notes w2

Sam Reeves

A Modern Approach to regression with R Ch 2.1

$$Y_i = E(Y|X = x) + \epsilon_i = \beta_0 + \beta_1 x + \epsilon_i$$

where ϵ_i is the random error in Y_i and is such that $E(\epsilon|X) = 0$.

We assume that:

$$Var(Y|X = x) = \sigma^2$$

In practice, we wish to minimize the difference between the actual value of $y(y_i)$ and the predicted value of $y(\hat{y}_i)$. This difference is called the residual, $\hat{\epsilon_i}$, that is:

$$\hat{\epsilon_i} = y_i - \hat{y}_i$$

A very popular method of choosing b 0 and b 1 is called the method of least squares. As the name suggests b 0 and b 1 are chosen to minimize the sum of squared residuals (or residual sum of squares [RSS]),

$$\text{RSS} = \sum_{i=1}^{n} \hat{\epsilon_i}^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

For a minimum we require:

$$\frac{\partial \text{RSS}}{\partial \beta_0} = -2 \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) = 0$$

and

$$\frac{\partial \text{RSS}}{\partial \beta_1} = -2 \sum_{i=1}^{n} x_i (y_i - \beta_0 - \beta_1 x_i) = 0$$

.

Simplifying, we get:

$$\sum_{i=1}^{n} y_i = \beta_0 n + \beta_1 \sum_{i=1}^{n} x_i$$

and

$$\sum_{i=1}^{n} x_i y_i = \beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2$$

.

These last two equations are called normal equations. Solving these equations for β_0 and β_1 gives the soc-called least squares estimates of the intercept:

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

and the slope:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{SXY}{SXX}$$

.

Consider the linear regression model with constant variance:

$$Y_i = \beta_0 + \beta 1 x_i + \epsilon_i (i = 1, 2, ..., n)$$

where the random error ϵ_i has amean 0 variance σ^2. We wish to estimate $\sigma^2 = Var(\epsilon)\ldots$ Notice that:

$$\epsilon_i = Y_i - (\beta_0 + \beta_1 x_i) = Y - ?x_i$$

The residuals in practice can be used to estimate σ^2.

$$S^2 = \frac{\text{RSS}}{n-2} = \frac{1}{n-2}\sum_{i=1}^{n} \hat{e_i}^2$$

Linear Models with R Ch 2

Estimation

A linear model:

$$\begin{aligned} Y &= f(X_1, X_2, X_3) + \epsilon \\ &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon \end{aligned}$$

Matrix Representation

We start with some data where we have a response Y and, say, three predictors, $X_1, X_2, \text{and} X_3$. The data might br presented in tabular form like this:

$$\begin{array}{cccc} y_1 & x_{11} & x_{12} & x_{13} \\ y_2 & x_{21} & x_{22} & x_{23} \\ \ldots & & \ldots & \\ y_n & x_{n1} & x_{n2} & x_{n3} \end{array}$$

where n is the number of observations or cases in the dataset.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_1 x_{i2} + \beta_1 x_{i3} + \epsilon_i i = 1, 2, ..., n$$

.

$$y = X\beta + \epsilon$$

where $y = (y_1, y_2, ..., y_n)^T$, $\epsilon = (\epsilon_1, \epsilon_2, ..., \epsilon_n)^T$, $\beta = ()\beta_0, \beta_1, ..., \beta_n)^T$ and:

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ ... & & ... & \\ 1 & x_{n1} & x_{n2} & x_{n3} \end{pmatrix}$$

A simple model is the null model where there is no predictor and just a mean $y = \mu + \epsilon$:

$$\begin{pmatrix} y1 \\ ... \\ y_n \end{pmatrix} = \begin{pmatrix} 1 \\ ... \\ 1 \end{pmatrix} \mu + \begin{pmatrix} \epsilon_1 1 \\ ... \\ \epsilon_n \end{pmatrix}$$

Least Squares

$$\sum \epsilon_i^2 = \epsilon^T \epsilon = (y - X\beta)^T (y - X\beta)$$

Differentiating with respect to β and setting to sero, we find $\hat{\beta}$:

$$X^T X \hat{\beta} = X^T y$$

These are normal equations... We can derive the same result using the geometrix approach. Now provided $X^T X$ is invertible:

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

$$H = X(X^T X)^{-1} X^T \text{ is called the hat matrix.}$$

The hat matrix is the orthogonal projection of y onto the space spanned by X. H is useful for theoretical manipulations, but you usually do not want to computer it explicitly, as it is an $n \times n$ matrix which could be uncomfortably large for some datasets.

The following useful quantities can now be used represented using H:

$$\hat{y} = Hy = X\hat{\beta}$$
$$\hat{\epsilon} = y - X\hat{\beta} = y - \hat{y} = (I - H)y$$

RSS:

$$\hat{\epsilon}^T \hat{\epsilon} = y^T (I - H)^T (I - H)y = y^T (I - H)y$$

Later we show that the least swuares estimate is the best possible estimate of β when the errors ϵ are uncorrelated and have equal variance, i.e., $Var(\epsilon) = \sigma^2 I$. $\hat{\beta}$ is a vector, its variance is a matrix.

$$\hat{\sigma}^2 = \frac{\hat{\epsilon}^T \hat{\epsilon}}{n-p} = \frac{\text{RSS}}{n-p}$$

n - p is degrees of freedom of the model.

$$se(\hat{\beta}_{i-1}) = \sqrt{(X^T X)^{-1}_{ii}}\hat{\sigma}$$

Calculating β

```
library(faraway)
data(gala, package = 'faraway')

head(gala[,-2])
```

```
##              Species  Area Elevation Nearest Scruz Adjacent
## Baltra            58 25.09       346     0.6   0.6     1.84
## Bartolome         31  1.24       109     0.6  26.3   572.33
## Caldwell           3  0.21       114     2.8  58.7     0.78
## Champion          25  0.10        46     1.9  47.4     0.18
## Coamano            2  0.05        77     1.9   1.9   903.82
## Daphne.Major      18  0.34       119     8.0   8.0     1.84
```

```
lmod <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent, data = gala)
sumary(lmod)
```

```
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  7.068221  19.154198  0.3690 0.7153508
## Area        -0.023938   0.022422 -1.0676 0.2963180
## Elevation    0.319465   0.053663  5.9532 3.823e-06
## Nearest      0.009144   1.054136  0.0087 0.9931506
## Scruz       -0.240524   0.215402 -1.1166 0.2752082
## Adjacent    -0.074805   0.017700 -4.2262 0.0002971
##
## n = 30, p = 6, Residual SE = 60.97519, R-Squared = 0.77
```

$$(X^T X)^{-1} X^T y$$

```
x <- model.matrix( ~ Area + Elevation + Nearest + Scruz + Adjacent, data = gala)
y <- gala$Species

xtxi <- solve(t(x) %*% x)
xtxi %*% t(x) %*% y
```

```
##                      [,1]
## (Intercept)  7.068220709
## Area        -0.023938338
## Elevation    0.319464761
## Nearest      0.009143961
## Scruz       -0.240524230
## Adjacent    -0.074804832
```

orrrr

```
solve(crossprod(x,x), crossprod(x,y))
```

```
##                      [,1]
## (Intercept)  7.068220709
## Area        -0.023938338
## Elevation    0.319464761
## Nearest      0.009143961
## Scruz       -0.240524230
## Adjacent    -0.074804832
```

We can estimate σ^2 or pull it from sumary:

```
lmodsum <- sumary(lmod)
```

```
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  7.068221  19.154198  0.3690 0.7153508
## Area        -0.023938   0.022422 -1.0676 0.2963180
## Elevation    0.319465   0.053663  5.9532 3.823e-06
## Nearest      0.009144   1.054136  0.0087 0.9931506
## Scruz       -0.240524   0.215402 -1.1166 0.2752082
## Adjacent    -0.074805   0.017700 -4.2262 0.0002971
##
## n = 30, p = 6, Residual SE = 60.97519, R-Squared = 0.77
```

```
sqrt(deviance(lmod) / df.residual(lmod))
```

```
## [1] 60.97519
```

```
lmodsum$sigma
```

```
## [1] 60.97519
```

```
xtxi <- lmodsum$cov.unscaled

sqrt(diag(xtxi)) * lmodsum$sigma
```

```
## (Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
## 19.15419782  0.02242235  0.05366280  1.05413595  0.21540225  0.01770019
```

```
lmodsum$coef[,2]
```

```
## (Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
## 19.15419782  0.02242235  0.05366280  1.05413595  0.21540225  0.01770019
```

QR Decomposition

Any design matrix X can be written as:

$$X = Q \begin{pmatrix} R \\ 0 \end{pmatrix} = Q_f R$$

Where Q is an $n \times n$ orthoganal matrix. $Q^T Q = QQ^T = I$ and R is a $p \times p$ upper triangular matrix. 0 is an $(n-p) \times p$ matrix of zeroes while Q_f is the first p columns of Q.