STAT 3340 Assignment 4 Solutions (out of 25 points)

1. The regression model $y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$ was fit to a data set with $n = 24$ observations, and resulted in MSE=16, with $\hat{\beta}$ and $(X'X)^{-1}$ given by:

```
hatbeta= c(1000, 100, -200)
xpxinverse=matrix(c(15,8,7,8,12,5,7,5,20), byrow=T,ncol=3)
```

- 1a) Construct a 95% confidence interval for β_1. (3 points)

```
hatbeta= c(1000, 100, -200)
xpxinverse=matrix(c(15,8,7,8,12,5,7,5,20), byrow=T,ncol=3)
n=24
p=3
MSE=16
alpha=.05
hatbeta[2]+c(-1,1)*qt(1-alpha/2,n-p)*sqrt(MSE)*sqrt(xpxinverse[2,2])
```

```
## [1]  71.18403 128.81597
```

- 1b) Construct simultaneous 90% confidence intervals for β_1 and β_2.

(4 points, 1 point for using the correct adjusted alpha=.05, 1 point for the interval for β_1 [OK to just say to use the interval from 1a], and 2 points for the interval for β_2.)

Here $\alpha = .1$. Because we are making 2 simultaneous intervals, we replace α by $\alpha/2 = .1/2 = .05$, and make two 95% confidence intervals.

The interval for β_1 is the interval in problem 1a. Full marks for just saying that. Or, carrying out the calculation, starting with $\alpha = .1$, and then moving to $\alpha/2$, then using $qt(1 - (\alpha/2)/2), n - p)$, the interval for β_1 is:

```
alpha=.1
alphaover2=alpha/2
hatbeta[2]+c(-1,1)*qt(1-alphaover2/2,n-p)*sqrt(MSE)*sqrt(xpxinverse[2,2])
```

```
## [1]  71.18403 128.81597
```

The interval for β_2 is

```
hatbeta[3]+c(-1,1)*qt(1-alphaover2/2,n-p)*sqrt(MSE)*sqrt(xpxinverse[3,3])
```

```
## [1] -237.2013 -162.7987
```

- 1c) Construct a 99% confidence interval for $\beta_0 - 2\beta_1 + \beta_2$. (3 points)

Let $x_0 = c(1, -2, 1)$, then $x_0'\beta = \beta_0 - 2\beta_1 + \beta_2$. The 99% confidence interval for $x_0'\beta$ is

```
x0=c(1,-2,1)
alpha=.01
t(x0)%*%hatbeta + c(-1,1)*qt(1-alpha/2,n-p)*sqrt(MSE)*sqrt(t(x0)%*%xpxinverse%*%x0)
```

```
## Warning in c(-1, 1) * qt(1 - alpha/2, n - p) * sqrt(MSE) * sqrt(t(x0) %*% : Recycling array of length
##   Use c() or as.vector() instead.
```

```
## Warning in t(x0) %*% hatbeta + c(-1, 1) * qt(1 - alpha/2, n - p) * sqrt(MSE) * : Recycling array of l
##   Use c() or as.vector() instead.
```

```
## [1] 524.0267 675.9733
```

- 1d) Is the value $\beta = c(950, 90, -230)$ contained in the 95% confidence ellipse for β? Calculate the value of $(\hat{\beta} - \beta)^T (X'X)(\hat{\beta} - \beta)$ and compare it to $pMSE\ qf(.95, p, n-p)$. Show your work. (3 points)

```
beta=c(950,90,-230)
LHS=t(hatbeta-beta)%*%solve(xpxinverse)%*%(hatbeta-beta)
RHS=p*MSE*qf(.95,p,n-p)
c(LHS,RHS,ifelse(LHS<RHS,"YES","NO"))
```

```
## [1] "207.92905581638" "147.47841534705" "NO"
```

2

. The following calculates the SSE and MSE for the model $y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \epsilon$, for the "cement" data set.

```
data=read.csv("http://chase.mathstat.dal.ca/~bsmith/stat3340/Data/cement.csv")
  #read data
 X=as.matrix(cbind(rep(1,13),data[,-c(1,2)])) #construct the X matrix
 y=matrix(data[,2],byrow=T,ncol=1)   #construct Y
  n=dim(X)[1]   #number of observations
  k=dim(X)[2]-1 #number of predictor variables in full model

beta=solve(t(X)%*%X)%*%t(X)%*%y  #calculate least squares estimates
SSEfull=t(y)%*%y-t(beta)%*%t(X)%*%y  #get the error SS for full model
MSEfull=SSEfull/(n-k-1)                  #MSE under full model
```

Answer the following questions by computing SSE for the appropriate reduced models using matrix operations, then calculating the F statistic for the partial F test, and calculating the p-value using the "pf" command. (You can check your results using the builtin "anova" function.)

- 2a) Test the hypothesis that $\beta_1 = 0$ (4 points)

```
X2=X[,c(1,3:5)]   #construct the X matrix for the reduced model
r=1
beta2=solve(t(X2)%*%X2)%*%t(X2)%*%y  #calculate least squares estimates
                                     #for the reduced model
SSE2=t(y)%*%y-t(beta2)%*%t(X2)%*%y   #get the error SS for the reduced model
SSE2
```

```
##          [,1]
## [1,] 73.81455
```

```
Fobs=((SSE2-SSEfull)/r)/MSEfull    #observed value of test statistic
Fobs
```

```
##          [,1]
## [1,] 4.337474
```

```
pv=1- pf(Fobs,r,n-k-1)     #p-value for the test
pv
```

```
##            [,1]
## [1,] 0.07082169
```

```
#verify with anova(lm(y~x_1+x_2+x_3+x_4,data=data),lm(y~x_2+x_3+x_4,data=data))
```

- 2b) Test the hypothesis that $\beta_1 = \beta_2 = 0$. (4 points)

```
X2=X[,c(1,4:5)]    #construct the X matrix for the reduced model
r=2
beta2=solve(t(X2)%*%X2)%*%t(X2)%*%y  #calculate least squares estimates
                                     #for the reduced model
SSE2=t(y)%*%y-t(beta2)%*%t(X2)%*%y   #get the error SS for the reduced model
SSE2
```

```
##         [,1]
## [1,] 175.738
```

```
Fobs=((SSE2-SSEfull)/r)/MSEfull    #observed value of test statistic
Fobs
```

```
##          [,1]
## [1,] 10.68656
```

```
pv=1- pf(Fobs,r,n-k-1)     #p-value for the test
pv
```

```
##             [,1]
## [1,] 0.005502501
```

```
#verify with anova(lm(y~x_1+x_2+x_3+x_4,data=data),lm(y~x_3+x_4,data=data))# enter your R commnds here
```

- 2c) Test the hypothesis that $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$. (4 points)

```
X2=X[,c(1)]    #construct the X matrix for the reduced model
r=4
beta2=solve(t(X2)%*%X2)%*%t(X2)%*%y  #calculate least squares estimates
                                     #for the reduced model
SSE2=t(y)%*%y-t(beta2)%*%t(X2)%*%y   #get the error SS for the reduced model
SSE2
```

```
##          [,1]
## [1,] 2715.763
```

```
Fobs=((SSE2-SSEfull)/r)/MSEfull    #observed value of test statistic
Fobs
```

```
##          [,1]
## [1,] 111.4792
```

```
pv=1- pf(Fobs,r,n-k-1)     #p-value for the test
pv
```

```
##              [,1]
## [1,] 4.756182e-07
```

```
#verify with anova(lm(y~x_1+x_2+x_3+x_4,data=data),lm(y~1,data=data))# enter your R commnds here
```