Chapter 11. Multiple linear regression

1. Model and estimation

1.1. Model:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} ... + \beta_k x_{ik} + \varepsilon_i , \qquad i = 1, 2, \ldots, n$$

1.2. Data format

i	***y***	x_1	x_2	…	x_k
1	y_1	x_{11}	x_{12}	…	x_{1k}
2	y_2	x_{21}	x_{22}	…	x_{2k}
…	…	…	…	…	…
n	y_n	x_{n1}	x_{n2}	…	x_{nk}

1.3. Least squares estimation

$$S(\beta_0, \beta_1, ..., \beta_k) = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} \left[y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} ... + \beta_k x_{ik}) \right]^2 \rightarrow \min$$

1.4. An easier approach is to write model (1.1) in matrix notation

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

where

$$y=\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}, \quad X=\begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1k} \\ 1 & x_{21} & x_{22} & \dots & x_{2k} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nk} \end{bmatrix}, \quad \beta=\begin{bmatrix} \beta_0 \\ \beta_1 \\ \dots \\ \beta_k \end{bmatrix}, \quad \varepsilon=\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \dots \\ \varepsilon_n \end{bmatrix}$$

1.5. Based on the matrix notation, the least squares estimator of $\boldsymbol{\beta}$ is:

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

If regressors x_i is highly correlated, the inverse $(\boldsymbol{X}'\boldsymbol{X})^{-1}$ may not exist (i.e., $\boldsymbol{X}'\boldsymbol{X}$ is singular). This is called collinearity problem.

R-code:

```
x1=rep(1,420)            #create a list containing 1, length=420
x=cbind(x1, dbh, htb)    #create a data frame for x
x=matrix(x,ncol=3)       #matrix with 3 columns
xx=t(x)%*%x              #matrix(3,3)
xx.inv=solve(xx)         #inverse of xx, i.e., (X'X)^-1
beta=xx.inv%*%t(x)%*%htt #obtain the estimates of beta
```

1.6. The fitted values are:

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

$\boldsymbol{H}$ is called hat matrix.

R-code:

```
yhat=x%*%xx.inv%*%t(x)*htt      #fitted tree height
```

1.7. Use ***lm***:

hl.lm_lm(htt~dbh+htb,data=hl.dat)

> **summary(hl.lm)** #view outputs

```
Call:
lm(formula = htt ~ dbh + htb, data = hl.dat)

Residuals:
    Min      1Q  Median      3Q     Max
-10.5166  -0.7098  -0.2194   0.5908  14.5024

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.04146    0.12683   8.211 2.66e-15 ***
dbh          0.65954    0.01492  44.198  < 2e-16 ***
htb          0.27073    0.04964   5.454 8.46e-08 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 1.758 on 417 degrees of freedom
Multiple R-Squared: 0.9192,     Adjusted R-squared: 0.9188
F-statistic:  2371 on 2 and 417 degrees of freedom,     p-value:     0
```

2. Hypothesis testing

2.1. The expectation and variance of $\hat{\boldsymbol{\beta}}$:

Unbiased estimator: $\boldsymbol{E}(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$

Covariance: $\boldsymbol{Cov}(\hat{\boldsymbol{\beta}}) = \boldsymbol{\sigma}^2(\boldsymbol{X}'\boldsymbol{X})^{-1}$

The variance of $\hat{\beta}_i$ is the diagonal element, the off-diagonal is the covariance.

Assume $\boldsymbol{C} = (\boldsymbol{X}'\boldsymbol{X})^{-1}$, the variance of $\hat{\beta}_i$ is:

$$V(\hat{\beta}_i) = \sigma^2 C_{ii}$$

The covariance of $\hat{\beta}_i$ and $\hat{\beta}_j$ is:

$$Cov(\hat{\beta}_i, \hat{\beta}_j) = \sigma^2 C_{ii}$$

2.2. Estimation of σ^2:

$$SS_E = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum \varepsilon_i^2 = \boldsymbol{\varepsilon}'\boldsymbol{\varepsilon}$$

or

$$SS_E = \boldsymbol{y}'\boldsymbol{y} - \hat{\boldsymbol{\beta}}'\boldsymbol{X}'\boldsymbol{y}$$

The unbiased estimate of σ^2 is:

$$\hat{\sigma}^2 = \frac{SS_E}{n - \#\ parameters} = \frac{SS_E}{n-(k+1)} = MS_E$$

2.3. Test for significance of regression

$H_0 : \beta_1 = \beta_2 = ... = \beta_k = 0$

$H_1 : \beta_i \neq 0$ for at least one *i*

Variation partition:

$$\sum (y_i - \bar{y})^2 = \sum (\hat{y}_i - \bar{y})^2 + \sum (y_i - \hat{y}_i)^2$$

	SS_T	=	SS_R	+	SS_E
df:	*n*-1		*k*		*n*-(*k*+*1*)

F-test:

$$F_0 = \frac{SS_R / k}{SS_E /(n-k-1)} = \frac{MS_R}{MS_E}$$

Reject H_0 is $F_0 > F_{\alpha,k,n-k-1}$, or p-value = $P(F_{k,\, n\text{-}k\text{-}1} > F_0)$. # small p-value is evidence against H_0.

(Note: R does not have outputs for this test. It is more informative to test for individual regression coefficients)

2.3. Test for individual regression coefs:

Including an extra regressor will always increase the regression sum of squares and reduce the residual sum of squares. Testing for individual coef will help determine if the benefit is sufficient to include that regressor. This is an important step in model building.

$$H_0 : \beta_i = 0$$
$$H_1 : \beta_i \neq 0$$

Under H_0 hypothesis, the test statistic is the same as in simple linear regression:

$$t_0 = \frac{\hat{\beta}_i}{\sqrt{MS_E C_{ii}}} \sim t_{n-k-1}$$

p-value = $P(t_{n\text{-}k\text{-}1} > t_0)$

Small p-value is against H_0.

(Note: To remember, this test is a test of the contribution of x_i given the other regressors already in the model. So it is really a partial or marginal test because the coef $\hat{\beta}_i$ depends on other regressors in the model.)

2.4. Extra-sum-of-squares to test $H_0 : \beta_2 = 0$ *vs* $H_1 : \beta_2 \neq 0$

Full model:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$$

Reduced model:

$$y_r = \beta_0 + \beta_1 x_1 + \varepsilon$$

To test $H_0 : \beta_2 = 0$, the regression sum of squares between the full model and the reduced model:

$$SS_R(\beta_2 \mid \beta_1) = SS_{Rf}(\beta_1, \beta_2 \mid \beta_0) - SS_{Rr}(\beta_1 \mid \beta_0)$$

This sum of squares is called the extra-sum-of-squares due to β_2 because it measures the increase in the regression sum of squares that results from adding the regressor x_2.

The test statistics:

$$F_0 = \frac{SSR(\beta_2 \mid \beta_1)/r}{MS_E} \sim F_{r,n-k-1}$$

p-value = $P(F_{r,\, n\text{-}k\text{-}1} > F_0)$. Small *p*-value is against H_0.

Compare the differences in the outputs of the ***R-codes***:

(1). **hl.lm1=lm(htt~dbh+htb)**
anova.lm(hl.lm1)

(2). **hl.lm2=lm(htt~htb+dbh)**
anova.lm(hl.lm2)

(Note: Here we are assessing the value of adding x_2 to a model that has not included x_2 yet. It is helpful to think this measure as the contribution of x_2 as if it were the *last*

variable added to the model. Therefore, the second SS_R is the extra contribution of that regressor given that x_1 is already in the model.)

3. Indicator variables

1.1. Parallel lines – x_2 is an indicator variable equaling 0 or 1:

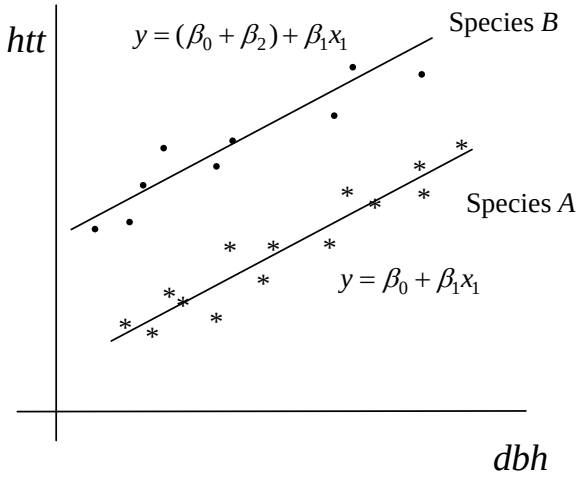
Model: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$

Species A ($x_2 = 0$):

$$y = \beta_0 + \beta_1 x_1 + \varepsilon$$

Species B ($x_2 = 1$):

$$y = (\beta_0 + \beta_2) + \beta_1 x_1 + \varepsilon$$



(Note: The procedure is virtually the same as fitting two regression lines to the two data. The advantages are: (1) there is only one model to work on, (2) a common error variance σ^2, and (3) more residual degrees of freedom)

1.2. Nonparallele lines – Model tree heights for Hemlock and Cedar species

Model: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \varepsilon$

(Note the interaction term for nonparallele lines)

Hemlock ($x_2 = 0$):

$$y = \beta_0 + \beta_1 x_1 + \varepsilon$$

Cedar ($x_2 = 1$):

$$y = (\beta_0 + \beta_2) + (\beta_1 + \beta_3) x_1 + \varepsilon$$

1.3. Testing two species have the same intercept: H_0: $\beta_2 = 0$.

Using the extra-sum-of-squares technique:

$$SS_R(\beta_2 \mid \beta_1, \beta_3) = SS_R(\beta_1, \beta_2, \beta_3 \mid \beta_0) - SS_R(\beta_1, \beta_3 \mid \beta_0)$$

Test statistic:

$$F_0 = \frac{SS_R(\beta_2 \mid \beta_1, \beta_3)/1}{MS_E}$$

1.4. R-code:

lm(htt~dbh+htb+sp+dbh*sp+htb*sp, data=hlcd.dat)

Exercise: Compare the outputs with two separate analyses:

lm(htt~dbh+htb, data=hl.dat)

and

lm(htt~dbh+htb, data=cd.dat)

4. Assessing adequacy of the model

Follow the same procedure as for the simple linear regression model using **plot.lm(**.lm)**

5. Model building and variable selection

5.1. Goal: To build an adequate but parsimonious model

5.2. Many criteria can be used to select variables, e.g., R^2, F statistic, etc.

5.3. Akaike Information Criterion (AIC) is one of the best

AIC = -2 maximized log-likelihood + 2 # parameters

AIC = $n \log(SS_E/n) + 2k + \text{const}$

(Note: It is desirable to have a small AIC. Large k will lead to small SS_E, but if trivial regressors are included, it will increase AIC. Thus, AIC trades off between SS_E and the number of parameters.)

Some programs use Mallow's C_p as a basis to select variables. C_p is closely related to AIC in the way as:

AIC = $SS_E\ (C_p + 1)$

5.4. R-code: **step**

(1). Start from the simplest model, only with constant:

step.lm=lm(htt~1,data=hlcd.dat)

(2). Add all terms:

step.lm=step(step.lm,~dbh+htb+sp+dbh*sp+htb*sp,data=hlcd.dat)

6. Special topics

6.1. **lowess** is a useful technique for EDA to detect patterns

plot(hldbh, hlhtt)

lines(lowess(hldbh,hlhtt),col=2)

Another function is: **loess**.

6.2. Reasons why regression coefs have the "wrong" sign:

(1). The range of some of the regressors is too small (this increases the variance of $\hat{\beta}$).

(2). Important regressors have not been included in the model:

x_1	x_2	Y
2	1	1
4	2	5
5	2	3
6	4	8
8	4	5
10	4	3
11	6	10
13	6	7

y versus x_1:

$$\hat{y} = 1.835 + 0.463x_1$$

y versus x_1 and x_2:

$$\hat{y} = 1.036 - 1.222x_1 + 3.649x_2$$

The sign for x_1 is reserved. The reason is that $\beta_1 = -1.222$ in the multiple model is a "partial" regression coef. which measures the effect of x_1 given that x_2 is also in the model.

(3). Multicollinearity is present.

(4). Computation errors have been made.