**Solution**

**1)**

**Theoretical Understandings of Box Cox Transformation:**

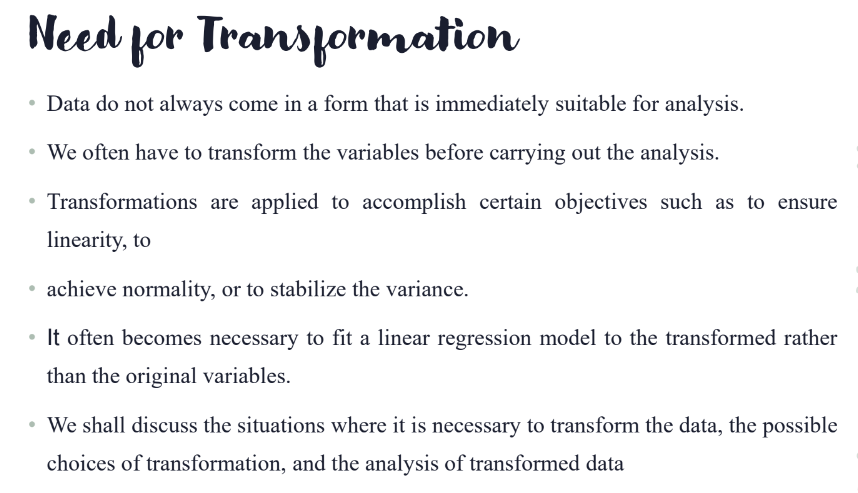
Conditions for using Box-Cox transformation

• The process data is not normally distributed

• Residuals are not normally distributed

• Residuals do not have constant variance

• All values must be greater than 0

****• Data must be continuous variables.

**R CODE:**

df <- data.frame(x= c(7, 7, 8, 3, 2, 4, 4, 6, 6, 7, 5, 3, 3, 5, 8),y= c(1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 6, 7, 8))

**# fitting the model**

*mod=lm(y ~ x,data=df)*

*summary(mod)*

*plot(mod)*

*plot(mod$residuals)*

*qqnorm(mod$residuals)*

*qqline(mod$residuals)*

*library(MASS)*

*bc=boxcox(y~x)*

**###Finding the optimal lambda value###**

*lambda=bc$x[which.max(bc$y)]*

*lambda*

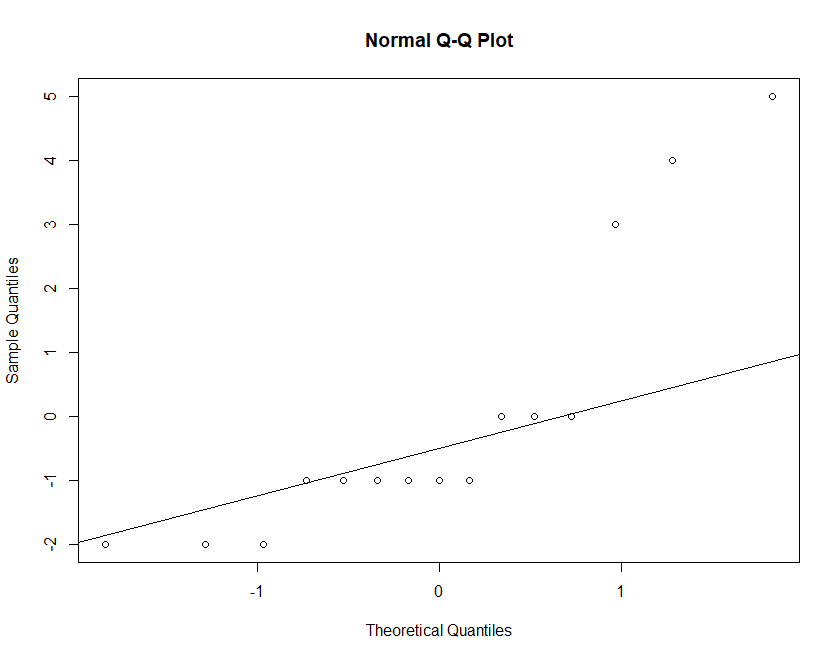
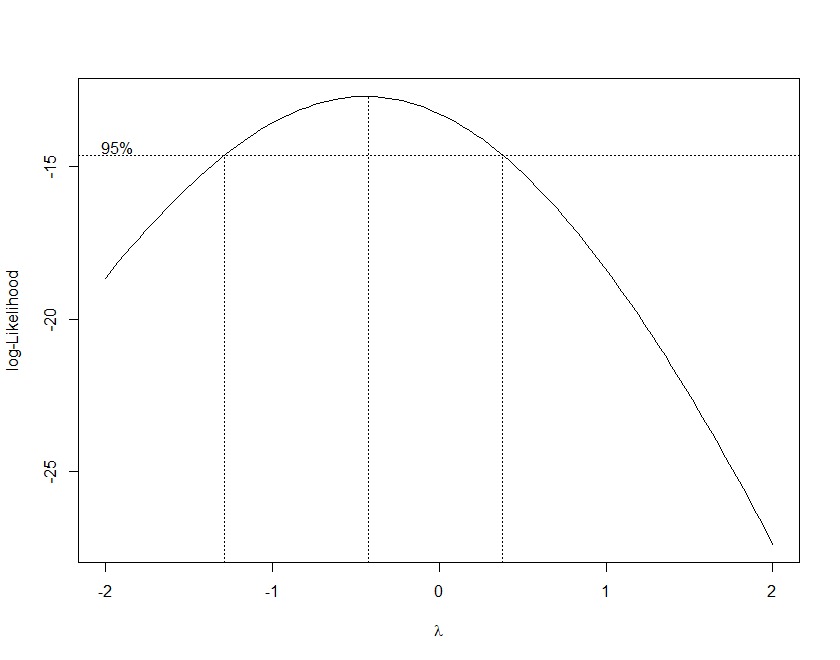
*new\_model=lm(((y^lambda-1)/lambda) ~ x)*

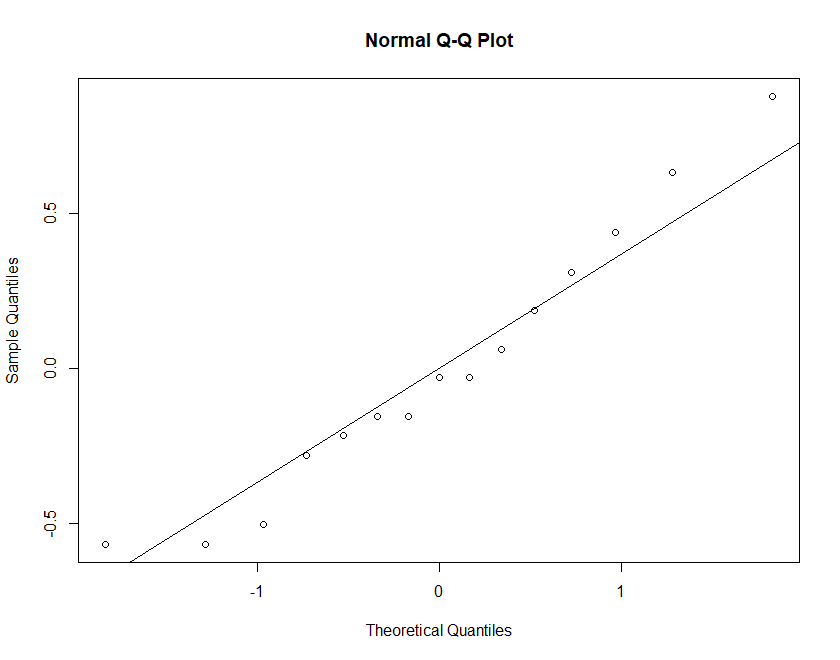
*new\_model*

*new\_model=lm(((y^lambda-1)/lambda) ~ x)*

*qqnorm(new\_model$residuals)*

*qqline(new\_model$residuals)*

**PLOTTING:**

**Normal QQ Plot of the transformed variables:**

2)

**Solution:**

**Theoretical understandings:**

BASIC CONCEPT

❖One of the assumption of standard regression is that the relationship between Y or

transformed Y is linear with X or transformed X.

❖If we regress Y on X’s when some of the X’s are entered in a transformed way, then the

nice properties of least squares like estimator being unbiased go away.

❖We are interested in the case where the assumptions of the classical regression model are

satisfied except that the regressor is not linear, but can be made linear using a suitable transform on X.

❖A solution to this problem is by using Box-Tidwell transformation, which is just the usual

regression model with powers on the X’s

❖Here BT transformation is applicable for X’s that are continuous and greater than 0.

❖Other conditions to run BT are the usual assumptions

▪ The error term is normally distributed with a constant variance σ2

▪ The errors are independent.

**R code:**

**##Calling the relevant R Library###**

*library(car)*

*data<-read\_xlsx("C:/Users/DELL/Desktop/Datareg.xlsx")*

**###Feature Engineering for the model###**

*y<-data$y*

*x1<-data$x1*

*x2<-data$x2*

*x3<-data$x3*

*x4<-data$x4*

*x5<-data$x5*

**###Creating data in a matrix form###**

*data<-data.frame(x1,x2,x3,x4,x5,y)*

*pairs(data)*

**###Fitting the model#####**

*fittrial<-lm(y~x1+x2+x3+x4+x5)*

**###check the normality using residual plot###**

*plot(fittrial$residuals)*

*abline(0,0)*

**##Better understanding of the violation of normality assumption###**

**##using QQ Norm line technique###**

*qqnorm(fittrial$residuals)*

*qqline(fittrial$residuals)*

**##Checking which variables are creating trouble####**

*plot(x1,fittrial$residuals)*

*plot(x2,fittrial$residuals)*

*plot(x3,fittrial$residuals)*

*plot(x4,fittrial$residuals)*

*plot(x5,fittrial$residuals)*

**####Finally fitting the Model using Box-Tidwel Transformation###**

*fit1<-boxTidwell(y~x4+x5+x1,~x2+x3)*

*fit1*

*summary(fit1)*

**###Finding the lambda values of the variables which need to be transformed##**

*al1<-fit1$result[3,1]; al4<-fit1$result[1,1]; al5<-fit1$result[2,1]*

*X1 = x1^al1; X4 = x4^al4; X5 = x5^al5*

**###Finally fitting the transformed model###**

*fitfinal<-lm(y~X1+x2+x3+X4+X5)*

*par(mfrow=c(1,2))*

*plot(fittrial$residuals)*

*abline(0,0)*

**###Checking the residual plot of transformed variables##**

*plot(fitfinal$residuals)*

*abline(0,0)*

**####Residual analysis of old model###**

*qqnorm(fittrial$residuals)*

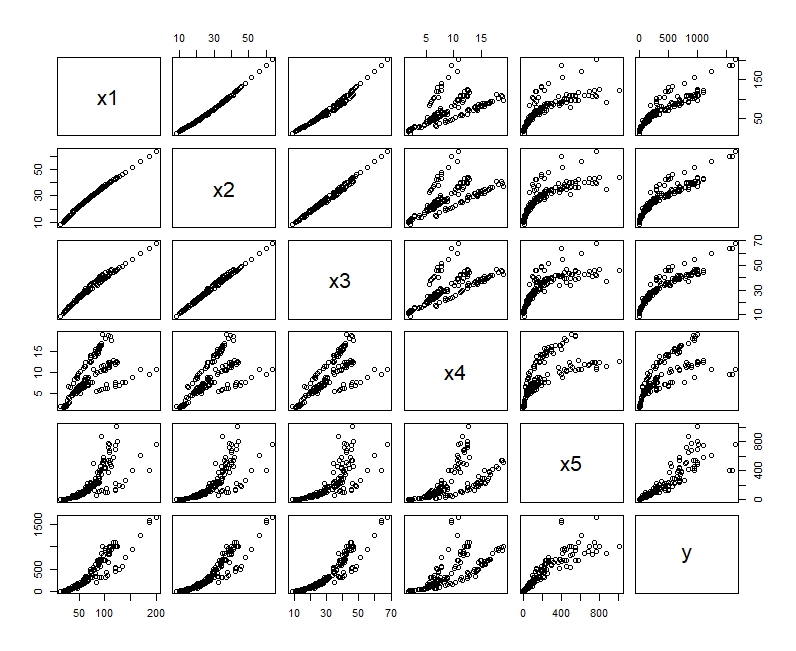
*qqline(fittrial$residuals)*

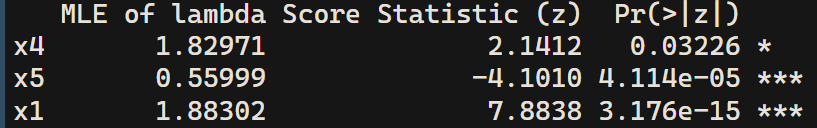
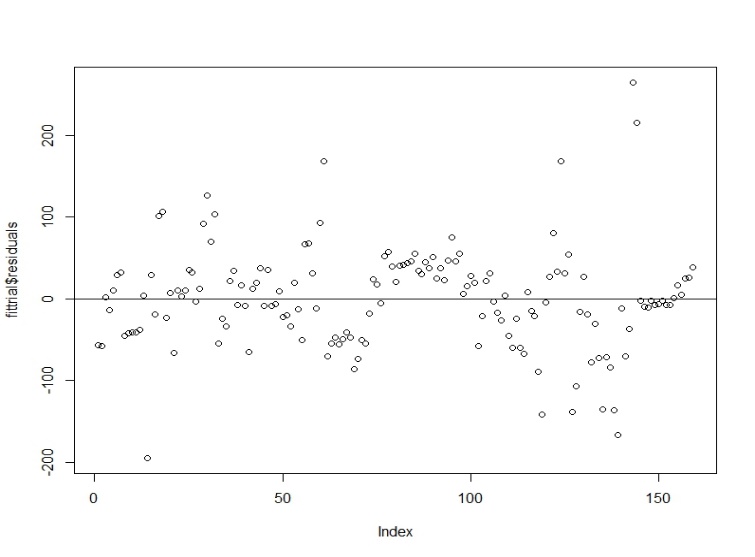
**####Residual plot of the new transformed model###**

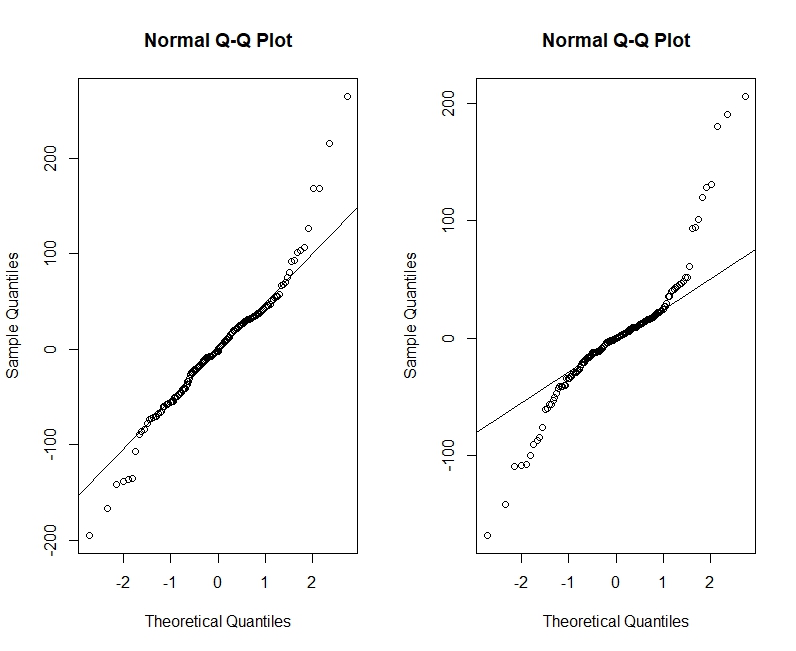
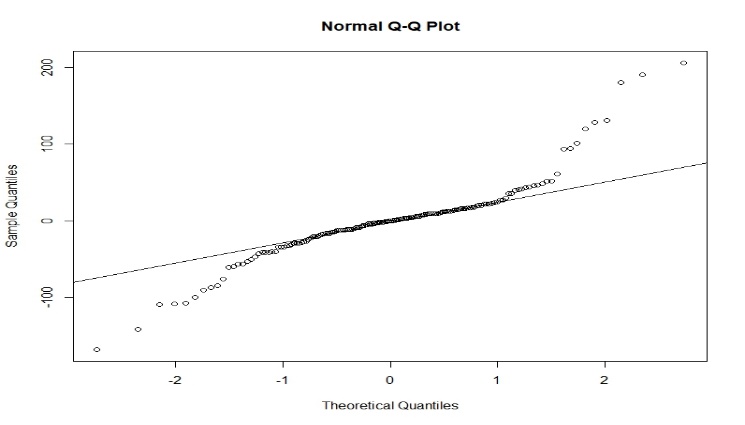
*qqnorm(fitfinal$residuals)*

*qqline(fitfinal$residuals)*

**Relevant Plotting and analysis of the data:**



**Lambda values**:

****

**Mt Car Analysis:**

**Implication of Multiple Linear Regression Model:**

R code:

**##Feature Engineering####**

*Mtcars*

*n=nrow(mtcars)*

*colnames(mtcars)*

*y=mtcars[,'mpg']*

*y*

*predictors\_nm=c('hp','disp','wt')*

*X=mtcars[,predictors\_nm]*

*X*

*str(X)*

*Int=rep(1,n)*

*X=as.matrix(cbind(Int,X))*

*X*

*str(X)*

**##calculate x'x**

*A=t(X)%\*%X*

*A*

**##calculate inv(X'x)**

*B= solve(A)*

*B*

**##Calculate Beta###**

*B%\*%t(X)%\*%y*

**#Basic structure:mpg=b0+b1\*hp+b2\*wt+b3\*disp+error###**

*fit=lm(mpg~hp+wt+disp, data=mtcars)*

*coef(fit)*

*cbind(coef(fit),B%\*%t(X)%\*%y)*

**####PLOTTING###**

*plot(fitted(fit))*

*abline(0,0)*

*boxplot(fitted(fit))*

*ggplot2::aes(fitted(fit))*

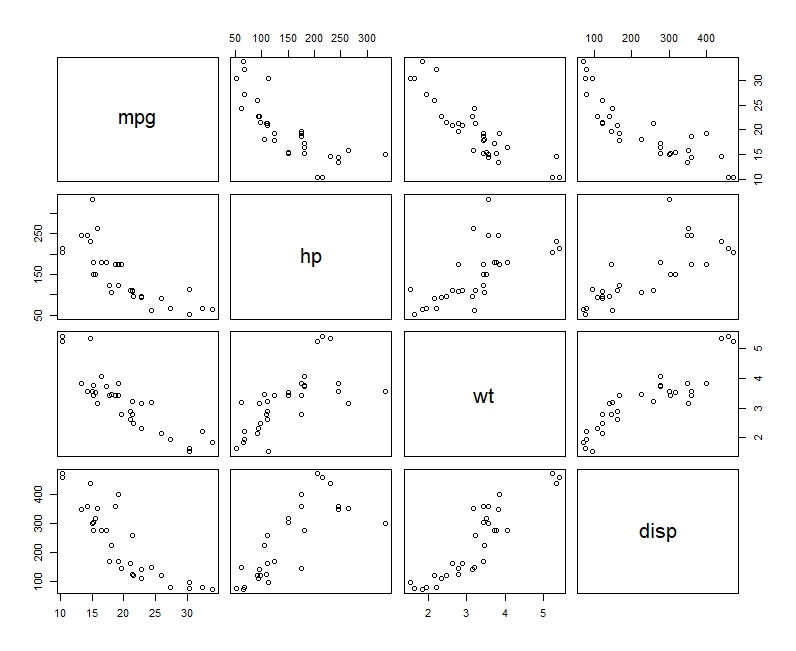
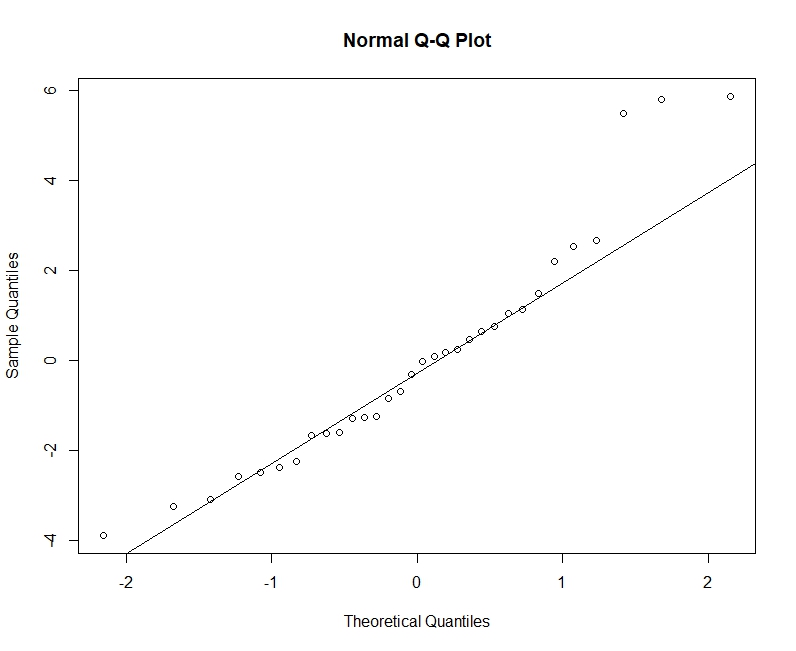
*plot(fit$model,col='black')*

*plot(fit$rank,col='black')*

*plot(fit$residuals)*

*qqnorm(fit$residuals)*

*qqline(fit$residuals)*

**Residual Plot:**