MTH 4230 R Notes 5

1 Multiple Regression (Cont'd)

1.1 Predicted Values

- We can compute the ***predicted value*** $\hat{Y}_h$ of a new individual response $Y_{h(New)}$, for a given **set** of values $X_1, X_2, \ldots, X_{p-1}$ of the $p-1$ predictors using the `predict()` function.
- For example, consider the following data:

```
x1 <- c(22,15,7,19,20,9,15,10,19,21)
x2 <- c(100, 101, 114, 123, 119, 104, 117, 119, 121, 118)
x3 <- c(4, 7, 11, 12, 7, 19, 23, 4, 17, 15)
y <- c(12.2,9.5,6.7,5.9,10.0,8.9,11.5,10.0,9.9,10.1)
```

We fit the multiple regression model with three predictors:

```
my.reg <- lm(y ~ x1 + x2 + x3)
```

Then to get the ***predicted value*** $\hat{Y}_h$ for when the values of the three predictors are $X_1 = 18, X_2 = 111$, and $X_3 = 13$, first create a data frame containing these values:

```
x.new <- data.frame(x1 = 14, x2 = 111, x3 = 13)
```

then type:

```
predict(my.reg, newdata = x.new)

##        1
## 9.471889
```

Thus the ***predicted value*** is $\hat{\boldsymbol{Y}}_{\boldsymbol{h}} = \mathbf{9.471889}$

1.2 Confidence Interval for a Mean Response and Prediction Interval

- We can use `predict()` to compute a ***confidence interval for a mean response*** $E(Y)$ for a given **set** of predictor values $X_1, X_2, \ldots, X_{p-1}$ by specifying `interval = "confidence"`.
- For example, to get a ***95% confidence interval for the mean response*** $E(Y)$ when the values of the three predictors are $X_1 = 18, X_2 = 111$, and $X_3 = 13$, first create the data frame `x.new` from above. Then type:

```
predict(my.reg, newdata = x.new, interval = "confidence")

##        fit      lwr      upr
## 1 9.471889 7.770421 11.17336
```

The ***confidence interval for the mean response*** is **(7.770421, 11.173357)**.

1.3 Confidence Interval for a Mean Response and Prediction Interval

- To compute a ***prediction interval*** for a new response $Y_{h(New)}$, for a given *set* of values $X_1, X_2, \ldots, X_{p-1}$ of the $p-1$ predictors, we specify `interval = "prediction"` in `predict()`.
- For example, to get a ***95% prediction interval*** for a new individual response $Y_{h(New)}$ when $X_1 = 18, X_2 = 111$, and $X_3 = 13$, first create the data frame `x.new` from above. Then type:

```
predict(my.reg, newdata = x.new, interval = "prediction")

##        fit      lwr      upr
## 1 9.471889 4.342207 14.60157
```

The ***prediction interval*** is **(4.342207, 14.601572)**.

1.4 Extra Sums of Squares and Partial *F* Tests

- To compute ***extra sums of squares*** in R, we use the `anova()` function.
- Consider again, for example, the *lm* object `my.reg` from above (using the vectors `x1`, `x2`, `x3`, and `y` also from above):

```
my.reg <- lm(y ~ x1 + x2 + x3)
```

We now pass `my.reg` to `anova()`:

```
anova(my.reg)

## Analysis of Variance Table
##
## Response: y
##           Df  Sum Sq Mean Sq F value Pr(>F)
## x1         1  4.9200  4.9200  1.2579 0.3049
## x2         1  4.6583  4.6583  1.1910 0.3170
## x3         1  0.4147  0.4147  0.1060 0.7558
## Residuals  6 23.4680  3.9113
```

The output above shows the ***extra sums of squares*** in **sequential order**, i.e. as X_1, X_2, and X_3 are added to the model **in that order**. In other words, it gives:

- $\mathbf{SSR}(\boldsymbol{X_1}) = \mathbf{4.92}$ with **1** ***degree of freedom***.
- $\mathbf{SSR}(\boldsymbol{X_2}|\boldsymbol{X_1}) = \mathbf{4.6583}$ with **1** ***degree of freedom***.
- $\mathbf{SSR}(\boldsymbol{X_3}|\boldsymbol{X_1}, \boldsymbol{X_2}) = \mathbf{0.4147}$ with **1** ***degree of freedom***.
- The `residual sum of squares` is the ***error sum of squares*** **SSE** for the **full model** containing **all three predictors** X_1, X_2, and X_3, i.e. $\mathbf{SSE}(\boldsymbol{X_1}, \boldsymbol{X_2}, \boldsymbol{X_3}) = \mathbf{23.468}$, and it has $\boldsymbol{n - p} = \mathbf{6}$ ***degree of freedom***.

In each case, the ***mean square*** is the sum of squares divided by its degrees of freedom.

The ***partial F test statistics*** (which are also ***general linear F test statistics***) are for **sequentially** testing whether a predictor X_k should be added to a model that **already** includes $X_1, X_2, \ldots, X_{k-1}$. In the output above:

- $\boldsymbol{F} = \frac{\mathbf{SSR}(\boldsymbol{X_1})/\mathbf{1}}{\mathbf{SSE}(\boldsymbol{X_1},\boldsymbol{X_2},\boldsymbol{X_3})/(\boldsymbol{n-4})} = \mathbf{1.2579}$, with **1** and **6** ***degree of freedom***, and the ***p-value*** is **0.3049**.
- $\boldsymbol{F} = \frac{\mathbf{SSR}(\boldsymbol{X_2}|\boldsymbol{X_1})/\mathbf{1}}{\mathbf{SSE}(\boldsymbol{X_1},\boldsymbol{X_2},\boldsymbol{X_3})/(\boldsymbol{n-4})} = \mathbf{1.191}$, with **1** and **6** ***degree of freedom***, and the ***p-value*** is **0.317**.
- $\boldsymbol{F} = \frac{\mathbf{SSR}(\boldsymbol{X_3}|\boldsymbol{X_1},\boldsymbol{X_2})/\mathbf{1}}{\mathbf{SSE}(\boldsymbol{X_1},\boldsymbol{X_2},\boldsymbol{X_3})/(\boldsymbol{n-4})} = \mathbf{0.106}$, with **1** and **6** ***degree of freedom***, and the ***p-value*** is **0.7558**.

1.4.1 An Alternative Approach to Sequential Sums of Squares

- Another way to get **sequential *extra sums of squares*** is by successively fitting models using `lm()`, each of which contains one more predictor than the last:

```
my.reg1 <- lm(y ~ 1)
my.reg2 <- lm(y ~ x1)
my.reg3 <- lm(y ~ x1 + x2)
my.reg4 <- lm(y ~ x1 + x2 + x3)
```

Now passing all these *lm* objects to `anova()` gives the following:

```
anova(my.reg1, my.reg2, my.reg3, my.reg4)

## Analysis of Variance Table
##
## Model 1: y ~ 1
## Model 2: y ~ x1
## Model 3: y ~ x1 + x2
## Model 4: y ~ x1 + x2 + x3
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1      9 33.461
## 2      8 28.541  1    4.9200 1.2579 0.3049
## 3      7 23.883  1    4.6583 1.1910 0.3170
## 4      6 23.468  1    0.4147 0.1060 0.7558
```

From the output, we get the ***error sums of squares***:

- $\mathbf{SSE}(\boldsymbol{X_1}) = \mathbf{28.541}$
- $\mathbf{SSE}(\boldsymbol{X_1}, \boldsymbol{X_2}) = \mathbf{23.883}$
- $\mathbf{SSE}(\boldsymbol{X_1}, \boldsymbol{X_2}, \boldsymbol{X_3}) = \mathbf{23.468}$

and also the (sequential) ***extra sums of squares*** are

- $\mathbf{SSR}(\boldsymbol{X_1}) = \mathbf{4.9200}$
- $\mathbf{SSR}(\boldsymbol{X_2}|\boldsymbol{X_1}) = \mathbf{4.6583}$
- $\mathbf{SSR}(\boldsymbol{X_3}|\boldsymbol{X_1}, \boldsymbol{X_2}) = \mathbf{0.4147}$

and the partial F test statistics are

- $\boldsymbol{F} = \frac{\mathbf{SSR}(\boldsymbol{X_1})/\mathbf{1}}{\mathbf{SSE}(\boldsymbol{X_1},\boldsymbol{X_2},\boldsymbol{X_3})/(\boldsymbol{n-4})} = \mathbf{1.2579}$
- $\boldsymbol{F} = \frac{\mathbf{SSR}(\boldsymbol{X_2}|\boldsymbol{X_1})/\mathbf{1}}{\mathbf{SSE}(\boldsymbol{X_1},\boldsymbol{X_2},\boldsymbol{X_3})/(\boldsymbol{n-4})} = \mathbf{1.19109}$
- $\boldsymbol{F} = \frac{\mathbf{SSR}(\boldsymbol{X_3}|\boldsymbol{X_1},\boldsymbol{X_2})/\mathbf{1}}{\mathbf{SSE}(\boldsymbol{X_1},\boldsymbol{X_2},\boldsymbol{X_3})/(\boldsymbol{n-4})} = \mathbf{0.1060}$

as before. Notice also that

$$\mathbf{SSTO} = \mathbf{33.461}$$

(the SSE from fitting a model with just an intercept).

1.5 General Linear F Test in Multiple Linear Regression

- *Partial F tests* can be viewed as ***general linear F tests***, in which a decision is made between a **full model** and a **reduced model**.

1.5.1 General Linear F Test for a Single β_k

- We use the *partial F test* approach from above to carry out a ***general linear F test*** to decide if a *single* $\beta_k = 0$.

  Define the ***full model*** (with all predictors) as

  $$Y_i = \beta_0 + \beta_1 X_{i1} + \ldots + \beta_{p-1} X_{i,p-1} + \epsilon_i$$

  and the ***reduced model*** (with the kth predictor omitted) as

  $$Y_i = \beta_0 + \beta_1 X_{i1} + \ldots + \beta_{k-1} X_{i,k-1} + \beta_{k+1} X_{i,k+1} + \ldots + \beta_{p-1} X_{i,p-1} \epsilon_i$$

- To carry out the test using `anova()`, we need to be careful to enter X_k **last** in the call to `lm()`.

- For example, with $p - 1 = 3$ predictors X_1, X_2, and X_3, suppose we want to test whether it's useful to add X_2 to the model that already includes X_1 and X_3.

  This means we want to test

$$\begin{aligned} H_0 &: \quad \beta_2 = 0 \\ H_a &: \quad \beta_2 \neq 0 \end{aligned}$$

  To carry out this ***general linear F test*** using the *partial F test* approach from above, we type:

```
my.reg <- lm(y ~ x1 + x3 + x2)                          # x2 is last
anova(my.reg)

## Analysis of Variance Table
##
## Response: y
##           Df  Sum Sq Mean Sq F value Pr(>F)
## x1         1  4.9200  4.9200  1.2579 0.3049
## x3         1  0.0041  0.0041  0.0010 0.9752
## x2         1  5.0689  5.0689  1.2960 0.2984
## Residuals  6 23.4680  3.9113
```

  Notice `x2` was entered into the model **last** in the *formula* supplied to `lm()`. From the output,

$$\boldsymbol{F = \frac{\mathbf{SSR}(X_2|X_1, X_3)/1}{\mathbf{SSE}(X_1, X_2, X_3)/(n-4)} = 1.296}$$

  with **1** and **6** ***degree of freedom***, and the ***p-value*** is **0.2984**.

1.5.2 An Alternative Approach to the General Linear F Test for a Single β_k

- Another way to carry out the ***general linear F test*** (which is also the *partial F test*) for a single β_k is by fitting models with and without X_k using `lm()` and then passing both *lm* objects to `anova()`.
- For example, to test

$$\begin{aligned} H_0 &: \quad \beta_2 = 0 \\ H_a &: \quad \beta_2 \neq 0 \end{aligned}$$

  we type:

```
my.reg.reduced <- lm(y ~ x1 + x3)                       # x2 is left out
my.reg.full <- lm(y ~ x1 + x2 + x3)                     # x2 is included
anova(my.reg.reduced, my.reg.full)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ x1 + x3
## Model 2: y ~ x1 + x2 + x3
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1      7 28.537
## 2      6 23.468  1    5.0689 1.296 0.2984
```

From the output, we get the test statistic $\boldsymbol{F = 1.2960}$, with 1 and 6 ***degrees of freedom***, and the ***p-value*** is **0.2984**, as in the previous section.

1.6 Partial F Test for Several β_k's

- The ***partial F test*** approach of Section 1.4 can be used to decide if a **group** of predictors should be included in a model.
- Suppose, for example, we have $p - 1 = 3$ predictors X_1, X_2, and X_3 and want to test whether X_2 and X_3 should be included in the model that already includes X_1, i.e. we want to test

$$\begin{aligned} H_0 &: \quad \beta_2 = \beta_3 = 0 \\ H_a &: \quad \beta_2 \text{ and } \beta_3 \text{ aren't both equal to } 0 \end{aligned}$$

We fit a ***full model*** (with all three predictors) and a ***reduced model*** (with both X_2 and X_3 omitted) using `lm()`, and then pass the two *lm* objects to `anova()`:

```
my.reg.reduced <- lm(y ~ x1)                    # x2 and x3 are left out
my.reg.full <- lm(y ~ x1 + x2 + x3)
anova(my.reg.reduced, my.reg.full)

## Analysis of Variance Table
##
## Model 1: y ~ x1
## Model 2: y ~ x1 + x2 + x3
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1      8 28.541
## 2      6 23.468  2     5.073 0.6485 0.5559
```

From the output above,

- $\boldsymbol{\text{SSE}(X_1) = 28.541}$
- $\boldsymbol{\text{SSE}(X_1, X_2, X_3) = 23.4681}$.

Also, the extra sum of squares associated with adding X_2 and X_3 to the model is

$$\boldsymbol{\text{SSR}(X_2, X_3 | X_1) = 5.073},$$

and the partial F test statistic is

$$\boldsymbol{F = \frac{\mathbf{SSR}(X_2, X_3|X_1)/2}{\mathbf{SSE}(X_1, X_2, X_3)/(n-4)} = 0.6485}$$

and the ***p-value*** (from the F distribution with **2** and **6** ***degrees of freedom***) is **0.5559**.

1.7 Lack of Fit Test

- To carry out a ***lack of fit F test*** for a *multiple regression model*, we view the test as a *general linear F test* with ***full model***

$$Y_i = \mu_i + \epsilon_i$$

and ***reduced model***

$$Y_i = \beta_0 + \beta_1 X_i + \ldots + \beta_{p-1} X_{p-1} + \epsilon_i$$

These models are specified in R as `y~factor(x1)*factor(x2)*...*factor(xpminus1)` and `y~x1 + x2 + ... + xpminus1`, respectively.

- For example, suppose we have the following data:

```
x1 <- c(10, 10, 10, 10, 20, 20, 20, 20, 10, 10, 10, 10, 20, 20, 20, 20)
x2 <- c(100, 102, 104, 106, 100, 102, 104, 106, 100, 102, 104, 106, 100, 102, 104, 106)
x3 <- c(4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5)
y <- c(12.2,9.5,6.7,5.9,10.0,8.9,11.5,10.0, 14.1, 13.4, 10.8, 6.6, 11.2, 7.9, 13.0, 14.0)
```

Below, we fit both models to the data using `lm()`:

```
my.reduced.reg <- lm(y~x1 + x2 + x3)
my.full.reg <- lm(y~factor(x1)*factor(x2)*factor(x3))
```

The ***lack of fit F test*** is the same as a *general linear F test* of

H_0 : the reduced model is sufficient
H_a : the full model is needed

Thus as before, to perform the test to decide between the **full** and **reduced models**, we use `anova()`:

```
anova(my.reduced.reg, my.full.reg)

## Analysis of Variance Table
##
## Model 1: y ~ x1 + x2 + x3
## Model 2: y ~ factor(x1) * factor(x2) * factor(x3)
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     12 86.523
## 2      8 28.405  4    58.118 4.0921 0.04283
```

From the output,

- The ***sum of squares for pure error*** is **SSPE = 28.405** (which is also **SSE(F)**) with **8** ***degrees of freedom***.
- The ***sum of squares for lack of fit*** is **SSLF = 86.523 − 28.405 = 58.118** (which is also **SSE(R) − SSE(F)**) with **4** ***degrees of freedom***.
- The ***F statistic*** for the ***lack of fit test*** is $\boldsymbol{F = 4.0921}$ and the ***p-value*** (from the F distribution with **4** and **8** ***degrees of freedom***) is **0.04283**.