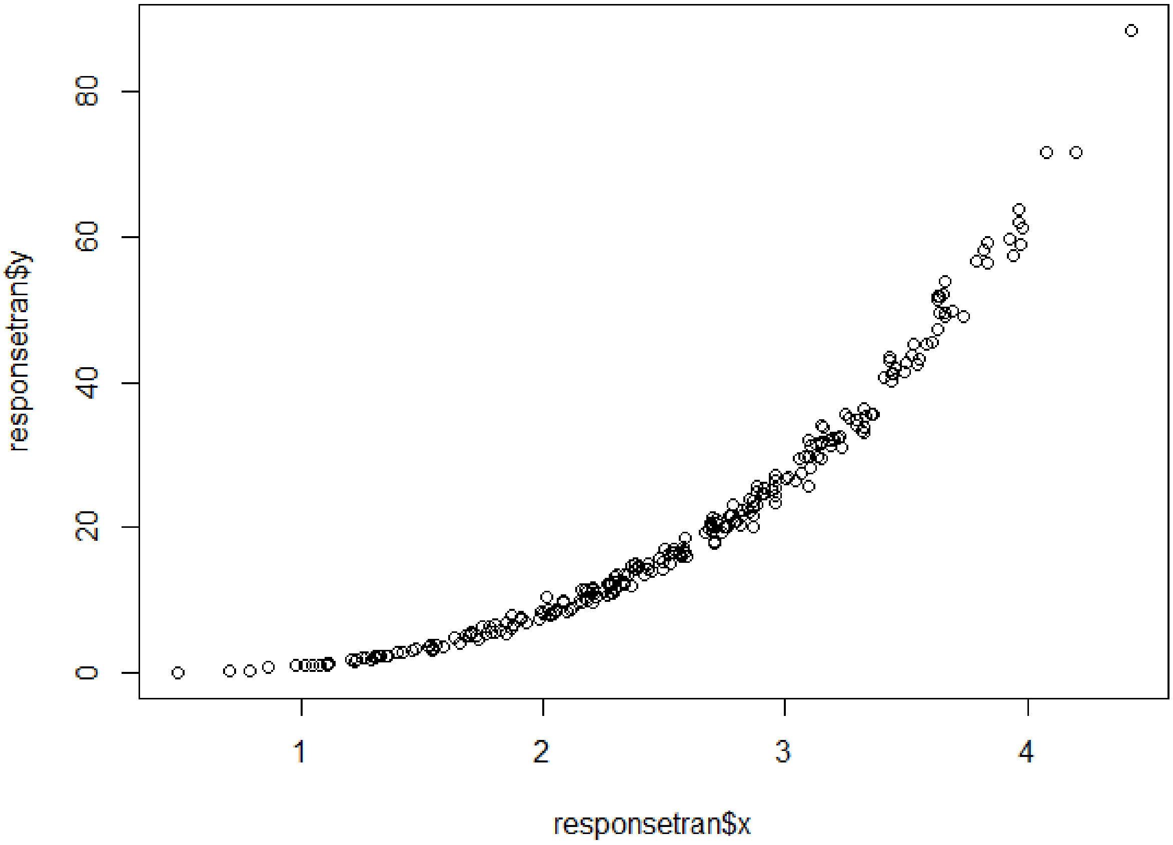
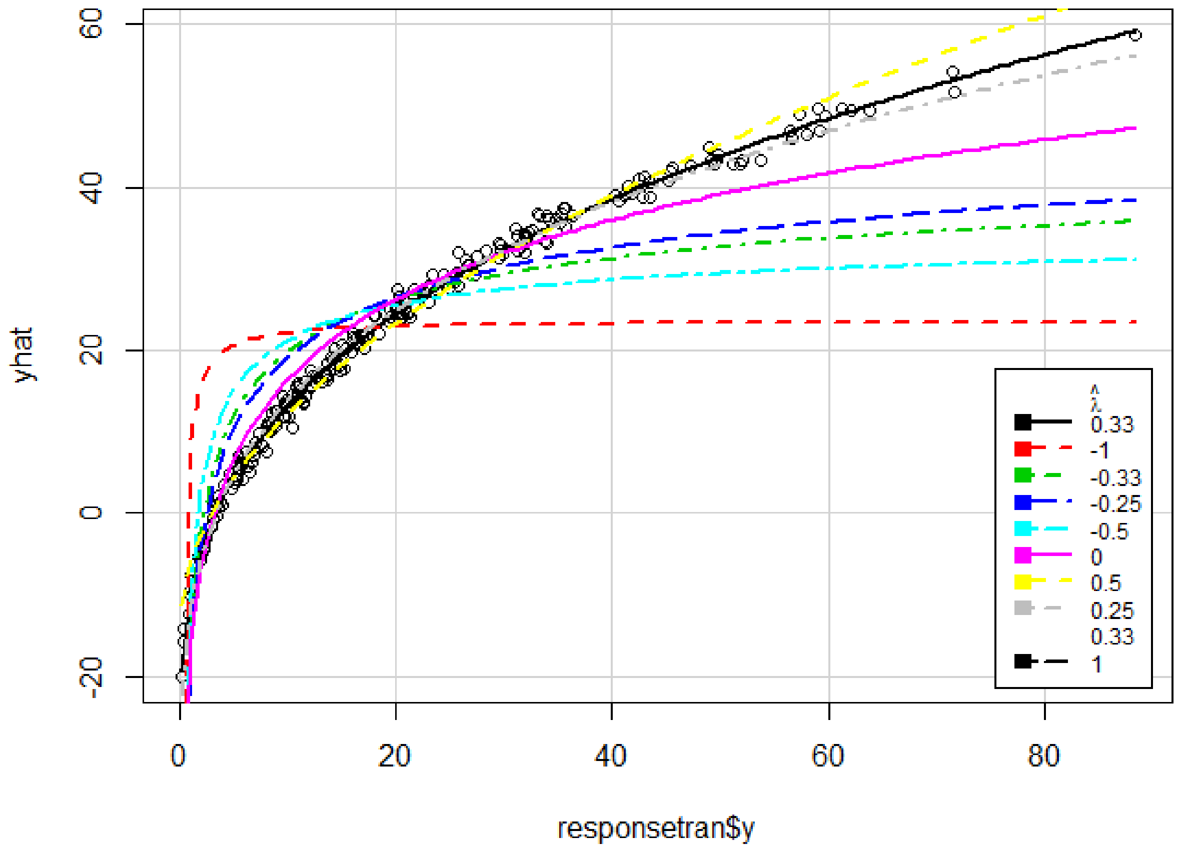
Fabricated data y=x^3



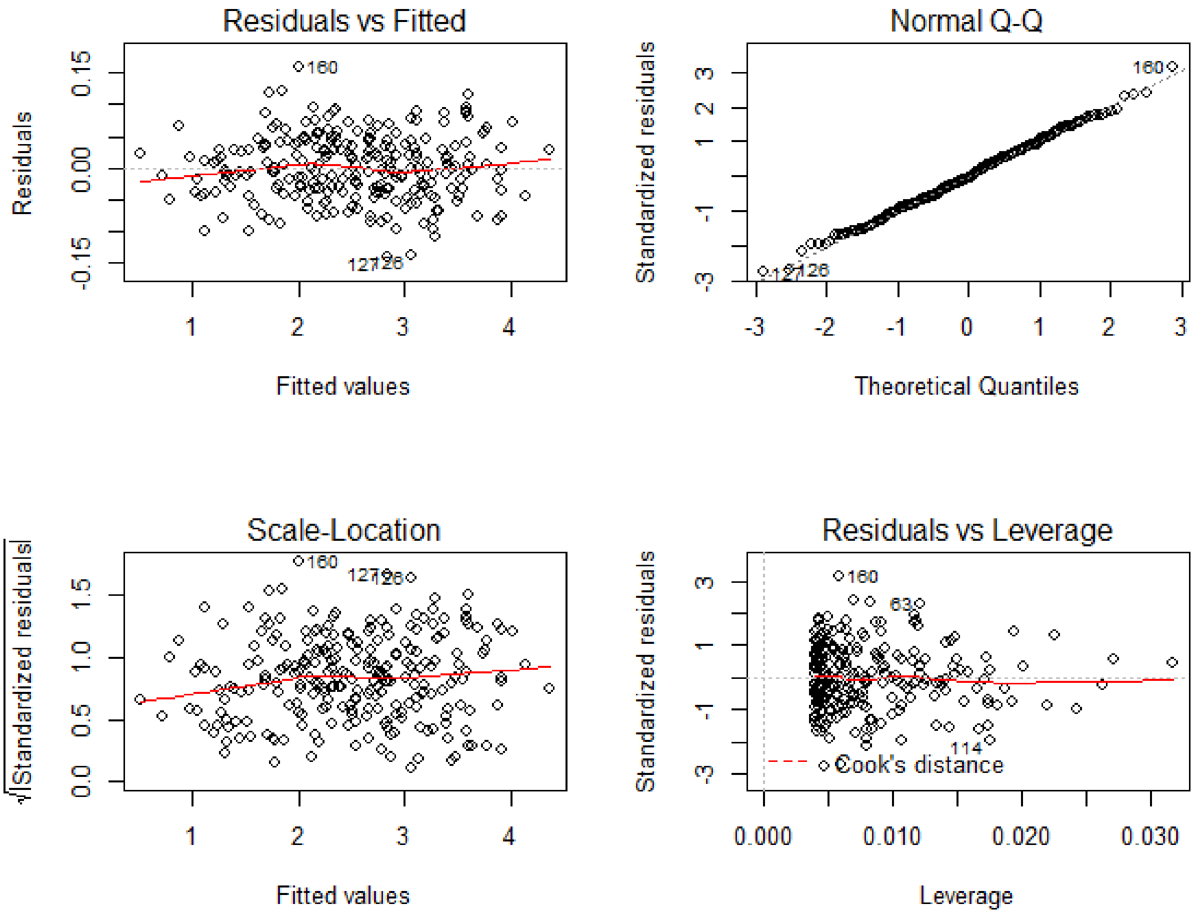
Fitted values plot versus Y (applicable when x, predictor has an **elliptical symmetrical distribution**. Assumption that univariate x has normal distr is much stronger than x having elliptically symmetric.)



Lamdahat=0.33 is the best transform

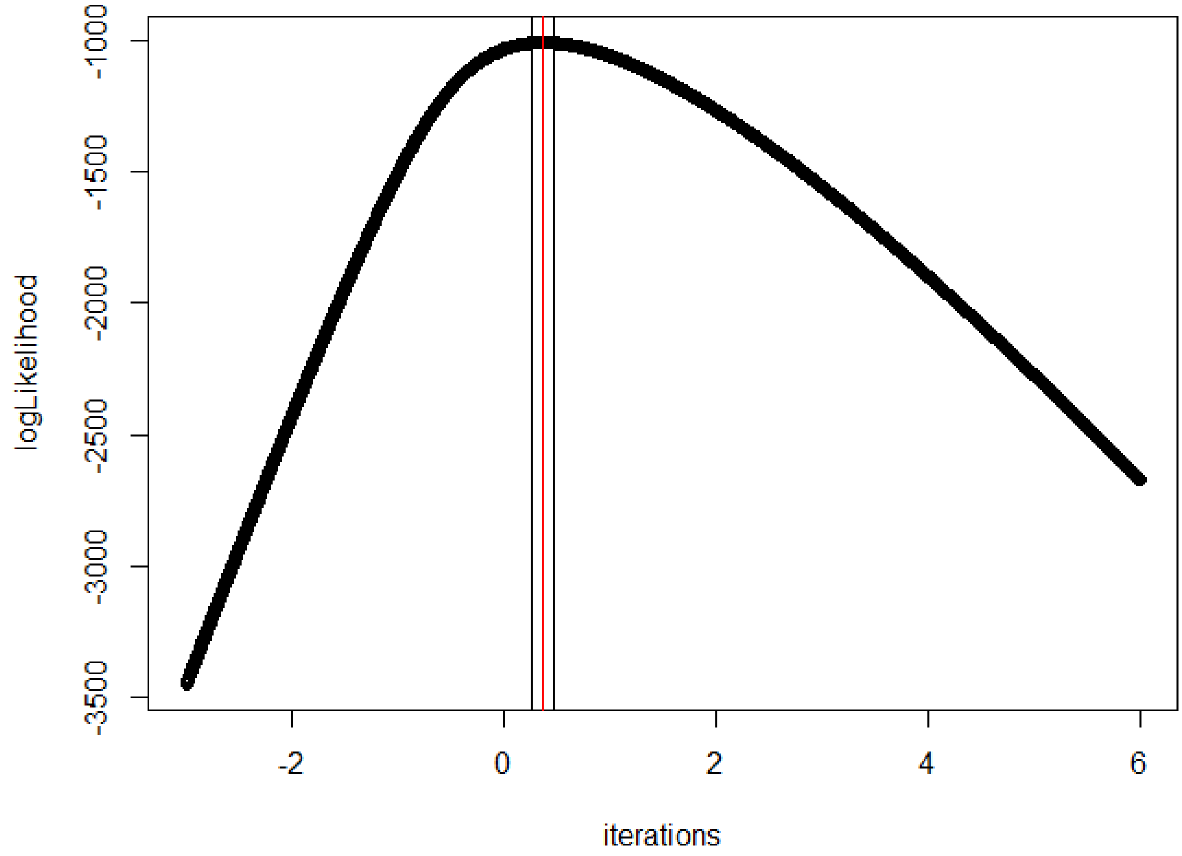
```
dev.off()
par(mfrow=c(2,2))
#par(mfrow=c(1,1))
responsetran <- read.csv('C:/Users/vigupta/OneDrive/Learning/DataScience/Statistics Texas A&M University/608/SheatherBook/Data/responsetransformation.txt', header = T, sep = '')

plot(responsetran$x, responsetran$y)

fit.1 <- lm(responsetran$y~responsetran$x)
lambda <- c(-1,-1/3,-1/4,-1/2,0,1/2,1/4,1/3,1)


library(alr3)
```

```
inverseResponsePlot(fit.1,lambda)
#we get lambda=1/3 as a good transformation.

responsetran$yt=responsetran$y^.33

fit.2 <-lm(responsetran$yt~responsetran$x)

#Asses validity of the model
plot(fit.2)
#Looks very good
```

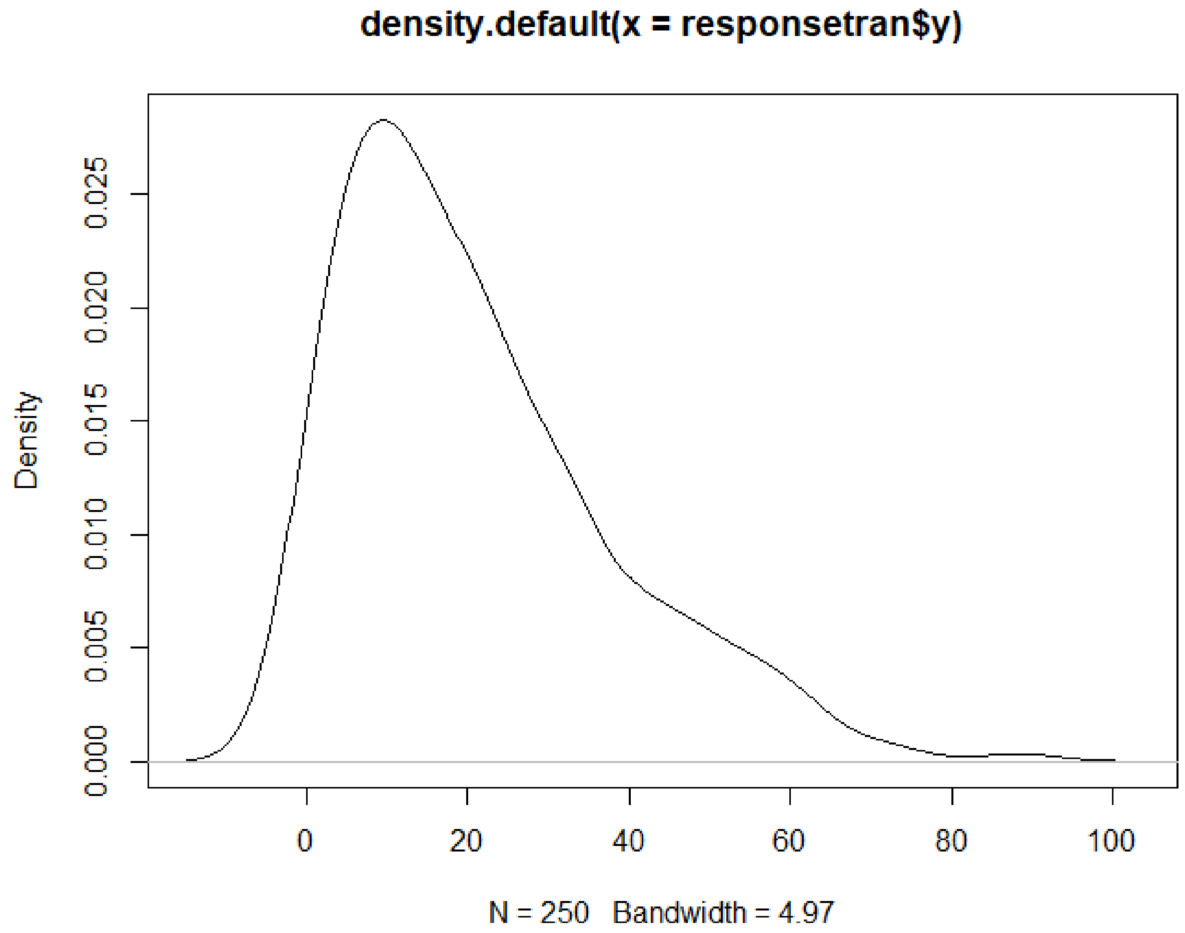


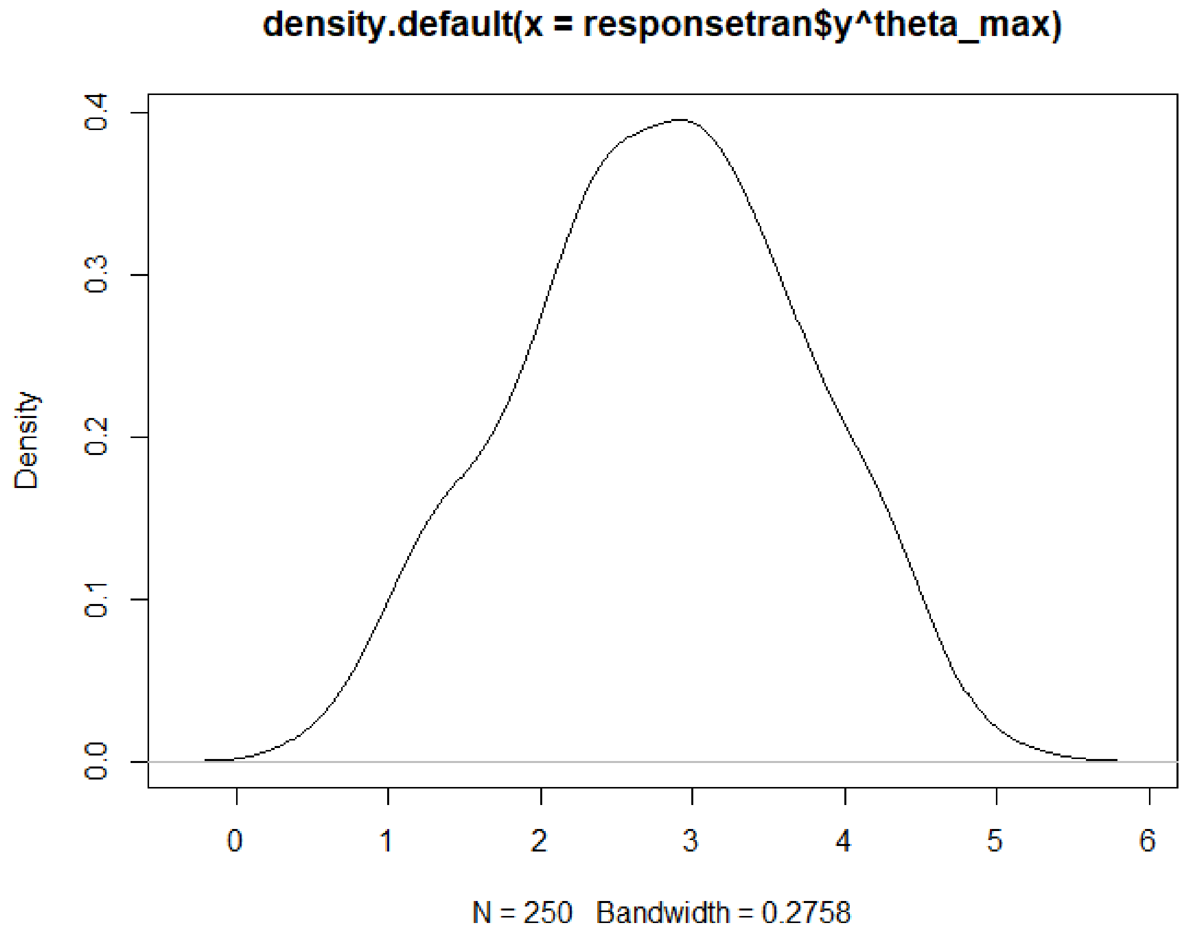
Trying with Box cox

If I can transform Y or X or both to have normality, as close as possible hen w ecan establish a linear relationship between Y and X. Note: normality may even not be possible if x does not explain everything.



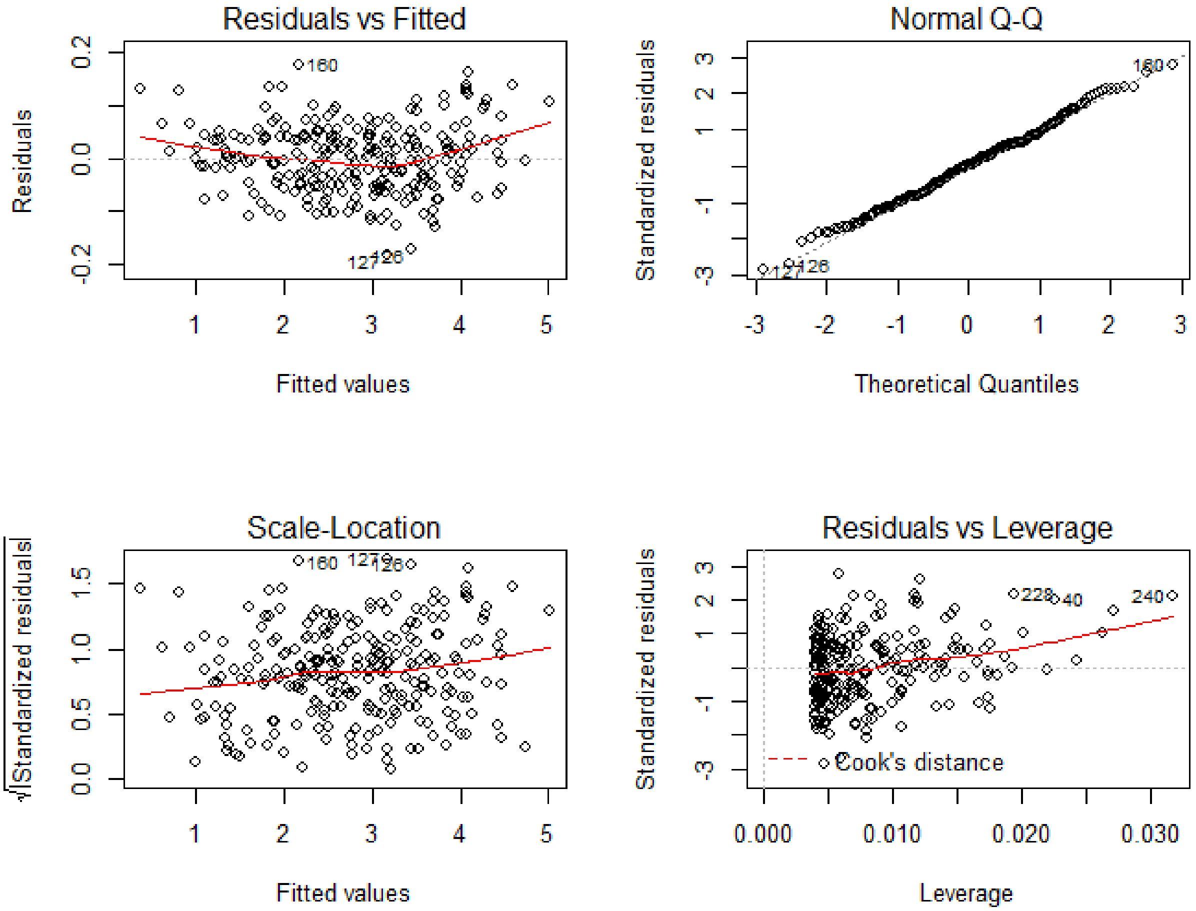
Above is a plot of power transform of Y (since it has a skewed dist) for various values of theta on x. Loglikehoood is plotted on y and we see theta ~ 0.364. We get some sort of normality in Y.


density.default(x = responsetran$y)
Density
0.000
0.005
0.010
0.015
0.020
0.025
0
20
40
60
80
100
N = 250   Bandwidth = 4.97



```
Shapiro-Wilk normality test

data:  responsetran$y^theta_max
W = 0.99269, p-value = 0.2563
```

Fairly good linear fit with BoxCox transform as well.



```
#Complete Code


dev.off()
par(mfrow=c(2,2))
#par(mfrow=c(1,1))
responsetran <- read.csv('C:/Users/vigupta/OneDrive/Learning/DataScience/Statistics Texas A&M
University/608/SheatherBook/Data/responsetransformation.txt', header = T, sep = '')

plot(responsetran$x, responsetran$y)

fit.1 <- lm(responsetran$y~responsetran$x)
lambda <- c(-1,-1/3,-1/4,-1/2,0,1/2,1/4,1/3,1)


library(alr3)
```

```
inverseResponsePlot(fit.1,lambda)
#we get lambda=1/3 as a good transformation.

responsetran$yt=responsetran$y^.33

fit.2 <-lm(responsetran$yt~responsetran$x)

#Asses validity of the model
plot(fit.2)
#Looks very good

#LETS TRY BOX COX

y1 <-  sort(responsetran$y)

  n <- length(y1)
theta <- -3 #starting seed , we will start with this seed and go to the +ve value of the seed.
iterations <- seq(theta, abs(theta)*2, 0.001) # this holds theta's, power transforms
yt0 <-log(y1)
var_yt0 <- var(yt0)
l0 <- (0-1)*sum(log(y1)) - 0.5*n*(log(2*pi*var_yt0)+1)
t0 <- 0
logLikelihood <- as.vector(rep(0,length(iterations)))# this holds logliklihoods 1:1 with power transforms
for (i in 1:length(logLikelihood)) {

  yt <- (y1^iterations[i] - 1)/iterations[i]
  var_yt <- var(yt)
  logLikelihood[i] <- (iterations[i]-1)*sum(log(y1)) - 0.5*n*(log(2*pi*var_yt)+1)
  if(abs(iterations[i]) < 1.0e-10) iterations[i] <- 0 # to cover for the iteration value when theta->0
  if(abs(iterations[i]) < 1.0e-10) logLikelihood[i] <-l0 # to cover for the iteration value when theta->0

}
plot(iterations,logLikelihood)

(theta_max <- iterations[which(logLikelihood==max(logLikelihood))])
(tU = max(logLikelihood)+.5*qchisq(.95,1)) #Upper bound on theta
(tL=  max(logLikelihood)-.5*qchisq(.95,1))#lower bound on theta
(tM=max(logLikelihood)) #theta max

(iL <- min(which((logLikelihood > tL) & (logLikelihood < tM)))) #index of lower bound on theta
(iU <- max(which((logLikelihood > tL) & (logLikelihood < tM))))#index of upper bound on theta

abline(v=iterations[iL])
abline(v=iterations[iU])
abline(v=theta_max, col=2)

plot(density(responsetran$y))
plot(density(responsetran$y^theta_max))
shapiro.test(responsetran$y^theta_max)

responsetran$yboxcox=responsetran$y^theta_max
fit.3 <- lm(responsetran$yboxcox~responsetran$x)
plot(fit.3)
```