EEB 581, Problem Set Eight Solutions
Due 23 March 2006

1 : Write each of these models in matrix form. $\mathbf{y}$ is the same for all models, while the $\mathbf{X}$ and $\mathbf{b}$ change over the models, and we will index these by the model

$$\mathbf{y} = \begin{pmatrix} 236 \\ 266 \\ 301 \\ 306 \\ 230 \\ 97 \\ 84 \\ 120 \\ 82 \\ 118 \end{pmatrix}. \text{ Mod 1: } \mathbf{a}_1 = \begin{pmatrix} a \\ b \end{pmatrix}, \ \mathbf{X}_1 = \begin{pmatrix} 1 & 10 \\ 1 & 12 \\ 1 & 13 \\ 1 & 14 \\ 1 & 10 \\ 1 & 12 \\ 1 & 13 \\ 1 & 20 \\ 1 & 11 \\ 1 & 19 \end{pmatrix} \quad \text{Mod 2:} \quad \mathbf{a}_2 = \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \ \mathbf{X}_2 = \begin{pmatrix} 1 & 10 & 1 \\ 1 & 12 & 1 \\ 1 & 13 & 1 \\ 1 & 14 & 1 \\ 1 & 10 & 1 \\ 1 & 12 & 0 \\ 1 & 13 & 0 \\ 1 & 20 & 0 \\ 1 & 11 & 0 \\ 1 & 19 & 0 \end{pmatrix}.$$

$$\text{Model 3:} \quad \mathbf{a}_2 = \begin{pmatrix} a \\ b \\ d \end{pmatrix}, \ \mathbf{X}_2 = \begin{pmatrix} 1 & 10 & 10 \\ 1 & 12 & 12 \\ 1 & 13 & 13 \\ 1 & 14 & 14 \\ 1 & 10 & 10 \\ 1 & 12 & 0 \\ 1 & 13 & 0 \\ 1 & 20 & 0 \\ 1 & 11 & 0 \\ 1 & 19 & 0 \end{pmatrix}. \quad \text{Model 4:} \quad \mathbf{a}_2 = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}, \ \mathbf{X}_2 = \begin{pmatrix} 1 & 10 & 1 & 10 \\ 1 & 12 & 1 & 12 \\ 1 & 13 & 1 & 13 \\ 1 & 14 & 1 & 14 \\ 1 & 10 & 1 & 10 \\ 1 & 12 & 0 & 0 \\ 1 & 13 & 0 & 0 \\ 1 & 20 & 0 & 0 \\ 1 & 11 & 0 & 0 \\ 1 & 19 & 0 & 0 \end{pmatrix}.$$

2 : For each model compute the OLS estimates of the model parameters, the error sum of squares, the estimated residual variance, and the estimated variance-covariance estimate of the parameters.

Data was entered as

```
y <- matrix(c(236, 266, 301, 306, 230, 97, 84, 120, 82, 118))
x <- matrix(c(10, 12, 13, 14, 10, 12, 13, 20, 11, 19))
s <- matrix(c(1, 1, 1, 1, 1, 0, 0, 0, 0, 0))
```

The **R** code was the same for all models, with only the matrix **X** (note UPPERCASE) varying. **X** was cnstructed by using the **cbnd** command to put the appropriate columns together for the **X** for the model under consideration.

```
N <- 10
OLS <-solve(t(X)%*%X)%*%t(X)%*%y
yhat <-X%*%OLS
SSE <- t(y-yhat)%*%(y-yhat)
errorVar <-SSE/(N-p)
var <- errorVar[1,1]  (why this ?)
Cov <-var*solve(t(X)%*%X)
```

(**why var** $<-$ **errorVar**[1,1]? **R** returns SSE as a 1 x 1 matrix (a scaler), so that when you try to multiply the 1 x 1 with the n x n matrix, **R** does not like this. Using the errorVar[1,1] returns the 1,1 element of the matix errorVar as a scaler)

Model 1: **p <- 2**, **X<-cbind(1,x)**

$$\widehat{b} = \begin{pmatrix} 283.75830 \\ -7.44465 \end{pmatrix}, \quad \text{SSE} = 70534.17, \quad \widehat{\sigma_e^2} = 8816.771, \quad \text{Cov} = \begin{pmatrix} 15486.284 & -1089.89604 \\ -1089.896 & 81.33553 \end{pmatrix}$$

Model 2: **p <- 3**, **X<-cbind(1,x,s)**

$$\widehat{b} = \begin{pmatrix} 3.497101 \\ 6.446860 \\ 188.229952 \end{pmatrix},\ \text{SSE} = 2876.266,\ \widehat{\sigma_e^2} = 410.8952,\ \text{Cov} = \begin{pmatrix} 1198.74199 & -74.437531 & -320.37913 \\ -74.43753 & 4.962502 & 15.88001 \\ -320.37913 & 15.880007 & 215.17409 \end{pmatrix}$$

Model 3: **p <- 3**, **X<-cbind(1,x,s*x)**

$$\widehat{b} = \begin{pmatrix} 38.965761 \\ 4.081647 \\ 15.311894 \end{pmatrix},\ \text{SSE} = 321.2258,\ \widehat{\sigma_e^2} = 45.8894,\ \text{Cov} = \begin{pmatrix} 119.767196 & -7.5167696 & -2.4497530 \\ -7.516770 & 0.5101650 & 0.1153490 \\ -2.449753 & 0.1153490 & 0.1532333 \end{pmatrix}$$

Model 4: **p <- 4**, **X<-cbind(1,x,s,x*s)**

$$\widehat{b} = \begin{pmatrix} 39.1285714 \\ 4.0714286 \\ -0.6910714 \\ 15.3660714 \end{pmatrix},\ \text{SSE} = 321.2,\ \widehat{\sigma_e^2} = 53.5,\ \text{Cov} = \begin{pmatrix} 182.774 & -11.47117 & -182.774 & 11.47117 \\ -11.471 & 0.76474 & 11.471 & -0.76474 \\ -182.774 & 11.47117 & 775.810 & -60.8211 \\ 11.47117 & -0.7647 & -60.8211 & 4.9469 \end{pmatrix}$$

FYI, the data were generated assuming a male slope of 5 ($\widehat{b} = 4.07$) and a female slope of 20 ($\widehat{b} + \widehat{d} = 19.43$)

3 : Given that models (i) – (iii) are subsets of the full model (iv), compute the F statistics, and their resulting p values, that the reduced model has the same fit as the full model.

Recall that

$$\frac{N-p}{p-q}\left(\frac{\text{SS}_{Er}}{\text{SS}_{Ef}} - 1\right) \sim F_{p-q,N-p}$$

Model 1 vs. full model, $N - p = 10 - 6 = 6,\ p - q = 4 - 2 = 2$

$$p = \Pr\left(F_{2,6} \geq \frac{6}{2}\left(\frac{70534.17}{321.1} - 1\right) = 655.9926\right) = 9.43457 \cdot 10^{-8}$$

Model 2 vs. full model, $N - p = 10 - 6 = 6,\ p - q = 4 - 3 = 1$

$$p = \Pr\left(F_{1,6} \geq \frac{6}{1}\left(\frac{2876.266}{321.1} - 1\right) = 47.74524\right) = 0.000454319$$

Model 3 vs. full model, $N - p = 10 - 6 = 6,\ p - q = 4 - 3 = 1$

$$p = \Pr\left(F_{1,6} \geq \frac{6}{1}\left(\frac{321.2258}{321.1} - 1\right) = 0.002350670\right) = 0.9629043$$

4 : Using your estimated variances from (2), perform t-tests for all of the parameters for each of the four models (note the df for each test is $N - p$). Note that these are two-sides tests, with the null hypothesis of a zero value, so that p is the probability of getting a value at least that extreme (positive or negative).

$$p = \Pr\left(t_{N-p} > \frac{|\,\widehat{\beta}_i\,|}{\sqrt{\sigma^2(\widehat{\beta}_i)}}\right) + \Pr\left(t_{N-p} < \frac{-|\,\widehat{\beta}_i\,|}{\sqrt{\sigma^2(\widehat{\beta}_i)}}\right) = 2 \cdot \Pr\left(t_{N-p} < \frac{-|\,\widehat{\beta}_i\,|}{\sqrt{\sigma^2(\widehat{\beta}_i)}}\right)$$

Since we will be using this a lot, let's make it a function, call it pval, that simply lets you just change i, the b value being tested,

```
pval <-function(i,OLS,Cov,df) {
pval <- 2 * pt(-abs(OLS[i,1])/sqrt(Cov[i,i]),df);
```

```
pval }
```

Using this, we find for Model 1 that

```
> pval(1,OLS,Cov,N-p)
[1] 0.05205421
> pval(2,OLS,Cov,N-p)
[1] 0.4330262
```

Model 2

```
> pval(1,OLS,Cov,N-p)
[1] 0.9223782
> pval(2,OLS,Cov,N-p)
[1] 0.02318367
> pval(3,OLS,Cov,N-p)
[1] 4.049889e-06
```

Model 3

```
> pval(1,OLS,Cov,N-p)
[1] 0.009212944
> pval(2,OLS,Cov,N-p)
[1] 0.0007244007
> pval(3,OLS,Cov,N-p)
[1] 1.858407e-09
```

Model 4

```
> pval(1,OLS,Cov,N-p)
[1] 0.02754016
> pval(2,OLS,Cov,N-p)
[1] 0.003482795
> pval(3,OLS,Cov,N-p)
[1] 0.9810103
> pval(4,OLS,Cov,N-p)
[1] 0.0004547192
```