MAST90104 - Lecture 6

Weichang Yu

Room 108, Old Geology South Bldg
School of Mathematics and Statistics, University of Melbourne

The less than full rank model

In previous sections we studied the linear model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

,
where $\mathbf{X}$, of dimension $n \times p$, is of full rank, i.e. $r(\mathbf{X}) = p$ under assumption II.

Assumption II is important because a full rank $\mathbf{X}$ implies that $\mathbf{X}^T\mathbf{X}$ is invertible, and therefore the normal equations

$$\mathbf{X}^T\mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{X}^T\mathbf{y}$$

have a unique solution.

The less than full rank model

Unfortunately, not all linear models fall into this category.

For example, consider the *the classification model with one factor (also known as the one-way ANOVA or one-way classification model) having k levels*.

In this model, samples come from k distinct populations, with different characteristics. The *factor* is the variable that gives the population.

The term *factor* is the one used in R.

We wish to investigate the differences between these populations.

One-way classification model

For example:

- A marketing manager compares satisfaction ratings of three different products (each customer rates only one product)
- An educational specialist compares the math aptitude of 10 different schools
- An engineer investigates the sulfur content in the five major coal seams in a particular geographic region.

One-way classification model

Let y_{ij} be the jth observation taken from the ith population. Then a natural model is

$$y_{ij} = \mu + \tau_i + \varepsilon_{ij},$$

for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, n_i$, where

- k is the number of populations/treatments;
- n_i is the number of observations from the ith population;
- μ is the overall mean;
- τ_i is a mean characteristic for population i.

One-way classification model

$$\begin{bmatrix} y_{11} \\ y_{12} \\ \vdots \\ y_{21} \\ y_{22} \\ \vdots \\ y_{k,n_k} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 0 & 1 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \vdots \\ \tau_k \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{12} \\ \vdots \\ \varepsilon_{21} \\ \varepsilon_{22} \\ \vdots \\ \varepsilon_{k,n_k} \end{bmatrix}$$

$$\mathbf{y} = \mathbf{X} \qquad \boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

The first column of $\mathbf{X}$ is the sum of the remaining columns, and therefore $\mathbf{X}$ is not of full rank. Assumption (II) is violated.

One-way classification model

Example. Three different treatment methods for removing organic carbon from tar sand wastewater are compared: airflotation, foam separation, and ferric-chloride coagulation. A study is conducted and the amounts of carbon removed are:

AF	FS	FCC
34.6	38.8	26.7
35.1	39.0	26.7
35.3	40.1	27.0

One-way classification model

The linear model is

$$\begin{bmatrix} 34.6 \\ 35.1 \\ 35.3 \\ 38.8 \\ 39.0 \\ 40.1 \\ 26.7 \\ 26.7 \\ 27.0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{12} \\ \varepsilon_{13} \\ \varepsilon_{21} \\ \varepsilon_{22} \\ \varepsilon_{23} \\ \varepsilon_{31} \\ \varepsilon_{32} \\ \varepsilon_{33} \end{bmatrix}$$

$$\mathbf{y} = \mathbf{X} \boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

The less than full rank model

The difficulty with a less than full rank model is that $\mathbf{X}^T\mathbf{X}$ is singular. This means that the normal equations do not have a unique solution.

However, the problem goes deeper than that: not only can we not estimate the parameters, but the parameters themselves are not well defined.

The less than full rank model

In a one-way classification model, the response variable from population i has a mean of $\mu + \tau_i$ for $i = 1, \ldots, 3$.

In our carbon removal example, suppose we have oracle knowledge that the population mean of methods 1, 2, and 3 are 35, 39, and 27 respectively. Thus, the model parameters must satisfy:

$$\begin{aligned} \mu + \tau_1 &= 35 \\ \mu + \tau_2 &= 39 \\ \mu + \tau_3 &= 27. \end{aligned}$$

So our parameters might be $\mu = 34, \tau_1 = 1, \tau_2 = 5, \tau_3 = -7$.

However, we can also have $\mu = 30, \tau_1 = 5, \tau_2 = 9, \tau_3 = -3$.

In fact, there are infinitely many solutions for $(\mu, \tau_1, \tau_2, \tau_3)$. We say that the model parameters are *unidentifiable*.

Reparametrization

Computer packages tackle the less than full rank model by converting it to a full rank model. We can then use all the machinery we have developed, in the knowledge that the least squares estimates and hypothesis tests have many desirable properties.

Example. Consider the one-way classification model with $k = 3$. The less than full rank model for this is

$$y_{ij} = \mu + \tau_i + \varepsilon_{ij},$$

for $i = 1, 2, 3$, $j = 1, 2, \ldots, n_i$.

However, we can write the mean of each population as

$$\mu_i = \mu + \tau_i.$$

The variable i is the *factor*. In the carbon removal example, the factor is the different methods of removing the organic carbon.

Reparametrization

Then we can recast the model as

$$y_{ij} = \mu_i + \varepsilon_{ij},$$

with corresponding matrices

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix}.$$

Reparametrization

The columns of $\mathbf{X}$ are now linearly independent, and so this is a full rank model that we can analyse. Simple matrix calculations give us

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} n_1 & 0 & 0 \\ 0 & n_2 & 0 \\ 0 & 0 & n_3 \end{bmatrix}, \quad (\mathbf{X}^T\mathbf{X})^{-1} = \begin{bmatrix} \frac{1}{n_1} & 0 & 0 \\ 0 & \frac{1}{n_2} & 0 \\ 0 & 0 & \frac{1}{n_3} \end{bmatrix}$$

$$\mathbf{X}^T\mathbf{y} = \begin{bmatrix} \sum_{j=1}^{n_1} y_{1j} \\ \sum_{j=1}^{n_2} y_{2j} \\ \sum_{j=1}^{n_3} y_{3j} \end{bmatrix}, \quad \widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \begin{bmatrix} \sum_{j=1}^{n_1} y_{1j} \big/ n_1 \\ \sum_{j=1}^{n_2} y_{2j} \big/ n_2 \\ \sum_{j=1}^{n_3} y_{3j} \big/ n_3 \end{bmatrix}.$$

Reparametrization

Therefore, the least squares estimates for each of the population means are the means of the samples drawn from that population:

$$\hat{\mu}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} y_{ij}.$$

Linear functions of the parameters, of the form $\mathbf{t}^T \boldsymbol{\beta}$, are estimated using $\mathbf{t}^T \widehat{\boldsymbol{\beta}}$. For example, the function $\mu_1 - \mu_2$ is estimated by

$$\frac{1}{n_1} \sum_{j=1}^{n_1} y_{1j} - \frac{1}{n_2} \sum_{j=1}^{n_2} y_{2j}.$$

Reparametrization

The standard assumption that the random error vector is normally distributed with mean $\mathbf{0}$ and variance $\sigma^2\mathbf{I}$ is interpreted in this context to mean that *all* populations have a common variance σ^2 (but different means). The standard estimator for this variance is the residual sum of squares divided by the degrees of freedom:

$$s^2 = \frac{\mathbf{y}^T(\mathbf{I} - \mathbf{H})\mathbf{y}}{n-p} = \frac{\mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\widehat{\boldsymbol{\beta}}}{n-3},$$

where $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is the hat matrix for which $\mathbf{H}\mathbf{y} = \mathbf{X}\widehat{\boldsymbol{\beta}}$.

Reparametrization

That is

$$
\begin{aligned}
& s^2 \\
&= \frac{1}{n-3}\left[\sum_{i=1}^{3}\sum_{j=1}^{n_i} y_{ij}^2 - \begin{bmatrix} \sum_{j=1}^{n_1} y_{1j} & \sum_{j=1}^{n_2} y_{2j} & \sum_{j=1}^{n_3} y_{3j} \end{bmatrix} \begin{bmatrix} \sum_{j=1}^{n_1} y_{1j}/n_1 \\ \sum_{j=1}^{n_2} y_{2j}/n_2 \\ \sum_{j=1}^{n_3} y_{3j}/n_3 \end{bmatrix}\right] \\
&= \frac{1}{n-3}\left[\sum_{i=1}^{3}\sum_{j=1}^{n_i} y_{ij}^2 - \sum_{i=1}^{3}\frac{1}{n_i}\left(\sum_{j=1}^{n_i} y_{ij}\right)^2\right] \\
&= \frac{1}{n-3}\sum_{i=1}^{3}\left[\sum_{j=1}^{n_i} y_{ij}^2 - \frac{1}{n_i}\left(\sum_{j=1}^{n_i} y_{ij}\right)^2\right].
\end{aligned}
$$

Reparametrization

This can be written as a 'pooled' variance

$$s^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2 + (n_3 - 1)s_3^2}{(n_1 - 1) + (n_2 - 1) + (n_3 - 1)}$$

where s_i^2 are the individual population variance estimators

$$s_i^2 = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} \left(y_{ij} - \frac{1}{n_i} \sum_{k=1}^{n_i} y_{ik} \right)^2 .$$

The case of 2 populations was discussed in MAST90105 and our full rank linear model theory provides a generalisation framework to $k > 2$ populations.

Reparametrization

In general, it is always possible to re-parameterise a less than full rank model into a full rank model.

There are a number of different ways to do this and each way will generate different parameters with different estimates.

So it is important to be precise about the way that this is done - we'll explore this through the lens of the practice in R.

Reparametrization

Example. Consider the *two-way* ANOVA model (without interaction), with two levels of each of the two factors:

$$y_{ijk} = \mu + \tau_i + \beta_j + \varepsilon_{ijk}, \quad i, j = 1, 2.$$

The design matrix for this model is

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

Reparametrization

It is obvious that the first column is the sum of the next two columns, and also the sum of the 4th and 5th columns. Thus $r(\mathbf{X}) = 3$.

Total number of columns in $\mathbf{X}$ is 5. This means that we have to reduce the number of columns of the design matrix by 2.

This can be done by creating linear combinations of the original columns

Framework for reparameterizing

Suppose there are n observations, p parameters, that the rank of the $\mathbf{X}$ matrix is $r < p$ and that $n \geq r$.

A linear combination of the columns is $\mathbf{Xc}$, for some $p-$dimensional column vector $\mathbf{c}$.

Hence r linear combinations of the columns can be expressed as $\mathbf{XC}$ where $\mathbf{C}$ is a $p \times r$ matrix.

The new product $\mathbf{XC}$ has dimensions n by r. Under what conditions can we be assured that $\mathbf{XC}$ is full rank?

XC is of full rank?

Mini-result: Let $\mathbf{X}$ and $\mathbf{C}$ be matrices of dimension n by p and p by r respectively. Suppose $r(\mathbf{X}) = r$ and $\mathbf{XC}$ is full rank. Then, $\mathbf{C}$ is full rank.

Proof: Note that the following inequality is always true:

$$r(\mathbf{XC}) \leq \min\{r, r(\mathbf{C})\}$$

Since $r \leq \min\{n, p\}$ and $\mathbf{XC}$ is full rank, we have $r(\mathbf{XC}) = r$. Hence, the rank of $\mathbf{C}$ satisfies

$$r \leq \min\{r, r(\mathbf{C})\}$$

and consequently

$$r(\mathbf{C}) \geq r$$

But $r(\mathbf{C}) \leq \min\{p, r\}$. Hence, $r(\mathbf{C}) = r$ and therefore $\mathbf{C}$ is full rank.

XC is of full rank?

The previous result can be summarised as: $\mathbf{XC}$ is full rank $\Rightarrow$ $\mathbf{C}$ is full rank. But is the converse true?

Suppose

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 1 & -1 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

Then

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

which has rank 1 and therefore isn't full rank. The condition cannot guarantee that $\mathbf{XC}$ is full rank.

Necessary and sufficient conditions for full rank $\mathbf{XC}$

Theorem 6.1

Suppose $\mathbf{X}$ *has* r *linearly independent columns and that the corresponding rows of* $\mathbf{C}$ *are also linearly independent. Rearrange the rows and columns of* $\mathbf{X}$ *and the rows of* $\mathbf{C}$ *such that*

$$\mathbf{X} = \left[\begin{array}{c|c} \mathbf{X}_r & \mathbf{X}_r\mathbf{D} \\ \hline \mathbf{X}_{n-r} & \mathbf{X}_{n-r}\mathbf{D} \end{array}\right] \quad \mathbf{C} = \left[\begin{array}{c} \mathbf{C}_r \\ \hline \mathbf{E}\mathbf{C}_r \end{array}\right],$$

$\mathbf{X}_r, \mathbf{X}_{n-r}, \mathbf{D}, \mathbf{C}_r, \mathbf{E}$ *are respectively*
$r \times r, n-r \times r, r \times p-r, r \times r$ & $p-r \times r$ *and* $\mathbf{X}_r, \mathbf{C}_r$ *are both rank* r.
Then $\mathbf{XC}$ *is full rank if and only if* $\mathbf{I}_r + \mathbf{DE}$ *has rank* r .

Reparameterisation through $\mathbf{C}$

If $\mathbf{X}$ & $\mathbf{C}$ satisfy the conditions of the previous theorem, then to change parameters from $\boldsymbol{\beta}$ to $\boldsymbol{\gamma}$ and predictors from $\mathbf{X}$ to $\widetilde{\mathbf{X}} = \mathbf{XC}$ with $\widetilde{\mathbf{X}}\boldsymbol{\gamma} = \mathbf{X}\boldsymbol{\beta}$, the first r equations in the ordering specified in the theorem are

$$\mathbf{X}_r(\mathbf{I}_r + \mathbf{DE})\mathbf{C}_r\boldsymbol{\gamma} = \left[\begin{array}{c|c} \mathbf{X}_r & \mathbf{X}_r\mathbf{D} \end{array} \right] \boldsymbol{\beta}.$$

The matrix on the left is non-singular since it is the product of the three non-singular matrices $\mathbf{X}_r, \mathbf{I}_r + \mathbf{DE}$ & $\mathbf{C}_r$. Hence there is a unique solution for $\boldsymbol{\gamma}$:

$$\boldsymbol{\gamma} = (\mathbf{X}_r(\mathbf{I}_r + \mathbf{DE})\mathbf{C}_r)^{-1} \left[\begin{array}{c|c} \mathbf{X}_r & \mathbf{X}_r\mathbf{D} \end{array} \right] \boldsymbol{\beta}. \tag{1}$$

in terms of $\boldsymbol{\beta}$ once $\mathbf{X}$ & $\mathbf{C}$ have been partitioned.

Using the formula for the inverse of a product of nonsingular matrices gives:

$$\begin{aligned} \boldsymbol{\gamma} &= \{(\mathbf{I}_r + \mathbf{DE})C_r\}^{-1}\mathbf{X}_r^{-1}\mathbf{X}_r \left[\begin{array}{c|c} \mathbf{I}_r & \mathbf{D} \end{array} \right] \boldsymbol{\beta}. \\ &= \{(\mathbf{I}_r + \mathbf{DE})C_r\}^{-1}(\boldsymbol{\beta}_r + \mathbf{D}\boldsymbol{\beta}_{p-r}) \end{aligned} \tag{2}$$

where

$$\boldsymbol{\beta} = \left[\begin{array}{c} \boldsymbol{\beta}_r \\ \boldsymbol{\beta}_{p-r} \end{array} \right]$$

and the dimensions of $\boldsymbol{\beta}_r, \boldsymbol{\beta}_{p-r}$ are $r \times 1, p - r \times 1$ respectively.

Different choices of $\mathbf{C}$

For any less than full rank model, there will be a number of different ways to choose the matrix C and each choice will correspond to a different reparameterisation.

Clearly, output from one of these choices will give different parameter estimates with different interpretations but:

Theorem 6.2

In Theorem 6.1, Restrict our choice of $\mathbf{C}$ to a full rank matrix. Then, the fitted values $\widehat{\mathbf{y}}$, the residuals $\mathbf{e}$, the residual sum of squares SS_{res}, the regression sum of squares SS_{reg}, and the mean-squared error s^2 are all invariant to the choice of $\mathbf{C}$.

i.e., the quantities $\widehat{\mathbf{y}}$, $\mathbf{e}$, SS_{res}, SS_{reg}, s^2 remain the same even if we change our choice of full rank $\mathbf{C}$.

Choices of **C** in R

The theorem allows us to use different choices of **C**, and the choice is one of convenience and interpretation in the context of the data.

The default option in R uses the command `contr.treatment`.

Consider a one-way ANOVA with a three-level factor, so in this case $p = 4$ and $r = 3$.

By default, R includes an intercept term so the matrix C is the following prefaced by a column which is the first unit vector in 4 dimensions and topped by a row which is the transpose of the first unit vector in 3 dimensions, so that the **XC** will have the first column of **X** and otherwise not the column of 1's in **X**

```
contr.treatment(3)

##   2 3
## 1 0 0
## 2 1 0
## 3 0 1
```

To simplify the presentation of our matrices, let $\mathbf{0}_m, \mathbf{1}_m$ denote column vectors of length m with entries $0, 1$ respectively. Let $\boldsymbol{v}_i$ be the unit column vector with 1 in the ith position and 0 elsewhere.

With parameter vector $\boldsymbol{\beta} = [\mu, \tau_1, \tau_2, \tau_3]^T$ with $n_i, i = 1, \cdots, 3$ observations at level i, the less than full rank model has matrix:

$$X = \begin{bmatrix} \mathbf{1}_{n_1} & \mathbf{1}_{n_1} & \mathbf{0}_{n_1} & \mathbf{0}_{n_1} \\ \mathbf{1}_{n_2} & \mathbf{0}_{n_2} & \mathbf{1}_{n_2} & \mathbf{0}_{n_2} \\ \mathbf{1}_{n_3} & \mathbf{0}_{n_3} & \mathbf{0}_{n_3} & \mathbf{1}_{n_3} \end{bmatrix}.$$

The matrix $\mathbf{C}$ is (before rerranging rows):

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

The effect of postmultiplying $\mathbf{X}$ by $\mathbf{C}$ is to form a new matrix the ith column of which is a linear combination of the columns of X whose weights are the ith column of C.

If a column of C is $\boldsymbol{v}_i$ for some i, then the corresponding linear combination of columns of $\mathbf{X}$ selects the ith column of $\mathbf{X}$.

In this case, the choice in R given above selects the first, third and fourth columns of $\mathbf{X}$.

Hence

$$\widetilde{\mathbf{X}} = \mathbf{XC} = \begin{bmatrix} \mathbf{1}_{n_1} & \mathbf{0}_{n_1} & \mathbf{0}_{n_1} \\ \mathbf{1}_{n_2} & \mathbf{1}_{n_2} & \mathbf{0}_{n_2} \\ \mathbf{1}_{n_3} & \mathbf{0}_{n_3} & \mathbf{1}_{n_3} \end{bmatrix}.$$

Treatment contrast parameters?

There are three parameters after reparameterization $\gamma_1, \gamma_2, \gamma_3$.

Since $\mathbf{X}\beta = \widetilde{\mathbf{X}}\boldsymbol{\gamma}$, picking rows $1, n_1 + 1, n_1 + n_2 + 1$ gives:

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \gamma_1 + \gamma_2 \\ \gamma_1 + \gamma_3 \end{bmatrix},$$

recalling that $\mu_i = \mu + \tau_i$ is the population mean for the ith population.

Rearanging gives $\gamma_1 = \mu + \tau_1$, $\gamma_2 = \tau_2 - \tau_1$, and $\gamma_3 = \tau_3 - \tau_1$.

That is, the intercept parameter is the population mean for the first level of the factor.

The other parameters are the *differences* between the population mean for the other levels with the population mean for the first level.

Exam marks example

We compare the marks of students in 3 different mathematics classes. There is another factor (IQ), but we ignore this for the time being.

```
maths <- read.csv("../data/maths.csv")
str(maths)

## 'data.frame': 30 obs. of  5 variables:
##  $ X      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ maths.y: int  81 84 81 79 78 79 81 85 72 79 ...
##  $ iq     : int  99 103 108 109 96 104 96 105 94 91 ...
##  $ class  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ class.f: int  1 1 1 1 1 1 1 1 1 1 ...

maths$class.f <- factor(maths$class.f)
```

Exam marks example

```
plot(maths$class.f, maths$maths.y)
```

Figure 1 shows the resulting boxplots.

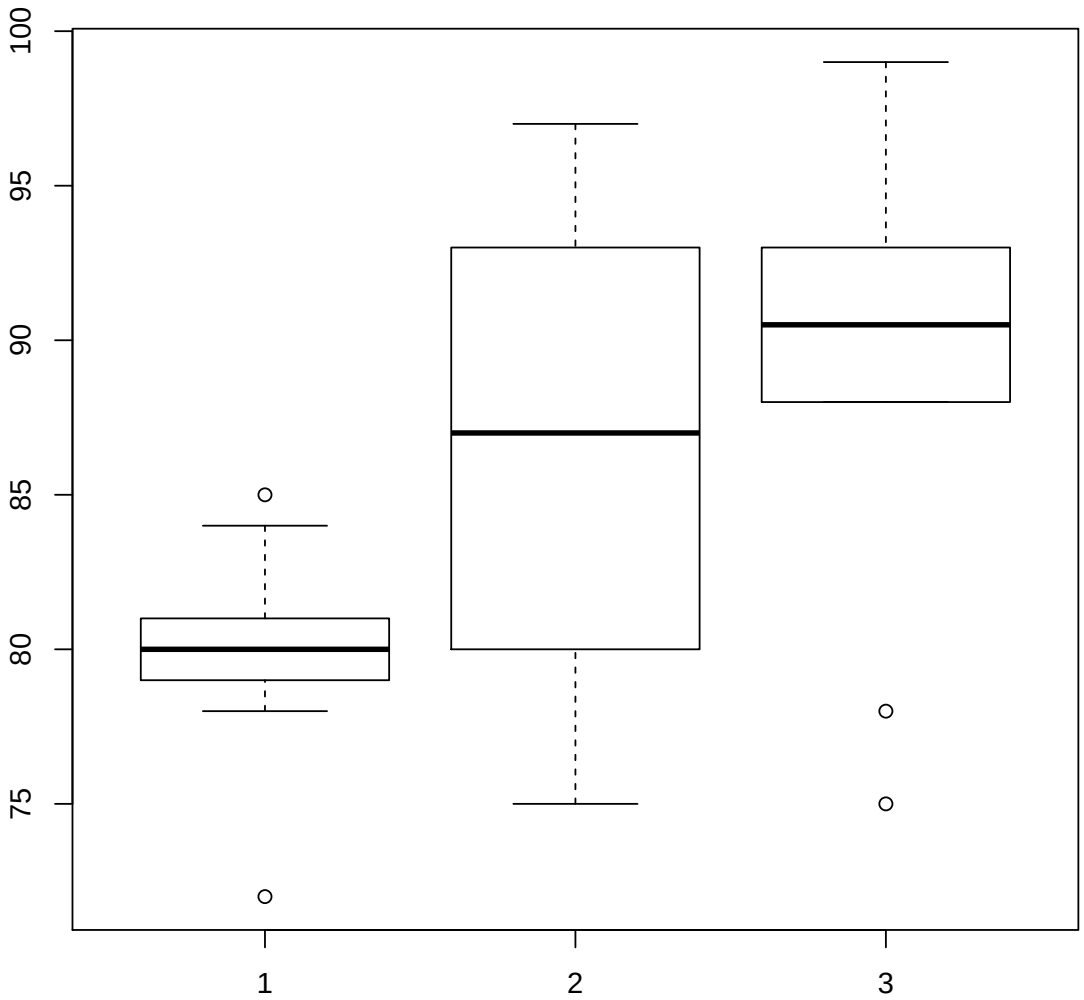


Figure: Boxplots of the marks in the three maths classes

Exam marks example

```
(y <- maths$maths.y)

#[1] 81 84 81 79 78 79 81 85 72 79 85 78 93 80 83 95 90 89 97 75 90 75 99
#[24] 97 93 91 88 93 90 78

n <- dim(maths)[1]
k <- length(levels(maths$class.f))
X <- matrix(0,n,k+1)
X[,1] <- 1
X[maths$class.f==1,2] <- 1
X[maths$class.f==2,3] <- 1
X[maths$class.f==3,4] <- 1
```

Exam marks example - extract of X matrix

```
##       [,1] [,2] [,3] [,4]
##  [1,]    1    1    0    0
##  [2,]    1    1    0    0
##  [3,]    1    1    0    0
##  [4,]    1    1    0    0
##  [5,]    1    1    0    0
##  [6,]    1    1    0    0
##  [7,]    1    1    0    0
##  [8,]    1    1    0    0
##  [9,]    1    1    0    0
## [10,]    1    1    0    0
## [11,]    1    0    1    0
## [12,]    1    0    1    0
## [13,]    1    0    1    0
## [14,]    1    0    1    0
## [15,]    1    0    1    0
## [16,]    1    0    1    0
## [17,]    1    0    1    0
## [18,]    1    0    1    0
## [19,]    1    0    1    0
## [20,]    1    0    1    0
## [21,]    1    0    0    1
## [22,]    1    0    0    1
## [23,]    1    0    0    1
## [24,]    1    0    0    1
## [25,]    1    0    0    1
## [26,]    1    0    0    1
## [27,]    1    0    0    1
## [28,]    1    0    0    1
## [29,]    1    0    0    1
## [30,]    1    0    0    1
```

Exam marks example: reparametrisation by omitting 1's

```
Xre <- X[,-1]
betahat <- solve(t(Xre) %*% Xre, t(Xre) %*% y)
```

```
##       [,1]
## [1,] 79.9
## [2,] 86.5
## [3,] 89.4
```

Exam marks example: reparametrisation using lm

```
modelre <- lm(y ~ 0 + X[,2] + X[,3] + X[,4])
summary(modelre)

##
## Call:
## lm(formula = y ~ 0 + X[, 2] + X[, 3] + X[, 4])
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14.40  -1.80   0.85   3.60  10.50
##
## Coefficients:
##        Estimate Std. Error t value Pr(>|t|)
## X[, 2]   79.900      2.053   38.92   <2e-16 ***
## X[, 3]   86.500      2.053   42.14   <2e-16 ***
## X[, 4]   89.400      2.053   43.55   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.492 on 27 degrees of freedom
## Multiple R-squared:  0.9948,Adjusted R-squared:  0.9942
## F-statistic:  1729 on 3 and 27 DF,  p-value: < 2.2e-16
```

The interpretation of the parameters is interesting here. They are the population means for students who would be taught in the three classes - identical conditions, same teachers.

The next slide shows what happens if the `lm` command is used in the usual way.

To make the message clear, the option for `contr.treatment` has been entered, but this is the default.

The resulting **C** matrix has already been shown and the parameters described as the population mean for the first class, and the differences between the population means for the second class with the first, and the third class with the first.

```r
modelstan <- lm(y ~ class.f, contrasts=contr.treatment,data = maths)
summary(modelstan)

##
## Call:
## lm(formula = y ~ class.f, data = maths, contrasts = contr.treatment)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14.40  -1.80   0.85   3.60  10.50
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   79.900      2.053  38.922  < 2e-16 ***
## class.f2       6.600      2.903   2.273  0.03117 *
## class.f3       9.500      2.903   3.272  0.00292 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.492 on 27 degrees of freedom
## Multiple R-squared:  0.2941,Adjusted R-squared:  0.2418
## F-statistic: 5.625 on 2 and 27 DF,  p-value: 0.009077
```

Estimability

Recall that in the less than full rank model, $\boldsymbol{\beta}$ cannot be estimated uniquely. So we cannot make inference on $\boldsymbol{\beta}$, but linear functions of $\boldsymbol{\beta}$, and in particular, functions that are invariant to the choice of solution $\widehat{\boldsymbol{\beta}}$.

Definition 6.3

In the general linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, a linear combination of the parameters $\mathbf{t}^T\boldsymbol{\beta}$ is said to be *estimable* if there exists a vector $\mathbf{c}$ such that $E[\mathbf{c}^T\mathbf{y}] = \mathbf{t}^T\boldsymbol{\beta}$, that is there exists an *unbiased* estimator of $\mathbf{t}^T\boldsymbol{\beta}$ based on a linear combination of the observations $\mathbf{y}$.

Estimability theorems

Theorem 6.4

Assume (I) and (III) holds. In the linear model $\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, *elements of* $X\boldsymbol{\beta}$ *are estimable.*

Proof. We know that $E[\mathbf{y}] = X\boldsymbol{\beta}$. Now take $\boldsymbol{v}_i$ to be the ith standard basis vector.

We have

$$\begin{aligned} (X\boldsymbol{\beta})_i &= \boldsymbol{v}_i^T X\boldsymbol{\beta} \\ &= \boldsymbol{v}_i^T E[\mathbf{y}] \\ &= E[\boldsymbol{v}_i^T \mathbf{y}] \end{aligned}$$

and so the ith element of $X\boldsymbol{\beta}$ is estimable.

Carbon example

Example. Consider the carbon removal example. We have

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}.$$

We know that we cannot estimate the parameter vector $\boldsymbol{\beta}$, because it is not uniquely determined.

Carbon example

However, the real quantities of interest are the mean responses from the three treatments. These are:

$$\begin{aligned}
\mu + \tau_1 &= \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \boldsymbol{\beta} \\
\mu + \tau_2 &= \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{\beta} \\
\mu + \tau_3 &= \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{\beta}
\end{aligned}$$

and each of these are elements of $X\boldsymbol{\beta}$. Therefore, they are estimable.

In a one-way classification model with any number of levels, $\mu + \tau_i$ is always estimable.

Independent of parameterization

Theorem 6.5

The BLUE estimator of an estimable combination of parameters is invariant to the choice of full rank parameterisation and is equal to the linear combination of the least squares estimates of its respective parameterisation.

Proof. Suppose $\mathbf{t}^T\beta$ is estimable. Then there is a column vector $\mathbf{c}$ so that $\mathbf{c}^T\mathbf{y}$ is an unbiased estimator of $\mathbf{t}^T\beta$.

But $E(\mathbf{c}^T y) = \mathbf{c}^T\mathbf{X}\beta$. Hence $\mathbf{t}^T\beta = \mathbf{c}^T\mathbf{X}\beta$ for all values of β.

Thus for a particular parameterisation, $\mathbf{y} = \widetilde{\mathbf{X}}\gamma + \epsilon$, we have $\mathbf{t}^T\beta = \mathbf{c}^T\mathbf{X}\beta = \mathbf{c}^T\widetilde{\mathbf{X}}\gamma$.

The BLUE of this is $\mathbf{c}^T\widetilde{\mathbf{X}}\mathbf{g}$, where $\mathbf{g} = (\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^T\mathbf{y}$. Note that the BLUE is a linear combination of the fitted values $\widehat{\mathbf{y}}$. Hence, it is invariant to the parameterization.

Linear combinations of estimable combinations

Theorem 6.6

Let $\mathbf{t}_1^T\boldsymbol{\beta}, \mathbf{t}_2^T\boldsymbol{\beta}, \ldots, \mathbf{t}_k^T\boldsymbol{\beta}$ *be estimable combinations of parameters, and let*

$$z = a_1\mathbf{t}_1^T\boldsymbol{\beta} + a_2\mathbf{t}_2^T\boldsymbol{\beta} + \ldots + a_k\mathbf{t}_k^T\boldsymbol{\beta}.$$

Then z *is estimable, and the best linear unbiased estimator for* z *is a linear combination of the least squares estimators in any parameterisation.*

Linear combinations proof

Proof. By definition,

$$z = (a_1\mathbf{t}_1 + a_2\mathbf{t}_2 + \ldots + a_k\mathbf{t}_k)^T\boldsymbol{\beta}.$$

Therefore z is estimable, with BLUE a corresponding linear combination of least squares estimates in any parameterization.

Treatment contrasts

Of particular interest in many studies is the way different populations compare against each other. In a one-way classification model suppose the mean response for the ith population is $\mu + \tau_i$. To attach a numerical value to these comparisons, we form linear combinations

$$a_1\tau_1 + a_2\tau_2 + \ldots + a_k\tau_k,$$

where $\sum_{i=1}^{k} a_i = 0$.

These are called *treatment contrasts*. They wipe out the effect of the overall mean response in describing the mean differences between populations as we shall see soon.

Treatment contrasts

In a one-way classification model (with the mean of the ith population being $\mu + \tau_i$), any treatment contrast is estimable.

If

$$z = a_1\tau_1 + a_2\tau_2 + \ldots + a_k\tau_k$$

is a treatment contrast, then

$$\begin{aligned} z &= \left\{\sum_{i=1}^{k} a_i\mu\right\} + a_1\tau_1 + a_2\tau_2 + \ldots + a_k\tau_k \\ &= a_1(\mu + \tau_1) + a_2(\mu + \tau_2) + \ldots + a_k(\mu + \tau_k) \end{aligned}$$

is a linear combination of the estimable functions $\mu + \tau_i$, and is therefore estimable.

Treatment contrasts

Of particular interest among treatment contrasts is the contrast of the form $\tau_i - \tau_j$, for some $i \neq j$. This is because

$$\tau_i - \tau_j = (\mu + \tau_i) - (\mu + \tau_j)$$

is the difference between the mean responses in populations i and j.

We would expect to estimate this contrast by the corresponding difference in sample means, $\bar{y}_i - \bar{y}_j$.

`contr.treatment` and `contr.sum` in R

For the less than full rank model, R uses *contrasts* to record parameter estimates for a factor with k levels. The two main contrast sets are `contr.treatment` and `contr.sum`.

Each contrast set has the maximum number, k, of parameter combinations to estimate.

Label	`contr.treatment`	`contr.sum`
Intercept	$\mu + \tau_1$	$\mu + \frac{1}{k}\sum \tau_i$
level 1		$\tau_1 - \frac{1}{k}\sum \tau_i$
level 2	$\tau_2 - \tau_1$	$\tau_2 - \frac{1}{k}\sum \tau_i$
level 3	$\tau_3 - \tau_1$	$\tau_3 - \frac{1}{k}\sum \tau_i$
$\vdots$	$\vdots$	$\vdots$
level k-1	$\tau_{k-1} - \tau_1$	$\tau_{k-1} - \frac{1}{k}\sum \tau_i$
level k	$\tau_k - \tau_1$	

`contr.treatment` and `contr.sum` in R

The intercept term for `contr.treatment` (the default) is estimable because it is an element of $X\boldsymbol{\beta}$.

The intercept term for `contr.sum` is estimable because it is the average of $\mu + \tau_i$.

The other terms are contrasts between estimable combinations (check this).

Illustration of contr.treatment

```r
contrasts(maths$class.f) <- contr.treatment(3)
model <- lm(maths.y ~ class.f, data = maths)
summary(model)
```

```
##
## Call:
## lm(formula = maths.y ~ class.f, data = maths)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14.40  -1.80   0.85   3.60  10.50
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   79.900      2.053  38.922  < 2e-16 ***
## class.f2       6.600      2.903   2.273  0.03117 *
## class.f3       9.500      2.903   3.272  0.00292 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.492 on 27 degrees of freedom
## Multiple R-squared:  0.2941,Adjusted R-squared:  0.2418
## F-statistic: 5.625 on 2 and 27 DF,  p-value: 0.009077
```

Exam marks example - illustration of `contr.sum`

```
contrasts(maths$class.f) <- contr.sum(3)
model2 <- lm(maths.y ~ class.f, data = maths)
summary(model2)
```

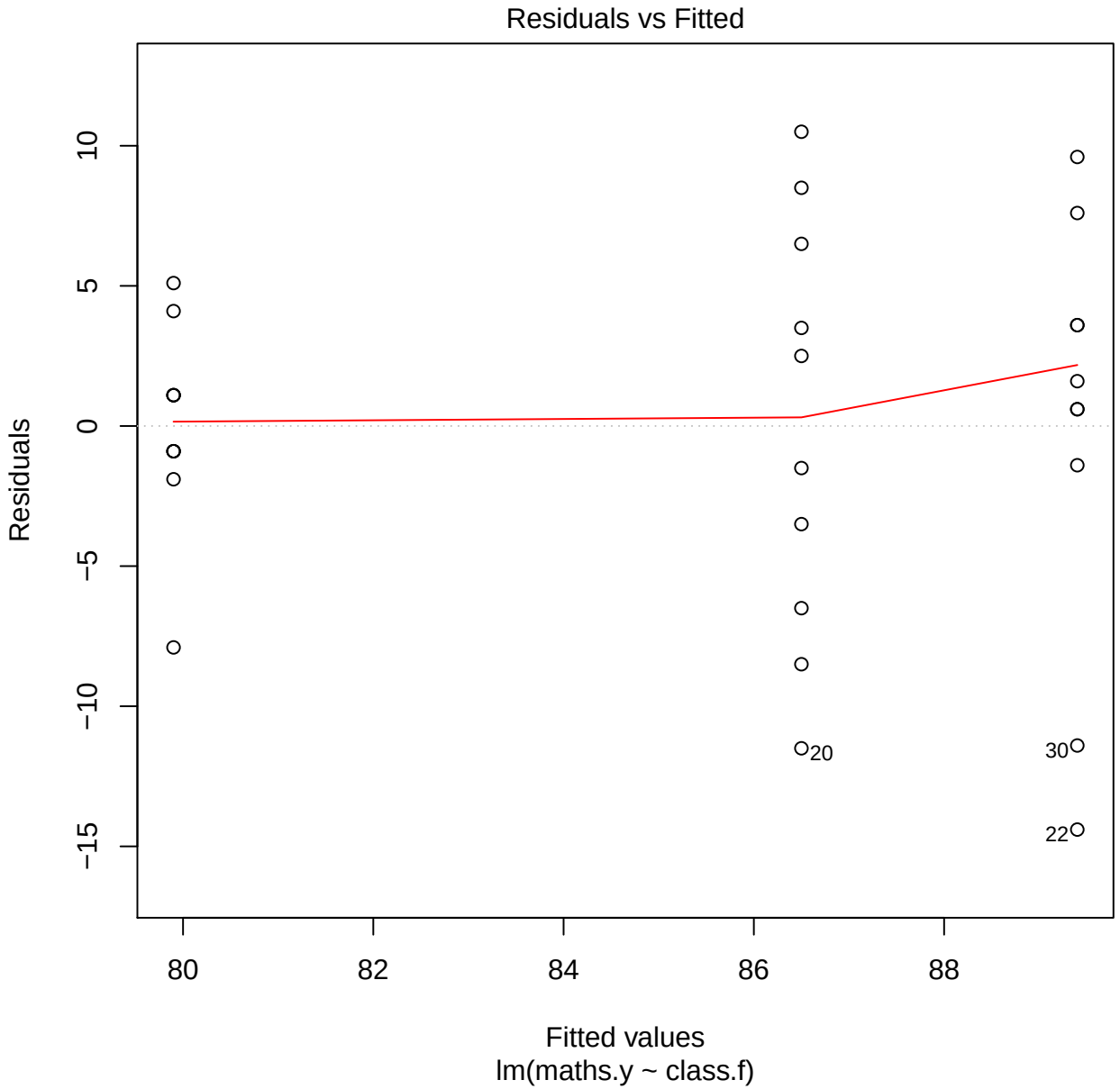
```
##
## Call:
## lm(formula = maths.y ~ class.f, data = maths)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14.40  -1.80   0.85   3.60  10.50
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   85.267      1.185  71.943  < 2e-16 ***
## class.f1      -5.367      1.676  -3.202  0.00348 **
## class.f2       1.233      1.676   0.736  0.46818
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.492 on 27 degrees of freedom
## Multiple R-squared:  0.2941,Adjusted R-squared:  0.2418
## F-statistic: 5.625 on 2 and 27 DF,  p-value: 0.009077
```

R commands to produce diagnostic plots

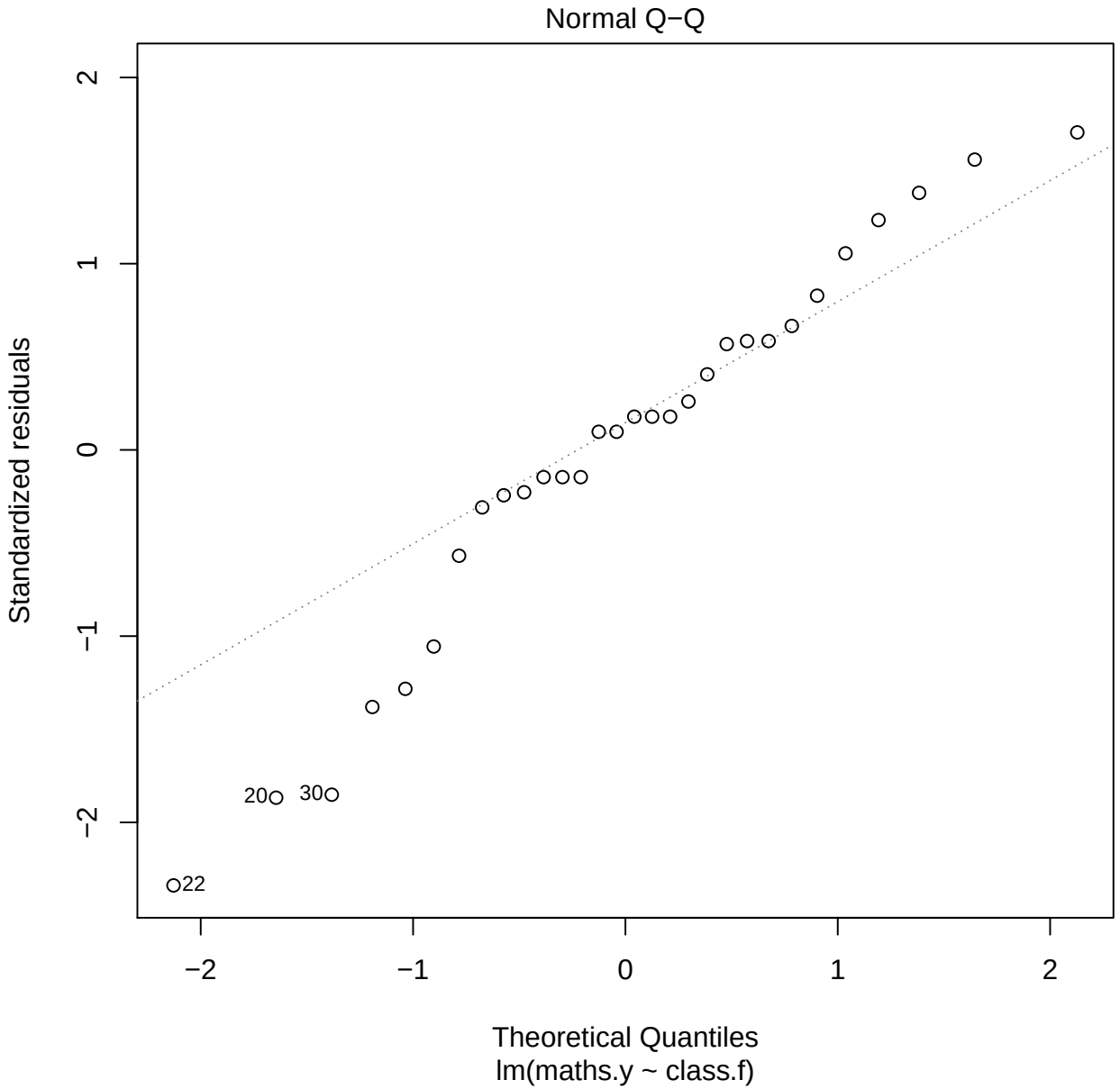
```
plot(model, which=1)
plot(model, which=2)
plot(model, which=3)
plot(model, which=5)
```

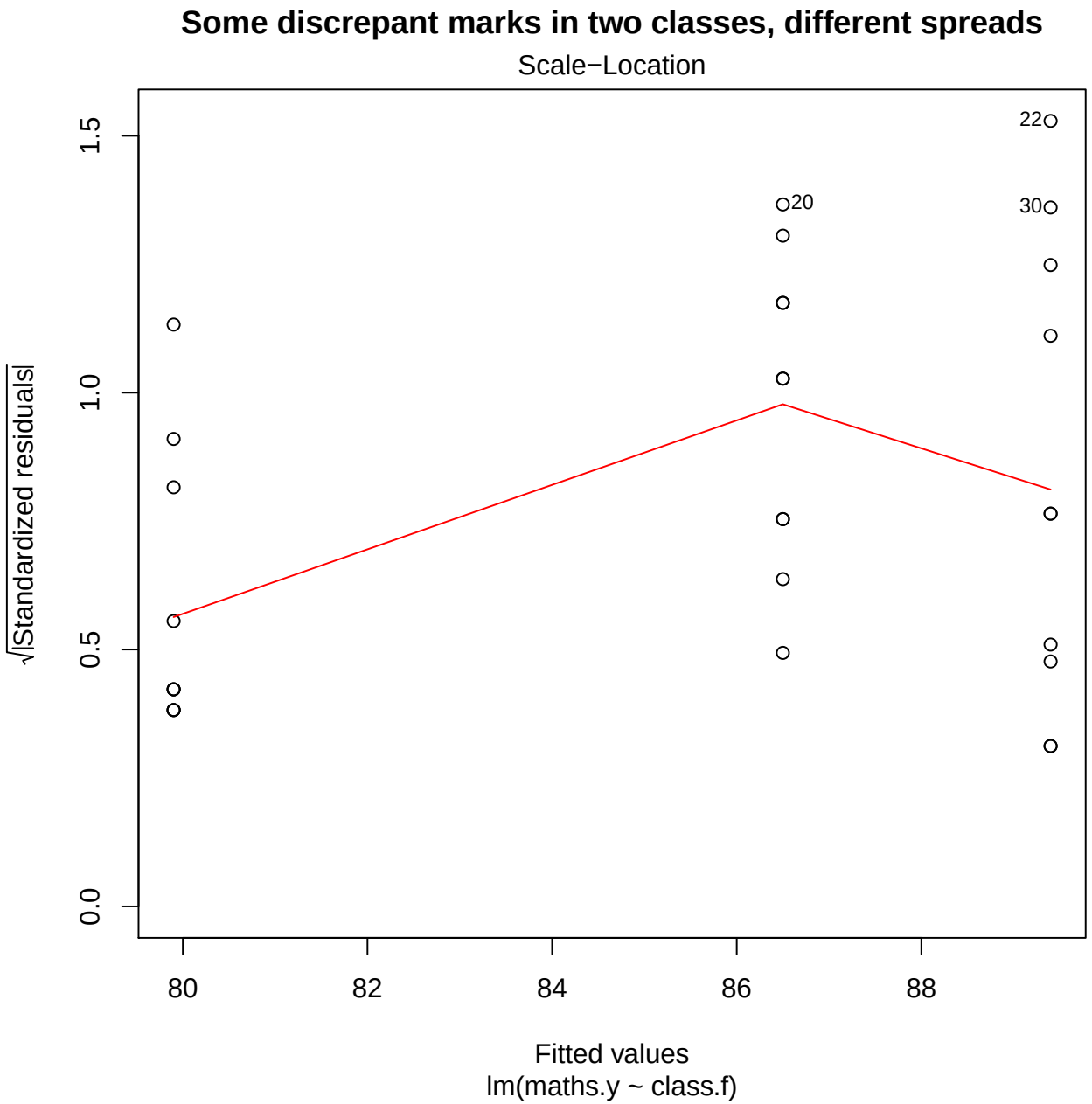
Figures 56, 57, 58 and 59 show the diagnostic plots for exam marks in the three classes.

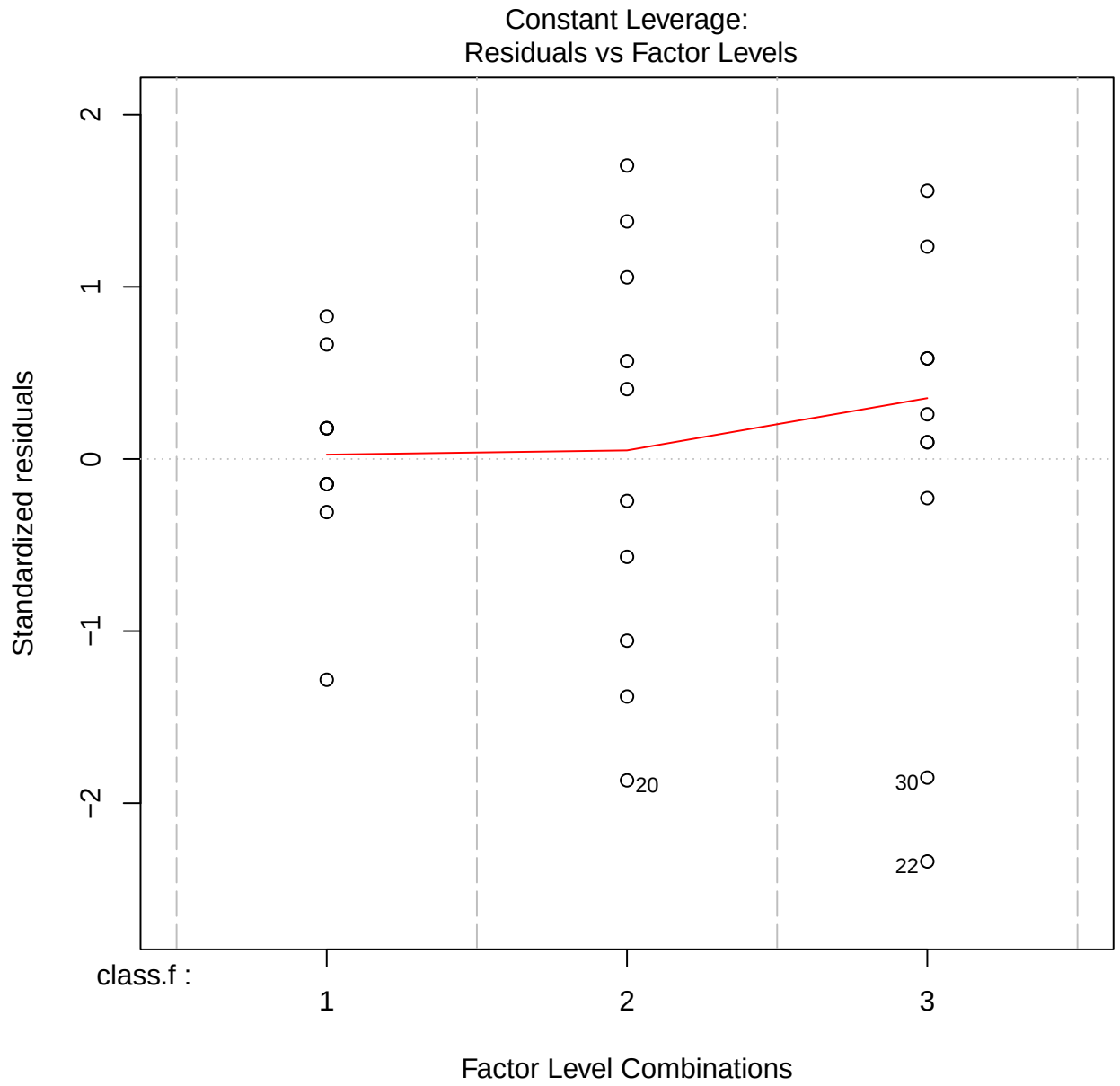
Different performance and spread for 3 classes?



Departures from normality at low end?




Some discrepant marks in two classes, different spreads
Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
20
22
30
80
82
84
86
88
Fitted values
lm(maths.y ~ class.f)


Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
2
1
0
−1
−2
20
30
22
class.f :
1
2
3
Factor Level Combinations

Testability

Note the notational change from $\mathbf{c}$ to $\mathbf{t}$

Definition 6.7

A hypothesis H_0 is testable if there exists a set of estimable functions $\mathbf{t}_1^T\boldsymbol{\beta}, \mathbf{t}_2^T\boldsymbol{\beta}, \ldots, \mathbf{t}_m^T\boldsymbol{\beta}$ such that H_0 is true if and only if

$$\mathbf{t}_1^T\boldsymbol{\beta} = \mathbf{t}_2^T\boldsymbol{\beta} = \ldots = \mathbf{t}_m^T\boldsymbol{\beta} = 0,$$

and $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_m$ are linearly independent.

Testability

That is, a testable hypothesis is of the form $H_0 : \mathbf{L}\boldsymbol{\beta} = \mathbf{0}$, where

$$\mathbf{L} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_m^T \end{bmatrix}$$

is $m \times p$ of rank m, and each $\mathbf{t}_i^T \boldsymbol{\beta}$ is estimable.

Testability

Example. Consider the one-way classification model with fixed effects and $k = 3$. The linear model that we use is

$$y_{ij} = \mu + \tau_i + \varepsilon_{ij}.$$

Consider the hypothesis that the means of all three populations are equal. This is equivalent to $H_0 : \tau_1 = \tau_2 = \tau_3$.

This hypothesis is true if and only if

$$\tau_1 - \tau_2 = 0$$

and

$$\tau_2 - \tau_3 = 0.$$

Testability

So we can express this hypothesis as $H_0 : \mathbf{L}\boldsymbol{\beta} = \mathbf{0}$, where

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}.$$

$\tau_1 - \tau_2$ is a contrast, so it is estimable. Similarly, $\tau_2 - \tau_3$ is estimable. The rows of $\mathbf{L}$ are obviously linearly independent, so H_0 is testable.

Testability

Once we have determined that a hypothesis is testable, how can we test it?

Answer: Take any full rank parameterisation, and the estimates of the parameters will be the same, and the tests of hypotheses will be the same.

The F statistic for this hypothesis (for the reparameterized full rank case) is

$$\frac{(\mathbf{L}\widehat{\beta})^T[\widetilde{\mathbf{L}}(\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{L}}^T]^{-1}\mathbf{L}\widehat{\beta}/m}{SS_{Res}/(n-r)},$$

which under the null hypothesis has an F distribution with m and $n-r$ degrees of freedom, where $m = r(\widetilde{\mathbf{L}})$ and $r = r(\widetilde{\mathbf{X}})$. Note that $\widetilde{\mathbf{L}}$ is a m by r full rank matrix such that $\widetilde{\mathbf{L}}\gamma = \mathbf{L}\beta$. More details in supplementary slides.

Carbon removal example

Example. Let us look at the carbon removal example from the previous section. We compare three methods of removing carbon from wastewater. The data is:

AF	FS	FCC
34.6	38.8	26.7
35.1	39.0	26.7
35.3	40.1	27.0

Carbon removal example

We test whether the populations have the same mean, i.e. $H_0 : \tau_1 = \tau_2 = \tau_3$.

This can be written in matrix form as $H_0 : \mathbf{L}\boldsymbol{\beta} = \mathbf{0}$, where

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}.$$

Carbon removal example

```
n <- 9
r <- 3
y <- c(34.6,35.1,35.3,38.8,39.0,40.1,26.7,26.7,27.0)
x <- factor(c(rep(1,3),rep(2,3),rep(3,3)))
anova(lm(y~x))

## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x          2 241.98 120.990  558.42 1.526e-07 ***
## Residuals  6   1.30   0.217
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Carbon removal example

```
anova(lm(y~x,contrast=contr.sum(x)))

## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x          2 241.98 120.990  558.42 1.526e-07 ***
## Residuals  6   1.30   0.217
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can reject H_0 firmly, so the populations are not all the same. It is still possible that *some* of the populations are the same, but not all of them.

Orthogonal contrasts

A concept which is similar in nature to orthogonality is that of *orthogonal contrasts*.

Definition 6.8

In a one-factor model, two treatment contrasts $\sum_{i=1}^{k} a_i \mu_i$ and $\sum_{i=1} b_i \mu_i$ are orthogonal if (and only if)

$$\sum_{i=1}^{k} \frac{a_i b_i}{n_i} = 0.$$

If each sub-population is equally sampled, this reduces to

$$\sum_{i=1}^{k} a_i b_i = 0.$$

Orthogonal contrasts

Orthogonal contrasts behave like orthogonal variables, in the sense that they can be tested independently of each other.

The sum of squares attributed to a hypothesis made up of multiple orthogonal contrasts can be broken down into sums of squares attributed to each of the individual contrasts.

Two-factor models

Example. We model the time taken to dissolve a capsule in a biological fluid. A study is conducted with 1 sample from each combination of factor levels and the following data found:

Time		Fluid type	
		Gastric	Duodenal
Capsule	A	39.5	31.2
Capsule	B	47.4	44

Two-factor models

The linear model is

$$\mathbf{y} = \begin{bmatrix} 39.5 \\ 47.4 \\ 31.2 \\ 44 \end{bmatrix}, \quad X = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \beta_1 \\ \beta_2 \end{bmatrix}.$$

We test the hypotheses that there is no difference in the response for the levels of each of the two factors.

Two-factor models

The first factor (fluid type) gives the null hypothesis against the alternative of different values of τ.

$$H_0 : \tau_1 = \tau_2 \text{ or } \begin{bmatrix} 0 & 1 & -1 & 0 & 0 \end{bmatrix} \boldsymbol{\beta} = \mathbf{0}.$$

The second factor (capsule type) gives the hypothesis of different values of β.

$$H_0 : \beta_1 = \beta_2 \text{ or } \begin{bmatrix} 0 & 0 & 0 & 1 & -1 \end{bmatrix} \boldsymbol{\beta} = \mathbf{0}.$$

The next slide gives the default option of `contr.treatment` for both τ and β, and the following one gives it for `contr.sum`. This is followed by the the model matrices and the analysis of variance.

```
##
## Call:
## lm(formula = y ~ tau + beta)
##
## Residuals:
##      1      2      3      4
##  1.225 -1.225 -1.225  1.225
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   38.275      2.122  18.039   0.0353 *
## tau2          -5.850      2.450  -2.388   0.2525
## beta2         10.350      2.450   4.224   0.1480
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.45 on 1 degrees of freedom
## Multiple R-squared:  0.9593,Adjusted R-squared:  0.8778
## F-statistic: 11.77 on 2 and 1 DF,  p-value: 0.2018
```

Sum contrasts

```
summary(models <- lm(y ~ tau + beta,
contrasts=
list(tau="contr.sum",beta="contr.sum")))

##
## Call:
## lm(formula = y ~ tau + beta, contrasts = list(tau = "contr.sum",
##     beta = "contr.sum"))
##
## Residuals:
##      1      2      3      4
##  1.225 -1.225 -1.225  1.225
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   40.525      1.225  33.082   0.0192 *
## tau1           2.925      1.225   2.388   0.2525
## beta1         -5.175      1.225  -4.224   0.1480
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.45 on 1 degrees of freedom
## Multiple R-squared:  0.9593,Adjusted R-squared:  0.8778
## F-statistic: 11.77 on 2 and 1 DF,  p-value: 0.2018
```

```r
anova(modelt)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df  Sum Sq Mean Sq F value Pr(>F)
## tau        1  34.222  34.222  5.7014 0.2525
## beta       1 107.123 107.123 17.8463 0.1480
## Residuals  1   6.002   6.002
```

```r
anova(models)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df  Sum Sq Mean Sq F value Pr(>F)
## tau        1  34.222  34.222  5.7014 0.2525
## beta       1 107.123 107.123 17.8463 0.1480
## Residuals  1   6.002   6.002
```

We cannot reject either null hypothesis, which is not surprising since there is only one data point for each combination of parameters.

In a two- or more-factor model, it is still possible to have orthogonal contrasts. This happens in particular if the design is *balanced*, which means that we have the same number of samples from each combination of factors.

In this case it is simple to show that any treatment contrast in the τs is orthogonal to any treatment contrast in the βs.

Interaction

In some cases, it is possible that *interaction* between factors may occur.

Interaction happens when one factor affects the effect of another factor.

For example, if the effect of factor 1 when factor 2 is at level 1 is different from the effect of factor 1 when factor 2 is at level 2, then there is interaction.

Interaction

Example. Suppose that we are studying the effect of pressure and temperature on viscosity, and the *actual* means of the response variable for each of the combinations are given by:

		Pressure			
		1	2	3	4
Temperature	1	4	6	4	3
	2	8	2	7	5

Interaction

When the pressure is at level 1, changing the temperature from level 1 to level 2 results in an increase of viscosity of 4.

However, when the pressure is at level 2, changing the temperature from level 1 to level 2 results in a *decrease* of viscosity of 4!

In this case, the factors interact.

Interaction

If, on the other hand, the actual means were:

		Pressure			
		1	2	3	4
Temperature	1	4	6	4	3
	2	8	10	8	7

then there would be no interaction between the factors. Even though the factors themselves are significant, the *combination* of factor levels has no effect apart from the individual factor effects.

Interaction

An additive model assumes that there is no interaction between the factors, so the effects of the factor levels can be measured in isolation from the other factor(s).

If we have interaction, or want to test whether there is interaction, we must use a different model:

$$y_{ijk} = \mu + \tau_i + \beta_j + \xi_{ij} + \varepsilon_{ijk},$$

where ξ_{ij} is an interaction term which quantifies the effect of factor 1 being at level i at the same time that factor 2 is at level j.

Interaction

Example. Consider the previous example (dissolving a capsule in fluid). If we allow an interaction term, $\mathbf{y}$ stays the same, but the linear model becomes

$$X = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \beta_1 \\ \beta_2 \\ \xi_{11} \\ \xi_{12} \\ \xi_{21} \\ \xi_{22} \end{bmatrix}.$$

Interaction

In a two-factor model with interaction, we are often interested in testing whether there is interaction or not.

However, testing the presence of interaction is not quite as straightforward as it may seem.

Interaction

Theorem 6.9

For the linear model

$$y_{ijk} = \mu + \tau_i + \beta_j + \xi_{ij} + \varepsilon_{ijk},$$

there is no interaction if and only if

$$(\xi_{ij} - \xi_{ij'}) - (\xi_{i'j} - \xi_{i'j'}) = 0,$$

for all $i \neq i', j \neq j'$.
Moreover these quantities are all estimable.

Interaction

Theorem 6.9 generates $IJ(I-1)(J-1)$ equations. However, it can be shown that all but $(I-1)(J-1)$ of them are redundant.

Example. In a two-factor design with two levels in each factor, Theorem 6.9 shows that there is no interaction if and only if

$$\begin{aligned}
(\xi_{11}-\xi_{12})-(\xi_{21}-\xi_{22}) &= 0\\
(\xi_{21}-\xi_{22})-(\xi_{11}-\xi_{12}) &= 0\\
(\xi_{12}-\xi_{11})-(\xi_{22}-\xi_{21}) &= 0\\
(\xi_{22}-\xi_{21})-(\xi_{12}-\xi_{11}) &= 0.
\end{aligned}$$

Interaction

It is easy to see that all of these equations are equivalent, so we need only test one.

This gives the hypothesis $H_0 : \mathbf{L}\boldsymbol{\beta} = \mathbf{0}$, where

$$\begin{aligned} \mathbf{L} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \end{bmatrix}, \\ \boldsymbol{\beta} &= \begin{bmatrix} \mu & \tau_1 & \tau_2 & \beta_1 & \beta_2 & \xi_{11} & \xi_{12} & \xi_{21} & \xi_{22} \end{bmatrix}^T. \end{aligned}$$

Interaction considerations

Some things to consider when testing for interaction:

1) If we have one sample per combination of factors, it is impossible to account for or test for interaction.

This is because $n = r = r(\widetilde{\mathbf{X}})$ and therefore the df of SS_{res} equals to $n - r = 0$.

Essentially we treat each combination of factors as a separate population. If we have one sample from each population, then we have no way to estimate the variance!

Interaction considerations

2) It is possible to have interaction between three or more factors.

However, this is hard to test for and hard to interpret. In practice most people only look at two-factor interactions.

Engine example

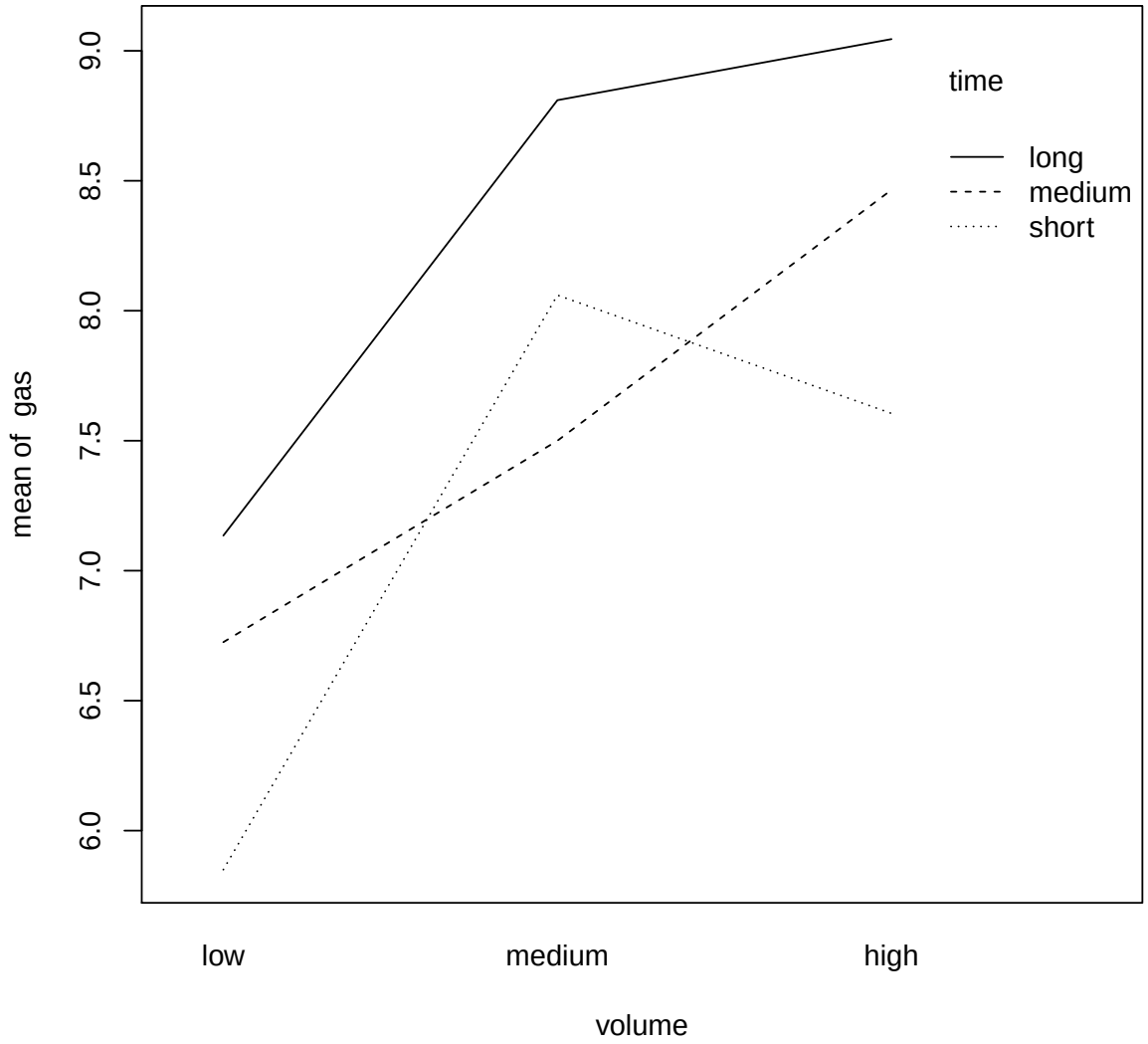
We look at the effect of pre-chamber volume ratio and injection timing on the emission of noxious gas from an engine. The factors have 3 levels each.

```
str(engine)

## 'data.frame': 18 obs. of  3 variables:
##  $ gas   : num  6.27 8.08 7.34 5.43 8.04 7.87 6.94 7.48 8.61 6.51 ...
##  $ volume: Factor w/ 3 levels "low","medium",..: 1 2 3 1 2 3 1 2 3 1 ...
##  $ time  : Factor w/ 3 levels "short","medium",..: 1 1 1 1 1 1 2 2 2 2 ...

means

##       [,1]  [,2]  [,3]
## [1,] 5.850 6.725 7.135
## [2,] 8.060 7.500 8.810
## [3,] 7.605 8.465 9.045
```

Engine example

```
with(engine, interaction.plot(time, volume, gas))
```

Engine example



ANCOVA

We can also do analysis of covariance (ANCOVA) using the linear model framework.

In this case we have one (or more) categorical predictors and one (or more) continuous predictors. For example:

$$y_{ij} = \mu + \tau_i + \beta x_{ij} + \xi_i x_{ij} + \varepsilon_{ij}.$$

We can think of this simple model as fitting several regression lines, one to each population (assuming equal variances across populations).

ANCOVA

Interaction in this case means that the slopes of the regression lines (effect of continuous predictor) are different for each population.

A model without interaction assumes that the slopes are the same (but the intercepts may be different):

$$y_{ij} = \mu + \tau_i + \beta x_{ij} + \varepsilon_{ij}.$$

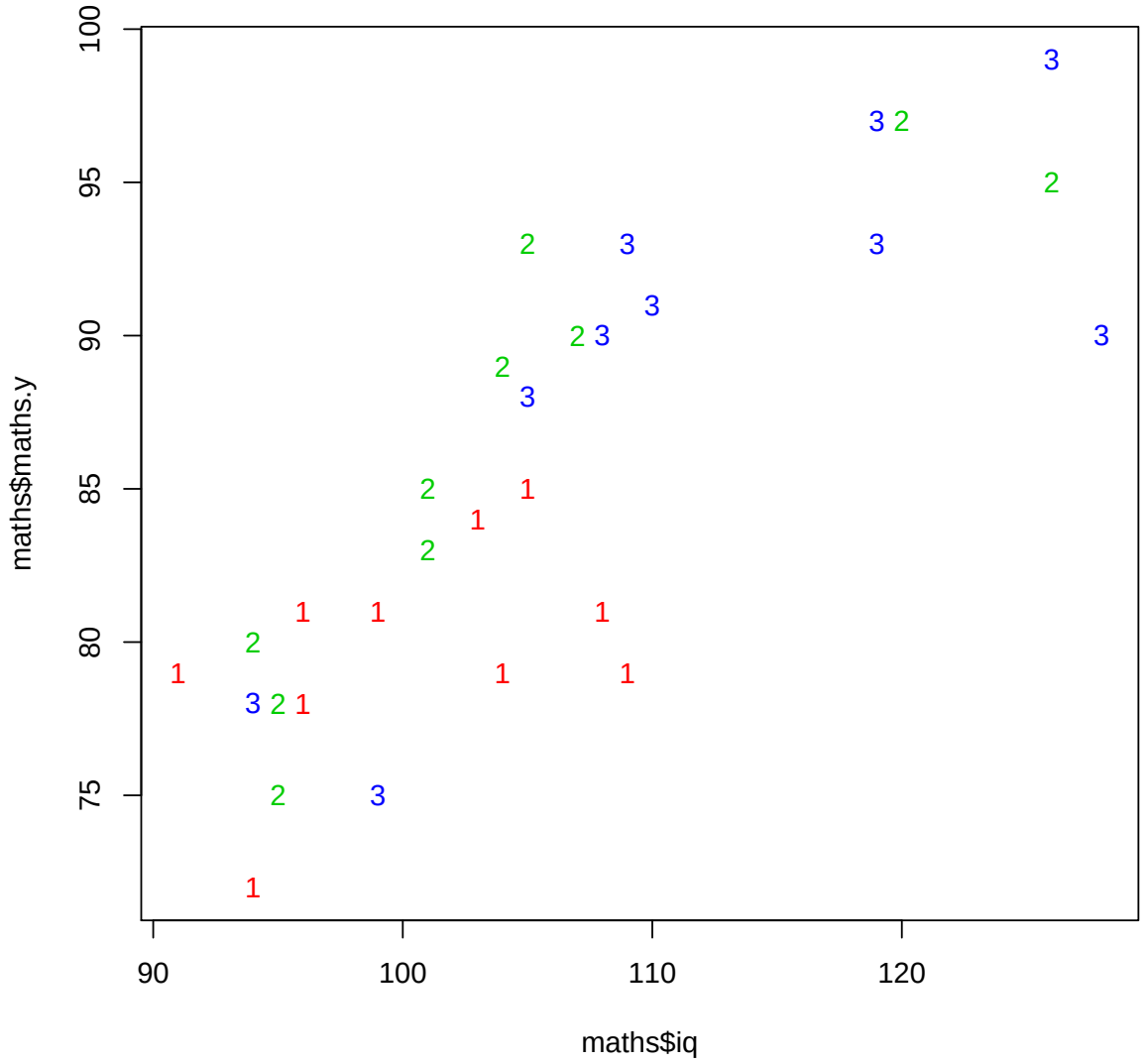
We fit these models using the less than full rank model.

Exam marks example

The `maths` dataset also has another component: the IQ of the student and we can plot the maths score versus IQ with class colour coded.

```
plot(maths$iq, maths$maths.y, pch=array(maths$class.f),
col=maths$class+1)
```

Plot of maths versus iq colour coded for class



The full model includes `class` as a factor, `iq` as a continuous predictor-variable, and allows for different slopes and intercepts for each class - $*$ indicates to allow for this.

```
contrasts(maths$class.f) <- contr.sum(3)
model <- lm(maths.y ~ class.f * iq, data=maths)
summary(model)
```

```
##
## Call:
## lm(formula = maths.y ~ class.f * iq, data = maths)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2507 -1.8312  0.9807  2.4711  6.3765
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  34.43315    9.68693   3.555  0.00161 **
## class.f1     18.32451   15.97135   1.147  0.26255
## class.f2    -12.63968   12.36337  -1.022  0.31681
## iq            0.47683    0.09327   5.112 3.13e-05 ***
## class.f1:iq  -0.20676    0.15688  -1.318  0.19996
## class.f2:iq   0.14060    0.11842   1.187  0.24674
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.043 on 24 degrees of freedom
## Multiple R-squared:  0.7566,Adjusted R-squared:  0.7059
## F-statistic: 14.92 on 5 and 24 DF,  p-value: 1.072e-06
```

Testing whether interaction is needed

```
amodel <- lm(maths.y ~ class.f + iq, data = maths)
anova(amodel, model)

## Analysis of Variance Table
##
## Model 1: maths.y ~ class.f + iq
## Model 2: maths.y ~ class.f * iq
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1     26 423.42
## 2     24 392.36  2    31.062 0.95 0.4008
```

Interaction is not significant, so we remove the interaction term and fit an additive model - this amounts to the same slope for each class but different intercepts.

Summary of chosen model

```
summary(amodel)

##
## Call:
## lm(formula = maths.y ~ class.f + iq, data = maths)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.137 -2.842  1.220  2.662  6.393
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 28.62522    8.58371   3.335  0.00257 **
## class.f1    -2.59713    1.12274  -2.313  0.02888 *
## class.f2     1.69790    1.04432   1.626  0.11605
## iq           0.53604    0.08093   6.623 5.03e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.036 on 26 degrees of freedom
## Multiple R-squared:  0.7373,Adjusted R-squared:  0.707
## F-statistic: 24.33 on 3 and 26 DF,  p-value: 1.032e-07
```

Is IQ necessary?

```
basemodel <- lm(maths.y ~ class.f, data = maths)
anova(basemodel,amodel)

## Analysis of Variance Table
##
## Model 1: maths.y ~ class.f
## Model 2: maths.y ~ class.f + iq
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     27 1137.80
## 2     26  423.42  1    714.38 43.866 5.032e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0
```

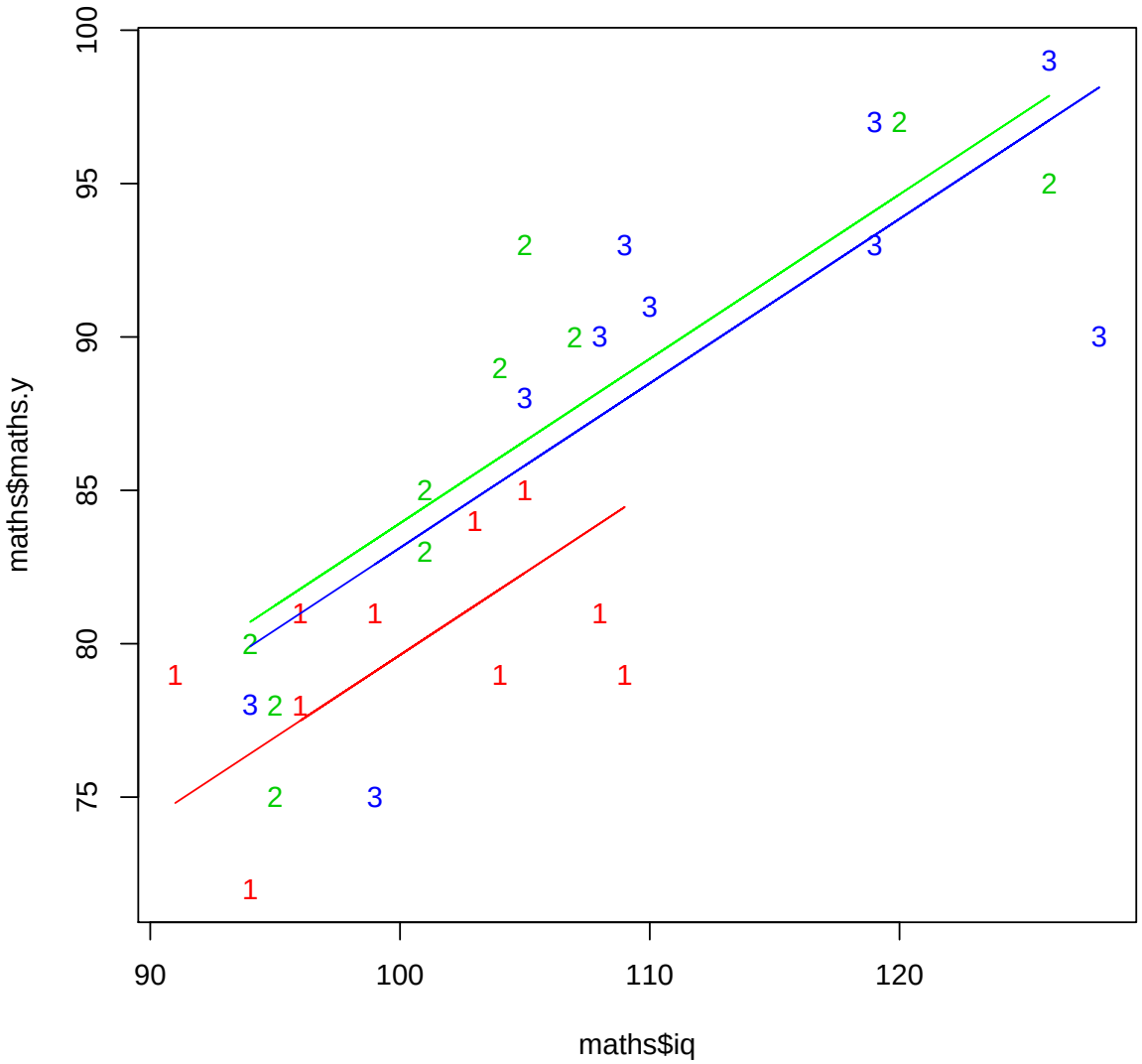
Clearly IQ is significant.

Exam marks example

```
basemodel <- lm(maths.y ~ iq, data = maths)
anova(basemodel, amodel)

## Analysis of Variance Table
##
## Model 1: maths.y ~ iq
## Model 2: maths.y ~ class.f + iq
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     28 518.13
## 2     26 423.42  2    94.707 2.9077 0.0725 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The class is not very significant. However, since it is so close, we will retain it. Remember that it was significant in the one-factor model!

The fitted ANCOVA model



Exam marks example

To spell out the full model including IQ as a less than full rank model:

```
maths <- read.csv("../data/maths.csv")
maths$class.f <- factor(maths$class)
y <- maths$maths.y
n <- 30
X <- matrix(0, n, 8)
X[,1] <- 1
X[cbind(1:n,maths$class+1)] <- 1
X[,5] <- maths$iq
X[cbind(1:n,maths$class+5)] <- maths$iq
r <- rankMatrix(X)[1]
```

Model matrix - less than full rank

```
X

##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
##  [1,]    1    1    0    0   99   99    0    0
##  [2,]    1    1    0    0  103  103    0    0
##  [3,]    1    1    0    0  108  108    0    0
##  [4,]    1    1    0    0  109  109    0    0
##  [5,]    1    1    0    0   96   96    0    0
##  [6,]    1    1    0    0  104  104    0    0
##  [7,]    1    1    0    0   96   96    0    0
##  [8,]    1    1    0    0  105  105    0    0
##  [9,]    1    1    0    0   94   94    0    0
## [10,]    1    1    0    0   91   91    0    0
## [11,]    1    0    1    0  101    0  101    0
## [12,]    1    0    1    0   95    0   95    0
## [13,]    1    0    1    0  105    0  105    0
## [14,]    1    0    1    0   94    0   94    0
## [15,]    1    0    1    0  101    0  101    0
## [16,]    1    0    1    0  126    0  126    0
## [17,]    1    0    1    0  107    0  107    0
## [18,]    1    0    1    0  104    0  104    0
## [19,]    1    0    1    0  120    0  120    0
## [20,]    1    0    1    0   95    0   95    0
## [21,]    1    0    0    1  108    0    0  108
## [22,]    1    0    0    1   99    0    0   99
## [23,]    1    0    0    1  126    0    0  126
## [24,]    1    0    0    1  119    0    0  119
## [25,]    1    0    0    1  109    0    0  109
## [26,]    1    0    0    1  110    0    0  110
## [27,]    1    0    0    1  105    0    0  105
## OUTPUT TRUNCATED
```

Model matrix - contr.treatment

```
model.matrix(~class.f*iq,data=maths)

##    (Intercept) class.f2 class.f3  iq class.f2:iq class.f3:iq
## 1            1        0        0  99           0           0
## 2            1        0        0 103           0           0
## 3            1        0        0 108           0           0
## 4            1        0        0 109           0           0
## 5            1        0        0  96           0           0
## 6            1        0        0 104           0           0
## 7            1        0        0  96           0           0
## 8            1        0        0 105           0           0
## 9            1        0        0  94           0           0
## 10           1        0        0  91           0           0
## 11           1        1        0 101         101           0
## 12           1        1        0  95          95           0
## 13           1        1        0 105         105           0
## 14           1        1        0  94          94           0
## 15           1        1        0 101         101           0
## 16           1        1        0 126         126           0
## 17           1        1        0 107         107           0
## 18           1        1        0 104         104           0
## 19           1        1        0 120         120           0
## 20           1        1        0  95          95           0
## 21           1        0        1 108           0         108
## 22           1        0        1  99           0          99
## 23           1        0        1 126           0         126
## 24           1        0        1 119           0         119
## 25           1        0        1 109           0         109
## 26           1        0        1 110           0         110
## 27           1        0        1 105           0         105
## OUTPUT TRUNCATED
```

Model matrix - contr.sum

```
model.matrix(~class.f*iq,data=maths,
contrasts.arg = list(class.f="contr.sum"))

##    (Intercept) class.f1 class.f2  iq class.f1:iq class.f2:iq
## 1            1        1        0  99          99           0
## 2            1        1        0 103         103           0
## 3            1        1        0 108         108           0
## 4            1        1        0 109         109           0
## 5            1        1        0  96          96           0
## 6            1        1        0 104         104           0
## 7            1        1        0  96          96           0
## 8            1        1        0 105         105           0
## 9            1        1        0  94          94           0
## 10           1        1        0  91          91           0
## 11           1        0        1 101           0         101
## 12           1        0        1  95           0          95
## 13           1        0        1 105           0         105
## 14           1        0        1  94           0          94
## 15           1        0        1 101           0         101
## 16           1        0        1 126           0         126
## 17           1        0        1 107           0         107
## 18           1        0        1 104           0         104
## 19           1        0        1 120           0         120
## 20           1        0        1  95           0          95
## 21           1       -1       -1 108        -108        -108
## 22           1       -1       -1  99         -99         -99
## 23           1       -1       -1 126        -126        -126
## 24           1       -1       -1 119        -119        -119
## 25           1       -1       -1 109        -109        -109
## 26           1       -1       -1 110        -110        -110
## 27           1       -1       -1 105        -105        -105
## OUTPUT TRUNCATED
```

Interaction in R

With factors and continous predictors, R uses the `contr` argument to determine the coding for the factors, and the resulting parameters for the factor.

With additional continous predictors, the product of the continous predictors with the factor is included when the * is used in the model formula.