Chapter 1

Linear Models

How old is the universe? The standard Big Bang model of the origin of the universe says that it expands uniformly, and locally, according to Hubble's law,

$$y = \beta x,$$

where y is the relative velocity of any two galaxies separated by distance x, and β is 'Hubble's constant' (in standard astrophysical notation $y \equiv v$, $x \equiv d$ and $\beta \equiv H_0$). β^{-1} gives the approximate age of the universe, but β is unknown and must somehow be estimated from observations of y and x, made for a variety of galaxies at different distances from us.

Figure 1.1 plots velocity against distance for 24 galaxies, according to measurements made using the Hubble Space Telescope. Velocities are assessed by measuring the Doppler effect red shift in the spectrum of light observed from the galaxies concerned, although some correction for 'local' velocity components is required. Distance measurement is much less direct, and is based on the 1912 discovery, by

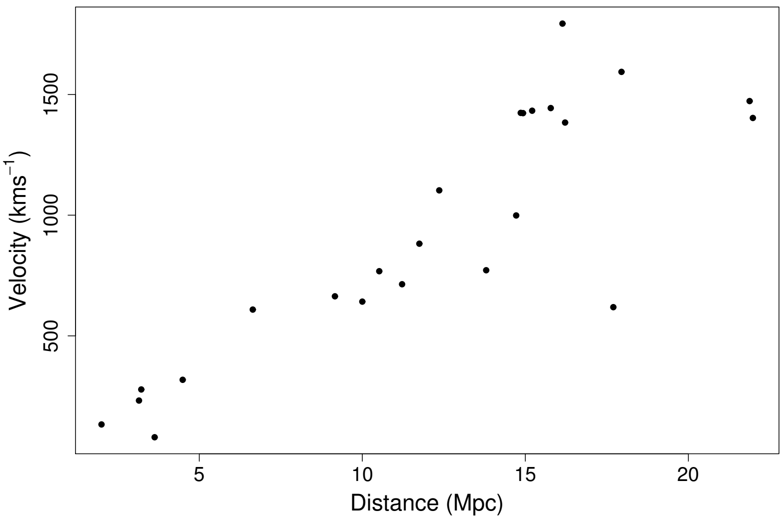

Figure 1.1 *A Hubble diagram showing the relationship between distance, x, and velocity, y, for 24 galaxies containing Cepheid stars. The data are from the Hubble Space Telescope key project to measure the Hubble constant as reported in Freedman et al. (2001).*

Henrietta Leavitt, of a relationship between the period of a certain class of variable stars, known as the Cepheids, and their luminosity. The intensity of Cepheids varies regularly with a period of between 1.5 and something over 50 days, and the mean intensity increases predictably with period. This means that, if you can find a Cepheid, you can tell how far away it is, by comparing its apparent brightness to its period predicted intensity.

It is clear, from the figure, that the observed data do not follow Hubble's law exactly, but given the measurement process, it would be surprising if they did. Given the apparent variability, what can be inferred from these data? In particular: (i) what value of β is most consistent with the data? (ii) what range of β values is consistent with the data and (iii) are some particular, theoretically derived, values of β consistent with the data? Statistics is about trying to answer these three sorts of questions.

One way to proceed is to formulate a linear statistical model of the way that the data were generated, and to use this as the basis for inference. Specifically, suppose that, rather than being governed directly by Hubble's law, the observed velocity is given by Hubble's constant multiplied by the observed distance plus a 'random variability' term. That is

$$y_i = \beta x_i + \epsilon_i, \quad i = 1 \ldots 24, \tag{1.1}$$

where the ϵ_i terms are independent random variables such that $\mathbb{E}(\epsilon_i) = 0$ and $\mathbb{E}(\epsilon_i^2) = \sigma^2$. The random component of the model is intended to capture the fact that if we gathered a replicate set of data, for a new set of galaxies, Hubble's law would not change, but the apparent random variation from it would be different, as a result of different measurement errors. Notice that it is not implied that these errors are completely unpredictable: their mean and variance are assumed to be fixed; it is only their particular values, for any particular galaxy, that are not known.

1.1 A simple linear model

This section develops statistical methods for a simple linear model of the form (1.1), allowing the key concepts of linear modelling to be introduced without the distraction of any mathematical difficulty.

Formally, consider n observations, x_i, y_i, where y_i is an observation on random variable, Y_i, with expectation, $\mu_i \equiv \mathbb{E}(Y_i)$. Suppose that an appropriate model for the relationship between x and y is:

$$Y_i = \mu_i + \epsilon_i \text{ where } \mu_i = x_i\beta. \tag{1.2}$$

Here β is an unknown parameter and the ϵ_i are mutually independent zero mean random variables, each with the same variance σ^2. So the model says that Y is given by x multiplied by a constant plus a random term. Y is an example of a *response variable*, while x is an example of a *predictor variable*. Figure 1.2 illustrates this model for a case where $n = 8$.

Simple least squares estimation

How can β, in model (1.2), be estimated from the x_i, y_i data? A sensible approach is to choose a value of β that makes the model fit closely to the data. To do this we

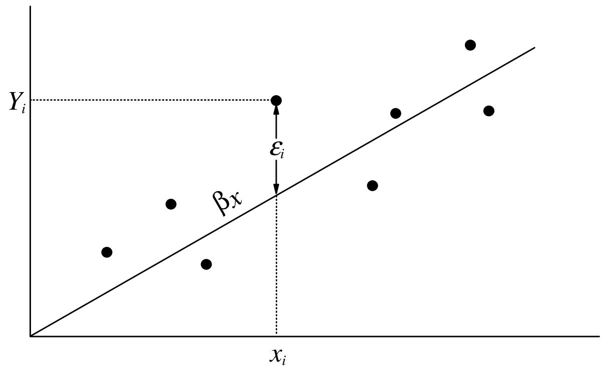

Figure 1.2 *Schematic illustration of a simple linear model with one explanatory variable.*

need to define a measure of how well, or how badly, a model with a particular β fits the data. One possible measure is the residual sum of squares of the model:

$$\mathcal{S} = \sum_{i=1}^{n}(y_i - \mu_i)^2 = \sum_{i=1}^{n}(y_i - x_i\beta)^2.$$

If we have chosen a good value of β, close to the true value, then the model predicted μ_i should be relatively close to the y_i, so that $\mathcal{S}$ should be small, whereas poor choices will lead to μ_i far from their corresponding y_i, and high values of $\mathcal{S}$. Hence β can be estimated by minimizing $\mathcal{S}$ with respect to (w.r.t.) β and this is known as the method of *least squares*.

To minimize $\mathcal{S}$, differentiate w.r.t. β,

$$\frac{\partial\mathcal{S}}{\partial\beta} = -\sum_{i=1}^{n} 2x_i(y_i - x_i\beta)$$

and set the result to zero to find $\hat\beta$, the least squares estimate of β:

$$-\sum_{i=1}^{n} 2x_i(y_i - x_i\hat\beta) = 0 \Rightarrow \sum_{i=1}^{n} x_i y_i - \hat\beta\sum_{i=1}^{n} x_i^2 = 0 \Rightarrow \hat\beta = \sum_{i=1}^{n} x_i y_i \Big/ \sum_{i=1}^{n} x_i^2.^*$$

1.1.1 Sampling properties of $\hat\beta$

To evaluate the reliability of the least squares estimate, $\hat\beta$, it is useful to consider the sampling properties of $\hat\beta$. That is, we should consider some properties of the distribution of $\hat\beta$ values that would be obtained from repeated independent replication of the x_i, y_i data used for estimation. To do this, it is helpful to introduce the concept of an *estimator*, obtained by replacing the observations, y_i, in the estimate of $\hat\beta$ by the random variables, Y_i:

$$\hat\beta = \sum_{i=1}^{n} x_i Y_i \Big/ \sum_{i=1}^{n} x_i^2.$$

[*] $\partial^2\mathcal{S}/\partial\beta^2 = 2\sum x_i^2$ which is clearly positive, so a minimum of $\mathcal{S}$ has been found.

Clearly the *estimator*, $\hat{\beta}$, is a random variable and we can therefore discuss its distribution. For now, consider only the first two moments of that distribution.

The expected value of $\hat{\beta}$ is obtained as follows:

$$\mathbb{E}(\hat{\beta}) = \mathbb{E}\left(\sum_{i=1}^{n} x_i Y_i / \sum_{i=1}^{n} x_i^2\right) = \sum_{i=1}^{n} x_i \mathbb{E}(Y_i) / \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i^2 \beta / \sum_{i=1}^{n} x_i^2 = \beta.$$

So $\hat{\beta}$ is an unbiased estimator — its expected value is equal to the true value of the parameter that it is supposed to estimate.

Unbiasedness is a reassuring property, but knowing that an estimator gets it right on average does not tell us much about how good any one particular estimate is likely to be. For this we also need to know how much estimates would vary from one replicate data set to the next — we need to know the estimator variance.

From general probability theory we know that if $Y_1, Y_2, \ldots, Y_n$ are *independent* random variables and $a_1, a_2, \ldots a_n$ are real constants then

$$\text{var}\left(\sum_i a_i Y_i\right) = \sum_i a_i^2 \text{var}(Y_i).$$

But we can write

$$\hat{\beta} = \sum_i a_i Y_i \;\text{ where }\; a_i = x_i / \sum_i x_i^2,$$

and from the original model specification we have that $\text{var}(Y_i) = \sigma^2$ for all i. Hence,

$$\text{var}(\hat{\beta}) = \sum_i x_i^2 / \left(\sum_i x_i^2\right)^2 \sigma^2 = \left(\sum_i x_i^2\right)^{-1} \sigma^2. \tag{1.3}$$

In most circumstances σ^2 is an unknown parameter and must also be estimated. Since σ^2 is the variance of the ϵ_i, it makes sense to estimate it using the variance of the 'estimated' ϵ_i, the model *residuals*, $\hat{\epsilon}_i = y_i - x_i\hat{\beta}$. An unbiased estimator of σ^2 is:

$$\hat{\sigma^2} = \frac{1}{n-1} \sum_i (y_i - x_i\hat{\beta})^2$$

(proof of unbiasedness is given later for the general case). Plugging this into (1.3) obviously gives an unbiased estimator of the variance of $\hat{\beta}$.

1.1.2 So how old is the universe?

The least squares calculations derived above are available as part of the statistical package and environment R. The function `lm` fits linear models to data, including the simple example currently under consideration. The Cepheid distance–velocity data shown in figure 1.1 are stored in a data frame[†] `hubble`. The following R code fits the model and produces the (edited) output shown.

[†] A data frame is just a two-dimensional array of data in which the values of different variables (which may have different types) are stored in different named columns.

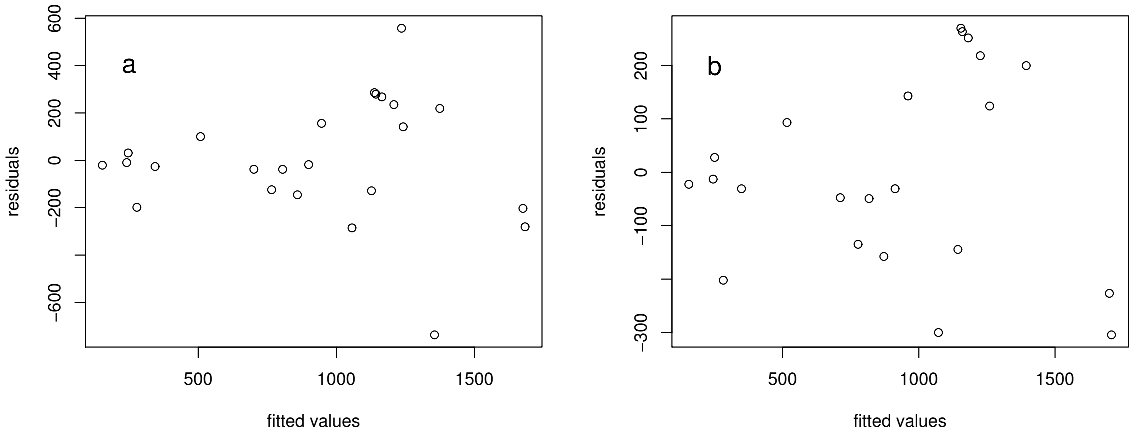

Figure 1.3 *Residuals against fitted values for* (a) *the model (1.1) fitted to all the data in figure 1.1 and* (b) *the same model fitted to data with two substantial outliers omitted.*

```
> library(gamair) ## contains 'hubble'
> data(hubble)
> hub.mod <- lm(y ~ x - 1, data=hubble)
> summary(hub.mod)

Call:
lm(formula = y ~ x - 1, data = hubble)

Coefficients:
  Estimate Std. Error
x   76.581      3.965
```

The call to `lm` passed two arguments to the function. The first is a *model formula*, `y ~ x - 1`, specifying the model to be fitted: the name of the response variable is to the left of '~' while the predictor variable is specified on the right; the '`-1`' term indicates that the model has no 'intercept' term, i.e., that the model is a straight line through the origin. The second (optional) argument gives the name of the data frame in which the variables are to be found. `lm` takes this information and uses it to fit the model by least squares: the results are returned in a 'fitted model object', which in this case has been assigned to an object called `hub.mod` for later examination. '`<-`' is the assignment operator, and `hub.mod` is created by this assignment (overwriting any previously existing object of this name).

The `summary` function is then used to examine the fitted model object. Only part of its output is shown here: $\hat{\beta}$ and the estimate of the standard error of $\hat{\beta}$ (the square root of the estimated variance of $\hat{\beta}$, derived above). Before using these quantities it is important to check the model assumptions. In particular we should check the plausibility of the assumptions that the ϵ_i are independent and all have the same variance. The way to do this is to examine residual plots.

The ‘fitted values’ of the model are defined as $\hat{\mu}_i = \hat{\beta} x_i$, while the residuals are simply $\hat{\epsilon}_i = y_i - \hat{\mu}_i$. A plot of residuals against fitted values is often useful and the following produces the plot in figure 1.3(a).

```
plot(fitted(hub.mod),residuals(hub.mod),xlab="fitted values",
     ylab="residuals")
```

What we would like to see, in such a plot, is an apparently random scatter of residuals around zero, with no trend in either the mean of the residuals, or their variability, as the fitted values increase. A trend in the mean violates the independence assumption, and is usually indicative of something missing in the model structure, while a trend in the variability violates the constant variance assumption. The main problematic feature of figure 1.3(a) is the presence of two points with very large magnitude residuals, suggesting a problem with the constant variance assumption. It is probably prudent to repeat the model fit, with and without these points, to check that they are not having undue influence on our conclusions.[‡] The following code omits the offending points and produces a new residual plot shown in figure 1.3(b).

```
> hub.mod1 <- lm(y ~ x - 1,data=hubble[-c(3,15),])
> summary(hub.mod1)

Call:
lm(formula = y ~ x - 1, data = hubble[-c(3, 15), ])

Coefficients:
  Estimate Std. Error
x    77.67       2.97

> plot(fitted(hub.mod1),residuals(hub.mod1),
+       xlab="fitted values",ylab="residuals")
```

The omission of the two large outliers has improved the residuals and changed $\hat{\beta}$ somewhat, but not drastically.

The Hubble constant estimates have units of $(\mathrm{km})\mathrm{s}^{-1}\ (\mathrm{Mpc})^{-1}$. A Mega-parsec is 3.09×10^{19}km, so we need to divide $\hat{\beta}$ by this amount, in order to obtain Hubble’s constant with units of s^{-1}. The approximate age of the universe, in seconds, is then given by the reciprocal of $\hat{\beta}$. Here are the two possible estimates expressed in years:

```
> hubble.const <- c(coef(hub.mod),coef(hub.mod1))/3.09e19
> age <- 1/hubble.const
> age/(60^2*24*365)
12794692825 12614854757
```

Both fits give an age of around 13 billion years. So we now have an idea of the best estimate of the age of the universe, but, given the measurement uncertainties, what age range would be consistent with the data?

[‡] The most common mistake made by students in first courses on regression is simply to drop data with large residuals, without further investigation. Beware of this.

1.1.3 Adding a distributional assumption

So far everything done with the simple model has been based only on the model equations and the two assumptions of independence and equal variance for the response variable. To go further and find confidence intervals for β, or test hypotheses related to the model, a further distributional assumption will be necessary.

Specifically, assume that $\epsilon_i \sim N(0, \sigma^2)$ for all i, which is equivalent to assuming $Y_i \sim N(x_i\beta, \sigma^2)$. We have already seen that $\hat{\beta}$ is just a weighted sum of the Y_i, but the Y_i are now assumed to be normal random variables, and a weighted sum of normal random variables is itself a normal random variable. Hence the estimator, $\hat{\beta}$, must be a normal random variable. Since we have already established the mean and variance of $\hat{\beta}$, we have that

$$\hat{\beta} \sim N\left(\beta, \left(\sum x_i^2\right)^{-1} \sigma^2\right). \tag{1.4}$$

Testing hypotheses about β

One thing we might want to do, is to try and evaluate the consistency of some hypothesized value of β with the data. For example, some Creation Scientists estimate the age of the universe to be 6000 years, based on a reading of the Bible. This would imply that $\beta = 163 \times 10^6$.[§] The consistency with data of such a hypothesized value for β can be based on the probability that we would have observed the $\hat{\beta}$ actually obtained, if the true value of β was the hypothetical one.

Specifically, we can test the null hypothesis, $\text{H}_0 : \beta = \beta_0$, versus the alternative hypothesis, $\text{H}_1 : \beta \neq \beta_0$, for some specified value β_0, by examining the probability of getting the observed $\hat{\beta}$, or one further from β_0, assuming H_0 to be true. If σ^2 were known then we could work directly from (1.4), as follows.

The probability required is known as the **p-value** of the test. It is the *probability of getting a value of $\hat{\beta}$ at least as favourable to* H_1 *as the one actually observed, if* H_0 *is actually true.*[¶] In this case it helps to distinguish notationally between the estimate, $\hat{\beta}_{\text{obs}}$, and estimator $\hat{\beta}$. The p-value is then

$$\begin{aligned} p &= \Pr\left(|\hat{\beta} - \beta_0| \geq |\hat{\beta}_{\text{obs}} - \beta_0| \,\middle|\, \text{H}_0\right) \\ &= \Pr\left(|\hat{\beta} - \beta_0|/\sigma_{\hat{\beta}} \geq |\hat{\beta}_{\text{obs}} - \beta_0|/\sigma_{\hat{\beta}} \,\middle|\, \text{H}_0\right) \\ &= \Pr(|Z| > |z|) \end{aligned}$$

where $Z \sim N(0, 1)$, $z = (\hat{\beta}_{\text{obs}} - \beta_0)/\sigma_{\hat{\beta}}$ and $\sigma_{\hat{\beta}}^2 = (\sum x_i^2)^{-1}\sigma^2$. Hence, having formed z, the p-value is easily worked out, using the cumulative distribution function for the standard normal built into any statistics package. Small p-values suggest that the data are inconsistent with H_0, while large values indicate consistency. 0.05 is often used as the arbitrary boundary between 'small' and 'large' in this context.

[§] This isn't really valid, of course, since the Creation Scientists are not postulating a Big Bang theory.

[¶] This definition holds for any hypothesis test, if the specific '$\hat{\beta}$' is replaced by the general '*a test statistic*'.

In reality σ^2 is usually unknown. Broadly the same testing procedure can still be adopted, by replacing σ with $\hat{\sigma}$, but we need to somehow allow for the extra uncertainty that this introduces (unless the sample size is very large). It turns out that if $\text{H}_0 : \beta = \beta_0$ is true then

$$T \equiv \frac{\hat{\beta} - \beta_0}{\hat{\sigma}_{\hat{\beta}}} \sim t_{n-1}$$

where n is the sample size, $\hat{\sigma}^2_{\hat{\beta}} = (\sum x_i^2)^{-1}\hat{\sigma}^2$, and t_{n-1} is the t-distribution with $n-1$ degrees of freedom. This result is proven in section 1.3. It is clear that large magnitude values of T favour H_1, so using T as the test statistic, in place of $\hat{\beta}$, we can calculate a p-value by evaluating

$$p = \Pr(|T| > |t|)$$

where $T \sim t_{n-1}$ and $t = (\hat{\beta}_{\text{obs}} - \beta_0)/\hat{\sigma}_{\hat{\beta}}$. This is easily evaluated using the c.d.f. of the t distributions, built into any decent statistics package. Here is some code to evaluate the p-value for H_0 : 'the Hubble constant is 163000000'.

```
> cs.hubble <- 163000000
> t.stat <- (coef(hub.mod1)-cs.hubble)/vcov(hub.mod1)[1,1]^0.5
> pt(t.stat,df=21)*2 # 2 because of |T| in p-value definition
3.906388e-150
```

i.e., as judged using t, the data would be hugely improbable if $\beta = 1.63 \times 10^8$. It would seem that the hypothesized value can be rejected rather firmly (in this case, using the data with the outliers increases the p-value by a factor of 1000 or so).

Hypothesis testing is useful when there are good reasons to stick with some null hypothesis until there are compelling grounds to reject it. This is often the case when comparing models of differing complexity: it is often a good idea to retain the simpler model until there is quite firm evidence that it is inadequate. Note one interesting property of hypothesis testing. If we choose to reject a null hypothesis whenever the p-value is less than some fixed level, α (often termed the *significance level* of a test), then we will inevitably reject a proportion, α, of correct null hypotheses. We could try and reduce the probability of such mistakes by making α very small, but in that case we pay the price of reducing the probability of rejecting H_0 when it is false.

Confidence intervals

Having seen how to test whether a *particular* hypothesized value of β is consistent with the data, the question naturally arises of what *range* of values of β would be consistent with the data? To answer this, we need to select a definition of 'consistent': a common choice is to say that any parameter value is consistent with the data if it results in a p-value of ≥ 0.05, when used as the null value in a hypothesis test.

Sticking with the Hubble constant example, and working at a significance level of 0.05, we would have rejected any hypothesized value for the constant that resulted in a t value outside the range $(-2.08, 2.08)$, since these values would result in p-values of less than 0.05. The R function `qt` can be used to find such ranges:

e.g., `qt(c(0.025,0.975),df=21)` returns the range of the middle 95% of t_{21} random variables. So we would accept any β_0 fulfilling:

$$-2.08 \le \frac{\hat{\beta} - \beta_0}{\hat{\sigma}_{\hat{\beta}}} \le 2.08$$

which rearranges to give the interval

$$\hat{\beta} - 2.08\hat{\sigma}_{\hat{\beta}} \le \beta_0 \le \hat{\beta} + 2.08\hat{\sigma}_{\hat{\beta}}.$$

Such an interval is known as a '95% confidence interval' for β.

The defining property of a 95% confidence interval is this: if we were to gather an infinite sequence of independent replicate data sets, and calculate 95% confidence intervals for β from each, then 95% of these intervals would include the true β, and 5% would not. It is easy to see how this comes about. By construction, a hypothesis test with a significance level of 5% rejects the correct null hypothesis for 5% of replicate data sets, and accepts it for the other 95% of replicates. Hence 5% of 95% confidence intervals must exclude the true parameter, while 95% include it.

A 95% CI for the Hubble constant (in the usual astrophysicists' units) is given by:

```
> sigb <- summary(hub.mod1)$coefficients[2]
> h.ci <- coef(hub.mod1)+qt(c(0.025,0.975),df=21)*sigb
> h.ci
[1] 71.49588 83.84995
```

This can be converted to a confidence interval for the age of the universe, in years:

```
> h.ci <- h.ci*60^2*24*365.25/3.09e19 # convert to 1/years
> sort(1/h.ci)
[1] 11677548698 13695361072
```

i.e., the 95% CI is (11.7,13.7) billion years. Actually this 'Hubble age' is the age of the universe if it has been expanding freely, essentially unfettered by gravitation and other effects since the Big Bang. In reality some corrections are needed to get a better estimate, and at time of writing this is about 13.8 billion years.[‖]

1.2 Linear models in general

The simple linear model, introduced above, can be generalized by allowing the response variable to depend on multiple predictor variables (plus an additive constant). These extra predictor variables can themselves be transformations of the original predictors. Here are some examples, for each of which a response variable datum, y_i, is treated as an observation on a random variable, Y_i, where $\mathbb{E}(Y_i) \equiv \mu_i$, the ϵ_i are zero mean random variables, and the β_j are model parameters, the values of which are unknown and will need to be estimated using data.

[‖] If that makes you feel young, recall that the stuff you are made of is also that old. Feeling small is better justified: there are estimated to be something like 10^{24} stars in the universe, which is approximately the number of full stops it would take to cover the surface of the earth (if it was a smooth sphere).

1. $\mu_i = \beta_0 + x_i\beta_1, \quad Y_i = \mu_i + \epsilon_i$, is a straight line relationship between y and predictor variable, x.
2. $\mu_i = \beta_0 + x_i\beta_1 + x_i^2\beta_2 + x_i^3\beta_3, \quad Y_i = \mu_i + \epsilon_i$, is a cubic model of the relationship between y and x.
3. $\mu_i = \beta_0 + x_i\beta_1 + z_i\beta_2 + \log(x_iz_i)\beta_3, \quad Y_i = \mu_i + \epsilon_i$, is a model in which y depends on predictor variables x and z and on the log of their product.

Each of these is a linear model because the ϵ_i terms and the model parameters, β_j, enter the model in a linear way. Notice that the predictor variables can enter the model non-linearly. Exactly as for the simple model, the parameters of these models can be estimated by finding the β_j values which make the models best fit the observed data, in the sense of minimizing $\sum_i (y_i - \mu_i)^2$. The theory for doing this will be developed in section 1.3, and that development is based entirely on re-writing the linear model using matrices and vectors.

To see how this re-writing is done, consider the straight line model given above. Writing out the μ_i equations for all n pairs, (x_i, y_i), results in a large system of linear equations:

$$\begin{aligned}
\mu_1 &= \beta_0 + x_1\beta_1 \\
\mu_2 &= \beta_0 + x_2\beta_1 \\
\mu_3 &= \beta_0 + x_3\beta_1 \\
. &\quad\ . \\
. &\quad\ . \\
\mu_n &= \beta_0 + x_n\beta_1
\end{aligned}$$

which can be re-written in matrix-vector form as

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ . \\ . \\ \mu_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ . & . \\ . & . \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}.$$

So the model has the general form $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$, i.e., the expected value vector $\boldsymbol{\mu}$ is given by a **model matrix** (also known as a design matrix), $\mathbf{X}$, multiplied by a parameter vector, $\boldsymbol{\beta}$. All linear models can be written in this general form.

As a second illustration, the cubic example, given above, can be written in matrix vector form as

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ . \\ . \\ \mu_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \\ . & . & . & . \\ . & . & . & . \\ 1 & x_n & x_n^2 & x_n^3 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}.$$

Models in which data are divided into different groups, each of which are assumed to have a different mean, are less obviously of the form $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$, but in fact they can be written in this way, by use of dummy indicator variables. Again, this is most easily seen by example. Consider the model

$$\mu_i = \beta_j \text{ if observation } i \text{ is in group } j,$$

and suppose that there are three groups, each with 2 data. Then the model can be re-written

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}.$$

Variables indicating the group to which a response observation belongs are known as *factor* variables. Somewhat confusingly, the groups themselves are known as *levels* of a factor. So the above model involves one factor, 'group', with three levels. Models of this type, involving factors, are commonly used for the analysis of designed experiments. In this case the model matrix depends on the design of the experiment (i.e., on which units belong to which groups), and for this reason the terms 'design matrix' and 'model matrix' are often used interchangeably. Whatever it is called, $\mathbf{X}$ is absolutely central to understanding the theory of linear models, generalized linear models and generalized additive models.

1.3 The theory of linear models

This section shows how the parameters, $\boldsymbol{\beta}$, of the linear model

$$\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}, \quad \mathbf{y} \sim N(\boldsymbol{\mu}, \mathbf{I}_n\sigma^2)$$

can be estimated by least squares. It is assumed that $\mathbf{X}$ is a full rank matrix, with n rows and p columns. It will be shown that the resulting estimator, $\hat{\boldsymbol{\beta}}$, is unbiased, has the lowest variance of any possible linear estimator of $\boldsymbol{\beta}$, and that, given the normality of the data, $\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\sigma^2)$. Results are also derived for setting confidence limits on parameters and for testing hypotheses about parameters — in particular the hypothesis that several elements of $\boldsymbol{\beta}$ are simultaneously zero.

In this section it is important not to confuse the *length* of a vector with its *dimension*. For example $(1, 1, 1)^\mathsf{T}$ has dimension 3 and length $\sqrt{3}$. Also note that no distinction has been made notationally between random variables and particular observations of those random variables: it is usually clear from the context which is meant.

1.3.1 *Least squares estimation of* $\boldsymbol{\beta}$

Point estimates of the linear model parameters, $\boldsymbol{\beta}$, can be obtained by the method of least squares, that is by minimizing

$$\mathcal{S} = \sum_{i=1}^{n} (y_i - \mu_i)^2,$$

with respect to $\boldsymbol{\beta}$. To use least squares with a linear model written in general matrix-vector form, first recall the link between the Euclidean length of a vector and the sum of squares of its elements. If $\mathbf{v}$ is any vector of dimension, n, then

$$\|\mathbf{v}\|^2 \equiv \mathbf{v}^{\mathsf{T}}\mathbf{v} \equiv \sum_{i=1}^{n} v_i^2.$$

Hence

$$\mathcal{S} = \|\mathbf{y} - \boldsymbol{\mu}\|^2 = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2.$$

Since this expression is simply the squared (Euclidian) length of the vector $\mathbf{y} - \mathbf{X}\boldsymbol{\beta}$, its value will be unchanged if $\mathbf{y} - \mathbf{X}\boldsymbol{\beta}$ is rotated. This observation is the basis for a practical method for finding $\hat{\boldsymbol{\beta}}$, and for developing the distributional results required to use linear models.

Specifically, like any real matrix, $\mathbf{X}$ can always be decomposed

$$\mathbf{X} = \mathbf{Q}\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_{\mathrm{f}}\mathbf{R} \tag{1.5}$$

where $\mathbf{R}$ is a $p \times p$ upper triangular matrix,[†] and $\mathbf{Q}$ is an $n \times n$ orthogonal matrix, the first p columns of which form $\mathbf{Q}_{\mathrm{f}}$ (see B.6). Recall that orthogonal matrices rotate or reflect vectors, but do not change their length. Orthogonality also means that $\mathbf{Q}\mathbf{Q}^{\mathsf{T}} = \mathbf{Q}^{\mathsf{T}}\mathbf{Q} = \mathbf{I}_n$. Applying $\mathbf{Q}^{\mathsf{T}}$ to $\mathbf{y} - \mathbf{X}\boldsymbol{\beta}$ implies that

$$\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 = \|\mathbf{Q}^{\mathsf{T}}\mathbf{y} - \mathbf{Q}^{\mathsf{T}}\mathbf{X}\boldsymbol{\beta}\|^2 = \left\|\mathbf{Q}^{\mathsf{T}}\mathbf{y} - \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}\boldsymbol{\beta}\right\|^2.$$

Writing $\mathbf{Q}^{\mathsf{T}}\mathbf{y} = \begin{bmatrix} \mathbf{f} \\ \mathbf{r} \end{bmatrix}$, where $\mathbf{f}$ is vector of dimension p, and hence $\mathbf{r}$ is a vector of dimension $n - p$, yields

$$\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 = \left\|\begin{bmatrix} \mathbf{f} \\ \mathbf{r} \end{bmatrix} - \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}\boldsymbol{\beta}\right\|^2 = \|\mathbf{f} - \mathbf{R}\boldsymbol{\beta}\|^2 + \|\mathbf{r}\|^2.^{[‡]}$$

The length of $\mathbf{r}$ does not depend on $\boldsymbol{\beta}$, while $\|\mathbf{f} - \mathbf{R}\boldsymbol{\beta}\|^2$ can be reduced to zero by choosing $\boldsymbol{\beta}$ so that $\mathbf{R}\boldsymbol{\beta}$ equals $\mathbf{f}$. Hence

$$\hat{\boldsymbol{\beta}} = \mathbf{R}^{-1}\mathbf{f} \tag{1.6}$$

[†] i.e., $R_{i,j} = 0$ if $i > j$.

[‡] If the last equality isn't obvious recall that $\|\mathbf{x}\|^2 = \sum_i x_i^2$, so if $\mathbf{x} = \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}$, $\|\mathbf{x}\|^2 = \sum_i v_i^2 + \sum_i w_i^2 = \|\mathbf{v}\|^2 + \|\mathbf{w}\|^2$.

is the least squares estimator of $\boldsymbol{\beta}$. Notice that $\|\mathbf{r}\|^2 = \|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2$, the *residual sum of squares* for the model fit.

1.3.2 The distribution of $\hat{\boldsymbol{\beta}}$

The distribution of the estimator, $\hat{\boldsymbol{\beta}}$, follows from that of $\mathbf{Q}^{\mathsf{T}}\mathbf{y}$. Multivariate normality of $\mathbf{Q}^{\mathsf{T}}\mathbf{y}$ follows from that of $\mathbf{y}$, and since the covariance matrix of $\mathbf{y}$ is $\mathbf{I}_n\sigma^2$, the covariance matrix of $\mathbf{Q}^{\mathsf{T}}\mathbf{y}$ is

$$\mathbf{V}_{\mathbf{Q}^{\mathsf{T}}\mathbf{y}} = \mathbf{Q}^{\mathsf{T}}\mathbf{I}_n\mathbf{Q}\sigma^2 = \mathbf{I}_n\sigma^2.$$

Furthermore,

$$\mathbb{E}\begin{bmatrix}\mathbf{f}\\ \mathbf{r}\end{bmatrix} = \mathbb{E}(\mathbf{Q}^{\mathsf{T}}\mathbf{y}) = \mathbf{Q}^{\mathsf{T}}\mathbf{X}\boldsymbol{\beta} = \begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}\boldsymbol{\beta}$$

$$\Rightarrow \mathbb{E}(\mathbf{f}) = \mathbf{R}\boldsymbol{\beta} \text{ and } \mathbb{E}(\mathbf{r}) = \mathbf{0},$$

i.e., we have that

$$\mathbf{f} \sim N(\mathbf{R}\boldsymbol{\beta}, \mathbf{I}_p\sigma^2) \text{ and } \mathbf{r} \sim N(\mathbf{0}, \mathbf{I}_{n-p}\sigma^2)$$

with both vectors independent of each other.

Turning to the properties of $\hat{\boldsymbol{\beta}}$ itself, unbiasedness follows immediately:

$$\mathbb{E}(\hat{\boldsymbol{\beta}}) = \mathbf{R}^{-1}\mathbb{E}(\mathbf{f}) = \mathbf{R}^{-1}\mathbf{R}\boldsymbol{\beta} = \boldsymbol{\beta}.$$

Since the covariance matrix of $\mathbf{f}$ is $\mathbf{I}_p\sigma^2$, it follows that the covariance matrix of $\hat{\boldsymbol{\beta}}$ is

$$\mathbf{V}_{\hat{\beta}} = \mathbf{R}^{-1}\mathbf{I}_p\mathbf{R}^{-\mathsf{T}}\sigma^2 = \mathbf{R}^{-1}\mathbf{R}^{-\mathsf{T}}\sigma^2. \tag{1.7}$$

Furthermore, since $\hat{\boldsymbol{\beta}}$ is just a linear transformation of the normal random variables $\mathbf{f}$, it must have a multivariate normal distribution,

$$\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \mathbf{V}_{\hat{\beta}}).$$

The foregoing distributional result is not usually directly useful for making inferences about $\boldsymbol{\beta}$, since σ^2 is generally unknown and must be estimated, thereby introducing an extra component of variability that should be accounted for.

1.3.3 $(\hat{\beta}_i - \beta_i)/\hat{\sigma}_{\hat{\beta}_i} \sim t_{n-p}$

Since the elements of $\mathbf{r}$ are i.i.d. $N(0, \sigma^2)$, the r_i/σ are i.i.d. $N(0, 1)$ random variables, and hence

$$\frac{1}{\sigma^2}\|\mathbf{r}\|^2 = \frac{1}{\sigma^2}\sum_{i=1}^{n-p} r_i^2 \sim \chi^2_{n-p}.$$

The mean of a χ^2_{n-p} r.v. is $n - p$, so this result is sufficient (but not necessary: see exercise 7) to imply that

$$\hat{\sigma}^2 = \|\mathbf{r}\|^2/(n-p) \tag{1.8}$$

is an unbiased estimator of σ^2.[§] The independence of the elements of $\mathbf{r}$ and $\mathbf{f}$ also implies that $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ are independent.

Now consider a single parameter estimator, $\hat{\boldsymbol{\beta}}_i$, with standard deviation, $\sigma_{\hat{\beta}_i}$, given by the square root of element i,i of $\mathbf{V}_{\hat{\beta}}$. An unbiased estimator of $\mathbf{V}_{\hat{\beta}}$ is $\hat{\mathbf{V}}_{\hat{\beta}} = \mathbf{V}_{\hat{\beta}}\hat{\sigma}^2/\sigma^2 = \mathbf{R}^{-1}\mathbf{R}^{-\mathsf{T}}\hat{\sigma}^2$, so an estimator, $\hat{\sigma}_{\hat{\beta}_i}$, is given by the square root of element i,i of $\hat{\mathbf{V}}_{\hat{\beta}}$, and it is clear that $\hat{\sigma}_{\hat{\beta}_i} = \sigma_{\hat{\beta}_i}\hat{\sigma}/\sigma$. Hence

$$\frac{\hat{\beta}_i - \beta_i}{\hat{\sigma}_{\hat{\beta}_i}} = \frac{\hat{\beta}_i - \beta_i}{\sigma_{\hat{\beta}_i}\hat{\sigma}/\sigma} = \frac{(\hat{\beta}_i - \beta_i)/\sigma_{\hat{\beta}_i}}{\sqrt{\frac{1}{\sigma^2}\|\mathbf{r}\|^2/(n-p)}} \sim \frac{N(0,1)}{\sqrt{\chi^2_{n-p}/(n-p)}} \sim t_{n-p}$$

(where the independence of $\hat{\beta}_i$ and $\hat{\sigma}^2$ has been used). This result enables confidence intervals for β_i to be found, and is the basis for hypothesis tests about individual β_i's (for example, $\mathrm{H}_0 : \beta_i = 0$).

1.3.4 F-ratio results I

Sometimes it is useful to be able to test $\mathrm{H}_0 : \mathbf{C}\boldsymbol{\beta} = \mathbf{d}$, where $\mathbf{C}$ is $q \times p$ and rank q ($< p$). Under H_0 we have $\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d} \sim N(\mathbf{0}, \mathbf{C}\mathbf{V}_{\hat{\beta}}\mathbf{C}^\mathsf{T},)$, from basic properties of the transformation of normal random vectors. Forming a Cholesky decomposition $\mathbf{L}^\mathsf{T}\mathbf{L} = \mathbf{C}\mathbf{V}_{\hat{\beta}}\mathbf{C}^\mathsf{T}$ (see B.7), it is then easy to show that $\mathbf{L}^{-\mathsf{T}}(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d}) \sim N(\mathbf{0}, \mathbf{I})$, so,

$$(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d})^\mathsf{T}(\mathbf{C}\mathbf{V}_{\hat{\beta}}\mathbf{C}^\mathsf{T})^{-1}(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d}) = \\ (\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d})^\mathsf{T}\mathbf{L}^{-1}\mathbf{L}^{-\mathsf{T}}(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d}) \sim \sum_{i=1}^{q} N(0,1)^2 \sim \chi^2_q.$$

As in the previous section, plugging in $\hat{\mathbf{V}}_{\hat{\beta}} = \mathbf{V}_{\hat{\beta}}\hat{\sigma}^2/\sigma^2$ gives the computable test statistic and its distribution under H_0:

$$\frac{1}{q}(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d})^\mathsf{T}(\mathbf{C}\hat{V}_{\hat{\beta}}\mathbf{C}^\mathsf{T})^{-1}(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d}) = \frac{\sigma^2}{q\hat{\sigma}^2}(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d})^\mathsf{T}(\mathbf{C}V_{\hat{\beta}}\mathbf{C}^\mathsf{T})^{-1}(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d}) \\ = \frac{(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d})^\mathsf{T}(\mathbf{C}V_{\hat{\beta}}\mathbf{C}^\mathsf{T})^{-1}(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d})/q}{\frac{1}{\sigma^2}\|\mathbf{r}\|^2/(n-p)} \sim \frac{\chi^2_q/q}{\chi^2_{n-p}/(n-p)} \sim F_{q,n-p}. \tag{1.9}$$

This result can be used to compute a p-value for the test.

1.3.5 F-ratio results II

An alternative F-ratio test derivation is also useful. Consider testing the simultaneous equality to zero of several model parameters. Such tests are useful for making

[§] Don't forget that $\|\mathbf{r}\|^2 = \|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2$.

inferences about factor variables and their interactions, since each factor or interaction is typically represented by several elements of $\boldsymbol{\beta}$. More specifically suppose that the model matrix is partitioned $\mathbf{X} = [\mathbf{X}_0 : \mathbf{X}_1]$, where $\mathbf{X}_0$ and $\mathbf{X}_1$ have $p - q$ and q columns, respectively. Let $\boldsymbol{\beta}_0$ and $\boldsymbol{\beta}_1$ be the corresponding sub-vectors of $\boldsymbol{\beta}$, and consider testing

$$\mathrm{H}_0 : \boldsymbol{\beta}_1 = \mathbf{0} \quad \text{versus} \quad \mathrm{H}_1 : \boldsymbol{\beta}_1 \neq \mathbf{0}.$$

Any test involving the comparison of a linear model with a simplified null version of the model can be written in this form, by re-ordering of the columns of $\mathbf{X}$ or by re-parameterization. Now

$$\mathbf{Q}^\mathsf{T}\mathbf{X}_0 = \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{R}_0$ is the first $p - q$ rows and columns of $\mathbf{R}$ ($\mathbf{Q}$ and $\mathbf{R}$ are from (1.5)). So rotating $\mathbf{y} - \mathbf{X}_0\boldsymbol{\beta}_0$ using $\mathbf{Q}^\mathsf{T}$ implies that

$$\|\mathbf{y} - \mathbf{X}_0\boldsymbol{\beta}_0\|^2 = \|\mathbf{Q}^\mathsf{T}\mathbf{y} - \mathbf{Q}^\mathsf{T}\mathbf{X}_0\boldsymbol{\beta}_0\|^2 = \|\mathbf{f}_0 - \mathbf{R}_0\boldsymbol{\beta}_0\|^2 + \|\mathbf{f}_1\|^2 + \|\mathbf{r}\|^2$$

where $\mathbf{f}$ has been partitioned so that $\mathbf{f} = \begin{bmatrix} \mathbf{f}_0 \\ \mathbf{f}_1 \end{bmatrix}$ ($\mathbf{f}_1$ being of dimension q). Hence $\|\mathbf{f}_1\|^2$ is the increase in residual sum of squares that results from dropping $\mathbf{X}_1$ from the model (i.e. setting $\boldsymbol{\beta}_1 = \mathbf{0}$).

Now, since $\mathbb{E}(\mathbf{f}) = \mathbf{R}\boldsymbol{\beta}$ and $\mathbf{R}$ is upper triangular, then $\mathbb{E}(\mathbf{f}_1) = \mathbf{0}$ if $\boldsymbol{\beta}_1 = \mathbf{0}$ (i.e. if H_0 is true). Also, we already know that the elements of $\mathbf{f}_1$ are independent normal r.v.s with variance σ^2. Hence, if H_0 is true, $\mathbf{f}_1 \sim N(\mathbf{0}, \mathbf{I}_q\sigma^2)$ and consequently

$$\frac{1}{\sigma^2}\|\mathbf{f}_1\|^2 \sim \chi^2_q.$$

So, forming an ‘F-ratio statistic’, F, assuming H_0, and recalling the independence of $\mathbf{f}$ and $\mathbf{r}$ we have

$$F = \frac{\|\mathbf{f}_1\|^2/q}{\hat{\sigma}^2} = \frac{\frac{1}{\sigma^2}\|\mathbf{f}_1\|^2/q}{\frac{1}{\sigma^2}\|\mathbf{r}\|^2/(n-p)} \sim \frac{\chi^2_q/q}{\chi^2_{n-p}/(n-p)} \sim F_{q,n-p}$$

which can be used to compute a p-value for H_0, thereby testing the significance of model terms. Notice that for comparing any null model with residual sum of squares, RSS_0, to a full model with residual sum of squares, RSS_1, the preceding derivation holds, but the test statistic can also be computed (without any initial re-ordering or re-parameterization) as

$$F = \frac{(\mathrm{RSS}_0 - \mathrm{RSS}_1)/q}{\mathrm{RSS}_1/(n-p)}.$$

In slightly more generality, if $\boldsymbol{\beta}$ is partitioned into sub-vectors $\boldsymbol{\beta}_0, \boldsymbol{\beta}_1 \ldots, \boldsymbol{\beta}_m$ (each usually relating to a different effect), of dimensions $q_0, q_1, \ldots, q_m$, then $\mathbf{f}$ can also be so partitioned, $\mathbf{f}^\mathsf{T} = [\mathbf{f}_0^\mathsf{T}, \mathbf{f}_1^\mathsf{T}, \ldots, \mathbf{f}_m^\mathsf{T}]$, and tests of

$$\mathrm{H}_0 : \boldsymbol{\beta}_j = \mathbf{0} \text{ versus } \mathrm{H}_1 : \boldsymbol{\beta}_j \neq \mathbf{0}$$

are conducted using the result that under H_0

$$F = \frac{\|\mathbf{f}_j\|^2/q_j}{\hat{\sigma}^2} \sim F_{q_j,n-p},$$

with F larger than this suggests, if the alternative is true. This is the result used to draw up sequential ANOVA tables for a fitted model, of the sort produced by a single argument call to `anova` in R. Note, however, that the hypothesis test about $\boldsymbol{\beta}_j$ is only valid in general if $\boldsymbol{\beta}_k = \mathbf{0}$ for all k such that $j < k \le m$: this follows from the way that the test was derived, and is the reason that the ANOVA tables resulting from such procedures are referred to as 'sequential' tables. The practical upshot is that, if models are reduced in a different order, the p-values obtained will be different. The exception to this is if the $\hat{\boldsymbol{\beta}}_j$'s are mutually independent, in which case all the tests are simultaneously valid, and the ANOVA table for a model is not dependent on the order of model terms: such independent $\hat{\boldsymbol{\beta}}_j$'s usually arise only in the context of balanced data, from designed experiments.

Notice that sequential ANOVA tables are very easy to calculate: once a model has been fitted by the QR method, all the relevant 'sums of squares' are easily calculated directly from the elements of $\mathbf{f}$, with $\|\mathbf{r}\|^2$ providing the residual sum of squares.

1.3.6 The influence matrix

One matrix that will feature extensively in the discussion of GAMs is the *influence matrix* (or *hat matrix*) of a linear model. This is the matrix which yields the fitted value vector, $\hat{\boldsymbol{\mu}}$, when post-multiplied by the data vector, $\mathbf{y}$. Recalling the definition of $\mathbf{Q}_\mathrm{f}$, as being the first p columns of $\mathbf{Q}$, $\mathbf{f} = \mathbf{Q}_\mathrm{f}^\mathsf{T}\mathbf{y}$ and so

$$\hat{\boldsymbol{\beta}} = \mathbf{R}^{-1}\mathbf{Q}_\mathrm{f}^\mathsf{T}\mathbf{y}.$$

Furthermore $\hat{\boldsymbol{\mu}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ and $\mathbf{X} = \mathbf{Q}_\mathrm{f}\mathbf{R}$ so

$$\hat{\boldsymbol{\mu}} = \mathbf{Q}_\mathrm{f}\mathbf{R}\mathbf{R}^{-1}\mathbf{Q}_\mathrm{f}^\mathsf{T}\mathbf{y} = \mathbf{Q}_\mathrm{f}\mathbf{Q}_\mathrm{f}^\mathsf{T}\mathbf{y}$$

i.e., the matrix $\mathbf{A} \equiv \mathbf{Q}_\mathrm{f}\mathbf{Q}_\mathrm{f}^\mathsf{T}$ is the influence (hat) matrix such that $\hat{\boldsymbol{\mu}} = \mathbf{A}\mathbf{y}$.

The influence matrix has a couple of interesting properties. Firstly, the trace of the influence matrix is the number of (identifiable) parameters in the model, since

$$\mathrm{tr}\left(\mathbf{A}\right) = \mathrm{tr}\left(\mathbf{Q}_\mathrm{f}\mathbf{Q}_\mathrm{f}^\mathsf{T}\right) = \mathrm{tr}\left(\mathbf{Q}_\mathrm{f}^\mathsf{T}\mathbf{Q}_\mathrm{f}\right) = \mathrm{tr}\left(\mathbf{I}_p\right) = p.$$

Secondly, $\mathbf{A}\mathbf{A} = \mathbf{A}$, a property known as *idempotency*. Again the proof is simple:

$$\mathbf{A}\mathbf{A} = \mathbf{Q}_\mathrm{f}\mathbf{Q}_\mathrm{f}^\mathsf{T}\mathbf{Q}_\mathrm{f}\mathbf{Q}_\mathrm{f}^\mathsf{T} = \mathbf{Q}_\mathrm{f}\mathbf{I}_p\mathbf{Q}_\mathrm{f}^\mathsf{T} = \mathbf{Q}_\mathrm{f}\mathbf{Q}_\mathrm{f}^\mathsf{T} = \mathbf{A}.$$

1.3.7 The residuals, $\hat{\epsilon}$, and fitted values, $\hat{\boldsymbol{\mu}}$

The influence matrix is helpful in deriving properties of the fitted values, $\hat{\boldsymbol{\mu}}$, and residuals, $\hat{\epsilon}$. $\hat{\boldsymbol{\mu}}$ is unbiased, since $\mathbb{E}(\hat{\boldsymbol{\mu}}) = \mathbb{E}(\mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{X}\mathbb{E}(\hat{\boldsymbol{\beta}}) = \mathbf{X}\boldsymbol{\beta} = \boldsymbol{\mu}$. The covariance matrix of the fitted values is obtained from the fact that $\hat{\boldsymbol{\mu}}$ is a linear transformation of the random vector $\mathbf{y}$, which has covariance matrix $\mathbf{I}_n\sigma^2$, so that

$$\mathbf{V}_{\hat{\boldsymbol{\mu}}} = \mathbf{A}\mathbf{I}_n\mathbf{A}^\mathsf{T}\sigma^2 = \mathbf{A}\sigma^2,$$

by the idempotence (and symmetry) of $\mathbf{A}$. The distribution of $\hat{\boldsymbol{\mu}}$ is degenerate multivariate normal.

Similar arguments apply to the residuals.

$$\hat{\boldsymbol{\epsilon}} = (\mathbf{I} - \mathbf{A})\mathbf{y},$$

so

$$\mathbb{E}(\hat{\boldsymbol{\epsilon}}) = \mathbb{E}(\mathbf{y}) - \mathbb{E}(\hat{\boldsymbol{\mu}}) = \boldsymbol{\mu} - \boldsymbol{\mu} = \mathbf{0}.$$

Proceeding as in the fitted value case we have

$$\mathbf{V}_{\hat{\boldsymbol{\epsilon}}} = (\mathbf{I}_n - \mathbf{A})\mathbf{I}_n(\mathbf{I}_n - \mathbf{A})^\mathsf{T}\sigma^2 = (\mathbf{I}_n - 2\mathbf{A} + \mathbf{A}\mathbf{A})\,\sigma^2 = (\mathbf{I}_n - \mathbf{A})\,\sigma^2.$$

Again, the distribution of the residuals will be degenerate normal. The results for the residuals are useful for model checking, since they allow the residuals to be standardized so that they should have constant variance if the model is correct.

1.3.8 Results in terms of $\mathbf{X}$

The presentation so far has been in terms of the method actually used to fit linear models in practice (the QR decomposition approach[¶]), which also greatly facilitates the derivation of the distributional results required for practical modelling. However, for historical reasons, these results are more usually presented in terms of the model matrix, $\mathbf{X}$, rather than the components of its QR decomposition. For completeness some of the results are restated here, in terms of $\mathbf{X}$.

Firstly consider the covariance matrix of $\boldsymbol{\beta}$. This turns out to be $(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\sigma^2$, which is easily seen to be equivalent to (1.7) as follows:

$$\mathbf{V}_{\hat{\boldsymbol{\beta}}} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\sigma^2 = \left(\mathbf{R}^\mathsf{T}\mathbf{Q}_\mathrm{f}^\mathsf{T}\mathbf{Q}_\mathrm{f}\mathbf{R}\right)^{-1}\sigma^2 = \left(\mathbf{R}^\mathsf{T}\mathbf{R}\right)^{-1}\sigma^2 = \mathbf{R}^{-1}\mathbf{R}^{-\mathsf{T}}\sigma^2.$$

The expression for the least squares estimates is $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}$, which is equivalent to (1.6):

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} = \mathbf{R}^{-1}\mathbf{R}^{-\mathsf{T}}\mathbf{R}^\mathsf{T}\mathbf{Q}_\mathrm{f}^\mathsf{T}\mathbf{y} = \mathbf{R}^{-1}\mathbf{Q}_\mathrm{f}^\mathsf{T}\mathbf{y} = \mathbf{R}^{-1}\mathbf{f}.$$

Given this last result, it is easy to see that the influence matrix can be written:

$$\mathbf{A} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}.$$

These results are of largely historical and theoretical interest: they should not generally be used for computational purposes, and derivation of the distributional results is much more difficult if one starts from these formulae.

[¶] Some software still fits models by solution of $\mathbf{X}^\mathsf{T}\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^\mathsf{T}\mathbf{y}$, but this is less computationally stable than the orthogonal decomposition method described here, although it is a bit faster.

1.3.9 The Gauss Markov Theorem: What's special about least squares?

How good are least squares estimators? In particular, might it be possible to find better estimators, in the sense of having lower variance while still being unbiased? The Gauss Markov theorem shows that least squares estimators have the lowest variance of all unbiased estimators that are linear (meaning that the data only enter the estimation process in a linear way).

Theorem 1. *Suppose that* $\boldsymbol{\mu} \equiv \mathbb{E}(\mathbf{Y}) = \mathbf{X}\boldsymbol{\beta}$ *and* $\mathbf{V}_y = \sigma^2\mathbf{I}$*, and let* $\tilde{\phi} = \mathbf{c}^\mathsf{T}\mathbf{Y}$ *be any* unbiased *linear estimator of* $\phi = \mathbf{t}^\mathsf{T}\boldsymbol{\beta}$*, where* $\mathbf{t}$ *is an arbitrary vector. Then:*

$$\mathrm{var}(\tilde{\phi}) \geq \mathrm{var}(\hat{\phi})$$

where $\hat{\phi} = \mathbf{t}^\mathsf{T}\hat{\boldsymbol{\beta}}$*, and* $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{Y}$ *is the least squares estimator of* $\boldsymbol{\beta}$*. Notice that, since* $\mathbf{t}$ *is arbitrary, this theorem implies that each element of* $\hat{\boldsymbol{\beta}}$ *is a minimum variance unbiased estimator.*

Proof. Since $\tilde{\phi}$ is a linear transformation of $\mathbf{Y}$, $\mathrm{var}(\tilde{\phi}) = \mathbf{c}^\mathsf{T}\mathbf{c}\sigma^2$. To compare variances of $\hat{\phi}$ and $\tilde{\phi}$ it is also useful to express $\mathrm{var}(\hat{\phi})$ in terms of $\mathbf{c}$. To do this, note that unbiasedness of $\tilde{\phi}$ implies that

$$\mathbb{E}(\mathbf{c}^\mathsf{T}\mathbf{Y}) = \mathbf{t}^\mathsf{T}\boldsymbol{\beta} \Rightarrow \mathbf{c}^\mathsf{T}\mathbb{E}(\mathbf{Y}) = \mathbf{t}^\mathsf{T}\boldsymbol{\beta} \Rightarrow \mathbf{c}^\mathsf{T}\mathbf{X}\boldsymbol{\beta} = \mathbf{t}^\mathsf{T}\boldsymbol{\beta} \Rightarrow \mathbf{c}^\mathsf{T}\mathbf{X} = \mathbf{t}^\mathsf{T}.$$

So the variance of $\hat{\phi}$ can be written as

$$\mathrm{var}(\hat{\phi}) = \mathrm{var}(\mathbf{t}^\mathsf{T}\hat{\boldsymbol{\beta}}) = \mathrm{var}(\mathbf{c}^\mathsf{T}\mathbf{X}\hat{\boldsymbol{\beta}}).$$

This is the variance of a linear transformation of $\hat{\boldsymbol{\beta}}$, and the covariance matrix of $\hat{\boldsymbol{\beta}}$ is $(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\sigma^2$, so

$$\mathrm{var}(\hat{\phi}) = \mathrm{var}(\mathbf{c}^\mathsf{T}\mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{c}^\mathsf{T}\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{c}\sigma^2 = \mathbf{c}^\mathsf{T}\mathbf{A}\mathbf{c}\sigma^2$$

(where $\mathbf{A}$ is the influence or hat matrix). Now the variances of the two estimators can be directly compared, and it can be seen that $\mathrm{var}(\tilde{\phi}) \geq \mathrm{var}(\hat{\phi})$ iff

$$\mathbf{c}^\mathsf{T}(\mathbf{I} - \mathbf{A})\mathbf{c} \geq 0.$$

This condition will always be met, because it is equivalent to:

$$\{(\mathbf{I} - \mathbf{A})\mathbf{c}\}^\mathsf{T}(\mathbf{I} - \mathbf{A})\mathbf{c} \geq 0$$

by the idempotency and symmetry of $\mathbf{A}$ and hence of $(\mathbf{I} - \mathbf{A})$, but this last condition is saying that a sum of squares can not be less than 0, which is clearly true. □

Notice that this theorem uses independence and equal variance assumptions, but does not assume normality. Of course there is a sense in which the theorem is intuitively rather unsurprising, since it says that the minimum variance estimators are those obtained by seeking to minimize the residual variance.

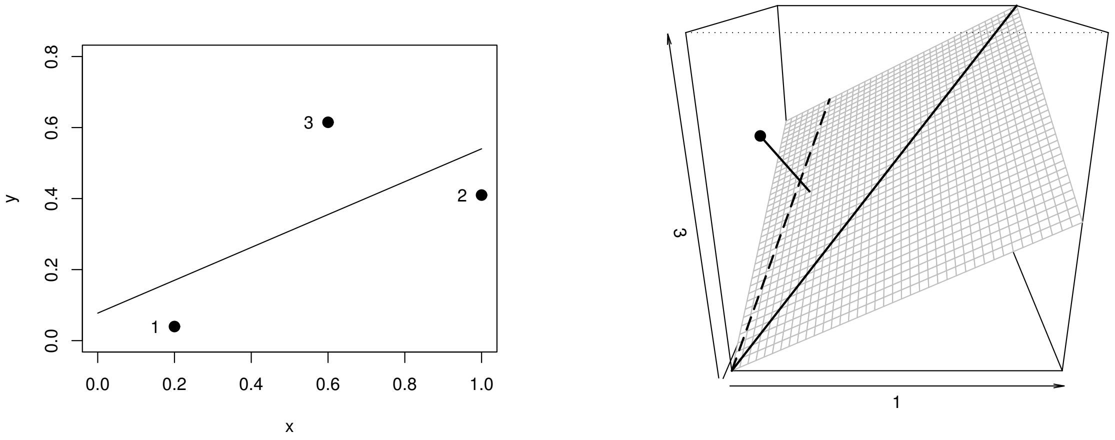

Figure 1.4 *The geometry of least squares. The left panel shows a straight line model fitted to 3 data by least squares. The right panel gives a geometric interpretation of the fitting process. The 3-dimensional space shown is spanned by 3 orthogonal axes: one for each response variable. The observed response vector,* $\mathbf{y}$*, is shown as a point (●) within this space. The columns of the model matrix define two directions within the space: the thick and dashed lines from the origin. The model states that* $\mathbb{E}(\mathbf{y})$ *could be any linear combination of these vectors, i.e., anywhere in the 'model subspace' indicated by the grey plane. Least squares fitting finds the closest point in the model subspace to the response data (●): the 'fitted values'. The short thick line joins the response data to the fitted values: it is the 'residual vector'.*

1.4 The geometry of linear modelling

A full understanding of what is happening when models are fitted by least squares is facilitated by taking a geometric view of the fitting process. Some of the results derived in the last few sections become rather obvious when viewed in this way.

1.4.1 Least squares

Again consider the linear model,

$$\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}, \quad \mathbf{y} \sim N(\boldsymbol{\mu}, \mathbf{I}_n\sigma^2),$$

where $\mathbf{X}$ is an $n \times p$ model matrix. But now consider an n-dimensional Euclidean space, $\Re^n$, in which $\mathbf{y}$ defines the location of a single point. The space of all possible linear combinations of the columns of $\mathbf{X}$ defines a subspace of $\Re^n$, the elements of this space being given by $\mathbf{X}\boldsymbol{\beta}$, where $\boldsymbol{\beta}$ can take any value in $\Re^p$: this space will be referred to as the *space of* $\mathbf{X}$ (strictly the *column* space). So, a linear model states that $\boldsymbol{\mu}$, the expected value of $\mathbf{Y}$, lies in the space of $\mathbf{X}$. Estimating a linear model by least squares, amounts to finding the point, $\hat{\boldsymbol{\mu}} \equiv \mathbf{X}\hat{\boldsymbol{\beta}}$, in the space of $\mathbf{X}$, that is closest to the observed data $\mathbf{y}$. Equivalently, $\hat{\boldsymbol{\mu}}$ is the orthogonal projection of $\mathbf{y}$ on to the space of $\mathbf{X}$. An obvious, but important, consequence of this is that the residual vector, $\hat{\boldsymbol{\epsilon}} = \mathbf{y} - \hat{\boldsymbol{\mu}}$, is orthogonal to all vectors in the space of $\mathbf{X}$.

Figure 1.4 illustrates these ideas for the simple case of a straight line regression model for 3 data (shown conventionally in the left hand panel). The response data

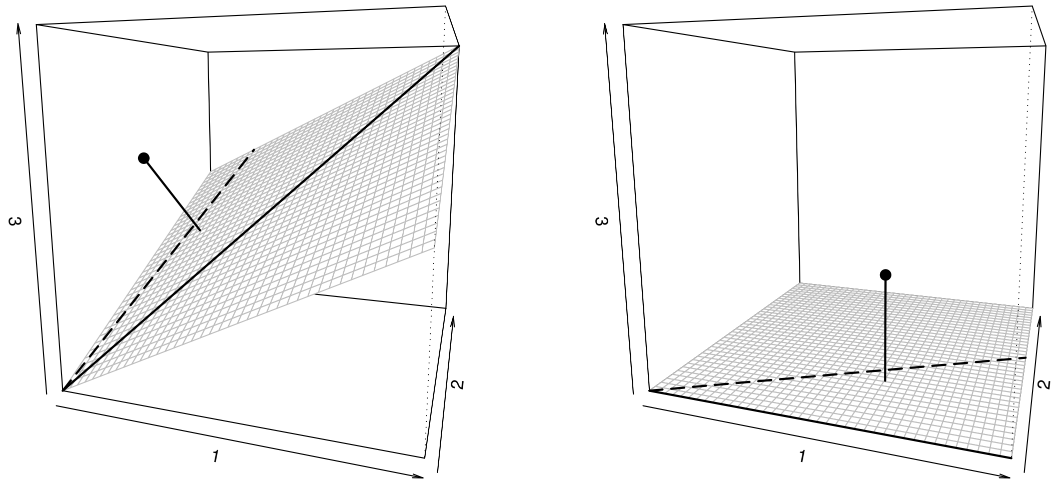

Figure 1.5 *The geometry of fitting via orthogonal decompositions. The left panel illustrates the geometry of the simple straight line model of 3 data introduced in figure 1.4. The right hand panel shows how this original problem appears after rotation by* $\mathbf{Q}^\mathsf{T}$*, the transpose of the orthogonal factor in a QR decomposition of* $\mathbf{X}$*. Notice that in the rotated problem the model subspace only has non-zero components relative to* p *axes (2 axes for this example), while the residual vector has only zero components relative to those same axes.*

and model are

$$\mathbf{y} = \begin{bmatrix} .04 \\ .41 \\ .62 \end{bmatrix} \text{ and } \boldsymbol{\mu} = \begin{bmatrix} 1 & 0.2 \\ 1 & 1.0 \\ 1 & 0.6 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}.$$

Since $\boldsymbol{\beta}$ is unknown, the model simply says that $\boldsymbol{\mu}$ could be any linear combination of the vectors $[1, 1, 1]^\mathsf{T}$ and $[.2, 1, .6]^\mathsf{T}$. As the right hand panel of figure 1.4 illustrates, fitting the model by least squares amounts to finding the particular linear combination of the columns of these vectors that is as close to $\mathbf{y}$ as possible (in terms of Euclidean distance).

1.4.2 Fitting by orthogonal decompositions

Recall that the actual calculation of least squares estimates involves first forming the QR decomposition of the model matrix, so that

$$\mathbf{X} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix},$$

where $\mathbf{Q}$ is an $n \times n$ orthogonal matrix and $\mathbf{R}$ is a $p \times p$ upper triangular matrix. Orthogonal matrices rotate vectors (without changing their length) and the first step in least squares estimation is to rotate both the response vector, $\mathbf{y}$, and the columns of the model matrix, $\mathbf{X}$, in exactly the same way, by pre-multiplication with $\mathbf{Q}^\mathsf{T}$.[‖]

[‖] In fact the QR decomposition is not uniquely defined, in that the sign of rows of $\mathbf{Q}$, and corresponding columns of $\mathbf{R}$, can be switched, without changing $\mathbf{X}$ — these sign changes are equivalent to reflections

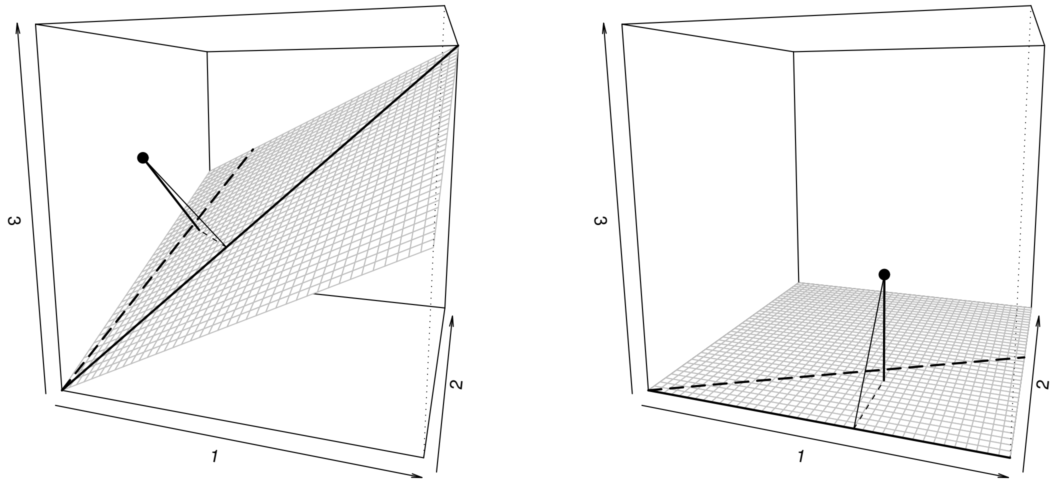

Figure 1.6 *The geometry of nested models.*

Figure 1.5 illustrates this rotation for the example shown in figure 1.4. The left panel shows the response data and model space, for the original problem, while the right hand panel shows the data and model space after rotation by $\mathbf{Q}^\mathsf{T}$. Notice that, since the problem has simply been rotated, the relative position of the data and basis vectors (columns of $\mathbf{X}$) has not changed. What has changed is that the problem now has a particularly convenient orientation relative to the axes. The first two components of the fitted value vector can now be read directly from axes 1 and 2, while the third component is simply zero. By contrast, the residual vector has zero components relative to axes 1 and 2, and its non-zero component can be read directly from axis 3. In terms of section 1.3.1, these vectors are $[\mathbf{f}^\mathsf{T}, \mathbf{0}^\mathsf{T}]^\mathsf{T}$ and $[\mathbf{0}^\mathsf{T}, \mathbf{r}^\mathsf{T}]^\mathsf{T}$, respectively.

The $\hat{\boldsymbol{\beta}}$ corresponding to the fitted values is now easily obtained. Of course we usually require fitted values and residuals to be expressed in terms of the un-rotated problem, but this is simply a matter of reversing the rotation using $\mathbf{Q}$, i.e.,

$$\hat{\boldsymbol{\mu}} = \mathbf{Q}\left[\begin{array}{c}\mathbf{f}\\ \mathbf{0}\end{array}\right], \text{ and } \hat{\boldsymbol{\epsilon}} = \mathbf{Q}\left[\begin{array}{c}\mathbf{0}\\ \mathbf{r}\end{array}\right].$$

1.4.3 Comparison of nested models

A linear model with model matrix $\mathbf{X}_0$ is nested within a linear model with model matrix $\mathbf{X}_1$ if they are models for the same response data, and the columns of $\mathbf{X}_0$ span a subspace of the space spanned by the columns of $\mathbf{X}_1$. Usually this simply means that $\mathbf{X}_1$ is $\mathbf{X}_0$ with some extra columns added.

The vector of the difference between the fitted values of two nested linear models is entirely within the subspace of the larger model, and is therefore orthogonal to the residual vector for the larger model. This fact is geometrically obvious, as figure

of vectors, and the sign leading to maximum numerical stability is usually selected in practice. These reflections don't introduce any extra conceptual difficulty, but can make plots less easy to understand, so I have suppressed them in this example.

1.6 illustrates, but it is a key reason why F-ratio statistics have a relatively simple distribution (under the simpler model).

Figure 1.6 is again based on the same simple straight line model that forms the basis for figures 1.4 and 1.5, but this time also illustrates the least squares fit of the simplified model

$$y_i = \beta_0 + \epsilon_i,$$

which is nested within the original straight line model. Again, both the original and rotated versions of the model and data are shown. This time the fine continuous line shows the projection of the response data onto the space of the simpler model, while the fine dashed line shows the vector of the difference in fitted values between the two models. Notice how this vector is orthogonal both to the reduced model subspace and the full model residual vector.

The right panel of figure 1.6 illustrates that the rotation, using the transpose of the orthogonal factor, $\mathbf{Q}$, of the full model matrix, has also lined up the problem very conveniently for estimation of the reduced model. The fitted value vector for the reduced model now has only one non-zero component, which is the component of the rotated response data (•) relative to axis 1. The residual vector has gained the component that the fitted value vector has lost, so it has zero component relative to axis 1, while its other components are the positions of the rotated response data relative to axes 2 and 3.

So, much of the work required for estimating the simplified model has already been done, when estimating the full model. Note, however, that if our interest had been in comparing the full model to the model

$$y_i = \beta_1 x_i + \epsilon_i,$$

then it would have been necessary to reorder the columns of the full model matrix, in order to avoid extra work in this way.

1.5 Practical linear modelling

This section covers practical linear modelling, via an extended example: the analysis of data reported by Baker and Bellis (1993), which they used to support a theory of ‘sperm competition’ in humans. The basic idea is that it is evolutionarily advantageous for males to (subconciously) increase their sperm count in proportion to the opportunities that their mate may have had for infidelity. Such behaviour has been demonstrated in a wide variety of other animals, and using a sample of student and staff volunteers from Manchester University, Baker and Bellis set out to see if there is evidence for similar behaviour in humans. Two sets of data will be examined: `sperm.comp1` contains data on sperm count, time since last copulation and proportion of that time spent together, for single copulations, from 15 heterosexual couples; `sperm.comp2` contains data on median sperm count, over multiple copulations, for 24 heterosexual couples, along with the weight, height and age of the male and female of each couple, and the volume of one teste of the male. From these data, Baker and Bellis concluded that sperm count increases with the proportion of time,

`lm`	Estimates a linear model by least squares. Returns a fitted model object of class `lm` containing parameter estimates plus other auxiliary results for use by other functions.
`plot`	Produces model checking plots from a fitted model object.
`summary`	Produces summary information about a fitted model, including parameter estimates, associated standard errors, p-values, r^2, etc.
`anova`	Used for model comparison based on F-ratio testing.
`AIC`	Extract Akaike's information criterion for a model fit.
`residuals`	Extract an array of model residuals from a fitted model.
`fitted`	Extract an array of fitted values from a fitted model object.
`predict`	Obtain predicted values from a fitted model, either for new values of the predictor variables, or for the original values. Standard errors of the predictions can also be returned.

Table 1.1 *Some standard linear modelling functions. Strictly all of these functions except* `lm` *itself end* `.lm`*, but when calling them with an object of class* `lm` *this may be omitted.*

since last copulation, that a couple have spent apart, and that sperm count increases with female weight.

In general, practical linear modelling is concerned with finding an appropriate model to explain the relationship of a response (random) variable to some predictor variables. Typically, the first step is to decide on a linear model that can reasonably be supposed capable of describing the relationship, in terms of the predictors included and the functional form of their relationship to the response. In the interests of ensuring that the model is not too restrictive this 'full' model is often more complicated than is necessary, in that the most appropriate value for a number of its parameters may, in fact, be zero. Part of the modelling process is usually concerned with 'model selection': that is deciding which parameter values ought to be zero. At each stage of model selection it is necessary to estimate model parameters by least squares fitting, and it is equally important to check the model assumptions (particularly equal variance and independence) by examining diagnostic plots. Once a model has been selected and estimated, its parameter estimates can be interpreted, in part with the aid of confidence intervals for the parameters, and possibly with other follow-up analyses. In R these practical modelling tasks are facilitated by a large number of functions for linear modelling, some of which are listed in table 1.1.

1.5.1 Model fitting and model checking

The first thing to do with the sperm competition data is to have a look at them.

```
library(gamair)
pairs(sperm.comp1[,-1])
```

produces the plot shown in figure 1.7. The columns of the data frame are plotted against each other pairwise (with each pairing transposed between lower left and upper right of the plot); the first column has been excluded from the plot as it sim-

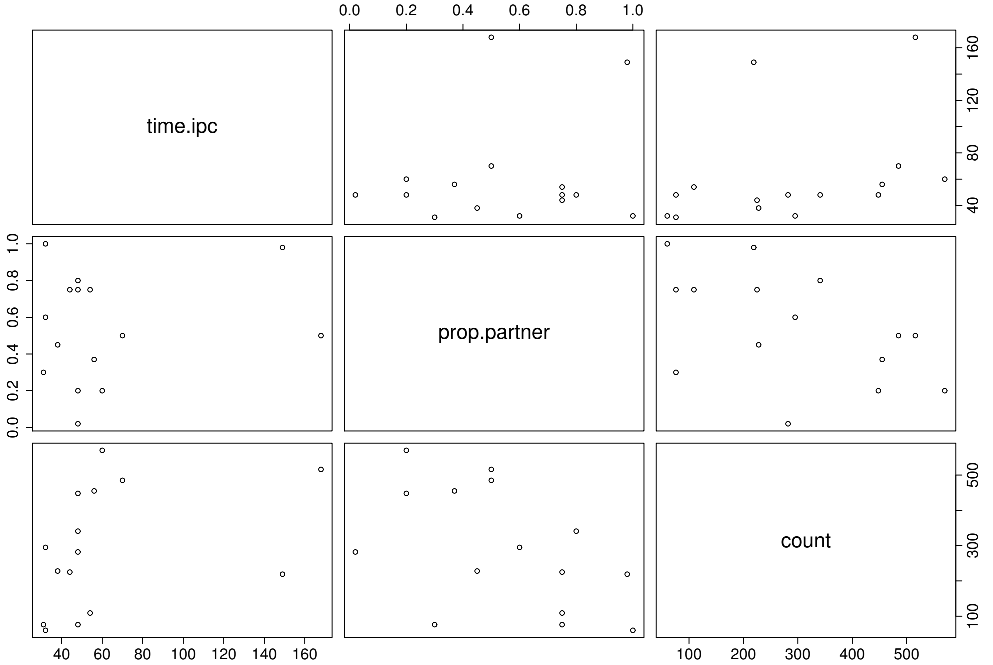

Figure 1.7 *Pairs plot of the sperm competition data from Baker and Bellis (1993). 'count' is sperm count (millions) from one copulation, 'time.ipc' is time (hours) since the previous copulation and 'prop.partner' is the proportion of the time since the previous copulation that the couple have spent together.*

ply contains subject identification labels. The clearest pattern seems to be of some decrease in sperm count as the proportion of time spent together increases.

Following Baker and Bellis, a reasonable initial model might be

$$y_i = \beta_0 + t_i\beta_1 + p_i\beta_2 + \epsilon_i, \tag{1.10}$$

where y_i is sperm count (`count`), t_i is the time since last inter-pair copulation (`time.ipc`) and p_i is the proportion of time, since last copulation, that the pair have spent together (`prop.partner`). As usual, the β_j are unknown parameters and the ϵ_i are i.i.d. $N(0, \sigma^2)$ random variables. Really this model defines the *class* of models thought to be appropriate: it is not immediately clear whether either of β_1 or β_2 are non-zero.

The following fits the model (1.10) and stores the results in an object called `sc.mod1`.

```
sc.mod1 <- lm(count ~ time.ipc + prop.partner, sperm.comp1)
```

The first argument to `lm` is a model formula, specifying the structure of the model to be fitted. In this case, the response (to the left of ~) is `count`, and this is to depend on variables `time.ipc` and `prop.partner`. By default, the model will include an intercept term, unless it is suppressed by a '`-1`' in the formula. The second argument to `lm` supplies a data frame within which the variables in the formula can be found.

The terms on the right hand side of the model formula specify how the model matrix, $\mathbf{X}$, is to be specified. In fact, in this example, the terms give the model matrix columns directly. It is possible to check the model matrix of a linear model:

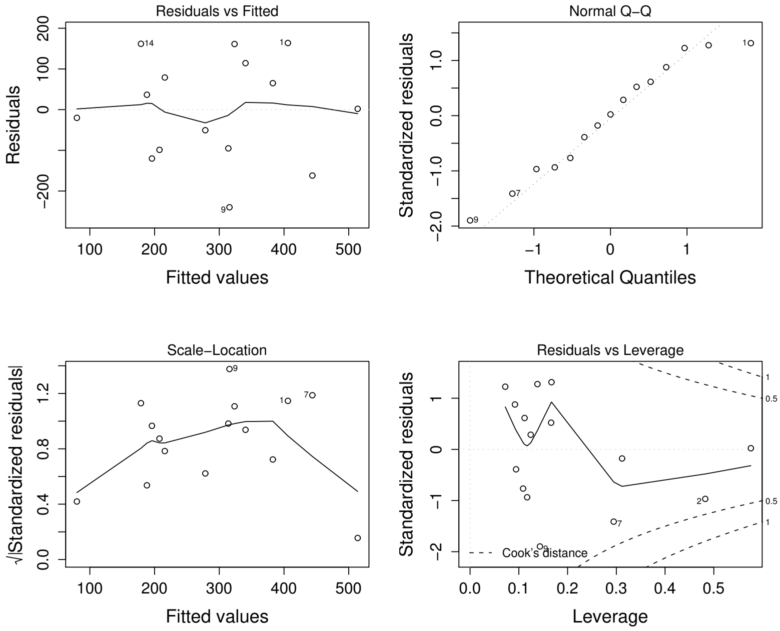

Figure 1.8 *Model checking plots for the linear model* `sc.mod1`.

```
> model.matrix(sc.mod1)
   (Intercept) time.ipc prop.partner
1            1       60         0.20
2            1      149         0.98
3            1       70         0.50
4            1      168         0.50
5            1       48         0.20
6            1       32         1.00
7            1       48         0.02
8            1       56         0.37
9            1       31         0.30
10           1       38         0.45
11           1       48         0.75
12           1       54         0.75
13           1       32         0.60
14           1       48         0.80
15           1       44         0.75
```

Having fitted the model, it is important to check the plausibility of the assumptions, graphically.

```
par(mfrow=c(2,2))   # split the graphics device into 4 panels
plot(sc.mod1)       # (uses plot.lm as sc.mod1 is class ‘lm’)
```

The resulting plots, shown in figure 1.8, require some explanation. In two of the plots the residuals have been scaled, by dividing them by their estimated standard deviation (see section 1.3.7). If the model assumptions are met, then this standardization should result in residuals that look like $N(0,1)$ random deviates.

- The upper left plot shows the model residuals, $\hat{\epsilon}_i$, against the model fitted values,

$\hat{\mu}_i$, where $\hat{\boldsymbol{\mu}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\epsilon}} = \mathbf{y} - \hat{\boldsymbol{\mu}}$. The residuals should be evenly scattered above and below zero (the distribution of fitted values is not of interest). A trend in the mean of the residuals would violate the assumption of independent response variables, and usually results from an erroneous model structure: e.g., assuming a linear relationship with a predictor, when a quadratic is required, or omitting an important predictor variable. A trend in the variability of the residuals suggests that the variance of the response is related to its mean, violating the constant variance assumption: transformation of the response, or use of a GLM, may help in such cases. The plot shown does not indicate any problem.

- The lower left plot is a scale-location plot. The raw residuals are standardized by dividing by their estimated standard deviation, $\hat{\sigma}\sqrt{1 - A_{ii}}$ (**A** is the influence matrix). The square root of the absolute value of each standardized residual is then plotted against the equivalent fitted value. It can be easier to judge the constant variance assumption from such a plot, and the square root transformation reduces the skew in the distribution, which would otherwise be likely to occur. Again, the plot shown gives no reason to doubt the constant variance assumption.
- The upper right panel is a normal QQ (quantile-quantile) plot. The standardized residuals are sorted and plotted against the quantiles of a standard normal distribution. If the residuals are normally distributed then the resulting plot should look like a straight line relationship, perturbed by some correlated random scatter. The current plot fits this description, so the normality assumption seems plausible.
- The lower right panel plots the standardized residuals against the *leverage* of each datum. The leverage is simply A_{ii}, which measures the *potential* for the i^{th} datum to influence the overall model fit. A large residual combined with high leverage implies that the corresponding datum has a substantial influence on the overall fit. A quantitative summary of how much influence each datum actually has is provided by its *Cook's distance*. If $\hat{\mu}_i^{[k]}$ is the i^{th} fitted value when the k^{th} datum is omitted from the fit, then Cook's distance is

$$d_k = \frac{1}{(p+1)\hat{\sigma}^2} \sum_{i=1}^{n} (\hat{\mu}_i^{[k]} - \hat{\mu}_i)^2, \tag{1.11}$$

  where p and n are the numbers of parameters and data, respectively. A large value of d_k indicates a point that has a substantial influence on the model results. If the Cook's distance values indicate that model estimates may be very sensitive to just a few data, then it usually prudent to repeat any analysis without the offending points, in order to check the robustness of the modelling conclusions. d_k can be shown to be a function of leverage and standardized residual, so contours of Cook's distance are shown on the plot, from which the Cook's distance for any datum can be read. In this case none of the points look wildly out of line.

The QQ-plot shows the reference line that a 'perfect' set of residuals would follow, while the other plots have a simple smooth overlaid, in order to guide the eye when looking for patterns in the residuals (these can be very useful for large datasets). By default the 'most extreme' three points in each plot are labelled with their row labels from the original data frame, so that the corresponding data can readily be checked. The 9^{th} datum is flagged in all four plots in figure 1.8. It should be checked:

```
> sperm.comp1[9,]
  subject time.ipc prop.partner count
9       P       31          0.3    76
```

This subject has quite a low count, but not the lowest in the frame. Examination of the plot of `count` against `prop.partner` indicates that the point adds substantially to the uncertainty surrounding the relationship, but it is hard to see a good reason to remove it, particularly since, if anything, it is obscuring the relationship, rather than exaggerating it.

Since the assumptions of model (1.10) appear reasonable, we can proceed to examine the fitted model object. Typing the name of an object in R causes the default print method for the object to be invoked (`print.lm` in this case).

```
> sc.mod1

Call:
lm(formula= count ~ time.ipc + prop.partner, data=sperm.comp1)

Coefficients:
 (Intercept)      time.ipc  prop.partner
     357.418         1.942      -339.560
```

The intercept parameter (β_0) is estimated to be 357.4. Notionally, this would be the count expected if `time.ipc` and `prop.partner` were zero, but the value is biologically implausible if interpreted in this way. Given that the smallest observed `time.ipc` was 31 hours we cannot really expect to predict the count near zero. The remaining two parameter estimates are $\hat{\beta}_1$ and $\hat{\beta}_2$, and are labelled by the name of the variable to which they relate. In both cases they give the expected increase in count for a unit increase in their respective predictor variable. Note the important point that the absolute values of the parameter estimates are only interpretable *relative* to the variable which they multiply. For example, we are not entitled to conclude that the effect of `prop.partner` is much greater than that of `time.ipc`, on the basis of the relative magnitudes of the respective parameters: they are measured in completely different units.

One point to consider is whether `prop.partner` is the most appropriate predictor variable. Perhaps the total time spent together (in hours) would be a better predictor.

```
sc.mod2 <- lm(count ~ time.ipc + I(prop.partner*time.ipc),
              sperm.comp1)
```

would fit such a model. The term `I(prop.partner*time.ipc)` indicates that, rather than use proportion of time together as a predictor, total time should be used. The `I()` function is used to 'protect' the product `prop.partner*time.ipc` within the model formula. This is necessary because symbols like `+` and `*` have special meanings within model formulae (see section 1.7): by protecting terms using `I()`, the usual arithmetic meanings are restored. Examination of diagnostic plots for `sc.mod2` shows that two points have much greater influence on the fit than the others, so for the purposes of this section it seems sensible to stick with the biologists' preferred model structure and use prop.partner.

1.5.2 *Model* summary

The summary[**] function provides more information about the fitted model.

```
> summary(sc.mod1)

Call:
lm(formula= count ~ time.ipc + prop.partner, data=sperm.comp1)

Residuals:
     Min       1Q   Median       3Q      Max
-239.740  -96.772    2.171   96.837  163.997

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   357.4184    88.0822   4.058  0.00159 **
time.ipc        1.9416     0.9067   2.141  0.05346 .
prop.partner -339.5602   126.2535  -2.690  0.01969 *
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 136.6 on 12 degrees of freedom
Multiple R-Squared: 0.4573,     Adjusted R-squared: 0.3669
F-statistic: 5.056 on 2 and 12 DF,  p-value: 0.02554
```

The explanations of the parts of this output are as follows:

Call simply reminds you of the call that generated the object being summarized.

Residuals gives a five figure summary of the residuals: this should indicate any gross departure from normality. For example, very skewed residuals might lead to very different magnitudes for Q1 and Q2, or to the median residual being far from 0 (the mean residual is always zero if the model includes an intercept: see exercise 6).

Coefficients gives a table relating to the estimated parameters of the model. The first two columns are the least squares estimates ($\hat{\beta}_j$) and the estimated standard errors associated with those estimates ($\hat{\sigma}_{\hat{\beta}_j}$), respectively. The standard error calculations follow sections 1.3.2 and 1.3.3. The third column gives the parameter estimates divided by their estimated standard errors: $T_j \equiv \hat{\beta}_j / \hat{\sigma}_{\hat{\beta}_j}$, which is a standardized measure of how far each parameter estimate is from zero. It was shown in section 1.3.3 (p. 13) that under $\text{H}_0 : \beta_j = 0$,

$$T_j \sim t_{n-p}, \tag{1.12}$$

[**] Note that calling the summary function with a fitted linear model object, x, actually results in the following: summary looks at the class of x, finds that it is "lm" and passes it to summary.lm; summary.lm calculates a number of interesting quantities from x which it returns in a list, y, of class lm.summary; unless y is assigned to an object, R prints it, using the print method print.lm.summary.

where n is the number of data and p the number of model parameters estimated. i.e., if the null hypothesis is true, then the observed T_j should be consistent with having been drawn from a t_{n-p} distribution. The final `Pr(>|t|)` column provides the measure of that consistency, namely the probability that the magnitude of a t_{n-p} random variable would be at least as large as the observed T_j. This quantity is known as the **p-value** of the test of $\text{H}_0 : \beta_j = 0$. A large p-value indicates that the data are consistent with the hypothesis, in that the observed T_j is quite a probable value for a t_{n-p} deviate, so that there is no reason to doubt the hypothesis underpinning (1.12). Conversely a small p-value suggests that the hypothesis is wrong, since the observed T_j is a rather improbable observation from the t_{n-p} distribution implied by $\beta_j = 0$. Various arbitrary **significance levels** are often used as the boundary p-values for deciding whether to accept or reject hypotheses. Some are listed at the foot of the table, and the p-values are flagged according to which, if any, they fall below.

`Residual standard error` gives $\hat{\sigma}$ where $\hat{\sigma}^2 = \sum \hat{\epsilon}_i^2/(n-p)$ (see section 1.3.3). $n - p$ is the 'residual degrees of freedom'.

`Multiple R-squared` is an estimate of the proportion of the variance in the data explained by the regression:

$$r^2 = 1 - \frac{\sum \hat{\epsilon}_i^2/n}{\sum (y_i - \bar{y})^2/n}$$

where $\bar{y}$ is the mean of the y_i. The fraction in this expression is basically an estimate of the proportion variance not explained by the regression.

`Adjusted R-squared`. The problem with r^2 is that it always increases when a new predictor variable is added to the model, no matter how useless that variable is for prediction. Part of the reason for this is that the variance estimates used to calculate r^2 are biased in a way that tends to inflate r^2. If unbiased estimators are used we get the adjusted r^2

$$r^2_{\text{adj}} = 1 - \frac{\sum \hat{\epsilon}_i^2/(n-p)}{\sum (y_i - \bar{y})^2/(n-1)}.$$

A high value of r^2_{adj} indicates that the model is doing well at explaining the variability in the response variable.

`F-statistic`. The final line, giving an F-statistic and p-value, is testing the null hypothesis that the data were generated from a model with only an intercept term, against the alternative that the fitted model generated the data. This line is really about asking if the whole model is of any use. The theory of such tests is covered in section 1.3.5.

Note that if the model formula contains '`-1`' to suppress the intercept, then `summary.lm` has the feature that r^2 is computed with the mean of the response data replaced by zero. This avoids r^2 being negative, but generally causes the r^2 to increase massively (and meaninglessly) if you drop the intercept from a model. The same feature applies to the F-ratio statistic — instead of comparing the fitted model to the model with just an intercept, it is compared to the model in which the mean is zero. Personally, when the model contains no intercept, I always re-compute the r^2 values using the observed mean of the data in place of R's default zero.

The summary of `sc.mod1` suggests that there is evidence that the model is better than one including just a constant (p-value = 0.02554). There is quite clear evidence that `prop.partner` is important in predicting sperm count (p-value = 0.019), but less evidence that `time.ipc` matters (p-value = 0.053). Indeed, using the conventional significance level of 0.05, we might be tempted to conclude that `time.ipc` does not affect count at all. Finally note that the model leaves most of the variability in count unexplained, since r^2_{adj} is only 37%.

1.5.3 Model selection

From the model summary it appears that `time.ipc` may not be necessary: the associated p-value of 0.053 does not provide strong evidence that the true value of β_1 is non-zero. By the 'true value' is meant the value of the parameter in the model imagined to have actually generated the data; or equivalently, the value of the parameter applying to the whole population of couples from which, at least conceptually, our particular sample has been randomly drawn. The question then arises of whether a simpler model, without any dependence on `time.ipc`, might be appropriate. This is a question of *model selection*. Usually it is a good idea to avoid overcomplicated models, dependent on irrelevant predictor variables, for reasons of interpretability and efficiency. Interpretations about causality will be made more difficult if a model contains spurious predictors, but estimates using such a model will also be less precise, as more parameters than necessary have been estimated from the finite amount of uncertain data available.

Several approaches to model selection are based on hypothesis tests about model terms, and can be thought of as attempting to find the simplest model consistent with a set of data, where consistency is judged relative to some threshold p-value. For the sperm competition model the p-value for `time.ipc` is greater than 0.05, so this predictor might be a candidate for dropping.

```
> sc.mod3 <- lm(count ~ prop.partner, sperm.comp1)
> summary(sc.mod3)
(edited)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    451.50      86.23   5.236 0.000161 ***
prop.partner  -292.23     140.40  -2.081 0.057727 .
---

Residual standard error: 154.3 on 13 degrees of freedom
Multiple R-Squared:  0.25,      Adjusted R-squared: 0.1923
F-statistic: 4.332 on 1 and 13 DF,  p-value: 0.05773
```

These results provide a good example of why it is dangerous to apply automatic model selection procedures unthinkingly. In this case dropping `time.ipc` has made the estimate of the parameter multiplying `prop.partner` less precise: indeed this term also has a p-value greater than 0.05 according to this new fit. Furthermore, the new model has a much reduced r^2, while the model's overall p-value does not give strong evidence that it is better than a model containing only an intercept. The only sensible choice here is to revert to `sc.mod1`. The statistical evidence indicates that it is better than the intercept only model, and dropping its possibly 'non-significant' term has led to a much worse model.

Hypothesis testing is not the only approach to model selection. One alternative is to try and find the model that gets as close as possible to the true model, rather than to find the simplest model consistent with data. In this case we can attempt to find the model which does the best job of predicting the $\mathbb{E}(y_i)$. Selecting models in order to minimize Akaike's Information Criterion (AIC) is one way of trying to do this (see section 1.8.6). In R, the `AIC` function can be used to calculate the AIC statistic for different models.

```
> sc.mod4 <- lm(count ~ 1, sperm.comp1) # null model
> AIC(sc.mod1,sc.mod3,sc.mod4)
        df      AIC
sc.mod1  4 194.7346
sc.mod3  3 197.5889
sc.mod4  2 199.9031
```

This alternative model selection approach also suggests that the model with both `time.ipc` and `prop.partner` is best.

So, on the basis of `sperm.comp1`, there seems to be reasonable evidence that sperm count increases with `time.ipc` but decreases with `prop.partner`: exactly as Baker and Bellis concluded.

1.5.4 Another model selection example

The second data set from Baker and Bellis (1993) is `sperm.comp2`. This gives median sperm count for 24 couples, along with ages (years), heights (cm) and weights (kg) for the male and female of each couple and volume (cm^3) of one teste for the male of the couple (`m.vol`). There are quite a number of missing values for the predictors, particularly for `m.vol`, but, for the 15 couples for which there is an `m.vol` measurement, the other predictors are also available. The number of copulations over which the median count has been taken varies widely from couple to couple. Ideally one should probably allow within couple and between couple components to the random variability component of the data to allow for this, but this will not be done here. Following Baker and Bellis it seems reasonable to start from a model including linear effects of all predictors, i.e.,

$$\begin{aligned}\texttt{count}_i = \beta_0 + \beta_1\texttt{f.age}_i + \beta_2\texttt{f.weight}_i + \beta_3\texttt{f.height}_i + \beta_4\texttt{m.age}_i \\ + \beta_5\texttt{m.weight}_i + \beta_6\texttt{m.height}_i + \beta_7\texttt{m.vol} + \epsilon_i\end{aligned}$$

The following estimates and summarizes the model, and plots diagnostics.

```
> sc2.mod1 <- lm(count ~ f.age + f.height + f.weight + m.age +
+                  m.height + m.weight + m.vol, sperm.comp2)
> plot(sc2.mod1)
> summary(sc2.mod1)
```

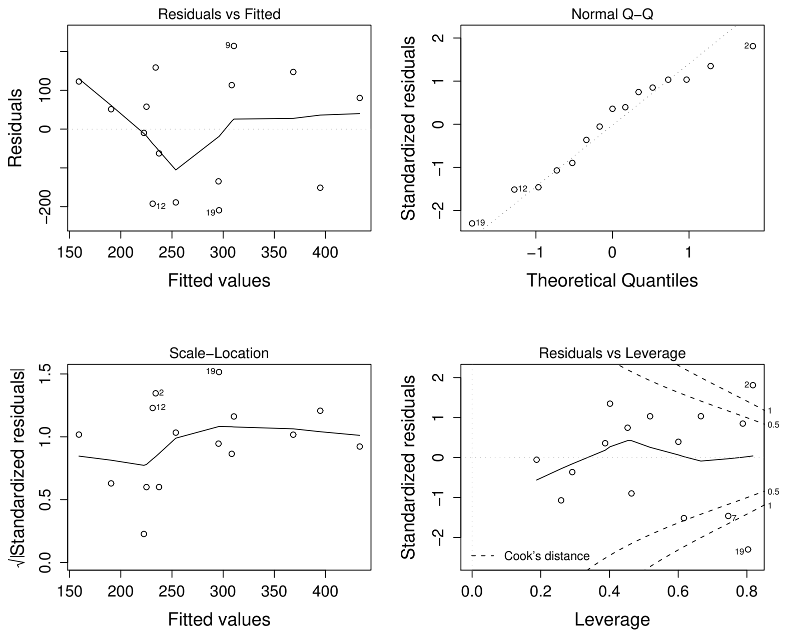

Figure 1.9 *Model checking plots for the* sc2.mod1 *model.*

```
[edited]
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1098.518   1997.984  -0.550    0.600
f.age          10.798     22.755   0.475    0.650
f.height       -4.639     10.910  -0.425    0.683
f.weight       19.716     35.709   0.552    0.598
m.age          -1.722     10.219  -0.168    0.871
m.height        6.009     10.378   0.579    0.581
m.weight       -4.619     12.655  -0.365    0.726
m.vol           5.035     17.652   0.285    0.784

Residual standard error: 205.1 on 7 degrees of freedom
Multiple R-Squared: 0.2192,     Adjusted R-squared: -0.5616
F-statistic: 0.2807 on 7 and 7 DF,  p-value: 0.9422
```

The resulting figure 1.9 looks reasonable, but datum 19 appears to produce the most extreme point on all 4 plots. Checking row 19 of the data frame shows that the male of this couple is rather heavy (particularly for his height), and has a large m.vol measurement, but a count right near the bottom of the distribution (actually down at the level that might be expected to cause fertility problems, if this is typical). Clearly, whatever we conclude from these data will need to be double-checked without this observation. Notice, from the summary, how poorly this model does at explaining the count variability: the adjusted r^2 is actually negative, an indication that we have a large number of irrelevant predictors in the model.

There are only 15 data from which to estimate the 8 parameters of the full model: it would be better to come up with something more parsimonious. One possibility would be to use the `step` function in R to perform model selection automatically. `step` takes a fitted model and repeatedly drops the term that leads to the largest decrease in AIC. By default it also tries adding back in each single term already dropped, to see if that leads to a reduction in AIC. See `?step` for more details. In this case using `AIC` suggests a rather complicated model with only `m.weight` dropped. It seems sensible to switch to hypothesis testing based model selection, and ask whether there is really good evidence that all these terms are necessary. One approach is to perform ‘backwards model selection’, by repeatedly removing the *single* term with highest p-value, above some threshold (e.g., 0.05), and then refitting the resulting reduced model, until all terms have significant p-values. For example the first step in this process would remove `m.age`:

```
> sc2.mod2 <- lm(count ~ f.age + f.height + f.weight +
+                m.height + m.weight + m.vol,sperm.comp2)
> summary(sc2.mod2)
[edited]
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1054.770   1856.843  -0.568    0.586
f.age           8.847     18.359   0.482    0.643
f.height       -5.119      9.871  -0.519    0.618
f.weight       20.259     33.334   0.608    0.560
m.height        6.033      9.727   0.620    0.552
m.weight       -4.473     11.834  -0.378    0.715
m.vol           4.506     16.281   0.277    0.789

Residual standard error: 192.3 on 8 degrees of freedom
Multiple R-Squared: 0.216,      Adjusted R-squared: -0.372
F-statistic: 0.3674 on 6 and 8 DF,  p-value: 0.8805
```

Relative to `sc2.mod1`, the reduced model has different estimates for each of the remaining parameter, as well as smaller standard error estimates for each parameter, and (consequently) different p-values. This is part of the reason for only dropping one term at a time: when we drop one term from a model, it is quite possible for some remaining terms to have their p-values massively reduced. For example, if two terms are highly correlated it is possible for both to be significant individually, but both to have very high p-values if present together. This occurs because the terms are to some extent interchangeable predictors: the information provided by one is much the same as the information provided by the other, so that one must be present in the model but both are not needed. If we were to drop several terms from a model at once, we might miss such effects.

Proceeding with backwards selection, we would drop `m.vol` next. This allows rather more of the couples to be used in the analysis. Continuing in the same way leads to the dropping of `m.weight`, `f.height`, `m.height` and finally `f.age` before arriving at a final model which includes only `f.weight`.

```
> sc2.mod7 <- lm(count ~ f.weight,sperm.comp2)
> summary(sc2.mod7)
[edited]
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1002.281    489.352  -2.048   0.0539 .
f.weight       22.397      8.629   2.595   0.0173 *

Residual standard error: 147.3 on 20 degrees of freedom
Multiple R-Squared: 0.252,      Adjusted R-squared: 0.2146
F-statistic: 6.736 on 1 and 20 DF,  p-value: 0.01730
```

This model does appear to be better than a model containing only a constant, according both to a hypothesis test at the 5% level and AIC.

Apparently then, only female weight influences sperm count. This concurs with the conclusion of Baker and Bellis (1993), who interpreted the findings to suggest that males might 'invest' more in females with a higher reproductive potential. However, in the light of the residual plots we need to re-analyze the data without observation 19, before having too much confidence in the conclusions. This is easily done:

```
> sc <- sperm.comp2[-19,]
> sc3.mod1 <- lm(count ~ f.age + f.height + f.weight + m.age +
+               m.height + m.weight + m.vol, sc)
> summary(sc3.mod1)
[edited]
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1687.406   1251.338   1.348   0.2262
f.age         55.248     15.991   3.455   0.0136 *
f.height      21.381      8.419   2.540   0.0441 *
f.weight     -88.992     31.737  -2.804   0.0310 *
m.age        -17.210      6.555  -2.626   0.0393 *
m.height     -11.321      6.869  -1.648   0.1504
m.weight       6.885      7.287   0.945   0.3812
m.vol         48.996     13.938   3.515   0.0126 *
--- [edited]
```

`m.vol` now has the lowest p-value. Repeating the whole backwards selection process, every term now drops out except for `m.vol`, leading to the much less interesting conclusion that the data only really supply evidence that size of testes influences sperm count. Given the rather tedious plausibility of this conclusion, it probably makes sense to prefer it to the conclusion based on the full data set.

A follow-up

Given the biological conclusions from the analysis of `sperm.comp2`, it would make sense to revisit the analysis of `sperm.comp1`. Baker and Bellis do not report `m.vol` values for these data, but the same couples feature in both datasets and are identified by label, so the required values can be obtained:

```
sperm.comp1$m.vol <-
  sperm.comp2$m.vol[sperm.comp2$pair %in% sperm.comp1$subject]
```

Repeating the same sort of backwards selection we end up selecting a 1 term model:

```
> sc1.mod1 <- lm(count ~ m.vol, sperm.comp1)
> summary(sc1.mod1)

Call:
lm(formula = count ~ m.vol, data = sperm.comp1)

Residuals:
     Min       1Q   Median       3Q      Max
-187.236  -55.028   -8.606   75.928  156.257

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -58.694    121.619  -0.483   0.6465
m.vol         23.247      7.117   3.266   0.0171 *
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 120.8 on 6 degrees of freedom
Multiple R-Squared:  0.64,      Adjusted R-squared:  0.58
F-statistic: 10.67 on 1 and 6 DF,  p-value: 0.01711
```

Although based on only 8 couples, this must call into question the original analysis, which concluded that time since last copulation and proportion of time spent together controlled sperm count. There is at least a suggestion that the explanation for sperm count variability may be rather more prosaic than the explanation suggested by sperm competition theory.

1.5.5 Confidence intervals

Exactly as in section 1.1.3 the results from section 1.3.3 can be used to obtain confidence intervals for the parameters. In general, for a p parameter model of n data, a $(1-2\alpha)100\%$ confidence interval for the $j^{\rm th}$ parameter is

$$\hat{\beta}_j \pm t_{n-p}(\alpha)\hat{\sigma}_{\hat{\beta}_j},$$

where $t_{n-p}(\alpha)$ is the value below which a t_{n-p} random variable would lie with probability α.

As an example of its use, the following calculates a 95% confidence interval for the mean increase in `count` per cm^3 increase in `m.vol`.

```
> sc.c <- summary(sc1.mod1)$coefficients
> sc.c   # check info extracted from summary
             Estimate Std. Error    t value   Pr(>|t|)
(Intercept) -58.69444 121.619433 -0.4826075 0.64647664
m.vol        23.24653   7.117239  3.2662284 0.01711481
> sc.c[2,1]+qt(c(.025,.975),6)*sc.c[2,2]
[1]  5.831271 40.661784   # 95% CI
```

1.5.6 Prediction

It is possible to predict the expected value of the response at new values of the predictor variables using the `predict` function. For example: what are the model predicted counts for `m.vol` values of 10, 15, 20 and 25? The following obtains the answer along with associated standard errors (see section 1.3.7).

```
> df <- data.frame(m.vol=c(10,15,20,25))
> predict(sc1.mod1,df,se=TRUE)
$fit
       1        2        3        4
173.7708 290.0035 406.2361 522.4688

$se.fit
       1        2        3        4
60.39178 43.29247 51.32314 76.98471
```

The first line creates a data frame containing the predictor variable values at which predictions are required. The second line calls `predict` with the fitted model object and new data from which to predict. `se=TRUE` tells the function to return standard errors along with the predictions.

1.5.7 Co-linearity, confounding and causation

Consider the case in which a predictor variable x is the variable which really controls response variable y, but at the same time, it is also highly correlated with a variable z, which plays no role at all in setting the level of y. The situation is represented by the following simulation:

```
set.seed(1); n <- 100; x <- runif(n)
z <- x + rnorm(n)*.05
y <- 2 + 3 * x + rnorm(n)
```

Despite the fact that z played no role in generating y, the correlation between x and z leads to the following

```
> summary(lm(y~z))

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.1628     0.2245   9.632 7.60e-16 ***
z             2.7342     0.3836   7.127 1.75e-10 ***
```

i.e., there seems to be strong evidence that z is predictive of y. Clearly there must be something wrong with interpreting this as implying that z is actually controlling y, since in this case we know that it isn't. Now the problem here is that both z and y are being controlled by a *confounding* variable that is absent from the model. Since the data were simulated we are in the happy position of knowing what the confounder is. What happens if we include it?

```
> summary(lm(y ~ x + z))

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

```
Residuals:
    Min      1Q  Median      3Q     Max
-2.8311 -0.7273 -0.0537  0.6338  2.3359

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.1305     0.2319   9.188  7.6e-15 ***
x             1.3750     2.3368   0.588    0.558
z             1.4193     2.2674   0.626    0.533
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.056 on 97 degrees of freedom
Multiple R-squared: 0.3437,     Adjusted R-squared: 0.3302
F-statistic:  25.4 on 2 and 97 DF,  p-value: 1.345e-09
```

In this case the linear model fitting has 'shared out' the real dependence on x between x and z (look at the estimated slope parameters), and has given x, the true predictor, a higher p-value than z, suggesting that if anything we should drop x! Notice also that both slope parameter estimates have very high standard errors. Because they are so highly correlated it is not possible to distinguish which is actually driving y, and this is reflected in the high standard errors for both parameters. Notice also the importance of dropping one of x and z here: if we don't then our inferences will be very imprecise.

In this simulated example we know what the truth is, but in most real data analyses we only have the data to tell us. We can not usually know whether there are hidden confounder variables or not. Similarly, even if we do have a complete set of possible predictors, we do not usually have special knowledge of which ones really control the response: we only have the statistical analysis to go on.

These issues are ubiquitous when trying to interpret the results of analysing observational data. For example, a majority of men find a low ratio of waist to hip size attractive in women. A recent study (Brosnan and Walker, 2009) found that men who did not share the usual preference were more likely to have autistic children. Now, if you view this association as causative, then you might be tempted to postulate all sorts of theories about the psychology of autistic brains and the inheritance of autism. However, it is known that being exposed to higher levels of testosterone in the womb is also associated with autism, and women with higher levels of testosterone are less likely to have a low waist to hip ratio. Since these women are more likely to select men who are attracted to them, the original association may have a rather obvious explanation.

So, correlation between known predictor variables, and with un-observed predictors, causes interpretational problems and tends to inflate variances. Does this mean that linear modelling is useless? No. Discovering *associations* can be a very useful part of the scientific process of uncovering *causation*, but it is often the case that we don't care about causation. If all I want to do is to be able to predict y, I don't much care whether I do so by using the variable that actually controlled y, or by using a variable that is a proxy for the controlling variable.

To really establish causation, it is usually necessary to do an experiment in which the putative causative variable is manipulated to see what effect it has on the response. In doing this, it is necessary to be very careful not to introduce correlations between the values of the manipulated variable and other variables (known or unknown) that might influence the response. In experiments carried out on individual units (e.g., patients in a clinical trial) the usual way that this is achieved is to allocate units randomly to the different values of the manipulated variable(s). This *randomization* removes any possibility that characteristics of the units that may effect the response can be correlated with the manipulated variable. Analysis of such randomized experiments often involves factor variables, and these will be covered next.

1.6 Practical modelling with factors

Most of the models covered so far have been for situations in which the predictor variables are variables that measure some quantity (*metric variables*), but there are many situations in which the predictor variables are more qualitative in nature, and serve to divide the responses into groups. Examples might be eye colour of subjects in a psychology experiment, which of three alternative hospitals were attended by patients in a drug trial, manufacturers of cars used in crash tests, etc. Variables like these, which serve to classify the units on which the response variable has been measured into distinct categories, are known as *factor variables*, or simply *factors*. The different categories of the factor are known as *levels* of the factor. For example, levels of the factor 'eye colour' might be 'blue', 'brown', 'grey' and 'green', so that we would refer to eye colour as a factor with four levels. Note that 'levels' is quite confusing terminology: there is not necessarily any natural ordering of the levels of a factor. Hence 'levels' of a factor might best be thought of as 'categories' of a factor, or 'groups' of a factor.

Factor variables are handled using dummy variables. Each factor variable can be replaced by as many dummy variables as there are levels of the factor — one for each level For each response datum, only one of these dummy variables will be non-zero: the dummy variable for the single level that applies to that response. Consider an example to see how this works in practice: 9 laboratory rats are fed too much, so that they divide into 3 groups of 3: 'fat', 'very fat' and 'enormous'. Their blood insulin levels are then measured 10 minutes after being fed a standard amount of sugar. The investigators are interested in the relationship between insulin levels and the factor 'rat size'. Hence a model could be set up in which the predictor variable is the factor 'rat size', with the three levels 'fat', 'very fat' and 'enormous'. Writing y_i for the i^{th} insulin level measurement, a suitable model might be:

$$\mathbb{E}(Y_i) \equiv \mu_i = \begin{cases} \beta_0 & \text{if rat is fat} \\ \beta_1 & \text{if rat is very fat} \\ \beta_2 & \text{if rat is enormous} \end{cases}$$

and this is easily written in linear model form, using a dummy predictor variable for each level of the factor:

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \\ \mu_8 \\ \mu_9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}.$$

A key difference between dummy variables and directly measured predictor variables is that the dummy variables, and parameters associated with a factor, are almost always treated as a group during model selection — it does not usually make sense for a subset of the dummy variables associated with a factor to be dropped or included on their own: either all are included or none. The F-ratio tests derived in section 1.3.5 are designed for hypothesis testing in this situation.

1.6.1 Identifiability

When modelling with factor variables, model 'identifiability' becomes an important issue. It is quite easy to set up models, involving factors, in which it is impossible to estimate the parameters uniquely, because an infinite set of alternative parameter vectors would give rise to exactly the same expected value vector. A simple example illustrates the problem. Consider again the fat rat example, but suppose that we wanted to formulate the model in terms of an overall mean insulin level, α, and deviations from that level, β_j, associated with each level of the factor. The model would be something like:

$$\mu_i = \alpha + \beta_j \text{ if rat } i \text{ is rat size level } j$$

(where j is 0, 1 or 2, corresponding to 'fat', 'very fat' or 'enormous'). The problem with this model is that there is not a one-to-one correspondence between the parameters and the fitted values, so that the parameters can not be uniquely estimated from the data. This is easy to see. Consider any particular set of α and $\boldsymbol{\beta}$ values, giving rise to a particular $\boldsymbol{\mu}$ value: any constant c could be added to α and simultaneously subtracted from each element of $\boldsymbol{\beta}$ without changing the value of $\boldsymbol{\mu}$. Hence there is an infinite set of parameters giving rise to each $\boldsymbol{\mu}$ value, and therefore the parameters can not be estimated uniquely. The model is not 'identifiable'.

This situation can be diagnosed directly from the model matrix. Written out in full, the example model is

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \\ \mu_8 \\ \mu_9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}.$$

But the columns of the model matrix are not independent, and this lack of full column rank means that the formulae for finding the least squares parameter estimates break down.[††] Identifiable models have model matrices of full column rank; unidentifiable models are column rank deficient.

The solution to the identifiability problem is to impose just enough linear constraints on the model parameters that the model becomes identifiable. For example, in the fat rat model we could impose the constraint that

$$\sum_{j=0}^{2} \beta_j = 0.$$

This would immediately remove the identifiability problem, but does require use of a linearly constrained least squares method (see sections 1.8.1 and 1.8.2). A simpler constraint is to set one of the unidentifiable parameters to zero, which requires only that the model is re-written without the zeroed parameter, rather than a modified fitting method. For example, in the fat rat case we could set α to zero, and recover the original identifiable model. This is perfectly legitimate, since the reduced model is capable of reproducing any expected values that the original model could produce.

In the one factor case, this discussion of identifiability may seem to be unnecessarily complicated, since it is so easy to write down the model directly in an identifiable form. However, when models involve more than one factor variable, the issue can not be avoided. For more on imposing constraints see sections 1.8.1 and 1.8.2.

1.6.2 Multiple factors

It is frequently the case that more than one factor variable should be included in a model, and this is straightforward to do. Continuing the fat rat example, it might be that the sex of the rats is also a factor in insulin production, and that

$$\mu_i = \alpha + \beta_j + \gamma_k \text{ if rat } i \text{ is rat size level } j \text{ and sex } k$$

[††] In terms of section 1.3.1, $\mathbf{R}$ will not be full rank, and will hence not be invertible; in terms of section 1.3.8, $\mathbf{X}^\mathsf{T}\mathbf{X}$ will not be invertible.

is an appropriate model, where k is 0 or 1 for male or female. Written out in full (assuming the rats are MMFFFMFMM) the model is

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \\ \mu_8 \\ \mu_9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta_0 \\ \beta_1 \\ \beta_2 \\ \gamma_0 \\ \gamma_1 \end{bmatrix}.$$

It is immediately obvious that the model matrix is of column rank 4, implying that two constraints are required to make the model identifiable. You can see the lack of column independence by noting that column 5 is column 1 minus column 6, while column 2 is column 1 minus columns 3 and 4. An obvious pair of constraints would be to set $\beta_0 = \gamma_0 = 0$, so that the full model is

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \\ \mu_8 \\ \mu_9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta_1 \\ \beta_2 \\ \gamma_1 \end{bmatrix}.$$

When you specify models involving factors in R, it will automatically impose identifiability constraints for you, and by default these constraints will be that the parameter for the 'first' level of each factor is zero. Note that 'first' is essentially arbitrary here — the order of levels of a factor is not important. However, if you need to change which level is 'first', in order to make parameters more interpretable, see the `relevel` function.

1.6.3 'Interactions' of factors

In the examples considered so far, the effect of factor variables has been purely additive, but it is possible that a response variable may react differently to the combination of two factors relative to what would be predicted by simply adding the effect of the two factors separately. For example, if examining patient blood cholesterol levels, we might consider the factors 'hypertensive' (yes/no) and 'diabetic' (yes/no). Being hypertensive or diabetic would be expected to raise cholesterol levels, but being both is likely to raise cholesterol levels much more than would be predicted from just adding up the apparent effects when only one condition is present. When the effect of two factor variables together differs from the sum of their separate effects, then they are said to *interact*, and an adequate model in such situations requires *interaction terms*. Put another way, if the effects of one factor change in response to another factor, then the factors are said to interact.

Let us continue with the fat rat example, but now suppose that how insulin level depends on size varies with sex. An appropriate model is then

$$\mu_i = \alpha + \beta_j + \gamma_k + \delta_{jk} \text{ if rat } i \text{ is rat size level } j \text{ and sex } k,$$

where the δ_{jk} terms are the parameters for the interaction of rat size and sex. Writing this model out in full it is clear that it is spectacularly unidentifiable:

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \\ \mu_8 \\ \mu_9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta_0 \\ \beta_1 \\ \beta_2 \\ \gamma_0 \\ \gamma_1 \\ \delta_{00} \\ \delta_{01} \\ \delta_{10} \\ \delta_{11} \\ \delta_{20} \\ \delta_{21} \end{bmatrix}.$$

In fact, for this simple example, with rather few rats, we now have more parameters than data. Of course there are many ways of constraining this model to achieve identifiability. One possibility (the default in R) is to set $\beta_0 = \gamma_0 = \delta_{00} = \delta_{01} = \delta_{10} = \delta_{20} = 0$. The resulting model can still produce any fitted value vector that the full model can produce, but all the columns of its model matrix are independent, so that the model is identifiable:

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \\ \mu_8 \\ \mu_9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta_1 \\ \beta_2 \\ \gamma_1 \\ \delta_{11} \\ \delta_{21} \end{bmatrix}.$$

Of course, the more factor variables are present, the more interactions are possible, and the higher the order of the possible interactions: for example if three factors are present then each factor could interact with each factor, giving three possible ‘two-way’ interactions, while all the factors could also interact together, giving a three-way interaction (e.g., the way in which insulin level’s dependence on rat size is influenced by sex is itself influenced by blood group — perhaps with interactions beyond two-way, equations are clearer than words).

Notice one very convenient fact about interactions. The model matrix columns corresponding to the interactions of two or more factors consist of the element-wise product of all possible combinations of the model matrix columns for the main effects of the factors. This applies with or without identifiability constraints, which makes the imposition of identifiability constraints on interactions especially convenient. In fact this 'column product' way of defining interactions is so convenient that in most software it is used as the definition of an interaction between factors and metric variables and, less naturally, even between different metric variables.

1.6.4 Using factor variables in R

It is very easy to work with factor variables in R. All that is required is that you let R know that a particular variable is a factor variable. For example, suppose z is a variable declared as follows:

```
> z <- c(1,1,1,2,2,1,3,3,3,3,4)
> z
 [1] 1 1 1 2 2 1 3 3 3 3 4
```

and it is to be treated as a factor with four levels. The functions `as.factor` or `factor` will ensure that `z` is treated as a factor:

```
> z <- as.factor(z)
> z
 [1] 1 1 1 2 2 1 3 3 3 3 4
Levels: 1 2 3 4
```

Notice that, when a factor variable is printed, a list of its levels is also printed — this provides an easy way to tell if a variable is a factor variable. Note also that the digits of `z` are treated purely as labels: the numbers 1 to 4 could have been any labels. For example, `x` could be a factor with 3 levels, declared as follows:

```
> x <- c("A","A","C","C","C","er","er")
> x
[1] "A"  "A"  "C"  "C"  "C"  "er" "er"
> x <- factor(x)
> x
[1] A  A  C  C  C  er er
Levels: A C er
```

Once a variable is declared as a factor variable, then R can process it automatically within model formulae, by replacing it with the appropriate number of binary dummy variables (and imposing any necessary identifiability constraints).

As an example of the use of factor variables, consider the `PlantGrowth` data frame supplied with R. These are data on the growth of plants under control conditions and two different treatment conditions. The factor `group` has three levels `cntrl`, `trt1` and `trt2`, and it is believed that the growth of the plants depends on this factor. First check that `group` is already a factor variable:

```
> PlantGrowth$group
 [1] ctrl ctrl ctrl ctrl ctrl ctrl ctrl ctrl ctrl ctrl trt1
```

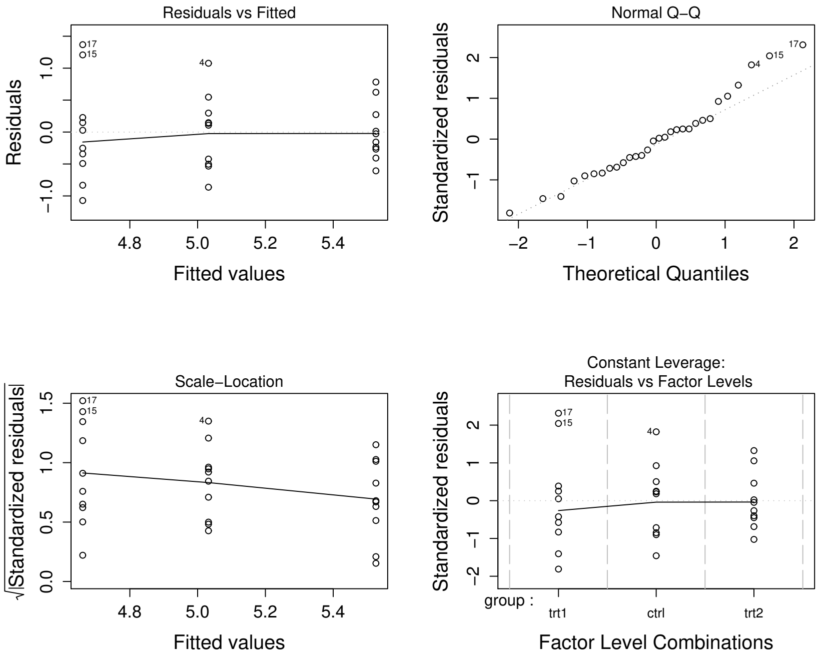

Figure 1.10 *Model checking plots for the plant growth example. Note that, since the leverages are all the same in this case, the lower right plot is now simplified.*

```
[12] trt1 trt1 trt1 trt1 trt1 trt1 trt1 trt1 trt1 trt2 trt2
[23] trt2 trt2 trt2 trt2 trt2 trt2 trt2 trt2
Levels: ctrl trt1 trt2
```

. . . since a list of levels is reported, it must be. If it had not been then

```
PlantGrowth$group <- as.factor(PlantGrowth$group)
```

would have converted it. The response variable for these data is `weight` of the plants at some set time after planting, and the aim is to investigate whether the `group` factor controls this, and if so to what extent.

```
> pgm.1 <- lm(weight ~ group, data=PlantGrowth)
> plot(pgm.1)
```

As usual, the first thing to do, after fitting a model, is to check the residual plots shown in figure 1.10. In this case there is some suggestion of decreasing variance with increasing mean, but the effect does not look very pronounced, so it is probably safe to proceed.

```
> summary(pgm.1)
[edited]
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.0320     0.1971  25.527   <2e-16 ***
grouptrt1    -0.3710     0.2788  -1.331   0.1944
grouptrt2     0.4940     0.2788   1.772   0.0877 .
---
[edited]
```

Notice how R reports an intercept parameter and parameters for the two treatment levels, but, in order to obtain an identifiable model, it has not included a parameter for the control level of the group factor. So the estimated overall mean weight (in the population that these plants represent, given control conditions) is 5.032, while treatment 1 is estimated to lower this weight by 0.37, and treatment 2 to increase it by 0.49. However, neither parameter individually appears to be significantly different from zero. (Don't forget that `model.matrix(pgm.1)` can be used to check up on the form of the model matrix used in the fit.)

Model selection based on the summary output is very difficult for models containing factors. It makes little sense to drop the dummy variable for just one level of a factor from a model, and if we did, what would we then do about the model identifiability constraints? Usually, it is only of interest to test whether the whole factor variable should be in the model or not, and this amounts to testing whether all its associated parameters are simultaneously zero or not. The F-ratio tests derived in section 1.3.5 are designed for just this purpose, and in R, such tests can be invoked using the `anova` function. For example, we would compare `pgm.1` to a model in which the expected response is given by a single parameter that does not depend on `group`:

```
> pgm.0 <- lm(weight ~ 1, data=PlantGrowth)
> anova(pgm.0,pgm.1)
Analysis of Variance Table

Model 1: weight ~ 1
Model 2: weight ~ group
  Res.Df     RSS Df Sum of Sq      F  Pr(>F)
1     29 14.2584
2     27 10.4921  2    3.7663 4.8461 0.01591 *
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
```

The output gives the F-ratio statistic used to test the null hypothesis that the simpler model is correct, against the alternative that `pgm.1` is correct. If the null is true then the probability of obtaining so large an F value is only 0.016, suggesting that the null hypothesis is not very plausible.

So we see that the data provide evidence for an effect of the `group` factor variable on `weight`, which appeared marginal or absent when we examined p-values for the individual model parameters. This comes about because we have, in effect, considered all the parameters associated with the factor simultaneously, thereby obtaining a more powerful test than any of the single parameter tests could be. In the light of this analysis, the most promising treatment to look at is clearly treatment 2, since this gives the largest and 'most significant' effect, and it is a positive effect.

Another useful function, particularly with models containing many terms, is `drop1` which drops each term singly from the full model, and then computes the p-value or AICs for comparing each reduced-by-one model and the full model.

1.7 General linear model specification in R

Having seen several examples of the use of `lm`, it is worth briefly reviewing the way in which models are specified using model formulae in R. The main components of a formula are all present in the following example

```
y ~ a*b + x:z + offset(v) -1
```

Note the following:

- `~` separates the response variable, on the left, from the 'linear predictor', on the right. So in the example `y` is the response and `a`, `b`, `x`, `z` and `v` are the predictors.
- + indicates that the response depends on what is to the left of + *and* what is to the right of it. Hence within formulae '+' should be thought of as 'and' rather than 'the sum of'.
- `:` indicates that the response depends on the *interaction* of the variables to the left and right of `:`. Interactions are obtained by forming the element-wise products of all model matrix columns corresponding to the two terms that are interacting and appending the resulting columns to the model matrix.
- `*` means that the response depends on whatever is to the left of `*` and whatever is to the right of it *and* their interaction, i.e. `a*b` is just a shorter way of writing `a + b + a:b`.
- `offset(v)` is equivalent to including `v` as a model matrix column with the corresponding parameter fixed at 1.
- `-1` means that the default intercept term should not be included in the model. Note that, for models involving factor variables, this often has no real impact on the model structure, but simply reduces the number of identifiability constraints by one, while changing the interpretation of some parameters.

Because of the way that some symbols change their usual meaning in model formulae, it is necessary to take special measures if the usual meaning is to be restored to arithmetic operations within a formula. This is accomplished by using the identity function `I()` which simply evaluates its argument and returns it. For example, if we wanted to fit the model:

$$y_i = \beta_0 + \beta_1(x_i + z_i) + \beta_2 v_i + \epsilon_i$$

then we could do so using the model formula

```
y ~ I(x+z) + v
```

Note that there is no need to 'protect' arithmetic operations within arguments to other functions in this way. For example

$$y_i = \beta_0 + \beta_1 \log(x_i + z_i) + \beta_2 v_i + \epsilon_i$$

would be fitted correctly by

```
y ~ log(x+z) + v
```

1.8 Further linear modelling theory

This section covers linear models with constraints on the parameters (including a discussion of contrasts), the connection with maximum likelihood estimation, AIC and Mallows' C_p, linear models for non-independent data with non-constant variance and non-linear models.

1.8.1 *Constraints I: General linear constraints*

It is often necessary, particularly when working with factor variables, to impose constraints on the linear model parameters of the general form

$$\mathbf{C}\boldsymbol{\beta} = \mathbf{0},$$

where $\mathbf{C}$ is an $m \times p$ matrix of known coefficients. The general approach to imposing such constraints is to rewrite the model in terms of $p - m$ unconstrained parameters. There are a number of ways of doing this, but a simple general approach uses the QR decomposition of $\mathbf{C}^\mathsf{T}$. Let

$$\mathbf{C}^\mathsf{T} = \mathbf{U}\begin{bmatrix} \mathbf{P} \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{U}$ is a $p \times p$ orthogonal matrix and $\mathbf{P}$ is an $m \times m$ upper triangular matrix. $\mathbf{U}$ can be partitioned $\mathbf{U} \equiv (\mathbf{D} : \mathbf{Z})$ where $\mathbf{Z}$ is a $p \times (p - m)$ matrix.

It turns out that

$$\boldsymbol{\beta} = \mathbf{Z}\boldsymbol{\beta}_z$$

will meet the constraints for any value of the $p - m$ dimensional vector $\boldsymbol{\beta}_z$. This is easy to see:

$$\mathbf{C}\boldsymbol{\beta} = \begin{bmatrix} \mathbf{P}^\mathsf{T} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{D}^\mathsf{T} \\ \mathbf{Z}^\mathsf{T} \end{bmatrix}\mathbf{Z}\boldsymbol{\beta}_z = \begin{bmatrix} \mathbf{P}^\mathsf{T} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{p-m} \end{bmatrix}\boldsymbol{\beta}_z = \mathbf{0}.$$

Hence to minimize $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$ w.r.t. $\boldsymbol{\beta}$, subject to $\mathbf{C}\boldsymbol{\beta} = \mathbf{0}$, the following algorithm can be used.

1. Find the QR decomposition of $\mathbf{C}^\mathsf{T}$: the final $p - m$ columns of the orthogonal factor define $\mathbf{Z}$.
2. Minimize the unconstrained sum of squares $\|\mathbf{y} - \mathbf{X}\mathbf{Z}\boldsymbol{\beta}_z\|^2$ w.r.t. $\boldsymbol{\beta}_z$ to obtain $\hat{\boldsymbol{\beta}}_z$.
3. $\hat{\boldsymbol{\beta}} = \mathbf{Z}\hat{\boldsymbol{\beta}}_z$.

Note that, in practice, it is computationally inefficient to form $\mathbf{Z}$ explicitly, when we only need to be able to post-multiply $\mathbf{X}$ by it, and pre-multiply $\boldsymbol{\beta}_z$ by it. The reason for this is that $\mathbf{Z}$ is completely defined as the product of m 'Householder rotations': simple matrix operations that can be applied very rapidly to any vector or matrix (see section 5.4.1, p. 211). R includes routines for multiplication by the orthogonal factor of a QR decomposition which makes use of these efficiencies. See B.5 and B.6 for further details on QR decomposition.

1.8.2 Constraints II: 'Contrasts' and factor variables

There is another approach to imposing identifiability constraints on models involving factor variables. To explain it, it is worth revisiting the basic identifiability problem with a simple example. Consider the model

$$y_i = \mu + \alpha_j + \epsilon_i \text{ if } y_i \text{ is from group } j$$

and suppose that there are three groups with two observations in each. The model matrix in this case is

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix},$$

but this is not of full column rank: any of its columns could be made up from a linear combination of the other 3. In geometric terms the model space is of dimension 3 and not 4, and this means that we can not estimate all 4 model parameters, but at most 3 parameters. Numerically this problem would manifest itself in the rank deficiency of $\mathbf{R}$ in equation (1.6), which implies that $\mathbf{R}^{-1}$ does not exist.

One approach to this issue is to remove one of the model matrix columns, implicitly treating the corresponding parameter as zero. This gives the model matrix full column rank, so that the remaining parameters are estimable, but since the model space is unaltered, we have not fundamentally changed the model. It can still predict every set of fitted values that the original model could predict. By default, R drops the model matrix column corresponding to the first level of each factor, in order to impose identifiability on models with factors. For the simple example, this results in

$$\mathbf{X}' = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

To generalize this approach, it helps to write out this deletion in a rather general way. For example, if we re-write the original model matrix in partitioned form, $\mathbf{X} = [\mathbf{1} : \mathbf{X}_1]$, where $\mathbf{1}$ is a column of 1s, then

$$\mathbf{X}' = [\mathbf{1} : \mathbf{X}_1\mathbf{C}_1] \text{ where } \mathbf{C}_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Now all that $\mathbf{C}_1$ has done is to replace the 3 columns of $\mathbf{X}_1$ by a 2 column linear combination of them, which cannot be combined to give $\mathbf{1}$. On reflection, *any* matrix which did these two things would have served as well as the particular $\mathbf{C}_1$ actually given, in terms of making the model identifiable. This observation leads to the idea of choosing alternative matrix elements for $\mathbf{C}_1$, in order to enhance the interpretability of the parameters actually estimated, for some models.

Matrices like $\mathbf{C}_1$ are known as *contrast matrices*[†] and several different types are available in R (see `?contrasts` and `options("contrasts")`). The degree of interpretability of some of the contrasts is open to debate. For the $\mathbf{C}_1$ given, suppose that parameters μ, α_2' and α_3' are estimated: μ would now be interpretable as the mean for group 1, while α_2' and α_3' would be the differences between the means for groups 2 and 3, and the mean for group 1.

With all contrast matrices, the contrast matrix itself can be used to transform back from the parameters actually estimated, to (non-unique) estimates in the original redundant parameterization, e.g.

$$\hat{\boldsymbol{\alpha}} = \mathbf{C}_1\hat{\boldsymbol{\alpha}}'.$$

The contrast approach generalizes easily to models with multiple factors and their interactions. Again a simple example makes things clear. Consider the model:

$$y_i = \mu + \alpha_j + \beta_k + \gamma_{jk} + \epsilon_i \text{ if } y_i \text{ is from groups } j \text{ and } k.$$

The unconstrained model matrix for this might have the form $\mathbf{X} = [\mathbf{1} : \mathbf{X}_\alpha : \mathbf{X}_\beta : \mathbf{X}_\alpha \odot \mathbf{X}_\beta]$, where $\mathbf{X}_\alpha$ and $\mathbf{X}_\beta$ are the columns generated by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and $\mathbf{X}_\alpha \odot \mathbf{X}_\beta$ are the columns corresponding to $\boldsymbol{\gamma}$, which are in fact generated by element-wise multiplication of all possible pairings of the column from $\mathbf{X}_\alpha$ and $\mathbf{X}_\beta$.

To make this model identifiable we would choose contrast matrices $\mathbf{C}_\alpha$ and $\mathbf{C}_\beta$ for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively, and then form the following identifiable model matrix:

$$\mathbf{X}' = [\mathbf{1} : \mathbf{X}_\alpha\mathbf{C}_\alpha : \mathbf{X}_\beta\mathbf{C}_\beta : (\mathbf{X}_\alpha\mathbf{C}_\alpha) \odot (\mathbf{X}_\beta\mathbf{C}_\beta)]$$

(in this case $\boldsymbol{\gamma} = \mathbf{C}_\alpha \otimes \mathbf{C}_\beta\gamma'$, where $\otimes$ is the Kronecker product, see B.4). Some further information can be found in Venables and Ripley (2002).

1.8.3 Likelihood

The theory developed in section 1.3 is quite sufficient to justify the approach of estimating linear models by least squares, but although it doesn't directly strengthen the case, it is worth understanding the link between the method of least squares and the method of maximum likelihood for normally distributed data.

The basic idea of likelihood is that, given some parameter values, a statistical model allows us to write down the probability, or probability density, of any set of data, and in particular of the set actually observed. In some sense, parameter values which cause the model to suggest that the observed data are probable, are more 'likely' than parameter values that suggest that what was observed was improbable. In fact it seems reasonable to use as estimates of the parameters those values which

[†] Actually, in much of the literature on linear models, the given $\mathbf{C}_1$ would not be called a 'contrast', as its columns are not orthogonal to $\mathbf{1}$, but R makes no terminological distinction.

maximize the probability of the data according to the model: these are the 'maximum likelihood estimates' of the parameters.

Appendix A covers the properties of maximum likelihood estimation in greater detail. For the moment consider the likelihood for the parameters of a linear model. According to the model, the joint p.d.f. of the response data is

$$f_{\boldsymbol{\beta},\sigma^2}(\mathbf{y}) = (2\pi\sigma^2)^{-n/2} e^{-\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2/(2\sigma^2)}.$$

Now suppose that the observed data are plugged into this expression and it is treated as a function of its parameters $\boldsymbol{\beta}$ and σ^2. This is known as the likelihood function

$$L(\boldsymbol{\beta}, \sigma^2) = (2\pi\sigma^2)^{-n/2} e^{-\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2/(2\sigma^2)},$$

and it is important to note that $\mathbf{y}$ is now representing the actual observed data, rather than arguments of a p.d.f. To estimate the parameters, L should be maximized w.r.t. them, and it is immediately apparent that the value of $\boldsymbol{\beta}$ maximizing L will be the value minimizing

$$\mathcal{S} = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

(irrespective of the value of σ^2 or its MLE).

In itself this connection is of little interest, but it suggests how to estimate linear models when data do not meet the constant variance assumption, and may not even be independent. To this end consider the linear model

$$\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}, \quad \mathbf{y} \sim N(\boldsymbol{\mu}, \mathbf{V}\sigma^2),$$

where $\mathbf{V}$ is any positive definite[‡] matrix. In this case the likelihood for $\boldsymbol{\beta}$ is

$$L(\boldsymbol{\beta}) = \frac{1}{\sqrt{(2\pi\sigma^2)^n|\mathbf{V}|}} e^{-(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\mathsf{T}\mathbf{V}^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})/(2\sigma^2)}$$

and if $\mathbf{V}$ is known then maximum likelihood estimation of $\boldsymbol{\beta}$ is achieved by minimizing

$$\mathcal{S}_v = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\mathsf{T}\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}).$$

In fact the likelihood approach can be taken further, since if $\mathbf{V}$ depends on unknown parameters then these too can be estimated by maximum likelihood estimation: this is what is done in linear mixed modelling, which is discussed in Chapter 2.

1.8.4 *Non-independent data with variable variance*

In the previous section a modified least squares criterion was developed for linear model parameter estimation, when data follow a general multivariate normal distribution, with unknown mean and covariance matrix known to within a constant of

[‡] A matrix $\mathbf{A}$ is positive definite iff $\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} > 0$ for any non-zero vector $\mathbf{x}$. Equivalent to this condition is the condition that all the eigenvalues of $\mathbf{A}$ must be strictly positive. Practical tests for positive definiteness are examination of the eigenvalues of the matrix or (more efficiently) seeing if a Cholesky decomposition of the matrix is possible (this must be performed without pivoting, otherwise only positive semi-definiteness is tested): see B.7.

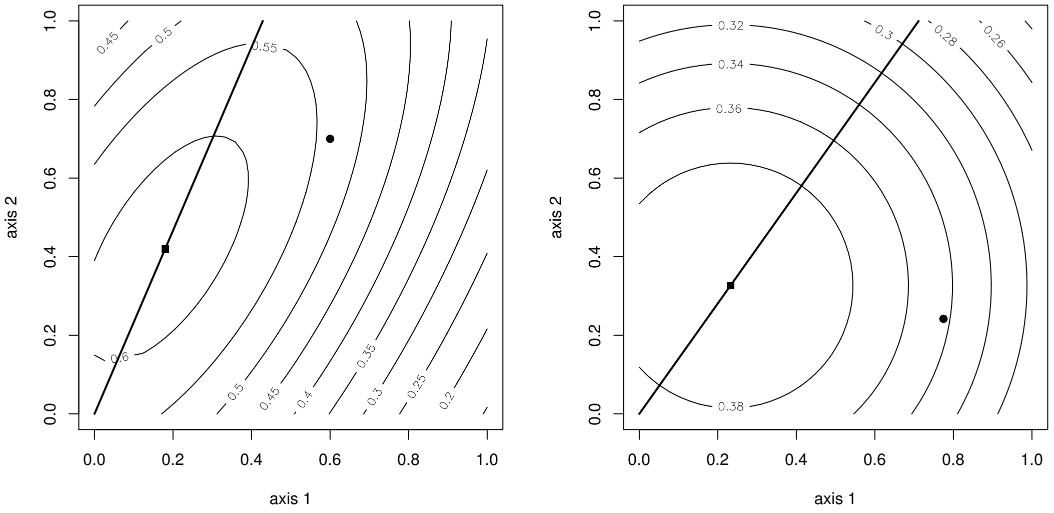

Figure 1.11 *Fitting a linear model to data that are not independent with constant variance. The example is a straight line through the origin, fit to two data. In both panels the response data values give the co-ordinates of the point ● and the straight line is the vector defining the 'model subspace' (the line along which the fitted values could lie). It is assumed that the data arise from the multivariate normal distribution contoured in the left panel. The right panel shows the fitting problem after transformation in the manner described in section 1.8.4: the response data and model subspace have been transformed and this implies a transformation of the distribution of the response. The transformed data are an observation from a radially symmetric multivariate normal density.*

proportionality. It turns out to be possible to transform this fitting problem so that it has exactly the form of the fitting problem for independent data with constant variance. Having done this, all inference about the model parameters can proceed using the methods of section 1.3.

First let $\mathbf{L}$ be any matrix such that $\mathbf{L}^\mathsf{T}\mathbf{L} = \mathbf{V}$: a Cholesky decomposition is usually the easiest way to obtain this (see section B.7). Then

$$\begin{aligned} \mathcal{S}_v &= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\mathsf{T}\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\ &= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\mathsf{T}\mathbf{L}^{-1}\mathbf{L}^{-\mathsf{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\ &= \|\mathbf{L}^{-\mathsf{T}}\mathbf{y} - \mathbf{L}^{-\mathsf{T}}\mathbf{X}\boldsymbol{\beta}\|^2, \end{aligned}$$

and this least squares objective can be minimized by the methods already met (i.e., form a QR decomposition of $\mathbf{L}^{-\mathsf{T}}\mathbf{X}$, etc.)

It is not just the model fitting that carries over from the theory of section 1.3. Since $\mathbf{L}^{-\mathsf{T}}\mathbf{y}$ is a linear transformation of a normal random vector, it must have a multivariate normal distribution, and it is easily seen that $\mathbb{E}(\mathbf{L}^{-\mathsf{T}}\mathbf{y}) = \mathbf{L}^{-\mathsf{T}}\mathbf{X}\boldsymbol{\beta}$ while the covariance matrix of $\mathbf{L}^{-\mathsf{T}}\mathbf{y}$ is

$$\mathbf{V}_{\mathbf{L}^{-\mathsf{T}}\mathbf{y}} = \mathbf{L}^{-\mathsf{T}}\mathbf{V}\mathbf{L}^{-1}\sigma^2 = \mathbf{L}^{-\mathsf{T}}\mathbf{L}^\mathsf{T}\mathbf{L}\mathbf{L}^{-1}\sigma^2 = \mathbf{I}\sigma^2,$$

i.e., $\mathbf{L}^{-\mathsf{T}}\mathbf{y} \sim N(\mathbf{L}^{-\mathsf{T}}\mathbf{X}\boldsymbol{\beta}, \mathbf{I}\sigma^2)$. In other words, the transformation has resulted in a new linear modelling problem, in which the response data are independent normal random variables with constant variance: exactly the situation which allows all the results from section 1.3 to be used for inference about $\boldsymbol{\beta}$.

Figure 1.11 illustrates the geometry of the transformation for a simple linear model,

$$y_i = \beta x_i + \epsilon_i, \quad \epsilon_i \sim N(\mathbf{0}, \mathbf{V}),$$

where $\mathbf{y}^\mathsf{T} = (.6, .7)$, $\mathbf{x}^\mathsf{T} = (.3, .7)$, $\beta = .6$ and $\mathbf{V} = \begin{bmatrix} .6 & .5 \\ .5 & 1.1 \end{bmatrix}$. The left panel shows the geometry of the original fitting problem, while the right panel shows the geometry of the transformed fitting problem.

1.8.5 *Simple AR correlation models*

Considerable computational simplification occurs when the correlation is given by a simple auto-regression (AR) model. For example residuals with a simple AR1 structure are generated as $\epsilon_i = \epsilon_{i-1} + \rho e_i$ where the e_i are independent $N(0, \sigma^2)$. Such a model is sometimes appropriate when the data are observed at regular time intervals. The covariance matrix for such a model is

$$\mathbf{V} = \sigma^2 \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 & . \\ \rho & 1 & \rho & \rho^2 & . \\ \rho^2 & \rho & 1 & \rho & . \\ . & . & . & . & . \end{bmatrix}$$

and the inverse of the Cholesky factor of $\mathbf{V}$ is

$$\mathbf{L}^{-1} = \begin{bmatrix} 1 & -\rho/\sqrt{1-\rho^2} & 0 & 0 & . \\ 0 & 1/\sqrt{1-\rho^2} & -\rho/\sqrt{1-\rho^2} & 0 & . \\ 0 & 0 & 1/\sqrt{1-\rho^2} & -\rho/\sqrt{1-\rho^2} & . \\ . & . & . & . & . \end{bmatrix} \sigma^{-1}.$$

Now because $\mathbf{L}^{-1}$ has only 2 non-zero diagonals the computation of $\mathbf{L}^{-\mathsf{T}}\mathbf{y}$ amounts to a simple differencing operation on the elements of $\mathbf{y}$ computable with $O(n)$ operations, while $\mathbf{L}^{-\mathsf{T}}\mathbf{X}$ simply differences the columns of $\mathbf{X}$ at $O(np)$ cost. Hence computing with such an AR1 model is very efficient. If the likelihood of the model is required note that $|\mathbf{V}|^{-1/2} = |\mathbf{L}^{-1}|$: the latter is just the product of the leading diagonal elements of $\mathbf{L}^{-1}$, which again has $O(n)$ cost. This computational convenience carries over to higher order AR models, which also have banded inverse Cholesky factors.

1.8.6 *AIC and Mallows' statistic*

Consider again the problem of selecting between nested models, which is usually equivalent to deciding whether some terms from the model should simply be set to zero. The most natural way to do this would be to select the model with the smallest residual sum of squares or largest likelihood, but this is always the largest model under consideration. Model selection methods based on F-ratio or t-tests address this problem by selecting the simplest model consistent with the data, where consistency is judged using some significance level (threshold p-value) that has to be chosen more or less arbitrarily.

In this section an alternative approach is developed, based on the idea of trying to select the model that should do the best job of predicting $\boldsymbol{\mu} \equiv \mathbb{E}(\mathbf{y})$, rather than the model that gets as close as possible to $\mathbf{y}$. Specifically, we select the model that minimizes an estimate of

$$K = \|\boldsymbol{\mu} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2.$$

We have

$$\begin{aligned} K = \|\boldsymbol{\mu} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2 &= \|\boldsymbol{\mu} - \mathbf{A}\mathbf{y}\|^2 = \|\mathbf{y} - \mathbf{A}\mathbf{y} - \boldsymbol{\epsilon}\|^2 \\ &= \|\mathbf{y} - \mathbf{A}\mathbf{y}\|^2 + \boldsymbol{\epsilon}^\mathsf{T}\boldsymbol{\epsilon} - 2\boldsymbol{\epsilon}^\mathsf{T}(\mathbf{y} - \mathbf{A}\mathbf{y}) \\ &= \|\mathbf{y} - \mathbf{A}\mathbf{y}\|^2 + \boldsymbol{\epsilon}^\mathsf{T}\boldsymbol{\epsilon} - 2\boldsymbol{\epsilon}^\mathsf{T}(\boldsymbol{\mu} + \boldsymbol{\epsilon}) + 2\boldsymbol{\epsilon}^\mathsf{T}\mathbf{A}(\boldsymbol{\mu} + \boldsymbol{\epsilon}) \\ &= \|\mathbf{y} - \mathbf{A}\mathbf{y}\|^2 - \boldsymbol{\epsilon}^\mathsf{T}\boldsymbol{\epsilon} - 2\boldsymbol{\epsilon}^\mathsf{T}\boldsymbol{\mu} + 2\boldsymbol{\epsilon}^\mathsf{T}\mathbf{A}\boldsymbol{\mu} + 2\boldsymbol{\epsilon}^\mathsf{T}\mathbf{A}\boldsymbol{\epsilon}. \end{aligned}$$

Now, $\mathbb{E}(\boldsymbol{\epsilon}^\mathsf{T}\boldsymbol{\epsilon}) = \mathbb{E}(\sum_i \epsilon_i^2) = n\sigma^2$, $\mathbb{E}(\boldsymbol{\epsilon}^\mathsf{T}\boldsymbol{\mu}) = \mathbb{E}(\boldsymbol{\epsilon}^\mathsf{T})\boldsymbol{\mu} = 0$ and $\mathbb{E}(\boldsymbol{\epsilon}^\mathsf{T}\mathbf{A}\boldsymbol{\mu}) = \mathbb{E}(\boldsymbol{\epsilon}^\mathsf{T})\mathbf{A}\boldsymbol{\mu} = 0$. Finally, using the fact that a scalar is its own trace:

$$\mathbb{E}[\text{tr}\left(\boldsymbol{\epsilon}^\mathsf{T}\mathbf{A}\boldsymbol{\epsilon}\right)] = \mathbb{E}[\text{tr}\left(\mathbf{A}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\mathsf{T}\right)] = \text{tr}\left(\mathbf{A}\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\mathsf{T}]\right) = \text{tr}\left(\mathbf{A}\mathbf{I}\right)\sigma^2 = \text{tr}\left(\mathbf{A}\right)\sigma^2.$$

Hence, replacing terms involving $\boldsymbol{\epsilon}$ by their expectation we can estimate K using

$$\hat{K} = \|\mathbf{y} - \mathbf{A}\mathbf{y}\|^2 - n\sigma^2 + 2\text{tr}\left(\mathbf{A}\right)\sigma^2. \tag{1.13}$$

Using the fact that $\text{tr}\left(\mathbf{A}\right) = p$ and dividing through by σ^2 we get 'Mallows' C_p' (Mallows, 1973)

$$C_p = \|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2/\sigma^2 + 2p - n.$$

Model selection by C_p minimization works well if σ^2 is known, but for most models σ^2 must be estimated, and using the estimate derived from the model fit has the unfortunate consequence that C_p ceases to depend on which model has been fitted, in any meaningful way. To avoid this, σ^2 is usually fixed at the estimate given by the fit of the largest candidate model (unless it really is known, of course).

An alternative that handles σ more satisfactorily is to replace K by the likelihood based equivalent

$$K' = \|\boldsymbol{\mu} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2/\sigma^2 + n\log(2\pi\sigma).$$

Writing $l(\boldsymbol{\beta}, \sigma)$ as the model log likelihood, and re-using the calculation that led to Mallows' C_p we have the estimate

$$\hat{K}' = -2l(\hat{\boldsymbol{\beta}}, \sigma^2) + 2p - n,$$

which is minimized by whichever model minimises Akaike's information criteria (Akaike, 1973)

$$\text{AIC} = -2l(\hat{\boldsymbol{\beta}}, \sigma^2) + 2p.$$

If we use the MLE , $\hat{\sigma}^2 = \|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2/n$, then

$$\text{AIC} = -2l(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2) + 2(p+1). \tag{1.14}$$

See section A.7 for justification in the context of general likelihoods.

Notice how the above derivation does not involve any assumption that directly implies that the models to be compared must be nested: however, a more detailed examination of the comparison of models by AIC would suggest that the comparison will be more reliable for nested models, since in that case some of the neglected terms in the approximation of K cancel. Notice also that if the true model is not in the set of models under consideration then the properties of (1.14) will be less ideal. Indeed in this case (1.14) would itself tend to favour more complex models, since it would be impossible to match $\boldsymbol{\mu}$ exactly: as sample size increases and estimates become more precise, this tendency starts to overcome the negative effects of overfitting and leads to more complex models being selected. The tendency for AIC to favour more complex models with increasing sample size is often seen in practice: presumably because the true model is rarely in the set of candidate models.

1.8.7 The wrong model

The derivation that led to Mallows' C_p implies that $\|\mathbf{y} - \hat{\boldsymbol{\mu}}\|^2 \to \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|^2 - 2\text{tr}(\mathbf{A})\sigma^2 + n\sigma^2$ as the sample size $n \to \infty$, and this applies whether or not $\mathbb{E}(\hat{\boldsymbol{\mu}}) = \boldsymbol{\mu}$. Hence in the large sample limit $\hat{\boldsymbol{\mu}}$ will minimize $\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|^2$, whether or not $\hat{\boldsymbol{\mu}}$ is biased. This result is useful when we consider the properties of regression splines.

1.8.8 Non-linear least squares

Some non-linear models can be estimated by iterative approximation by a linear model. At each iterate the fitted approximating linear model suggests improved parameter estimates, and at convergence the parameter estimates are least squares estimates. In addition, the approximating linear model at convergence can be used as the basis for approximate inference about the parameters.

Formally, consider fitting the model:

$$\mathbb{E}(\mathbf{y}) \equiv \boldsymbol{\mu} = \mathbf{f}(\boldsymbol{\beta})$$

to response data $\mathbf{y}$, when $\mathbf{f}$ is a non-linear vector valued function of $\boldsymbol{\beta}$. An obvious fitting objective is:

$$\mathcal{S} = \sum_{i=1}^{n} \{y_i - f_i(\boldsymbol{\beta})\}^2 = \|\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})\|^2$$

and, if the functions, f_i, are sufficiently well behaved, then this non-linear least squares problem can be solved by iterative linear least squares. To do this, we start from a guess at the best fit parameters, $\hat{\boldsymbol{\beta}}^{[k]}$, and then take a first order Taylor expansion of f_i around $\hat{\boldsymbol{\beta}}^{[k]}$ so that the fitting objective becomes

$$\mathcal{S} \approx \mathcal{S}^{[k]} = \|\mathbf{y} - \mathbf{f}(\hat{\boldsymbol{\beta}}^{[k]}) + \mathbf{J}^{[k]}\hat{\boldsymbol{\beta}}^{[k]} - \mathbf{J}^{[k]}\boldsymbol{\beta}\|^2$$

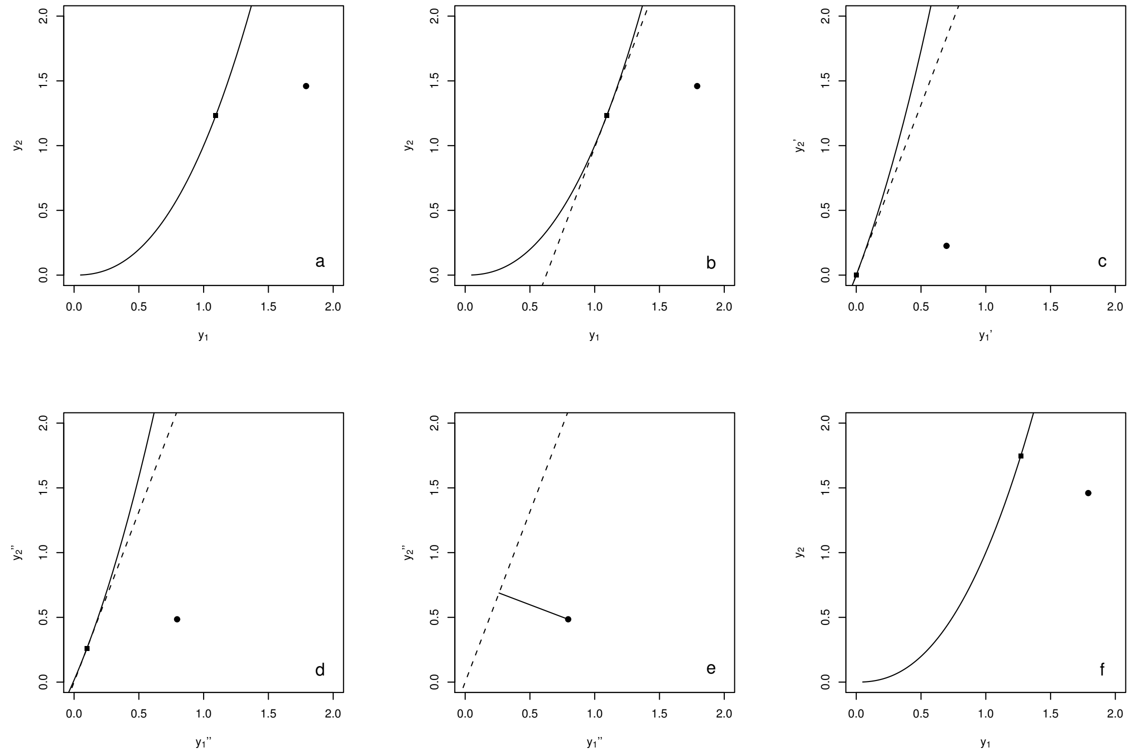

Figure 1.12 *Geometry of a single iteration of the Gauss-Newton approach to non-linear model fitting. The example is fitting the model* $\mathbb{E}(y_i) = \exp(\beta x_i)$ *to* x_i*,* y_i *data (*$i = 1, 2$*).* (a) *plots* y_2 *against* y_1 *(●) with the curve illustrating the possible values for the expected values of the response variable: as the value of* β *changes* $\mathbb{E}(\mathbf{y})$ *follows this curve – the 'model manifold'. An initial guess at the parameter gives the fitted values plotted as ■.* (b) *The tangent space to the model manifold is found and illustrated by the dashed line.* (c) *The model, tangent and data are linearly translated so that the current estimate of the fitted values is at the origin.* (d) *The model, tangent and data are linearly translated so that* $\mathbf{J}\hat{\boldsymbol{\beta}}^{[k]}$ *gives the location of the current estimate of the fitted values.* (e) $\hat{\boldsymbol{\beta}}^{[k+1]}$ *is estimated by finding the closest point in the tangent space to the response data, by least squares.* (f) *shows the original model and data, with the next estimate of the fitted values, obtained using* $\hat{\boldsymbol{\beta}}^{[k+1]}$*. The steps can now be repeated until the estimates converge.*

where $\mathbf{J}^{[k]}$ is the 'Jacobian' matrix such that $J_{ij}^{[k]} = \partial f_i / \partial \beta_j$ (derivatives evaluated at $\hat{\boldsymbol{\beta}}^{[k]}$, of course). Defining a vector of *pseudodata*,

$$\mathbf{z}^{[k]} = \mathbf{y} - \mathbf{f}(\hat{\boldsymbol{\beta}}^{[k]}) + \mathbf{J}^{[k]}\hat{\boldsymbol{\beta}}^{[k]},$$

the objective can be re-written

$$\mathcal{S}^{[k]} = \|\mathbf{z}^{[k]} - \mathbf{J}^{[k]}\boldsymbol{\beta}\|^2,$$

and, since this is a linear least squares problem, it can be minimized with respect to $\boldsymbol{\beta}$ to obtain an improved estimated parameter vector $\hat{\boldsymbol{\beta}}^{[k+1]}$. If $\mathbf{f}(\boldsymbol{\beta})$ is not too non-linear then this process can be iterated until the $\hat{\boldsymbol{\beta}}^{[k]}$ sequence converges, to the final least squares estimate $\hat{\boldsymbol{\beta}}$.

Figure 1.12 illustrates the geometrical interpretation of this method. Because the non-linear model is being approximated by a linear model, large parts of the theory of linear models carry over as approximations in the current context. For example, under the assumption of equal variance, σ^2, and independence of the response variable, the covariance matrix of the parameters is simply: $(\mathbf{J}^\mathsf{T}\mathbf{J})^{-1}\sigma^2$, where $\mathbf{J}$ is evaluated at convergence.

If $\mathbf{f}$ is sufficiently non-linear that convergence does not occur, then a simple 'step reduction' approach will stabilize the fitting. The vector $\Delta = \hat{\boldsymbol{\beta}}^{[k+1]} - \hat{\boldsymbol{\beta}}^{[k]}$ is treated as a trial step. If $\hat{\boldsymbol{\beta}}^{[k+1]}$ does not decrease $\mathcal{S}$, then trial steps $\hat{\boldsymbol{\beta}}^{[k]} + \alpha\Delta$ are taken, with ever decreasing α, until $\mathcal{S}$ does decrease (of course, $0 < \alpha < 1$). Geometrically, this is equivalent to performing an updating step by fitting to the average $\mathbf{y}\alpha + (1 - \alpha)\mathbf{f}(\boldsymbol{\beta}^{[k]})$, rather than original data $\mathbf{y}$: viewed in this way it is clear that for small enough α, each iteration must decrease $\mathcal{S}$, until a minimum is reached. It is usual to halve α each time that a step is unsuccessful, and to start each iteration with twice the α value which finally succeeded at the previous step (or $\alpha = 1$ if this is less). If it is *necessary* to set $\alpha < 1$ at the final step of the iteration then it is likely that inference based on the final approximating linear model will be somewhat unreliable.

1.8.9 Further reading

The literature on linear models is rather large, and there are many book length treatments. For a good introduction to linear models with R, see Faraway (2014). Other sources on the linear modelling functionality in R are Chambers (1993) and Venables and Ripley (2002). Numerical estimation of linear models is covered in Golub and Van Loan (2013, chapter 5). Dobson and Barnett (2008), McCullagh and Nelder (1989) and Davison (2003) also consider linear models as part of broader treatments. For R itself see R Core Team (2016).

1.9 Exercises

1. Four car journeys in London of length 1, 3, 4 and 5 kilometres took 0.1, 0.4, 0.5 and 0.6 hours, respectively. Find a least squares estimate of the mean journey speed.
2. Given n observations x_i, y_i, find the least squares estimate of β in the linear model: $y_i = \mu_i + \epsilon_i$, $\mu_i = \beta$.
3. Which, if any, of the following common linear model assumptions are required for $\hat{\boldsymbol{\beta}}$ to be unbiased: (i) The Y_i are independent, (ii) the Y_i all have the same variance, (iii) the Y_i are normally distributed?
4. Write out the following three models in the form $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, ensuring that all the parameters left in the written out model are identifiable. In all cases y is the response variable, ϵ the residual 'error' and other Greek letters indicate model parameters.
   (a) The 'balanced one-way ANOVA model':
$$y_{ij} = \alpha + \beta_i + \epsilon_{ij}$$

where $i = 1 \dots 3$ and $j = 1 \dots 2$.

(b) A model with two explanatory factor variables and only one observation per combination of factor variables:

$$y_{ij} = \alpha + \beta_i + \gamma_j + \epsilon_{ij}.$$

The first factor (β) has 3 levels and the second factor has 4 levels.

(c) A model with two explanatory variables: a factor variable and a continuous variable, x.

$$y_i = \alpha + \beta_j + \gamma x_i + \epsilon_i \quad \text{if obs. } i \text{ is from factor level } j.$$

Assume that $i = 1 \dots 6$, that the first two observations are for factor level 1 and the remaining 4 for factor level 2 and that the x_i's are 0.1, 0.4, 0.5, 0.3, 0.4 and 0.7.

5. Consider some data for deformation (in mm), y_i, of 3 different types of alloy, under different loads (in kg), x_i. When there is no load, there is no deformation, and the deformation is expected to vary linearly with load, in exactly the same way for all three alloys. However, as the load increases the three alloys deviate from this ideal linear behaviour in slightly different ways, with the relationship becoming slightly curved (possibly suggesting quadratic terms). The loads are known very precisely, so errors in x_i's can by ignored, whereas the deformations, y_i, are subject to larger measurement errors that do need to be taken into account. Define a linear model suitable for describing these data, assuming that the same 6 loads are applied to each alloy, and write it out in the form $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$.

6. Show that for a linear model with model matrix $\mathbf{X}$, fitted to data $\mathbf{y}$, with fitted values $\hat{\boldsymbol{\mu}}$,

$$\mathbf{X}^{\mathsf{T}}\hat{\boldsymbol{\mu}} = \mathbf{X}^{\mathsf{T}}\mathbf{y}.$$

What implication does this have for the residuals of a model which includes an intercept term?

7. Equation (1.8) in section 1.3.3 gives an unbiased estimator of σ^2, but in the text unbiasedness was only demonstrated assuming that the response data were normally distributed. By considering $\mathbb{E}(r_i^2)$, show that the estimator (1.8) is unbiased whatever the distribution of the response, provided that the response data are independent with constant variance.

8. The `MASS` library contains a data frame `Rubber` on wear of tyre rubber. The response, `loss`, measures rubber loss in grammes per hour. The predictors are `tens`, a measure of the tensile strength of the rubber with units of kgm^{-2}, and `hard`, the hardness of the rubber in Shore[§] units. Modelling interest focuses on predicting wear from hardness and tensile strength.

(a) Starting with a model in which `loss` is a polynomial function of `tens` and

[§] Measures hardness as the extent of the rebound of a diamond tipped hammer, dropped on the test object.

hard, with all terms up to third order present, perform backwards model selection, based on hypothesis testing, to select an appropriate model for these data.

(b) Note the AIC scores of the various models that you have considered. It would also be possible to do model selection based on AIC, and the step function in R provides a convenient way of doing this. After reading the step help file, use it to select an appropriate model for the loss data, by AIC.

(c) Use the contour and predict functions in R to produce a contour plot of model predicted loss, against tens and hard. You may find functions seq and rep helpful, as well.

9. The R data frame warpbreaks gives the number of breaks per fixed length of wool during weaving, for two different wool types, and three different weaving tensions. Using a linear model, establish whether there is evidence that the effect of tension on break rate is dependent on the type of wool. If there is, use interaction.plot to examine the nature of the dependence.

10. This question is about modelling the relationship between stopping distance of a car and its speed at the moment that the driver is signalled to stop. Data on this are provided in R data frame cars. It takes a more or less fixed 'reaction time' for a driver to apply the car's brakes, so that the car will travel a distance directly proportional to its speed before beginning to slow. A car's kinetic energy is proportional to the square of its speed, but the brakes can only dissipate that energy, and slow the car, at a roughly constant rate per unit distance travelled: so we expect that once braking starts, the car will travel a distance proportional to the square of its initial speed, before stopping.

(a) Given the information provided above, fit a model of the form

$$\texttt{dist}_i = \beta_0 + \beta_1 \texttt{speed}_i + \beta_2 \texttt{speed}_i^2 + \epsilon_i$$

to the data in cars, and from this starting model, select the most appropriate model for the data using both AIC and hypothesis testing methods.

(b) From your selected model, estimate the average time that it takes a driver to apply the brakes (there are 5280 feet in a mile).

(c) When selecting between different polynomial models for a set of data, it is often claimed that one should not leave in higher powers of some continuous predictor, while removing lower powers of that predictor. Is this sensible?

11. This question relates to the material in sections 1.3.1 to 1.3.5, and it may be useful to review sections B.5 and B.6 of Appendix B. R function qr computed the QR decomposition of a matrix, while function qr.qry provides a means of efficiently pre-multiplying a vector by the **Q** matrix of the decomposition and qr.R extracts the **R** matrix. See ?qr for details.
The question concerns calculation of estimates and associated quantities for the linear model, $y_i = \mathbf{X}_i \boldsymbol{\beta} + \epsilon_i$, where the ϵ_i are i.i.d. $N(0, \sigma^2)$.

(a) Write an R function which will take a vector of response variables, y, and a model matrix, `X`, as arguments, and compute the least squares estimates of associated parameters, $\boldsymbol{\beta}$, based on QR decomposition of `X`.

(b) Test your function by using it to estimate the parameters of the model

$$\texttt{dist}_i = \beta_0 + \beta_1 \texttt{speed}_i + \beta_2 \texttt{speed}_i^2 + \epsilon_i$$

for the data found in R data frame `cars`. Note that:

```
X <- model.matrix(dist ~ speed + I(speed^2),cars)
```

will produce a suitable model matrix. Check your answers against those produced by the `lm` function.

(c) Extend your function to also return the estimated standard errors of the parameter estimators, and the estimated residual variance. Again, check your answers against what `lm` produces, using the `cars` model. Note that `solve(R)` or more efficiently `backsolve(R,diag(ncol(R)))` will produce the inverse of an upper triangular matrix `R`.

(d) Use R function `pt` to produce p-values for testing the null hypothesis that each β_i is zero (against a two sided alternative). Again check your answers against a `summary` of an equivalent `lm` fit.

(e) Extend your fitting function again, to produce the p-values associated with a sequential ANOVA table for your model (see section 1.3.5). Again test your results by comparison with the results of applying the `anova` function to an equivalent `lm` fit. Note that `pf` is the function giving the c.d.f. of F-distributions.

12. R data frame `InsectSprays` contains counts of insects in each of several plots. The plots had each been sprayed with one of six insecticides. A model for these data might be

$$y_i = \mu + \beta_j \text{ if } i^{\text{th}} \text{ observation is for spray } j.$$

A possible identifiability constraint for this model is that $\sum_j \beta_j = 0$. In R, construct the rank deficient model matrix, for this model, and the coefficient matrix for the sum to zero constraint on the parameters. Using the methods of section 1.8.1, impose the constraint via QR decomposition of the constraint matrix, fit the model (using `lm` with your constrained model matrix), and then obtain the estimates of the original parameters. You will need to use R functions `qr`, `qr.qy` and `qr.qty` as part of this. Confirm that your estimated parameters meet the constraint.

13. The R data frame `trees` contains data on `Height`, `Girth` and `Volume` of 31 felled cherry trees. A possible model for these data is

$$\texttt{Volume}_i = \beta_1 \texttt{Girth}_i^{\beta_2} \texttt{Height}_i^{\beta_3} + \epsilon_i,$$

which can be fitted by non-linear least squares using the method of section 1.8.8.

(a) Write an R function to evaluate (i) the vector of $\mathbb{E}(\texttt{Volume})$ estimates given a vector of β_j values and vectors of `Girth` and `Height` measurements and (ii) the 31×3 'Jacobian' matrix with $(i, j)^{\text{th}}$ element $\partial \mathbb{E}(\texttt{Volume}_i)/\partial \beta_j$, returning these in a two item list. (Recall that $\partial x^y / \partial y = x^y \log(x)$.)

(b) Write R code to fit the model, to the `trees` data, using the method of section 1.8.8. Starting values of .002, 2 and 1 are reasonable.

(c) Evaluate approximate standard error estimates for your estimated model parameters.