be considered as a 'candidate model' that should be scrutinized using the residual analysis ideas described in Section 4. Each candidate model also needs to satisfy the notion that it is actually an appropriate model to choose.

15.1 Stepwise regression algorithms

Performing a manual search over all possible models can prove time-consuming when there are too many explanatory variables to choose among. Stepwise algorithms allow a structured search for *candidate models* that are then available for the user to choose among. The first algorithm to consider is the *backwards elimination* method. This method inserts every predictor into the regression model and then successively removes the variable that is the least significant. See how the algorithm drops terms in the model for a piecewise linear function to explain the winning speeds in the Indianapolis 500, done manually in Exhibit 44. Note that at each iteration we use the `anova` command to see how the current model looks and then the `drop1` command to see the impact of dropping each term from the current model. A much more efficient method of building a regression model this way is to use the Rcmdstep command, shown in Exhibit 15.1. It carries out the process

Exhibit 44 Stepwise Regression using backwards elimination on the Indianapolis 500 data.

```
> Indy500.back = Indy500.lm6
> anova(Indy500.back)

Analysis of Variance Table

Response: Speed
          Df Sum Sq Mean Sq F value  Pr(>F)
Yr         1  28632   28632 4265.46 < 2e-16 ***
PostWW1    1      4       4    0.62 0.43490
AfterWW1   1     25      25    3.79 0.05744 .
PostWW2    1    147     147   21.90 2.3e-05 ***
AfterWW2   1     94      94   13.95 0.00049 ***
Residuals 49    329       7
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> drop1(Indy500.back, test="F")

Single term deletions

Model:
Speed ~ Yr + PostWW1 + AfterWW1 + PostWW2 + AfterWW2
         Df Sum of Sq RSS AIC F value  Pr(>F)
<none>                329 110
Yr        1     108.0 437 124   16.09 0.00021 ***
PostWW1   1      18.1 347 111    2.69 0.10743
AfterWW1  1      25.7 355 112    3.82 0.05622 .
PostWW2   1     119.0 448 125   17.73 0.00011 ***
AfterWW2  1      93.6 423 122   13.95 0.00049 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> Indy500.back = update(Indy500.back, .~.-PostWW1)
> anova(Indy500.back)

Analysis of Variance Table

Response: Speed
          Df Sum Sq Mean Sq F value  Pr(>F)
Yr         1  28632   28632 4126.05 < 2e-16 ***
AfterWW1   1      0       0    0.06    0.81
PostWW2    1    126     126   18.14 9.0e-05 ***
AfterWW2   1    126     126   18.13 9.1e-05 ***
Residuals 50    347       7
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> drop1(Indy500.back, test="F")

Single term deletions

Model:
Speed ~ Yr + AfterWW1 + PostWW2 + AfterWW2
         Df Sum of Sq RSS AIC F value  Pr(>F)
<none>                347 111
Yr        1     162.2 509 130   23.38 1.3e-05 ***
AfterWW1  1       7.8 355 110    1.12 0.29527
PostWW2   1     106.9 454 124   15.41 0.00027 ***
AfterWW2  1     125.8 473 126   18.13 9.1e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Continued on page 53

Exhibit 44 (continued) Stepwise Regression using the backwards elimination method on the Indianapolis 500 data.

Continued from page 52

```
> Indy500.back = update(Indy500.back, .~.-AfterWW1)
> anova(Indy500.back)

Analysis of Variance Table

Response: Speed
          Df Sum Sq Mean Sq F value  Pr(>F)
Yr         1  28632   28632  4116.5 < 2e-16 ***
PostWW2    1    112     112    16.1 0.00019 ***
AfterWW2   1    132     132    19.0 6.3e-05 ***
Residuals 51    355       7
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> drop1(Indy500.back, test="F")

Single term deletions

Model:
Speed ~ Yr + PostWW2 + AfterWW2
         Df Sum of Sq  RSS AIC F value  Pr(>F)
<none>                 355 110
Yr        1      3777 4132 244   543.1 < 2e-16 ***
PostWW2   1       149  504 128    21.4 2.6e-05 ***
AfterWW2  1       132  487 126    19.0 6.3e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

.

```
> step(Indy500.lm6)

Start:  AIC=110.4
Speed ~ Yr + PostWW1 + AfterWW1 + PostWW2 + AfterWW2

           Df Sum of Sq RSS AIC
<none>                 329 110
- PostWW1   1      18.1 347 111
- AfterWW1  1      25.7 355 112
- AfterWW2  1      93.6 423 122
- Yr        1     108.0 437 124
- PostWW2   1     119.0 448 125

Call:
lm(formula = Speed ~ Yr + PostWW1 + AfterWW1 + PostWW2 + AfterWW2,
    data = Indy500)

Coefficients:
(Intercept)           Yr      PostWW1     AfterWW1      PostWW2
     47.388        2.484       -4.805       -1.222       -7.036
   AfterWW2
      0.396
```

Exhibit 45 Stepwise Regression using the forwards insertion method on the Indianapolis 500 data.

```
> Indy500.fwd = lm(Speed ~1, data=Indy500)
> anova(Indy500.fwd)

Analysis of Variance Table

Response: Speed
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 54  29231     541

> add1(Indy500.fwd, scope = ~Yr+PostWW1+AfterWW1+PostWW2+AfterWW2,  test="F")

Single term additions

Model:
Speed ~ 1
         Df Sum of Sq   RSS AIC F value  Pr(>F)
<none>                29231 347
Yr        1     28632   599 135  2532.9 < 2e-16 ***
PostWW1   1      8309 20922 331    21.1 2.8e-05 ***
AfterWW1  1     28485   746 147  2025.0 < 2e-16 ***
PostWW2   1     21077  8153 279   137.0 2.6e-16 ***
AfterWW2  1     24193  5038 252   254.5 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> Indy500.fwd = update(Indy500.fwd, .~.+Yr)
> anova(Indy500.fwd)

Analysis of Variance Table

Response: Speed
          Df Sum Sq Mean Sq F value Pr(>F)
Yr         1  28632   28632    2533 <2e-16 ***
Residuals 53    599      11
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> add1(Indy500.fwd, scope = ~Yr+PostWW1+AfterWW1+PostWW2+AfterWW2,  test="F")

Single term additions

Model:
Speed ~ Yr
         Df Sum of Sq RSS AIC F value Pr(>F)
<none>                599 135
PostWW1   1       4.2 595 137    0.36 0.5491
AfterWW1  1       0.4 599 137    0.03 0.8537
PostWW2   1     112.2 487 126   11.99 0.0011 **
AfterWW2  1      95.3 504 128    9.84 0.0028 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

followed manually, but instead of using the p-values of the predictors, it employs the AIC, or *Akaike information criterion*, to gauge the value of removing terms. The AIC should be minimized and is particularly useful in comparing models like the ones presented as it penalizes the inclusion of unnecessary terms in our models.

The opposite strategy, known as *forwards insertion*, starts with no explanatory variables in the model and then builds the model up by successively adding variables. It is important to note that forwards insertion will not necessarily come up with the same set of candidate models as the backwards elimination method. Exhibit 45 shows this algorithm applied to the winning speeds in the Indianapolis 500.

A third option exists which takes the two other strategies and applies them in con-

junction. *Stepwise regression* starts with no variables and successively adds in the best predictor, but at each step it considers the removal of any term already in the model that has been superseded by other variables. Exhibit 46 gives this output for the winning speeds in the Indianapolis 500. Note that in this example, the full stepwise algorithm has reached the same conclusions as the forwards insertion method, but that it has explicitly looked at the possible removal of terms already in the model. It also doesn't force development of models that have terms that do not surpass its minimum significance criteria.

Exercise 6 *Investigate the model chosen through using the stepwise regression algorithm on the Steam data, using the data found in Steam.csv or on page 64. Would you use this model for this data? (Answer on page 62)*

15.2 Best subsets regression

A best subsets regression considers all the possible models with p predictors. The process therefore considers $2^p - 1$ possible models and based on several criteria, chooses the best models with p predictors. These criteria are:

1. The standard deviation of residuals s.
2. The coefficient of determination R^2 as discussed on page 33.
3. The adjusted coefficient of determination which is also printed in every regression analysis.

The adjusted coefficient of determination is calculated using

$$\text{Adjusted } R^2 = 1 - \frac{n-1}{n-(p+1}\left(1-R^2\right) \tag{46}$$

and is more appropriate than the plain coefficient of determination when models with different numbers of predictors are being compared. As the adjusted coefficient of determination rises as the residual mean-square decreases, it gives the same model selection as the standard deviation of residuals. Also note that the adjusted coefficient of determination is always less than or equal to the coefficient of determination.

Investigating all possible models can be done by many statistical software packages. Note however, that while most software shows which variables are included in each model, we are not usually given the coefficients or their significance. You will still need to investigate the best of these candidate models via the normal regression model-building techniques discussed in this document. Residual analyses are also required to ultimately show that the selected model is adequate. The best subsets regression approach is not shown here as the majority of models you are likely to come across can be handled effectively using the stepwise approaches discussed above, or through your own manual efforts.

16 So what model do I choose?

This is not a simple question to answer. If we were to find all models that ar possible to fit to any set of data, we would probably have hundreds of models to consider; after all, linear regression is just one way of finding a model (!). First of all, we must remember the assumptions, especially the fact that our model must be suitable for the data. You should not use some models on statistical grounds — a straight line for curved relationships; but you should also not use a model just because it does work on statistical grounds. Use of the average rainfall in Spain for each month as an predictor in your model of monthly

Exhibit 46 Stepwise Regression on the Indianapolis 500 data.

```
> Indy500.step = lm(Speed ~1, data=Indy500)
> step(Indy500.step, scope = ~Yr+PostWW1+AfterWW1+PostWW2+AfterWW2, test="F")

Start:  AIC=347.2
Speed ~ 1

           Df Sum of Sq   RSS AIC F value  Pr(>F)
+ Yr        1     28632   599 135  2532.9 < 2e-16 ***
+ AfterWW1  1     28485   746 147  2025.0 < 2e-16 ***
+ AfterWW2  1     24193  5038 252   254.5 < 2e-16 ***
+ PostWW2   1     21077  8153 279   137.0 2.6e-16 ***
+ PostWW1   1      8309 20922 331    21.1 2.8e-05 ***
<none>                 29231 347
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Step:  AIC=135.3
Speed ~ Yr

           Df Sum of Sq   RSS AIC F value Pr(>F)
+ PostWW2   1       112   487 126   11.99 0.0011 **
+ AfterWW2  1        95   504 128    9.84 0.0028 **
<none>                  599 135
+ PostWW1   1         4   595 137    0.36 0.5491
+ AfterWW1  1         0   599 137    0.03 0.8537
- Yr        1     28632 29231 347 2532.91 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Step:  AIC=125.9
Speed ~ Yr + PostWW2

           Df Sum of Sq  RSS AIC F value  Pr(>F)
+ AfterWW2  1       132  355 110   19.00 6.3e-05 ***
+ PostWW1   1        47  439 122    5.50  0.0229 *
<none>                  487 126
+ AfterWW1  1        14  473 126    1.52  0.2237
- PostWW2   1       112  599 135   11.99  0.0011 **
- Yr        1      7667 8153 279  818.84 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Step:  AIC=110.5
Speed ~ Yr + PostWW2 + AfterWW2

           Df Sum of Sq  RSS AIC F value  Pr(>F)
<none>                  355 110
+ AfterWW1  1         8  347 111    1.12    0.30
+ PostWW1   1         0  355 112    0.02    0.89
- AfterWW2  1       132  487 126   19.00 6.3e-05 ***
- PostWW2   1       149  504 128   21.43 2.6e-05 ***
- Yr        1      3777 4132 244  543.06 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Call:
lm(formula = Speed ~ Yr + PostWW2 + AfterWW2, data = Indy500)

Coefficients:
(Intercept)           Yr      PostWW2     AfterWW2
     63.627        1.275       -7.221        0.384
```

production output of your industrial plant in Palmerston North is totally nonsensical, but there is always a chance that the relationship is linear.

You need to think about the eventual use of your model. A model for description could be very detailed if it is to get into the nitty-gritty of some phenomenon, but a model used for prediction needs to be more effective than actually measuring the phenomenon itself. A model for prediction purposes trades off the cost of estimating something against the cost of obtaining the actual results. You need to collect enough data to make a model valid in the first instance, and then be able to formulate a model that works. Then you apply that model enough times to make the exercise valuable by saving the expense of collecting exact results in future.

In many instances a simple model is all that is required. Over-complicated models may just cloud a relationship that is valued by its ability to elucidate some relationship. We use this notion of keeping models as simple as possible and call it *parsimony*. A parsimonious model does enough to statisfy our requirements without wasting effort in future. This is best illustrated with a trivial example. Let's say you want to predict the water usage of a manufacturing plant. The output of the plant is heavily dependent on the main product line and a number of minor product lines, each using a known quantity of water. Other tasks in the plant use an unquantifiable amount of water, but basically the more output the more water used in general. If you know how much the plant will make of its major product line each month, and the other lines you might be able to set up a model that has an R^2 of 95%. If you ignored the minor product lines though, the model would have an R^2 of 90%. You need to decide if collecting all the information on the minor product lines is actually of much use given that obtaining one set of monthly output projections will do a pretty good job anyway. It might be simpler to use this single variable linear regression model and incorporate some extra safety factor on the predicted water use, than getting a more accurate prediction by going to all the effort of finding a whole bunch of projections.

In the end, model selection is not just a science. There is certainly some art to finding the right model. In some situations the same data set might lead to a number of models that get used for different purposes. The candidate models found via manual searches or stepwise algorithms should be checked for their validity and then evaluated for their applicability. There is no reason to believe that a great model will be found, but there is no reason to suggest that only one model will be found either. George Box is credited with the saying "All models are wrong, but some are useful". It's just a matter of choice which models are the least wrong and the most useful.

17 References and other recommended reading

The library is full of books that have regression examples. I recommend looking for books that were written more recently than 1980, and perhaps that have one of the words 'applied' or 'applications' in their titles.

A search for regression as a keyword will give you far too many items to investigate. It's worth just walking into the section where all the statistics books are stored and let your eyes wander over the titles. Most regression books have a call number in the 519 range.

Particular textbooks used in preparation of this manual were:

Draper, N.R. and H. Smith (1998) *Applied Regression Analysis*, (3rd edition), John Wiley & Sons Inc., New York. Note: This edition and its predecessors are available in the Massey library at 519.5DRA.

Mendenhall, W. (1968) *Introduction to Linear Models and the Design and Analysis of Experiments*, Duxbury Press, Belmont, California.

Mendenhall, W., and T. Sincich (2003) it A Second Course in Statistics: Regression Analysis, (6th edition), Pearson Education Inc., New Jersey. Note: The 5th edition is available in the Massey library at 519.5MEN.

18 Solutions to exercises

Solution to Exercise 1 on page 5

The first task is to make the Arrhenius equation into a linear form. Taking the natural logarithm of Eq 7 gives

$$\ln k = \ln A - \frac{E_a}{RT} \tag{47}$$

If we make a new set of values $T^\star$, by taking the reciprocals of the temperatures multiplied by the universal gas constant R ($T^\star = 1/RT$), we can work with

$$\ln k = \ln A - E_a T^\star$$

We also need to make a new set of values that are the natural logarithms of the reaction rates k so that the model

$$y = \beta_0 + \beta_1 x$$

can be fitted using $y = \ln k$ and $x = T^\star$. The unknown parameters in Eq 7 are found using the estimated β_0 and β_1, using $\beta_0 = \ln A$ and $\beta_1 = -E_a$. The original Arrhenius equation with the new parameter estimates will then be

$$k = e^{\beta_0} e^{\beta_1 / RT}$$

Solution to Exercise 2 on page 6

You should have created a scatter plot of the data using `Hardness` as the response (dependent) variable and `Cement` as the predictor or explanatory variable. A fitted line could be added to the plot to give a useful alternative, and is presented in Exhibit 47. Whether it was by eyeballing the raw data, the scatter plot, or the fitted line plot, you should have decided that a straight line is a possible model to employ.

Solution to Exercise 3 on page 8

The R code and output is presented in Exhibit 48. The negative intercept of this line really has no interpretation. It would mean (if taken literally) that adding no cement results in negative hardness. Clearly, these experimental results are only valid for the range of cement quantities tested.

The positive slope of this line means that as cement is added to the mix it gets harder. For each unit of cement added to the mix, we can use the slope to indicate the change in its hardness. Note that this conclusion is only valid for the range of cement actually tested. At some point adding too much cement will have a detrimental effect.

Solution to Exercise 4 on page 19

To investigate the quality of your model (given in Exhibit 48, you need to create the following graphs:

- A histogram of the residuals,
- A normal probability plot,

Exhibit 47 A fitted line plot showing how the amount of cement in a mixture affects its hardness.

```
> plot(Cement$CementUsed, Cement$Hardness, xlab= "Cement Used", ylab = "Hardness")
> Cement.lm = lm(Hardness~CementUsed, data=Cement)
> abline(Cement.lm)
```

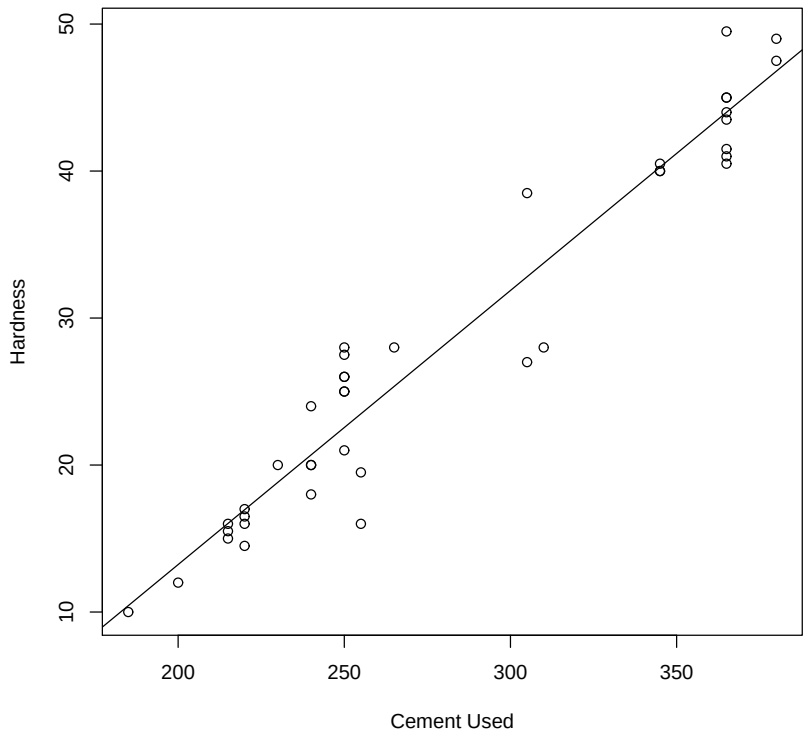
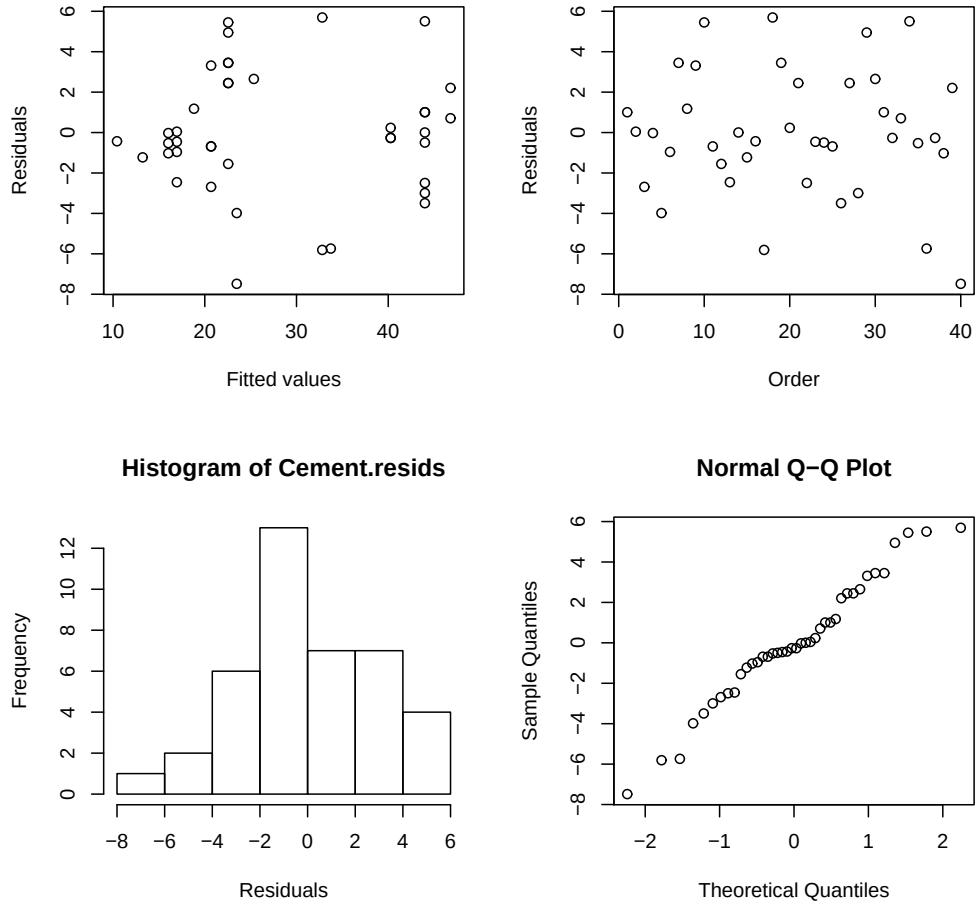


Exhibit 48 Regression Analysis: Hardness versus Cement

```
> Cement.lm = lm(Hardness~CementUsed, data=Cement)
> summary(Cement.lm)

Call:
lm(formula = Hardness ~ CementUsed, data = Cement)

Residuals:
    Min      1Q  Median      3Q     Max
-7.4835 -1.3085 -0.2659  2.2679  5.6929

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -24.066558   2.297903  -10.47  9.3e-13 ***
CementUsed    0.186471   0.007974   23.38  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.106 on 38 degrees of freedom
Multiple R-squared: 0.935,        Adjusted R-squared: 0.9333
F-statistic: 546.8 on 1 and 38 DF,  p-value: < 2.2e-16
```

Exhibit 49 Diagnostic plots for the residuals arising after fitting the model shown in Exhibit 48.



- A plot of residuals versus fitted values, and
- A plot showing the residuals against the order of the experimental runs.

R can create all four of these plots in a single graphic and an example appears in Exhibit 49.

It is perhaps easier and sometimes more useful to let R do the work for you in a single command. The plots generated by the `plot` command are not the same as you would have seen in other packages but they are actually more informative for smaller data sets commonly arising in technology and Engineering contexts. Compare the set of graphics in Exhibit 50 to that found in Exhibit 49 above.

Solution to Exercise 5 on page 34

You should have produced the R output shown in Exhibit 51. The Q-ratio of interest is $19.523/0.695 = 28.1$. Since $F_{2,22}$ for $\alpha = 0.05$ is 3.4, then the model as a whole is significant. But if we do a partial F-test on the quadratic term (assuming that the first order term is required) we get an interesting result. The error variance estimate is 0.695 (from the quadratic model) and the decrease in the sum of squared residuals (sometimes called the 'extra sum-of-squares') is $16.905 - 15.294 = 1.611$. The appropriate Q-ratio is $1.611/0.695 = 2.32$. $F_{1,22}$ for $\alpha = 0.05$ is 4.3, so we conclude that the quadratic term is

Exhibit 50

```
> par(mfrow=c(2,2))
> plot(Cement.lm)
```

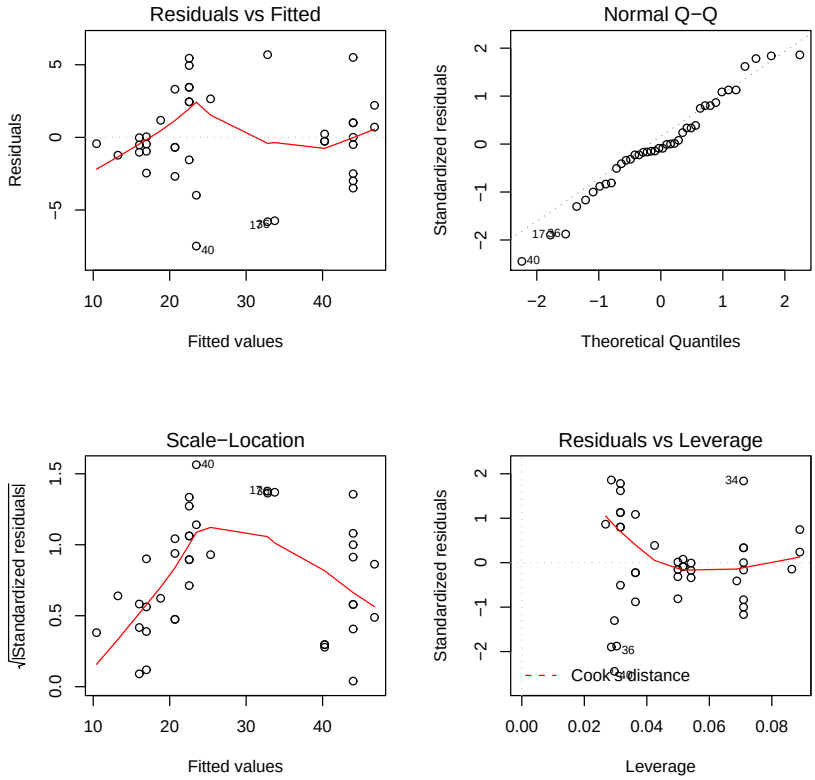


Exhibit 51 Regression Analysis

```
> Steam.lm5 = lm(SteamUse~poly(Temp,2), data=Steam)
> anova(Steam.lm5)

Analysis of Variance Table

Response: SteamUse
             Df Sum Sq Mean Sq F value    Pr(>F)
poly(Temp, 2)  2 46.894  23.447  30.483 4.558e-07 ***
Residuals     22 16.922   0.769
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(Steam.lm1, Steam.lm5)

Analysis of Variance Table

Model 1: SteamUse ~ Temp
Model 2: SteamUse ~ poly(Temp, 2)
  Res.Df     RSS Df Sum of Sq      F Pr(>F)
1     23 18.2234
2     22 16.9222  1    1.3012 1.6917 0.2068
```

not significant, even though the model as a whole is. The quadratic term is therefore not required.

Solution to Exercise 6 on page 55

First of all we need to find the model chosen using the stepwise algorithm.

```
> Steam.step = lm(SteamUse ~1, data=Steam)
> step(Steam.step, scope = ~Storage+Glycerin+Wind+CalDays+OpDays+ColdDays+Temp+Startups, test="F")

Start:  AIC=25.43
SteamUse ~ 1

           Df Sum of Sq    RSS    AIC F value     Pr(F)
+ Temp      1    45.592 18.223 -3.904 57.5428 1.055e-07 ***
+ ColdDays  1    26.192 37.624 14.219 16.0112 0.0005606 ***
+ OpDays    1    18.342 45.473 18.956  9.2774 0.0057364 **
+ Wind      1    14.357 49.459 21.057  6.6762 0.0165988 *
+ Storage   1     9.370 54.446 23.458  3.9581 0.0586652 .
+ Startups  1     9.318 54.497 23.482  3.9327 0.0594216 .
+ Glycerin  1     5.958 57.858 24.978  2.3683 0.1374645
<none>                 63.816 25.428
+ CalDays   1     1.193 62.623 26.956  0.4382 0.5145564
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Step:  AIC=-3.9
SteamUse ~ Temp

           Df Sum of Sq     RSS     AIC F value     Pr(F)
+ Storage   1     9.292   8.931 -19.733 22.8885 8.896e-05 ***
+ OpDays    1     8.595   9.629 -17.853 19.6375 0.0002103 ***
+ Glycerin  1     8.438   9.785 -17.451 18.9726 0.0002530 ***
+ CalDays   1     2.623  15.600  -5.790  3.6997 0.0674577 .
+ Startups  1     2.236  15.988  -5.176  3.0764 0.0933599 .
+ ColdDays  1     1.712  16.511  -4.371  2.2816 0.1451497
<none>                  18.223  -3.904
+ Wind      1     0.224  17.999  -2.213  0.2737 0.6060815
- Temp      1    45.592  63.816  25.428 57.5428 1.055e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Step:  AIC=-19.73
SteamUse ~ Temp + Storage

           Df Sum of Sq     RSS     AIC  F value     Pr(F)
+ OpDays    1     1.247   7.685 -21.491   3.4064    0.0791 .
<none>                    8.931 -19.733
+ Startups  1     0.319   8.612 -18.643   0.7790    0.3874
+ CalDays   1     0.238   8.693 -18.409   0.5758    0.4564
+ ColdDays  1     0.032   8.899 -17.823   0.0758    0.7857
+ Glycerin  1     0.004   8.927 -17.745   0.0102    0.9204
+ Wind      1 0.0003845   8.931 -17.734   0.0009    0.9763
- Storage   1     9.292  18.223  -3.904  22.8885 8.896e-05 ***
- Temp      1    45.515  54.446  23.458 112.1134 4.223e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Step:  AIC=-21.49
SteamUse ~ Temp + Storage + OpDays

           Df Sum of Sq     RSS     AIC  F value     Pr(F)
+ CalDays   1     0.883   6.802 -22.541   2.5954   0.12284
+ Startups  1     0.752   6.933 -22.064   2.1685   0.15643
<none>                    7.685 -21.491
+ Glycerin  1     0.529   7.156 -21.272   1.4774   0.23834
+ ColdDays  1     0.176   7.509 -20.070   0.4692   0.50124
+ Wind      1     0.155   7.530 -20.001   0.4125   0.52800
- OpDays    1     1.247   8.931 -19.733   3.4064   0.07909 .
- Storage   1     1.944   9.629 -17.853   5.3120   0.03149 *
- Temp      1    37.758  45.443  20.940 103.1806 1.472e-09 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Step:  AIC=-22.54
SteamUse ~ Temp + Storage + OpDays + CalDays

           Df Sum of Sq     RSS     AIC  F value     Pr(F)
- Storage   1     0.542   7.344 -22.625   1.5931   0.22141
<none>                   6.802 -22.541
+ Startups  1     0.343   6.459 -21.836   1.0096   0.32762
+ Glycerin  1     0.258   6.544 -21.509   0.7503   0.39720
- CalDays   1     0.883   7.685 -21.491   2.5954   0.12284
+ ColdDays  1     0.118   6.684 -20.980   0.3368   0.56849
+ Wind      1     0.040   6.762 -20.688   0.1120   0.74151
- OpDays    1     1.891   8.693 -18.409   5.5597   0.02867 *
- Temp      1    37.507  44.310  22.308 110.2821 1.381e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Step:  AIC=-22.63
SteamUse ~ Temp + OpDays + CalDays

           Df Sum of Sq     RSS     AIC  F value     Pr(F)
<none>                   7.344 -22.625
+ Storage   1     0.542   6.802 -22.541   1.5931    0.2214
+ ColdDays  1     0.319   7.025 -21.736   0.9085    0.3519
+ Glycerin  1     0.182   7.162 -21.253   0.5090    0.4838
+ Wind      1     0.130   7.213 -21.073   0.3616    0.5544
+ Startups  1     0.021   7.323 -20.697   0.0573    0.8132
- CalDays   1     2.285   9.629 -17.853   6.5334    0.0184 *
- OpDays    1     8.256  15.600  -5.790  23.6084 8.370e-05 ***
- Temp      1    37.116  44.460  20.393 106.1349 1.147e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Call:
lm(formula = SteamUse ~ Temp + OpDays + CalDays, data = Steam)

Coefficients:
(Intercept)         Temp       OpDays      CalDays
   -2.96806     -0.07392      0.19892      0.40205
```

We then need to look at the residuals arising from this model to satisfy ourselves that the assumptions of the linear model are met.

19 Data sets

The following data sets may be of use to you if you wish to re-produce the output given in this manual, or if you just want to practice applying the various modelling approaches.

Data set 4: A data set from Draper and Smith (1998), given in the file steam.csv, for 25 observations of nine variables. The variables are:

- SteamUse — pounds of steam used monthly.
- Storage — pounds of real fatty acid in storage per month.
- Glycerin — pounds of crude glycerin made.
- Wind — average wind velocity in miles per hour.
- Caldays — calendar days per month.
- Opdays — operating days per month.
- Colddays — days below freezing point.

Data Set 4 — Steam.csv (in full) from Draper and Smith (1998)

SteamUse	*Storage*	*Glycerin*	*Wind*	*Caldays*	*Opdays*	*Colddays*	*Temp*	*Startup*
10.98	*5.2*	*0.61*	*7.4*	*31*	*20*	*22*	*35.3*	*4*
11.13	*5.12*	*0.64*	*8*	*29*	*20*	*25*	*29.7*	*5*
12.51	*6.19*	*0.78*	*7.4*	*31*	*23*	*17*	*30.8*	*4*
8.4	*3.89*	*0.49*	*7.5*	*30*	*20*	*22*	*58.8*	*4*
9.27	*6.28*	*0.84*	*5.5*	*31*	*21*	*0*	*61.4*	*5*
8.73	*5.76*	*0.74*	*8.9*	*30*	*22*	*0*	*71.3*	*4*
6.36	*3.45*	*0.42*	*4.1*	*31*	*11*	*0*	*74.4*	*2*
8.5	*6.57*	*0.87*	*4.1*	*31*	*23*	*0*	*76.7*	*5*
7.82	*5.69*	*0.75*	*4.1*	*30*	*21*	*0*	*70.7*	*4*
9.14	*6.14*	*0.76*	*4.5*	*31*	*20*	*0*	*57.5*	*5*
8.24	*4.84*	*0.65*	*10.3*	*30*	*20*	*11*	*46.4*	*4*
12.19	*4.88*	*0.62*	*6.9*	*31*	*21*	*12*	*28.9*	*4*
11.88	*6.03*	*0.79*	*6.6*	*31*	*21*	*25*	*28.1*	*5*
9.57	*4.55*	*0.6*	*7.3*	*28*	*19*	*18*	*39.1*	*5*
10.94	*5.71*	*0.7*	*8.1*	*31*	*23*	*5*	*46.8*	*4*
9.58	*5.67*	*0.74*	*8.4*	*30*	*20*	*7*	*48.5*	*4*
10.09	*6.72*	*0.85*	*6.1*	*31*	*22*	*0*	*59.3*	*6*
8.11	*4.95*	*0.67*	*4.9*	*30*	*22*	*0*	*70*	*4*
6.83	*4.62*	*0.45*	*4.6*	*31*	*11*	*0*	*70*	*3*
8.88	*6.6*	*0.95*	*3.7*	*31*	*23*	*0*	*74.5*	*4*
7.68	*5.01*	*0.64*	*4.7*	*30*	*20*	*0*	*72.1*	*4*
8.47	*5.68*	*0.75*	*5.3*	*31*	*21*	*1*	*58.1*	*6*
8.86	*5.28*	*0.7*	*6.2*	*30*	*20*	*14*	*44.6*	*4*
10.36	*5.36*	*0.67*	*6.8*	*31*	*20*	*22*	*33.4*	*4*
11.08	*5.87*	*0.7*	*7.5*	*31*	*22*	*28*	*28.6*	*5*

- Temp — average atmospheric temperature in degrees Fahrenheit.
- Startup — number of startups.

Data for 40 measurements of concrete hardness appear in Data Set 5. The experiment was carried out by an Auckland concrete manufacturer to determine how the amount of cement in the mix affects the hardness of the final product.

Data Set 5 — Cement.csv from Wild and Seber (2000)

Cement	*Hardness*	*Cement*	*Hardness*	*Cement*	*Hardness*	*Cement*	*Hardness*
365	*45*	*240*	*20*	*250*	*25*	*365*	*45*
220	*17*	*250*	*21*	*365*	*41.5*	*345*	*40*
240	*18*	*220*	*14.5*	*220*	*16.5*	*380*	*47.5*
215	*16*	*365*	*44*	*365*	*43.5*	*365*	*49.5*
255	*19.5*	*200*	*12*	*240*	*20*	*215*	*15.5*
220	*16*	*185*	*10*	*365*	*40.5*	*310*	*28*
250	*26*	*305*	*27*	*250*	*25*	*345*	*40*
230	*20*	*305*	*38.5*	*365*	*41*	*215*	*15*
240	*24*	*250*	*26*	*250*	*27.5*	*380*	*49*
250	*28*	*345*	*40.5*	*265*	*28*	*255*	*16*

Data Set 6 — gaskets.csv from Mendenhall and Sincich (2003)

Speed	*Defectives*				
100	*15*	*23*	*11*	*14*	*18*
150	*19*	*29*	*20*	*35*	*24*
200	*26*	*48*	*27*	*38*	*39*

Data Set 7 — Indianapolis.csv

Year	*Speed*	*Year*	*Speed*	*Year*	*Speed*
1911	*74.59*	*1932*	*104.114*	*1955*	*128.209*
1912	*78.72*	*1933*	*104.162*	*1956*	*128.49*
1913	*75.931*	*1934*	*104.863*	*1957*	*135.601*
1914	*82.47*	*1935*	*106.24*	*1958*	*133.791*
1915	*89.84*	*1936*	*109.069*	*1959*	*138.857*
1916	*84*	*1937*	*113.58*	*1960*	*138.767*
1919	*88.05*	*1938*	*117.2*	*1961*	*139.13*
1920	*88.62*	*1939*	*115.035*	*1962*	*140.293*
1921	*89.62*	*1940*	*114.277*	*1963*	*143.137*
1922	*94.48*	*1941*	*115.117*	*1964*	*147.35*
1923	*90.95*	*1946*	*114.82*	*1965*	*151.388*
1924	*98.23*	*1947*	*116.338*	*1966*	*144.317*
1925	*101.13*	*1948*	*119.814*	*1967*	*151.207*
1926	*95.904*	*1949*	*121.327*	*1968*	*152.882*
1927	*97.545*	*1950*	*124.002*	*1969*	*156.867*
1928	*99.482*	*1951*	*126.244*	*1970*	*155.749*
1929	*97.585*	*1952*	*128.922*	*1971*	*157.735*
1930	*100.448*	*1953*	*128.74*		
1931	*96.629*	*1954*	*130.84*		

Index