```r
#R Code for discussion 3                        #Sta108, Fall2007, Utts

#Continue ExampleData from class

#read the data into R
Data = read.table("~/Documents/School/Sta108utts/wtheightm.txt", header=TRUE)
Data

#fit the regression model and see summary of the fit
Htwt = lm(Weight ~ Height, data=Data)
summary(Htwt)
  #gives: coefficients estimates, standard errors, t statistics, p-values
  #also gives: sqrt(MSE), df, R-squared

#get 95% CIs for Beta0 and Beta1:
confint(Htwt, level=0.95)
  #level=0.95 is by default, can be removed from above

#get 95% CI for mean response at Height=72
Xh = data.frame(Height=72)
predict(Htwt, Xh, interval="confidence", se.fit=TRUE, level=0.95)
  #se.fit is the standard error estimate, optional, can be removed from above
  #level=0.95 is by default, can be removed from above

#get 95% prediction interval for mean response at Height=72
Xh = data.frame(Height=72)
predict(Htwt, Xh, interval="prediction", se.fit=TRUE, level=0.95)
  #se.fit is the standard error estimate to be printed, optional, can be removed from above
  #level=0.95 is by default, can be removed from above

#consider a CI:  [point estimate] +/- t(1-alpha/2, n-2)*se(point estimate)
#get the multiplier used in the 95% CI, i.e. t-value, i.e. the critical value under t-
distribution
  #at n-2 degrees of freedom with left-tail probability (1-0.05/2)
qt(1-0.05/2, 43-2)

#print ANOVA table
anova(Htwt)
  #gives: df, SSR,SSE, MSR,MSE, F statistic, p-value

#get R-squared
summary(Htwt)$r.squared

#get correlation coefficient, r
cor(Data)
  #In the case of this example, the 1st column of Data is a column of text: "Male"
  #for this reason, the above code results in an error
  #to modify, we need to exclude the first column, and use only columns 2 and 3
  #the Square brackets after Data will access the elements of the table Data
  #try
  Data[1,2]  #to access the element in Row=1, Column=2
  Data[1,]  #to access the ALL elements of Row=1, Column=ALL
  Data[,2]  #to access the ALL elements of Column=2, Row=ALL
  Data[,c(1,3)]  #to access the ALL elements of Column=(1 and 3), Row=ALL
  Data[,c(2,3)]  #to access the ALL elements of Column=(2 and 3), Row=ALL
  Data[,2:3]  #identical to above, to access the ALL elements of Column=(2 and 3), Row=ALL
              #here, notation ":" is used to sequence integers from 2 to 3
              #try typing a command like    1:10
```

```r
cor(Data[,2:3])
  #r is the off-diagonal value
  #also, r=sqrt(R-squared)




##########################################
#Diagnostics
#Refer to pages 102-114 of your textbook (Section 3.2, 3.3) for departures and diagnostics

#Departures:
#1. Regression function is not linear
#2. Error terms do not have constant variance
#3. Error terms are not independent
#4. Model fits all but one or few outlying observations
#5. Error terms are not normally distributed
#6. One or more important predictor variables have been omitted from the model

#You will use:
Htwt$residuals
Htwt$fitted.values

#stem-and-leaf plot of residuals  => Departure #5
stem(Htwt$residuals, scale=2)
  #scale is optional, tells how to group the leafs

#boxplot of residuals  => Departure #5
boxplot(Htwt$residuals, ylab="residuals", pch=19)
  #ylab is to label y-axis
  #pch=19 is to plot the outlying observaitons as filled circles

#histogram of residuals  => Departure #5
hist(Htwt$residuals, xlab="residuals", main="Histogram of residuals")
  #xlab is to label x-axis
  #main is to create a proper title of the plot

#plot residuals against predictor X=Height  => Departure #1,2,4, somewhat 3,6
plot(Data$Height, Htwt$residuals, main="Residuals vs. Predictor", xlab="Height",
ylab="Residuals", pch=19)
abline(h=0)     #adds the reference line, horizontal line at y=0

#plot residuals against fitted values Y-hat-h  => Departure #1,2,4, somewhat 3,6
plot(Htwt$fitted.values, Htwt$residuals, main="Residuals vs. Predictor", xlab="Fitted values",
ylab="Residuals", pch=19)
abline(h=0)     #adds the horizontal line at y=0

#normal probability plot, or QQ-plot  => Departure #5
qqnorm(Htwt$residuals, main="Normal Probability Plot", pch=19)
qqline(Htwt$residuals)      #adds the reference line through first and third quartiles

#Departure #3 (and somewhat #6) are studied by a Sequence plot of residuals
#where residuals are plotted against the time order
#Sequence plot is NOT appropriate for this data, where observations are not taken over time, or
in a sequence
```

```r
########################################
#Transformations

#Reference Sections 3.8, 3.9 in your textbook

#Transformation to Y, response variable, are useful to treat Departures #2,5
#Transformation to X, predictor variable, are useful to treat Departures #1

#create square response variable: Y^2, add it to the Data, title/name it "Wt.squared"
Data = cbind(Data, Wt.squared = Data$Weight^2)
    #The names are completely user defined, consider: "Yprime", "Y.prime", "Weight.Sq", and so
on

#take a square-root of response variable: sqrt(Y)
Data = cbind(Data, sqrt.Wt = sqrt(Data$Weight))

#take a natural logarithm of response variable: log{base e}(Y), aka: ln(Y)
Data = cbind(Data, log.Wt = log(Data$Weight))

#take a common logarithm of response variable: log{base 10}(Y)
Data = cbind(Data, log10.Wt = log10(Data$Weight))

#take a resiprocal of response variable: 1/Y
Data = cbind(Data, recip.Wt = 1/Data$Weight)

#take a reciprocal square-root of response variable: 1/sqrt(Y)
Data = cbind(Data, recip.sqrt.Wt = 1/sqrt(Data$Weight))

#Boxcox procedure: consider to transform: Y' = Y^(lambda)
library(MASS)   #to load the package into R which has a proper boxcox function
boxcox(Htwt)    #shows the ideal value of lambda (by dashed line)
boxcox(Htwt, lambda = seq(0, 1, 0.1))   #redefines the location around lambda
  #Choose lambda = 0.75
Data = cbind(Data, Wt.prime = Data$Weight^0.75)
NewModel = lm(Wt.prime ~ Height, data=Data)
   #Now, go through the diagnostics again

#Consider transforming the predictor variable: X=Height
#transformations to X are done the similar way as to Y:
#create square predictor variable: X^2, add it to the Data, name it "Ht.squared"
Data = cbind(Data, Ht.squared = Data$Height^2)
  #Remember that the names are user defined
```