STA 108 Discussion 8: Multiple linear regression

Yuanyuan Li

Reference: Textbook Chapter 5, 6, 7.

1. Prerequisite: Some algebra (Chapter 5)

1.1 Matrix Multiplication

If A is an $m \times n$ matrix and B is an $n \times p$ matrix, the matrix product $C = AB$ (denoted without multiplication signs or dots) is defined to be the $m \times p$ matrix. The entry c_{ij} of the product is obtained by multiplying term-by-term the entries of the ith row of A and the jth column of B, and summing these n products.

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}_{2\times 3} \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}_{3\times 2} = \begin{pmatrix} 14 & 32 \\ 32 & 77 \end{pmatrix}_{2\times 2}$$

```
A=matrix(1:6, nrow=2, ncol=3,byrow = T)
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
```

```
B=matrix(1:6, nrow=3, ncol=2)
B
```

```
##      [,1] [,2]
## [1,]    1    4
## [2,]    2    5
## [3,]    3    6
```

```
A %*% B
```

```
##      [,1] [,2]
## [1,]   14   32
## [2,]   32   77
```

If $p = 1$, the product C is a $p \times 1$column vector:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}_{2\times 3} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}_{3\times 1} = \begin{pmatrix} 14 \\ 32 \end{pmatrix}_{2\times 1}$$

```r
A=matrix(1:6, nrow=2, ncol=3,byrow = T)
B= c(1,2,3)
A %*% B
```

```
##      [,1]
## [1,]   14
## [2,]   32
```

If $m = 1$ and $p = 1$, the matrix product becomes the dot(inner) product of two vectors:

$$\begin{pmatrix}1 & 2 & 3\end{pmatrix}_{1\times 3}\begin{pmatrix}1\\2\\3\end{pmatrix}_{3\times 1} = 14$$

```r
t(c(1,2,3)) %*% c(1,2,3)# t() is the transpose function
```

```
##      [,1]
## [1,]   14
```

If $n = 1$, the product of a column vector with row vector is a matrix:

$$\begin{pmatrix}1\\2\\3\end{pmatrix}_{3\times 1}\begin{pmatrix}1 & 2 & 3\end{pmatrix}_{1\times 3} = \begin{pmatrix}1 & 2 & 3\\2 & 4 & 6\\3 & 6 & 9\end{pmatrix}_{3\times 3}$$

```r
c(1,2,3) %*% t(c(1,2,3))
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    2    4    6
## [3,]    3    6    9
```

1.2 Matrix Inverse

The calculation of matrix inverse can be very intensive, we will use the *solve()* function in R to calculate it.

```r
A=matrix(1:4,2,2)
solve(A)
```

```
##      [,1] [,2]
## [1,]   -2  1.5
## [2,]    1 -0.5
```

2. Express multiple linear regression model

Consider a linear regression model with $p-1$ predictor variables $X_1, \cdots, X_p$. The model can be written as **Option 1**:

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \cdots + \beta_{p-1} X_{i,p-1} + \epsilon_i, \ i = 1, \cdots, n$$

Option 2(Matrix Expression):

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}_{n\times 1} = \begin{pmatrix} 1 & X_{1,1} & \cdots & X_{1,p-1} \\ 1 & X_{2,1} & \cdots & X_{2,p-1} \\ \vdots & \vdots & & \vdots \\ 1 & X_{n,1} & \cdots & X_{n,p-1} \end{pmatrix}_{n\times p} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{pmatrix}_{p\times 1} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}_{n\times 1}$$

$$\iff Y = X\beta + \epsilon$$

where Y is the nth vector of response, X is the $n \times p$ matrix of constants, β is the pth vector of parameters, ϵ is a nth vector of independent normal random variables with expectation $E(\epsilon) = \mathbf{0}$ and variance-covariance matrix

$$\text{Var}(\epsilon) = \Sigma = \begin{pmatrix} \sigma^2 & 0 & 0 & 0 \\ 0 & \sigma^2 & 0 & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{pmatrix}_{n\times n} = \sigma^2 I_{n\times n}$$

3. Estimation and Inference

Least squares (LS) estimate: Find β that minimizes $||Y - X\beta||^2 = \sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 X_{i,1} - \beta_2 X_{i,2} + \cdots - \beta_p X_{i,p-1})^2$, the solution is $\hat{\beta}$ as below, along with other estimators related to the model estimation:

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$
$$\hat{Y} = X\hat{\beta}$$
$$MSE = \frac{1}{n-p}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

```
iris1 = iris[,1:3]  # "iris" is a built-in dataset
colnames(iris1) = c("y","x1","x2")
fit = lm(y~x1+x2, data=iris1)
X= model.matrix(fit)
head(X)
```

```
##   (Intercept)  x1  x2
## 1           1 3.5 1.4
## 2           1 3.0 1.4
## 3           1 3.2 1.3
## 4           1 3.1 1.5
## 5           1 3.6 1.4
## 6           1 3.9 1.7
```

```
n=nrow(X)
p=ncol(X)
Y = iris1$y
beta=fit$coefficients #or solve(t(X) %*% X) %*% t(X) %*% Y
beta
```

```
## (Intercept)          x1          x2
##   2.2491402   0.5955247   0.4719200
```

```
MSE= summary(fit)$sigma^2
```

3.1 Inferences about β_k

Test for $H_0 : \beta_k = \beta_{k0}$ v.s. $H_a : \beta_k \neq \beta_{k0}$. The test statistic is

$$t = \frac{\hat{\beta}_k - \beta_{k0}}{\text{s.e.}(\hat{\beta}_k)}$$

$$\text{s.e.}(\hat{\beta}_k) = \sqrt{MSE(X'X)^{-1}_{k+1,k+1}},\ 0 \leq k \leq p-1,$$

where $(X'X)^{-1}_{k+1,k+1}$ is the $k+1$th diagonal element of $(X'X)^{-1}$, where the index starts from 1.

Under H_0, $t \sim t_{n-p}$. The critical value is $t_{1-\alpha/2;n-p}$.

The $(1-\alpha)\%$ confidence interval for β_k is $\hat{\beta}_k \pm t_{1-\alpha/2;n-p}\text{s.e.}(\hat{\beta}_k)$.

```
k=1# inference on beta_1
beta_k= beta[k+1]#index always starts from 1
xx_inverse= solve(t(X) %*% X) #p*p matrix
se_betak= sqrt(MSE*diag(xx_inverse)[k+1])#index always starts from 1
se_betak
```

```
##         x1
## 0.06932816
```

Or you can find s.e.$(\hat{\beta}_k)$ in "Std. Error" column, and the t value and p-value for testing $H_0 : \beta_k = 0$ v.s. $H_a : \beta_k \neq 0$ in last two columns of the "summary" output.

```
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = iris1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.96159 -0.23489  0.00077  0.21453  0.78557
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.24914    0.24797    9.07 7.04e-16 ***
```

```
## x1           0.59552    0.06933    8.59 1.16e-14 ***
## x2           0.47192    0.01712   27.57  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3333 on 147 degrees of freedom
## Multiple R-squared:  0.8402, Adjusted R-squared:  0.838
## F-statistic: 386.4 on 2 and 147 DF,  p-value: < 2.2e-16
```

```
se_betak= summary(fit)$coefficients[k+1,"Std. Error"]
null_value=0#test \beta_k=0
t_stat=(beta_k-null_value)/se_betak
t_stat
```

```
##      x1
## 8.58994
```

```
alpha=0.05
t_crit=qt(1-alpha/2,df=n-p)
t_stat>t_crit#test conclusion
```

```
##   x1
## TRUE
```

```
library(knitr)
kable(data.frame(lb=beta_k-t_crit*se_betak,ub=beta_k+t_crit*se_betak))#CI
```

	lb	ub
x1	0.4585161	0.7325334

3.2 Estimation of Mean Response and Prediction of New Observation

$$\hat{Y}_h = x_h'\hat{\beta}$$
$$\text{s.e.}(\hat{Y}_h) = \sqrt{MSE(x_h'(X'X)^{-1}x_h)}$$
$$\text{p.s.e.}(\hat{Y}_h) = \sqrt{MSE(1 + x_h'(X'X)^{-1}x_h)}$$

```
xh = c(1, 3, 1.5)#given a new X_h with x1=3, x2=1.5
yh = t(xh) %*% beta
se_yh =sqrt(MSE* t(xh) %*% xx_inverse %*% xh)
pse_yh= sqrt(MSE* (1+ t(xh) %*% xx_inverse %*% xh))
kable(data.frame(terms=c("CI for E(Yh)","PI for Yh"),
                 lb=c(yh-t_crit*se_yh,yh-t_crit*pse_yh),
                 ub=c(yh+t_crit*se_yh, yh+t_crit*pse_yh)))
```

terms	lb	ub
CI for E(Yh)	4.647144	4.840045
PI for Yh	4.077918	5.409271

The `predict` function also works for multiple linear regression model:

```
kable(predict(fit, newdata=data.frame(x1=3, x2=1.5),
              interval="confidence",level = 0.95))#Confidence interval
```

fit	lwr	upr
4.743595	4.647144	4.840045

```
kable(predict(fit, newdata=data.frame(x1=3, x2=1.5),
              interval="predict",level = 0.95))#prediction interval
```

fit	lwr	upr
4.743595	4.077918	5.409271

3.3 Simultaneous Inference

Same as the simultaneous inference for simple linear regression (discussion 7), with s.e.$(\hat{Y}_h)$ and p.s.e.$(\hat{Y}_h)$ calculated based on multiple linear regression model, and 2 is replaced by p in calculating "df" of multiples.

4. Anova and Extra SS

4.1 ANOVA table

$$\text{SSTO} = \text{SSE} + \text{SSR},$$

$$\text{Total Sum of Squares:} \quad \text{SSTO} = \sum (Y_i - \bar{Y})^2, \ df = n-1,$$

$$\text{Error Sum of Squares:} \quad \text{SSE} = \sum (Y_i - \hat{Y}_i)^2, \ df = n-p, \ \text{MSE} = \frac{SSE}{n-p}$$

$$\text{Regression Sum of Squares:} \quad \text{SSR} = \sum (\hat{Y}_i - \bar{Y})^2, \ df = p-1, \ \text{MSR} = \frac{SSR}{p-1}$$

```
#Compute manually:
y_hat = fit$fitted.values
SSTO = sum((Y-mean(Y))^2)
SSE = sum((Y-y_hat)^2)
SSR = sum((y_hat-mean(Y))^2)
MSR = SSR/(p-1)
MSE = SSE/(n-p)
F_stat=MSR/MSE
result=data.frame(Source=c("Regression", "Error", "Total"),
```

```
                  Df=c(p-1, n-p,n-1), SS=c(SSR, SSE, SSTO),
                  MS=c(MSR, MSE,NA), F_value=c(F_stat,NA,NA))
kable(result)
```

Source	Df	SS	MS	F_value
Regression	2	85.83957	42.91978	386.3862
Error	147	16.32876	0.11108	
Total	149	102.16833		

F Test for Regression Relation

To test whether there is a regression relation between the response variable Y and the set of X variables $X_1, \cdots, X_{p-1}$, i.e., to choose between the alternatives:

$$H_0 : \beta_1 = \cdots = \beta_{p-1} = 0$$
$$H_a : \text{not all } \beta_k (k = 1, \cdots, p-1) \text{ equal zero}$$

we use the test statistic:

$$F = \frac{MSR}{MSE}$$

Decision rule: If $F > F_{1-\alpha;\ p-1, n-p}$, reject H_0 and conlude the existence of a regression relation.

```
qf(1-alpha, p-1,n-p)#Critical value
```

```
## [1] 3.057621
```

```
1-pf(F_stat, p-1,n-p)#p-value
```

```
## [1] 0
```

4.2 ANOVA table with Extra SS(Chapter 7)

An extra sum of squares involves the difference between the two corresponding regression sums of squares, e.g.,

$$SSR(X_2|X_1) = SSR(X_1, X_2) - SSR(X_1) = SSE(X_1) - SSE(X_1, X_2), df = 1,$$
$$MSR(X_2|X_1) = \frac{SSR(X_2|X_1)}{df}$$

Then we get

$$SSTO = SSR(X_1) + SSE(X_1) = SSR(X_1) + SSR(X_2|X_1) + SSE(X_1, X_2)$$

The *anova* function in R gives the ANOVA table including extra SS for adding each predictor conditioning on existed predictors. For example, "Sum Sq" below denotes $SSR(X_1)$, $SSR(X_2|X_1)$, $SSE(X_1, X_2)$, hence the sum of this column is SSTO.

```
kable(anova(fit))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
x1	1	1.412238	1.412238	12.71369	0.0004902
x2	1	84.427332	84.427332	760.05861	0.0000000
Residuals	147	16.328764	0.111080		

Generally, for any extra SS, we can apply the reduced model/full model trick to calculate it.

$$SSR(X_3, X_4|X_1, X_2) = SSE(X_1, X_2) - SSE(X_1, X_2, X_3, X_4),\ df = 2,$$
$$MSR(X_3, X_4|X_1, X_2) = \frac{SSR(X_3, X_4|X_1, X_2)}{df}$$

Using CDI data as an example and *anova* function, the SSE under reduced/full model can be found in the column "RSS" with dfs($= df_R, df_F$) in the column "Res.Df", the extra SS can be found in the column "Sum of Sq", i.e., $SSR(V_6, V_7|V_4, V_5)$ with df($= df_R - df_F$) in the column "Df" in the table below.

```
setwd("~/books/108s21/UCDSTA108-master/datasets")#set working directory to "datasets" folder
CDI = read.table("CDI.txt")
reduced=lm(V8~V4+V5, data=CDI)
full= lm(V8~V4+V5+V6+V7, data=CDI)
kable(anova(reduced,full))
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
437	152633499				
435	144686245	2	7947254	11.94673	8.9e-06

Uses of Extra Sums of Squares in Tests for Regression Coefficients

Based on Extra SS, we can test whether several $\beta_k = 0$. In multiple regression we are frequently interested in whether several terms in the regression model can be dropped. For example, we may wish to know whether both X_3 and X_4 can be dropped from the full model with X_1, X_2, X_3 and X_4.

$$H_0 : \beta_3 = \beta_4 = 0$$
$$H_a : \text{not both } \beta_3 \text{ and } \beta_4 \text{ equal zero}$$

The statistic is

$$F = \frac{MSR(X_3, X_4|X_1, X_2)}{MSE(X_1, X_2, X_3, X_4)} = \frac{SSR(X_3, X_4|X_1, X_2)/2}{SSE(X_1, X_2, X_3, X_4)/435} = \frac{(SSE_R - SSE_F)/(df_R - df_F)}{SSE_F/df_F}$$

The F value can be found in previous ANOVA table. The p-value for this test is $P(F_{df_R - df_F,\ df_F} > F)$. The critical value for a given α is $F_{1-\alpha;\ df_R - df_F,\ df_F}$.

```r
Fv= anova(reduced,full)$'F'[2]
qf(1-alpha, 2, 435)#Critical value
```

```
## [1] 3.016458
```

```r
1-pf(Fv, 2, 435)#p-value
```

```
## [1] 8.893672e-06
```

- *Question*: Could you use the reduced/full model trick to do **F Test for Regression Relation**?

Yes, you can! Let the reduced model be $Y = \beta_0 + \epsilon_{ij}$, and full model be $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_{p-1} X_{p-1} + \epsilon_{ij}$,

$$F = \frac{(SSE_R - SSE_F)/(df_R - df_F)}{SSE_F/df_F} = \frac{(SSTO - SSE_F)/(n-1-(n-p))}{SSE_F/n-p} = \frac{MSR_F}{MSE_F}.$$

4.3 Coefficients of Partial Determination

Recall that the coefficient of multiple determination, R^2, measures the proportionate reduction in the variation of Y achieved by the introduction of the entire set of X variables considered in the model. A coefficient of partial determination, in contrast, measures the marginal contribution of one X variable when all others are already included in the model.

$$R^2_{Y3|12} = \frac{SSR(X_3|X_1, X_2)}{SSE(X_1, X_2)}$$

Coefficients of Partial Correlation: $(r_{Y3|12})^2 = R^2_{Y3|12}$.

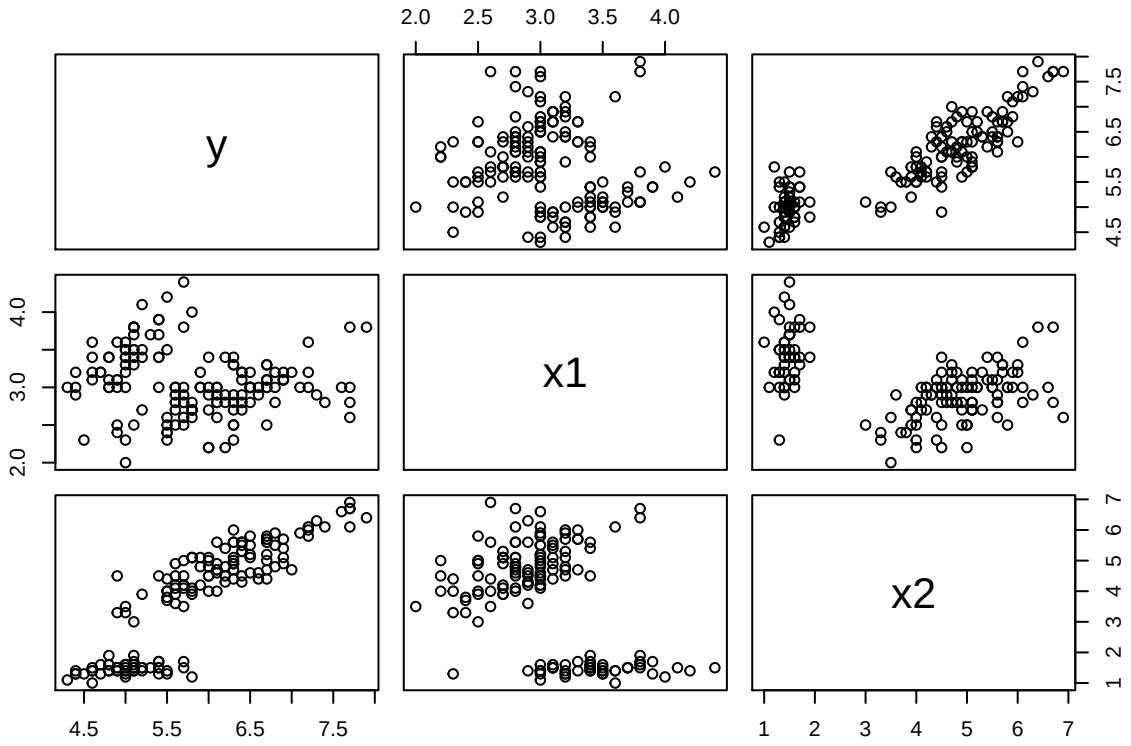
5. Diagnosis

The residual plots and most diagnosis tests for multiple linear regression are similar to those for simple linear regression, we just add some special methods for multiple linear regression in this part. We continue to use "iris1" data.

5.1 Scatterplot matrix

Scatter plots of the response variable against each predictor variable can aid in determining the nature and strength of the bivariate relationships between each of the predictor variables and the response variable and in identifying gaps in the data points as well as outlying data points. Scatter plots of each predictor variable against each of the other predictor variables are helpful for studying the bivariate relationships among the predictor variables and for finding gaps and detecting outliers.

```r
pairs(iris1)
```



5.2 Correlation matrix

A complement to the scatter plot matrix that may be useful at times is the correlation matrix. This matrix contains the coefficients of simple correlation r_{Y1}, r_{Y2}, $\ldots$, $r_{Y,p-1}$ between Y and each of the predictor variables, as well as all of the coefficients of simple correlation among the predictor variables: r_{12} between X_1 and X_2.

```
cor(iris1)
```

```
##             y         x1         x2
## y   1.0000000 -0.1175698  0.8717538
## x1 -0.1175698  1.0000000 -0.4284401
## x2  0.8717538 -0.4284401  1.0000000
```

5.3 F-test for lack-of-fit

The lack of fit F test described in Chapter 3 for simple linear regression can be carried over to test whether the multiple regression response function is an appropriate response surface. Repeat observations in multiple regression are replicate observations on Y corresponding to levels of each of the X variables that are constant from trial to trial. Thus, with two predictor variables, repeat observations require that X_1 and X_2 each remain at given levels from trial to trial.

$$H_0 : E(Y) = \beta_0 + \beta_1 X_1 + \cdots \beta_{p-1} X_{p-1}$$
$$H_a : E(Y) \neq \beta_0 + \beta_1 X + \cdots \beta_{p-1} X_{p-1}$$

Under H_0, we get the reduced model with $df_R = n - p$. While under H_a, we get the full model $df_F = n - c$, where c is the the number of groups with distinct sets of levels for the X variables. The appropriate test statistic is:

$$F = \frac{(SSE_R - SSE_F)/(df_R - df_F)}{SSE_F/df_F}.$$

The p-value for this test is $P(F_{df_R - df_F,\ df_F} > F)$. The critical value for a given α is $F_{1-\alpha;\ df_R - df_F,\ df_F}$.

```
reduce.model = lm(y~x1+x2, data=iris1)
full.model = lm(y~0+as.factor(x1):as.factor(x2), data=iris1)
kable(anova(reduce.model, full.model))
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
147	16.328764				
28	1.636667	119	14.6921	2.112198	0.0121407

Explanation for the dfs:

```
Xs=cbind(iris1$x1, iris1$x2); Xlevels= unique(Xs)
c = nrow(Xlevels)
df_r= n-p;df_f= n-c
c(df_r,df_f)
```

```
## [1] 147  28
```