```
#R code: Discussion 6.                          Sta108, Fall 2007, Utts

### Multiple Regression and Extra Sum of Squares

#Example: Grocery Retailer: Problem 6.9
Data = read.table("CH06PR09.txt")
names(Data) = c("Hours","Cases","Costs","Holiday")

#scatterplot matrix for ALL variables in dataset
pairs(Data, pch=19)
   #look for association between:
   #1. response variable and any of predictor variables
   #2. any two predictor variables

#correlation matrix for ALL variables in dataset
cor(Data)
   #gives correlation, r, measure of linear relationship b/w each pair of variables

#fit multiple regression model
Fit = lm(Hours~Cases+Costs+Holiday, data=Data)
summary(Fit)
   #gives: parameter restimates, standard errors, t-statistic with p-value for testing
      #Ho: Beta.k = 0
      #Ha: Beta.k not= 0,  while all other parameters are kept in model
   #also, gives: sqrt(MSE), df, R-Sq, F-statistic with df and p-value for testing
      #Ho: Beta.k=0 for all k=1,2,..,p-1
      #Ha: not all Beta.k are zero
      #(Test for regression relation)

#variance-covariance matrix for vector of parameters, Beta
vcov(Fit)

#95% Confidence Intervals for vector of parameters, Beta
confint(Fit, level=0.95)

#print anova table
anova(Fit)
   #shows decomposition of SSR into Extra Sums of Squares (ESS) (Table 7.3)

#save quantities for future use: SSTO, MSE,
SSTO = sum( anova(Fit)[,2] )
MSE = anova(Fit)[4,3]

#get the cumulated SSR for all predictors, F-statistic, to come up with
#ANOVA table without separation into ESS (Table 6.1)
SSR = sum( anova(Fit)[1:3,2] )
F = (SSR / 3) / MSE   #this F-statistic is also shown in the summary() above

#Estimate mean response, Yhat.h, for given vector Xh
Xh = c(1, 24500, 7.40, 0)
Yhat = t(Xh) %*% Fit$coefficients  #formula (6.55)

#95% Confidence Interval for mean response E{Yhat.h}
s = sqrt( t(Xh) %*% vcov(Fit) %*% Xh )  #formula (6.58)
t = qt(1-0.05/2, 52-4)                  #t-value  t(1-alpha/2, n-p)
c( Yhat - t*s, Yhat + t*s )             #formula (6.59)

#95% Prediction Interval for new observation Yhat.h.new
spred = sqrt( MSE + s^2)               #formula (6.63)
c( Yhat - t*spred, Yhat + t*spred )
```

```
#Diagnostics are done the same way as before.

#Extra Sums of Squares
anova(Fit)
  #SumSq “Cases” is SSR( X1 )
  #SumSq “Costs” is SSR( X2 | X1 )
  #SumSq “Holiday” is SSR( X3 | X1, X2 )
SSR = sum( anova(Fit)[1:3,2] )  #SSR(X1, X2, X3), by summing above three SSR
MSR = SSR / 3                   #MSR(X1, X2, X3) = SSR / df
SSE = anova(Fit)[4,2]           #SSE(X1, X2, X3)
MSE = anova(Fit)[4,3]           #MSE(X1, X2, X3)

#attain alternate decompositions of Extra Sums of Squares:
#get SSR(X3), SSR(X1|X3) and SSR(X2|X1,X3)
Model2 = lm( Hours ~ Holiday+Cases+Costs, data=Data)
anova(Model2)

#to get SSR( X2, X3 | X1 ) = SSE( X1 ) – SSE( X1, X2, X3 ),
#use equation (7.4b). You need:
#run a linear model involving only X1 to obtain SSE( X1 ).
Model3 = lm( Hours ~ Cases, data=Data)
SSE.x1 = anova(Model3)[1,3]
#then calculate needed SSR
SSE.x1 - SSE

#similarly, you can apply all formulae given in section 7.1 by fitting reduced models
#(where you leave out one or two variables), and computing SSR and SSE.

#Gerenal Linear Test
#consider dropping Costs ( X2 ) from the model
#test H0: Beta2 = 0    vs.   Ha: Beta2 not= 0
Reduced = lm( Hours ~ Cases + Holiday, data=Data)
#perform F-test, comparing Reduced and Full models
anova(Reduced, Fit)
```