Multiple Linear Regression Models

CHRIS SUTTON, MARCH 2024

Topics we have covered so far in this Statistical Modelling module

1. Principles of statistical modelling
2. The Simple Linear Regression Model
3. Least Squares estimation
4. Properties of estimators
5. Assessing the model
6. Inference about the model parameters
7. Matrix approaches to simple linear regression
8. Maximum Likelihood Estimation

Modelling more complex relationships between variables

Simple Linear Regression

Multiple Linear Regression

Simple Linear Regression isn't the end

Whenever we model using Simple Linear Regression some of the variability in the response (y_i) is left unexplained

- R^2 is less than 100%
- there can be a number of different reasons for this
- one reason might be that there is more than one explanatory variable we need to take account of to better understand the response
- this leads to Multiple Linear Regression models

2 explanatory variables

- with 2 explanatory variables X_1 and X_2 and a response variable Y

  $i = 1, 2, \ldots, n$ observations of the form (x_{1i} , x_{2i} , y_i)

- the multiple linear regression model here is

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i$$

More generally

model with p – 1 explanatory variables X_1, X_2, … , X_{p-1}

$$y_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_{p-1} x_{p-1\,i} + \varepsilon_i$$

$var(\varepsilon_i) = \sigma^2$ for all $i = 1, \ldots, n$

$cov(\varepsilon_i, \varepsilon_j) = 0$ for all $i \neq j$

Can also be written as

model with $p - 1$ explanatory variables $X_1, X_2, \ldots, X_{p-1}$

$$E[y_i] = \mu_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_{p-1} x_{p-1\,i}$$

$var(y_i) = \sigma^2$ for all $i = 1, \ldots, n$

$cov(y_i, y_j) = 0$ for all $i \neq j$

This is an equivalent way of writing the same multiple linear regression model

Normal linear regression

We usually have the additional assumption of the normal distribution

Can be written as

$y_i \sim N(\mu_i, \sigma^2)$

or

$\varepsilon_i \sim N(0, \sigma^2)$

Matrix form

$$\mathbf{Y} = \mathbf{X}\,\boldsymbol{\beta} + \varepsilon$$

$\boldsymbol{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ the vector of responses

$\boldsymbol{X} = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{p-1,1} \\ \vdots & \vdots & \dots & \vdots \\ 1 & x_{1,n} & \dots & x_{p-1n} \end{pmatrix}$ the *design matrix*

$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_{p-1} \end{pmatrix}$ the vector of parameters

$\varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}$ the vector of random errors

3 reasons we might be interested in multiple linear regression

- A
  - Improve a simple linear regression model
- B
  - We know there is a multi-variable relationship
- C
  - We don’t know which variables are explanatory

Least squares estimation

Algebraically it is easier to work with the matrix form

Here we seek estimates for the elements of vector $\boldsymbol{\beta}$

We will find that the results are very similar to those for the simple linear regression model

Our approach is to minimise the sums of squares of residuals

Sum of squares of residuals

We seek parameter estimates that minimise S($\boldsymbol{\beta}$) where

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_{1i} + \cdots + \beta_{p-1} x_{p-1\ i}))^2$$

Alternatively written as

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{n} \varepsilon_i{}^2$$

Or in vector form

$$S(\boldsymbol{\beta}) = \varepsilon^T \varepsilon$$

Least squares estimators

We know from our matrix work in weeks 5 and 6 that the least squares estimators here are in the form

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

This is the same result as for the simple linear regression model except that then the matrix ***X*** *had 2 columns whereas now the identity matrix* ***X*** *has p columns for p – 1 explanatory variables (and p beta parameters)*

Properties of the least squares estimators

Again these flow from our work on the simple linear regression model

- $\hat{\boldsymbol{\beta}}$ is an *unbiased estimator* for $\boldsymbol{\beta}$
- Var[$\hat{\boldsymbol{\beta}}$] = $\sigma^2 \, (\boldsymbol{X}^T\boldsymbol{X})^{-1}$
- If $\mathbf{Y} = \mathbf{X}\,\boldsymbol{\beta} + \varepsilon$ with $\varepsilon \sim N(0, \sigma^2 \boldsymbol{I})$ then $\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2 \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1})$

Fitted values and hat matrix

In finding the vector of fitted values $\hat{\boldsymbol{Y}}$ we can use the *hat matrix* $\boldsymbol{H}$ where

$$\hat{\boldsymbol{Y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{Y}$$

where

$$\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$$

Residuals in multiple linear regression

$$\boldsymbol{e} = \boldsymbol{Y} - \widehat{\boldsymbol{Y}} = \boldsymbol{Y} - \boldsymbol{HY} = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{Y}$$

With

$$E[\boldsymbol{e}] = 0$$

$$var(\boldsymbol{e}) = \sigma^2(\boldsymbol{I} - \boldsymbol{H})$$

The sum of the elements in $\boldsymbol{e}$ is zero as before

The sum of squares of residuals in matrix form is $\boldsymbol{e}^T\boldsymbol{e} = \boldsymbol{Y}^T(\boldsymbol{I} - \boldsymbol{H})\,\boldsymbol{Y}$

Multiple linear regression in R

We will spend more time on this in the forthcoming IT labs

But again multiple linear regression model building and analysis is a straightforward extension of simple linear regression in R

Response variable observations in vector `y`

If we have four explanatory variables with their observations in vectors

`x1 x2 x3 x4`

Multiple linear regression in R

To construct the multiple linear regression in an R object called `mlrm` (for example) and then display the results

```
mlrm <- lm(y ~ x1 + x2 + x3 + x4)
```

```
summary(mlrm)
```

To calculate the fitted values and store them as `yhat` and the standardised residuals and store them as `d`

```
yhat <- fitted(mlrm)
```

```
d <- rstandard(mlrm)
```