SMIII lectures

Week 3

Symbolic specification of linear models

Consider the model

$$\boldsymbol{Y} = X\boldsymbol{\beta} + \boldsymbol{\mathcal{E}}$$

How does R construct the matrix X?

Multiple regression models in R

When we use a command of the form `lm(y~x, data = df)` where

- `y` is a vector containing the y-values and
- `x` is a vector containing the x-values.

The argument `y ~ x` is called the **model formula**

The term to the left of `~` is the response variables, while the stuff to the right of `~` is used to get the design matrix X.

Example

See `SMC3-model-formula.Rmd`

Multiple predictors

Multiple regression models can be specified by including additional terms in the right hand side of the model formula, separated by the `+` symbol.

Functions

Functions can be applied directly to any terms in the model formula.

- Examples:

```
log(FVC) ~ Height + Weight
log(FVC) ~ log(Height) + Weight
```

`I()` operator

Arithmetic calculations, involving addition, subtraction, multiplication, division can also be incorporated into the model formula but must be enclosed within the `I` (inhibit) function.

The reason is that the operators `+`, `-`, `*`, `/` and `^` have different interpretations in the context of a model formula.

Factors

A categorical predictor variable is called a **factor**.

Example - Poison

In a certain experiment, animals were exposed to one of three different poisons and four different treatments and the survival time recorded for each animal. Four animals were observed at each treatment/poison combination so that the experiment would be described as a 3×4 factorial experiment with replication 4.

See `SM_W3_poisson.Rmd`

Box and Cox(1964) have previously determined that if

$$y = \frac{1}{\log(\text{Survival Time})}$$

then the additive model

$$y_{ijk} = \mu + \alpha_i + \beta_j + e_{ijk}$$

with $i =$ poison, $j =$ treatment and $k =$ replicate, provides an adequate description of the data.

Constraints

The additive formulation shown above cannot be used directly because the parameters α_i and β_j are not identifiable.

- That is, if we tried to construct the model matrix X, the columns would not be linearly independent.
- For this reason, it is necessary to impose certain constraints on the parameters.

Approach 1: zero sum

The classical approach is to impose the side conditions,

$$\sum_{i=1}^{r} \alpha_i = \sum_{j=1}^{s} \beta_j = 0.$$

Approach 2: reference level

For most computing packages (including R) the default parameterisation is to use the **treatment contrasts** defined by

$$\alpha_1 = \beta_1 = 0.$$

It can be checked mathematically that both approaches are equivalent in terms of the way that the model constrains the fitted values η_{ijk}.

Interpretation

For practical purposes it is essential to understand that the interpretation of the numerical parameter estimates depends upon the constraints used.

- [Zero Sum Constraints:] α_i indicates how far the mean for Treatment i lies above or below the average for all treatments.
- [Treatment Contrasts:] α_i how far the mean for Treatment i lies above or below the mean for the reference level, Treatment 1.

Interactions

For models involving two or more factors, it is necessary to consider also **interactions**.

For example, in the poison data, the most general model is

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \mathcal{E}_{ijk}$$

where

$$\alpha_1 = \beta_1 = \gamma_{i1} = \gamma_{1j} = 0.$$

The term γ_{ij} is called the interaction and allows for possible non-additive treatment and poison effects.

Factors in R

The system of model formulae implemented in R provides for the automatic generation of blocks of columns in the model matrix corresponding to individual factors (main effects) and interactions.

Such variables must be declared as factors using the `factor()` function.

The + operator

The addition operator operator is used to add extra terms to a model.

Its action is to concatenate the columns corresponding to the expressions on either side of the `+` operator.

For example, the model

$$\eta_{ij} = \mu + \alpha_i + \beta_j$$

with $\alpha_1 = \beta_1 = 0$ is represented as `~ A + B`.

The + operator - parallel regression

Similarly, the parallel regression model

$$\eta_{ij} = \mu + \alpha_i + \beta x_{ij}$$

with $\alpha_1 = 0$ is represented by `~A+x`.

Finally, the + operator automatically eliminates terms that are algebraically redundant.

For example, `~ A + A` is automatically recognised and reduced to `~ A`

The : operator

The interaction operator, `:`, is used to generate interactions between factors or between factors and covariates.

For example, the two-factor model with interaction

$$\eta_{ij} = \mu + \alpha_i + \beta_j + \gamma_{ij}$$

with $\alpha_1 = \beta_1 = \gamma_{i1} = \gamma_{1j} = 0$ is represented by `~ A + B + A:B`.

The : operator - separate regressions

The separate regressions model,

$$\eta_{ij} = \mu + \alpha_i + \beta x_{ij} + \beta_i x_{ij}$$

with $\alpha_1 = \beta_1 = 0$ is represented by `~A + x + A:x`.

Note that this model allows for regressions with different slopes and intercepts within each group.

- It is parameterised so that β is the slope for Group 1, β_2 is the difference in slopes for Group~2 and Group~1, and β_3 is the difference in slopes for Group 3 and Group 1.
- The hypothesis $H_0 : \beta_2 = \beta_3 = 0$ is therefore the hypothesis of parallel regressions.

In general the `A:B` operation works by pointwise multiplication of each column associated with `A` with each column associated with `B`.

In the preceding example, where `A` and `B` are both factors with three levels, there are two columns associated with each and therefore 4 columns associated with the interaction term.

The same principle can also be seen to apply to the parallel regression example.

The interaction operator can be applied to an arbitrary number of terms.

As with the + operator, the : operator also recognises and eliminates certain redundancies. For example, `~A:A` is automatically reduced to `~A` and `~A:B:A` is reduced to `~A:B`.

The * operator

The interaction operator `:` is not usually used directly as it is laborious to include all main effects and interactions explicitly.

The `*` operator is used to generate those terms automatically.

For example, `A*B` expands to `A + B + A:B` and `A*x` expands to `A + x + A:x`.

The `*` operator can be applied to more than two arguments so, for example,`A*B*C` expands to

$$`A + B + C + A : B + A : C + B : C + A : B : C`.$$

The ^ operator

The exponentiation operator provides a convenient notation for iteration of the `*` operator so, for example,

`(A+B+C)^2` is expanded to `(A+B+C)*(A+B+C)`.

Recalling that, redundant terms are automatically eliminated, this expands to

$$`A + B + C + A : B + A : C + B : C`.$$

The primary purpose of this operator is therefore to specify models including all interactions up to a given order. As a further example,

- `(A+B+C+D)^2` expands to

$$`A + B + C + D + A : B + A : C + A : D + B : C + B : D + C : D`.$$

The - operator

The subtraction operator `-` is used to remove terms already included in the model formula.

The most common applications are:

- To perform regression through the origin by removal of the intercept.
- In particular, the model $\eta_j = \beta x_j$ is specified as `~x - 1`.
- To remove higher order interaction terms. For example, the model of no three factor interaction `A+B+C+A:B+A:C+B:C` can be specified as `A*B*C-A:B:C`.

It should be noted that the `-` operator is interpreted in context as the formula is parsed from left to right, so that `A+B-A+A` is equivalent to `B+A`.

If the term being subtracted is not present then the operation has no effect and does not generate an error.

The / operator

The nesting operator `/` specifies models that are equivalent to the `*` operator but parameterised differently.