Math 5364

Data Mining 2

Homework 23

Mary Barker

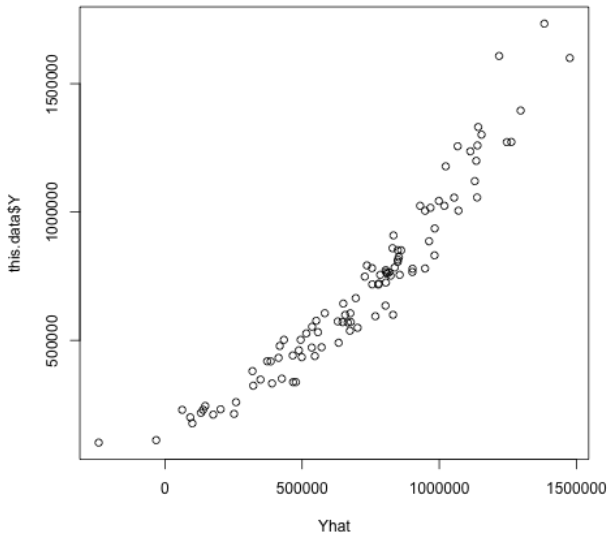
1. Import the file math5305Lab6Data.txt, whose columns are the variables Y, X1, X2, and X3. The goal of these first three problems is to perform diagnostics to assess the assumptions of normality and constancy of variabce for a model predicting Y from the Xj's, to transform Y if necessary, to assess the transformed model using diagnostics, adn to compare the original and transformed models via residual sums of squares.

   We begin by fitting a model and assessing it with diagnostics.

   (a) Fit model = lm(Y X1+X2+X3), compute the fitted values Yhat, and the residuals e. You can use the command Yhat = predict(model) to obtain Yhat.

   (b) Plot Y vs Yhat. If the model were valid, what would you expect this plot to look like? Does the plot suggest the existence of curvature in the model?

   If this were a valid model, then the predictions $\hat{Y}$ would be roughly equal to the Y values. Therefore the plot of $\hat{Y}$ vs Y for a valid model should be a diagonal line.

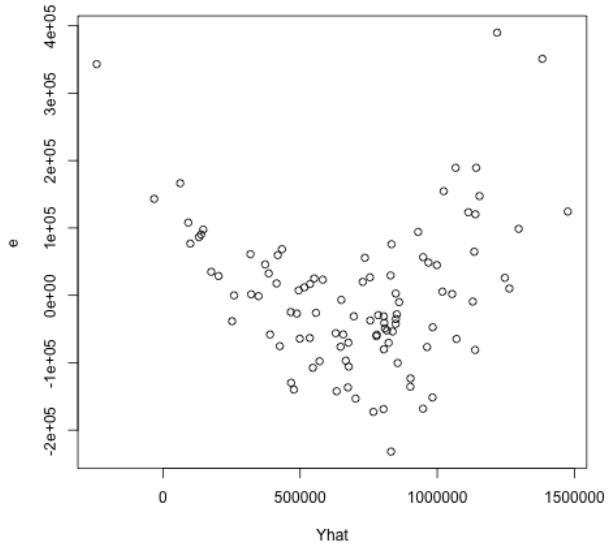


The plot does have a mainly diagonal line, but the curvature suggests that the model is not accurately representing some relationship in the predictor variables.

(c) Plot e vs Yhat. If the model were valid, what would you expect this plot to look like? Does the plot suggest the existence of curvature in the model?

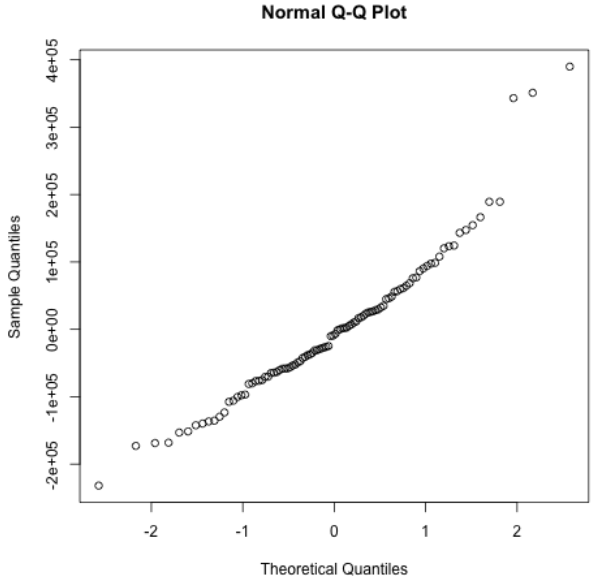
The plot for a valid model should be a banded straight line. The residuals should be randomly distributed about 0.

This is very much not the case for the example plotted below. There seems to be some curvature that is not being picked up by the model.



(d) Now that we know curvature is present, there are two courses of action we can take: transform Y or transform the Xj's, or both. Generally, if there are problems with the errors, we should transform Y, and if the errors are ok, we should transform the Xj's. Let's investigate the errors.

(e) Plot a qq-plot to check normality of the error terms using the qqnorm command.



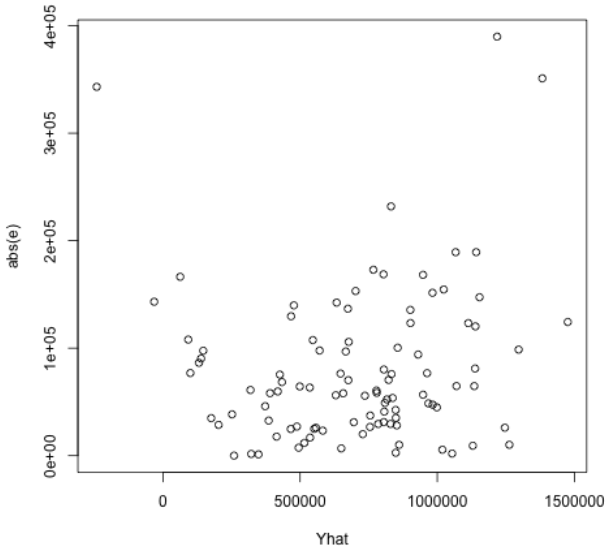
(f) Perform the Shapiro-Wilks test to check normality of the error terms using the shapiro.test command.

Using the Shapiro-Wilks test gave a p-value of 0.0002678

(g) Based on the results in parts (e) and (f), do the error terms for this model appear to be normal?

No. Both the plot and the results of the test give a strong indication that the errors are not normally distributed.

(h) Check constancy of error variance by plotting |e| vs. Yhat.



(i) Check constancy of error variance by performing the Brown-Forsythe test.

The output of this test was: Test Statistic = 1.841, p-value = 0.1779

(j) Based on the results in parts (h) and (i), do the error terms for this model appear to have constant variance?

No. The Brown-Forsythe test does not have a small p-value, which means that the error terms might have constant variance, but the plot does not support such an assumption.

(k) Does a transformation of Y appear to be necessary?

Yes.

(l) Finally, calculate the residual sum of squares $\|e\|^2$. Note that this value is $\|e\|^2 = \sum(Y_i - \hat{Y}_i)^2, i = 1, ..., n$ so it is similar to a prediciton sum of squares. It measures the sum of square errors between the predictions Yhati and the actual observations Yi. This number is very large, so to put it in perspective, calculate

$$\frac{\|e\|^2}{\|Y - \bar{Y}\|^2} \tag{1}$$

Assessing the model by this criterion is equivalent to using

$$R^2 = 1 - \frac{\|e\|^2}{\|Y - \bar{Y}\|^2} \tag{2}$$

The computed sum of squares error is 1.148277e+12

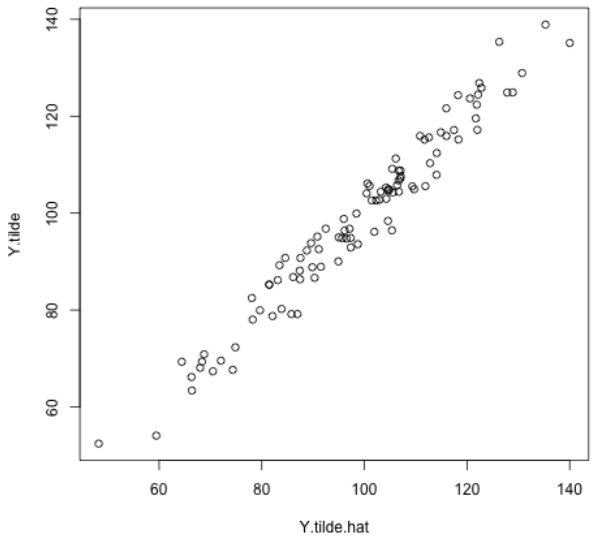
1 - R2 = 0.09342021

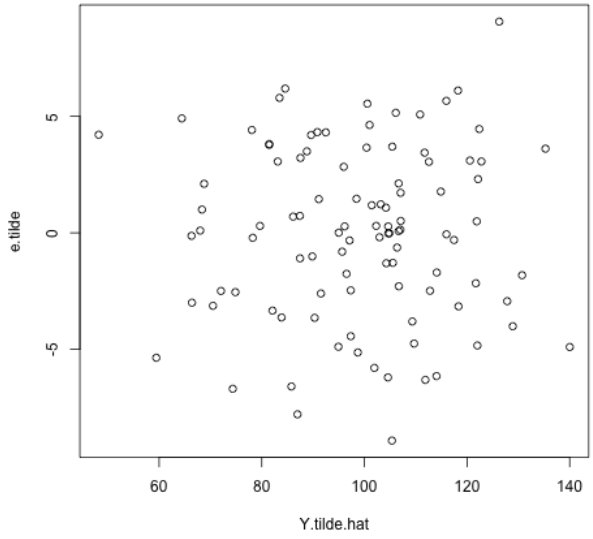
R2 = 0.9065798

2. Let lambda be the optimal value produced by the Box-Cox transformation. Transform Y by defining Y.tilde.i = (Y.i)^lambda for i = 1, ... , 100.

   (a) Fit a model tmodel by regressing y.tilde on X1, X2, and X3, and find the corresponding fitted values Y.tilde.hat and e.tilde

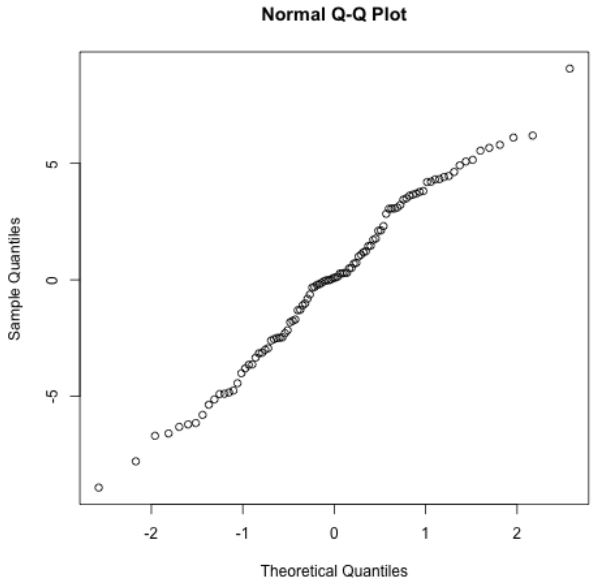
   (b) Plot Y.tilde vs Y.tilde.hat and e.tilde vs. Y.tilde.hat. How do these plots compare to those from problem 2? Does curvature appear to exist in the transformed model?





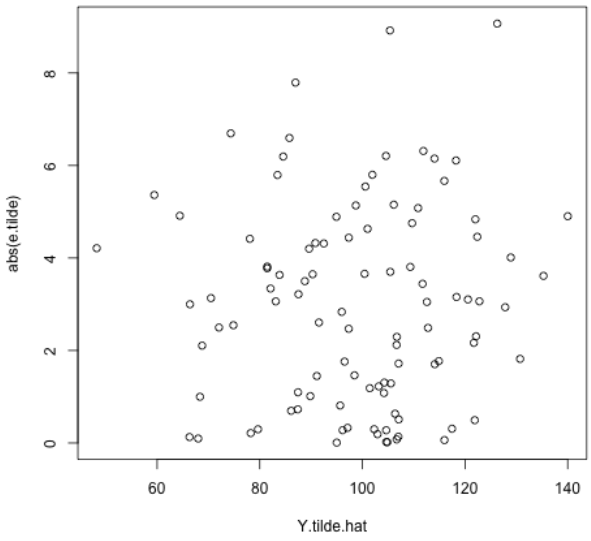
There does not appear to be any curvature in the transformed model. The plot of $\tilde{Y}$ and $\hat{\tilde{Y}}$ is a good indication that the predicted values are close to the actual values. There is no curvature also, and the residual plot shows random behavior.

(c) Investigate normality of the errors for the transformed model.



The Shapiro-Wilks test gave a p-value of 0.3862

(d) Investigate constancy of error variance for the transformed mode.



The results for The Brown-Forsythe test are Test Statistic = 0.15792, p-value = 0.6919.

(e) Do the errors for the transformed model appear to satisfy the assumptions of normality and constant error variance? How do your results compare to Those from problem 2?

The Shapiro-Wilks test indicates that the errors are not normal, however, looking at the QQ plot, this might be due to the outlier at the upper end. This test is very sensitive, and the plot shows very normal behavior except for a few outliers near the top of the plot that might skew the results.

The Brown-Forsythe test again supports the assumption that the errors have constant variance. This is also supported be the plot of the errors this time, which do display a very constant variance.

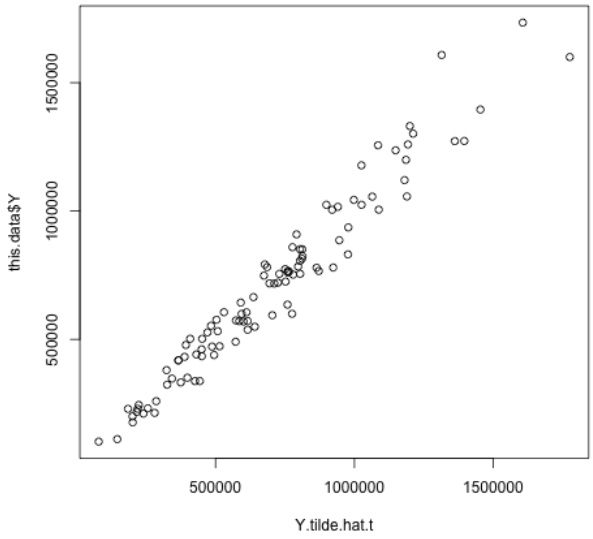
3. Now let's apply the results from the transformed model to the original variable Y.

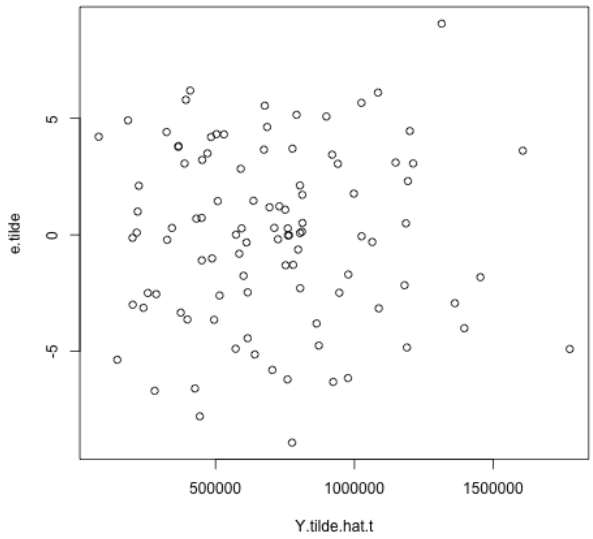
(a) First, create a vector of fitted values for Y by defining

```
Yhat_i = (Y_tilde_hat_i)^(1 / lambda)
```

for i = 1, ... , 100, and create a vector of residuals by defining e.i = Yi - Yhati, for i = 1, ..., 100. These are predicted values and residuals for the original model, but they take advantage of the information from the transformed model.

(b) Plot Y vs. Yhat and e vs. Yhat. Did the transformation appear to correct problems with the functional form?





The transformation does seem to have fixed the problems with the original model.

(c) Finally, calculate

```
||e||2, ||e||2 / (||Y - Ybar||2)
```

and

```
R2 = 1 - ||e||2 / (||Y - Ybar||2)
```

as in question 2. Which model fits the data better/has a lower residual sum of squares?

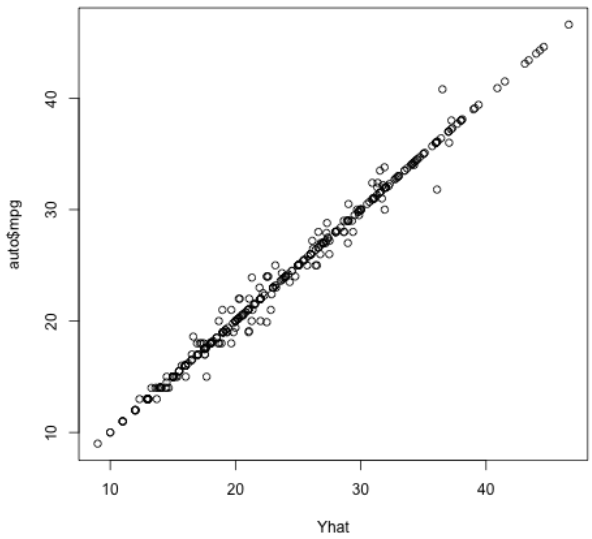
The sum of squares error for this model is 1355.552.

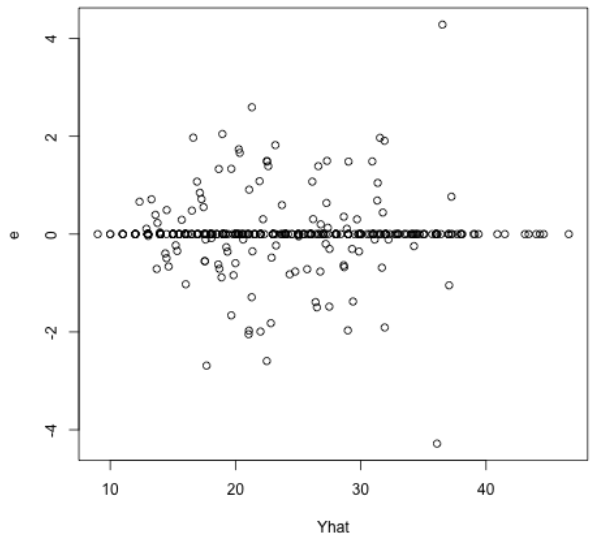
1 - R^2 = 1.162408e-10

$R^2 = 1$

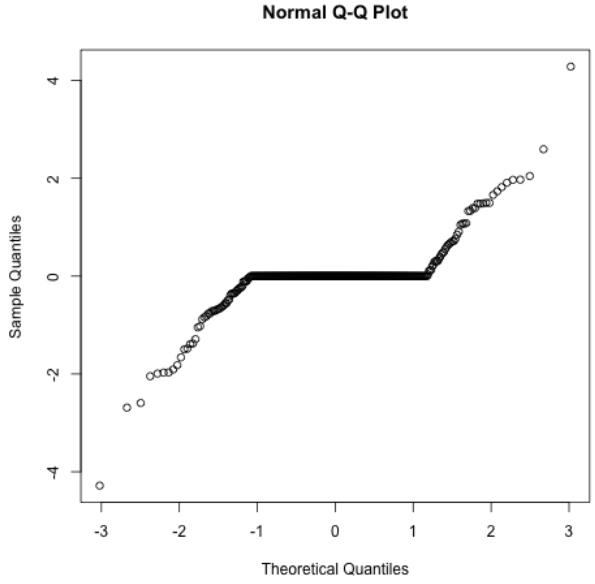
The transformed model has the lowest sum of squares error, the better R^2 value, and appears to satisfy the model assumptions the best.

4. Import the UCI Machine Learning Repository's Auto-MPG data set and create the best possible linear regression model for predicting mpg from the other variables. Use diagnostics and remedial measures to investigate curvature and assumptions related to the design matrix and error terms.





The predicted values match the actual values for mpg very well for this model. There does not appear to be any curvature involved in this case. Instead of having a random and evenly distributed range of values about 0, the residual plot shows a strong bias to 0. This is supported also by the QQ plot for residuals shown below.

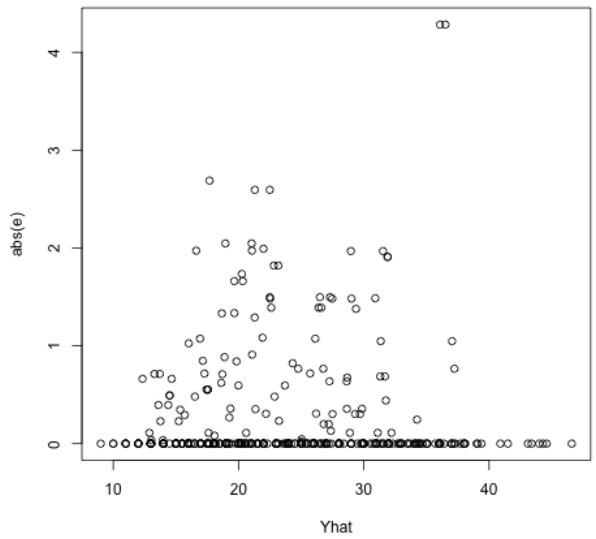


The errors do not appear to be normally distributed. The Shapiro-Wilks test gave a p-value $< 2.2e-16$, suggesting very strongly that the residuals are not normally distributed.

The Brown-Forsythe test gave the following results for this model:

Test Statistic = 0.0026272, p-value = 0.9591

The plot of $\hat{Y}$ vs residuals is shown below. This shows very strongly the errors do not have constant variance.



The sum of squares error for this model is 159.0592. The R^2 statistic is ≈ 1, and $1 - R^2$ is 1.294054e-11.

```
#data mining hw 23
path = '~/Dropbox/Tarleton/data_mining/hw23/'
library(MASS)
library(lawstat)

# 1 Import the file math5305Lab6Data.txt, whose columns are the variables
#   Y, X1, X2, and X3. The goal of these first three problems is to
#   perform diagnostics to assess the assumptions of normality and constancy
#   of variabce for a model predicting Y from the Xj's, to transform Y if
#   necessary, to assess the transformed model using diagnostics, adn to
#   compare the original and transformed models via residual sums of squares.

this.data <- read.csv('~/Dropbox/Tarleton/data_mining/dfiles/5305Lab6',
                                                header=F, sep=',', col.names=c('Y', 'X1', 'X2', 'X3'))

#   We begin by fitting a model and assessing it with diagnostics.

#   (a) Fit model = lm(Y~X1+X2+X3), compute the fitted values Yhat, and the
#       residuals e. You can use the command Yhat = predict(model) to obtain
#       Yhat.

                model = lm(Y~., data=this.data)
                Yhat = predict(model)
                e = model$residuals

#   (b) Plot Y vs Yhat. If the model were valid, what would you expect this
#       plot to look like? Does the plot suggest the existence of curvature
#       in the model?

                plot(Yhat, this.data$Y)
                dev.copy(png, paste0(path, 'Yhat_vs_Y.png'))
                dev.off()

#   (c) Plot e vs Yhat. If the model were valid, what would you expect this
#       plot to look like? Does the plot suggest the existence of curvature
#       in the model?

                plot(Yhat, e)
                dev.copy(png, paste0(path, 'Yhat_vs_e.png'))
                dev.off()
```

```
#   (d) Now that we know curvature is present, there are two courses of
#       action we can take: transform Y or transform the Xj's, or both.
#       Generally, if there are problems with the errors, we should transform
#       Y, and if the errors are ok, we should transform the Xj's. Let's
#       investigate the errors.

#   (e) Plot a qq-plot to check normality of the error terms using the qqnorm
#       command.

                qqnorm(e)
                dev.copy(png, paste0(path, 'residuals_qq_plot.png'))
                dev.off()

#   (f) Perform the Shapiro-Wilks test to check normality of the error terms
#       using the shapiro.test command.

                shapiro.test(e)

#   (g) Based on the results in parts (e) and (f), do the error terms for this
#       model appear to be normal?

#   (h) Check constancy of error variance by plotting |e| vs. Yhat.

                plot(Yhat, abs(e))
                dev.copy(png, paste0(path,'Yhat_vs_abs_e.png'))
                dev.off()

#   (i) Check constancy of error variance by performing the Brown-Forsythe test.

                levene.test(e, as.factor(Yhat <= median(Yhat)))

#   (j) Based on the results in parts (h) and (i), do the error terms for this
#       model appear to have constant variance?

#   (k) Does a transformation of Y appear to be necessary?


#   (l) Finally, calculate the residual sum of squares ||e||2. Note that this
#       value is
```

```
#                 || e ||2 = sum(Yi - Yhati)2, i = 1, ... , n
#       so it is similar to a prediciton sum of squares. It measures the sum of
#       square errors between the predictions Yhati and the actual observations Yi.
#       This number is very large, so to put it in perspective, calculate
#                           ||e||2
#                       -------------
#                       || Y - Ybar||2
#       Assessing the model by this criterion is equivalent to using
#                                ||e||2
#                     R2 = 1 - ------------.
#                                ||Y - Ybar||2
                sum(e*e)

                R2 <- 1.0 - sum(e*e) /
                                (sum( (this.data$Y - mean(this.data$Y)) *
                                          (this.data$Y - mean(this.data$Y)) ))

                R2


# 2 Let lambda be the optimal value produced by the Box-Cox transformation.
#   Transform Y by defining Y_tilde_i = (Y_i)^lambda for i = 1, ... , 100.

                boxcox.results = boxcox(model)
                lambda = boxcox.results$x[which.max(boxcox.results$y)]
                Y.tilde <- this.data$Y^lambda

#   (a) Fit a model tmodel by regressing y_tilde on X1, X2, and X3, and find the
#       corresponding fitted values Y_tilde_hat and e_tilde

                model.t <- lm(Y.tilde~this.data$X1+this.data$X2+this.data$X3)
                Y.tilde.hat <- predict(model.t)
                e.tilde <- model.t$residuals

#   (b) Plot Y_tilde vs Y_tilde_hat and e_tilde vs. Y_tilde_hat. How do these plots
#       compare to those from problem 2? Does curvature appear to exist in the
#       transformed model?

                plot(Y.tilde.hat, Y.tilde)
                dev.copy(png, paste0(path,'Ytilde_vs_Ytildehat.png'))
```

```
                dev.off()
                plot(Y.tilde.hat, e.tilde)
                dev.copy(png, paste0(path,'etilde_vs_Ytildehat.png'))
                dev.off()

#   (c) Investigate normality of the errors for the transformed model.

                qqnorm(e.tilde)
                dev.copy(png, paste0(path, 'residuals_qq_plot_t.png'))
                dev.off()

                shapiro.test(e.tilde)


#   (d) Investigate constancy of error variance for the transformed mode.

                plot(Y.tilde.hat, abs(e.tilde))
                dev.copy(png, paste0(path,'Yhat_t_vs_abs_e_t.png'))
                dev.off()

                levene.test(e.tilde,as.factor(Y.tilde.hat <= median(Y.tilde.hat)))

#   (e) Do the errors for the transformed model appear to satisfy the assumptions
#       of normality and constant error variance? How do your results compare to
#       Those from problem 2?

# 3 Now let's apply the results from the transformed model to the original variable Y.

#   (a) First, create a vector of fitted values for Y by defining
#       Yhat_i = (Y_tilde_hat_i)^(1 / lambda), for i = 1, ... , 100, and create a
#       vector of residuals by defining e_i = Yi - Yhati, for i = 1, ..., 100. These
#       are predicted values and residuals for the original model, but they take
#       advantage of the information from the transformed model.

                Y.tilde.hat.t = (Y.tilde.hat)^(1/lambda)

#   (b) Plot Y vs. Yhat and e vs. Yhat. Did the transformation appear to correct
#       problems with the functional form?

                plot(Y.tilde.hat.t, this.data$Y)
```

```
                dev.copy(png, paste0(path,'Y_vs_Ytildehat_t.png'))
                dev.off()
                plot(Y.tilde.hat.t, e.tilde)
                dev.copy(png, paste0(path,'e_vs_Ytildehat_t.png'))
                dev.off()

#   (c) Finally, calculate ||e||2, ||e||2 / (||Y - Ybar||2), and
#       R2 = 1 - ||e||2 / (||Y - Ybar||2) as in question 2. Which model fits the data
#       better/has a lower residual sum of squares?
                sum(e.tilde*e.tilde)

                R2 <- 1.0 - sum(e.tilde*e.tilde) /
                                (sum( (Y.tilde.hat.t - mean(Y.tilde.hat.t)) *
                                        (Y.tilde.hat.t - mean(Y.tilde.hat.t)) ))

                1 - R2

                R2



# 4 Import the UCI Machine Learning Repository's Auto-MPG data set and create the best
#   possible linear regression model for predicting mpg from the other variables. Use
#   diagnostics and remedial measures to investigate curvature and assumptions related
#   to the design matrix and error terms.

                auto <- read.csv('~/Dropbox/Tarleton/data_mining/dfiles/auto_data.csv',
                                        header=T, na.strings='?', dec='.', strip.white=T)

                names(auto) <- c('mpg', 'cyl', 'displ', 'hp', 'wt', 'accel', 'year', 'orig', 'name')
                model = lm(mpg~., data=auto)
                Yhat = predict(model)
                e = model$residuals

                plot(Yhat, auto$mpg)
                dev.copy(png, paste0(path, 'auto_Yhat_vs_Y.png'))
                dev.off()

                plot(Yhat, e)
                dev.copy(png, paste0(path, 'auto_Yhat_vs_e.png'))
```

```
dev.off()

qqnorm(e)
dev.copy(png, paste0(path, 'auto_residuals_qq_plot.png'))
dev.off()

shapiro.test(e)

plot(Yhat, abs(e))
dev.copy(png, paste0(path,'auto_Yhat_vs_abs_e.png'))
dev.off()

levene.test(e,as.factor(Yhat <= median(Yhat)))

sum(e*e)

R2 <- 1.0 - sum(e*e) /
                (sum( (this.data$Y - mean(this.data$Y)) *
                          (this.data$Y - mean(this.data$Y)) ))

R2
```