```r
#R code: Discussion 4.                    Sta108, Fall 2007, Utts

#Lack-of-fit test
#Ho: usual linear model holds
#Ha: population average GPA at ACT level j is mu_j
#consider similar hypotheses to (3.19) in your textbook

Data = read.table("~/Documents/School/Sta108utts/CH01PR19.txt")
names(Data)=c("GPA","ACT")
Reduced = lm( GPA ~ ACT, data = Data )
Full = lm( GPA ~ 0 + as.factor(ACT), data = Data )
anova(Reduced, Full)

#If p-value > 0.05, don't reject the null hypothesis; the model model appears to be
adequate.


#Matrices in R

#Create vectors
vec = c( 1.9, -2.8, 5.6, 0, 3.4, -4.2 ) #concatenate numbers of your choice
vec = -2:5                   #sequence integers in increasing order
vec = 5:-2                   #sequence integers in decreasing order
vec = seq( -3.1, 2.2, by=0.1 )            #sequence numbers with increments=0.1
vec = rep( 1, 8 )                         #repeat {1} value 8 times
vec = rep( c(1,0), 5 )                    #repeat {1,0} string 8 times
vec = c( rep(2,4), 1:3, c(8,1,9) )        #concatenate numbers together
vec = Data$ACT               #assign the column ACT from the dataset to a vector
length(vec)                  #get the length of vector
vec[5]                       #get the 5th element of the vector
vec[2:7]                     #get elements 2 through 7 of the vector
vec
sum(vec)                     #sum all elements of the vector
sum(vec^2)                   #square every element of the vector, sum the resulting vector
M = matrix( 1:6, nrow=2 )                 #create 2x3 matrix with integers 1 through 6
M = matrix( 1:6, ncol=3 )                 #same result
M = matrix( 1:6, ncol=3, byrow=TRUE )     #2x3 matrix is filled row by row
A = matrix(c(2,3,5,4,1,5,7,8), ncol=2)
B = matrix(c(6,9,3,1), nrow=4)
C = matrix(c(3,8,5,2,8,6,1,4), ncol=2)
N = matrix(c(3,8,5,6,9,3,1,2,8), ncol=3) #this matrix is a square 3x3 matrix
X = Data$ACT
Y = Data$GPA
t(M)                         #M': take transpose of M
A + C                        #Matrix Addition
A - C                        #Matrix Subtraction
A %*% M                      #Matrix Multiplication
A * A                        #Elementwise Multiplication
t(B) %*% A
I = diag(5)                  #create Identity matrix of dimensions 5x5
det(N)                       #take determinant of a square matrix
solve(N)                     #N^(-1): take inverse of a square matrix
cbind(A,B,C)                 #concatenate columns together
```

```r
rbind(A,C)                  #concatenate rows together
ncol(Data)                  #get number of columns in dataset
nrow(Data)                  #get number of rows in dataset
dim(Data)                   #get dimentions (rows, columns) in dataset

n=nrow(Data)
cbind(rep(1,n), Data$ACT)   #Create X matrix for linear regression
X=cbind(rep(1,n), X1=Data$ACT)
Y=Data$GPA
#In linear regression, get Beta parameter estimates:
b = solve( t(X) %*% X ) %*% t(X) %*% Y         #b = (X'X)^(-1)X'Y
b
lm(Y ~ 0 + X)      #here, X matrix already contains the column for the intercept term
                   #so, you need to set Beta0=0 to prevent R fitting additional intercept
lm(GPA ~ ACT,data=Data) # Check the result
```