Lab #3

Mario Arce Acosta

2024-05-31

# Chunks 2,3,4 display plots for each model in sequential order of assignment

# 1.) See Chunk 1

# 2.) See Chunk 2 for Lin Model and summary, lacked in assumptions and R squared value. An increase of 1 unit in disp (displacement/volume) leads to -0.041 change in mpg units (miles per gallon)

# 7.) See Chunk 1 for lin-log transformation and summary, better in assumptions and best R squared value of all 3. An increase of 1 percent in disp (displacement/volume) leads to -9.29 change in mpg (miles per gallon)

# 9.) See Chunk 1 for log-log transformation and summary, best satisfaction of assumptions but slightly worse in R squared value than lin-log model. An increase of 1 percent in disp (displacement/volume) leads to -45.9 percent change in mpg (miles per gallon)

# Chunk 1

library(stargazer)

##   
## Please cite as:

## Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.

## R package version 5.2.3. https://CRAN.R-project.org/package=stargazer

#Load and see structure of data  
myData<-mtcars  
structure(myData)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2  
## Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4  
## Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4  
## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3  
## Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3  
## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3  
## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4  
## Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4  
## Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4  
## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1  
## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1  
## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2  
## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2  
## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4  
## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2  
## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2  
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2

# Chunk 2

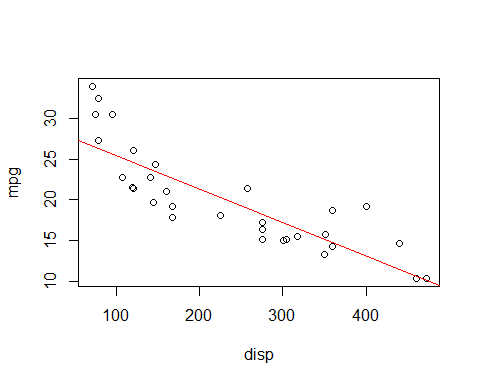
Build, summarize, and interpret models  
  
#Linear model  
model1<-lm(mpg~disp,myData)  
#Lin-Log model  
model2<-lm(mpg~log(disp),myData)  
#Log-Log model  
model3<-lm(log(mpg)~log(disp),myData)  
  
stargazer(model1, model2, model3,covariate.labels=c("disp"), type="text", out="models1.text")

##   
## =============================================================  
## Dependent variable:   
## -------------------------------  
## mpg log(mpg)   
## (1) (2) (3)   
## -------------------------------------------------------------  
## disp -0.041\*\*\*   
## (0.005)   
##   
## log(disp) -9.293\*\*\* -0.459\*\*\*   
## (0.787) (0.039)   
##   
## Constant 29.600\*\*\* 69.205\*\*\* 5.381\*\*\*   
## (1.230) (4.185) (0.208)   
##   
## -------------------------------------------------------------  
## Observations 32 32 32   
## R2 0.718 0.823 0.821   
## Adjusted R2 0.709 0.817 0.815   
## Residual Std. Error (df = 30) 3.251 2.579 0.128   
## F Statistic (df = 1; 30) 76.513\*\*\* 139.350\*\*\* 137.343\*\*\*  
## =============================================================  
## Note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

#We see that coefficients are stat significant for each model but, Lin-Log performs better with a higher R^2 and F-stat (more sample variance explained) despite its distribution not being as "normal" as Model 3's

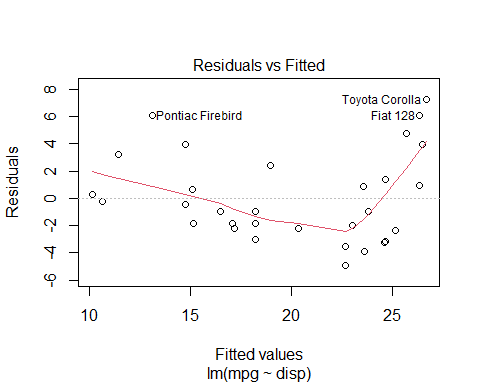
#Plot with relationship  
plot(myData$disp,myData$mpg, xlab="disp",ylab="mpg")  
abline(model1,col="red")

# 3a.) Linear Model plot and data for visualization

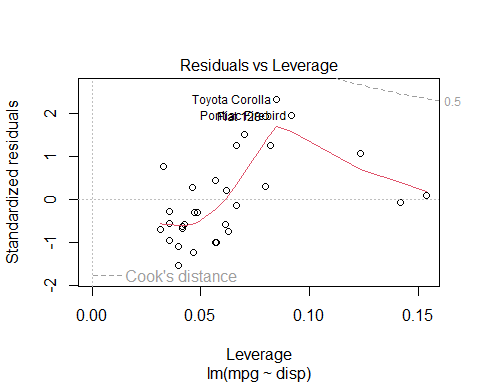
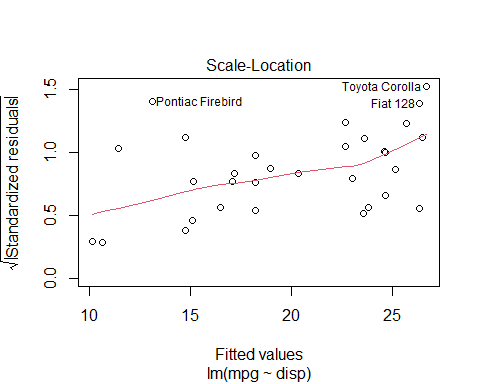
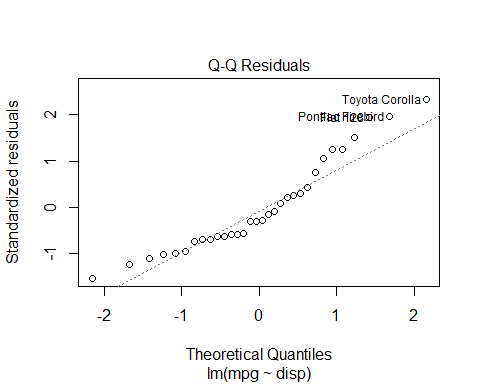


#Plot with residuals and fitted values, with histogram, and Q-Q  
plot(model1)

# 4a.) Residual vs. Fitted for Lin Model was plotted in Chunk 2, and it appears to be relatively homoscedastic and linear with residuals error spread evenly and around a line

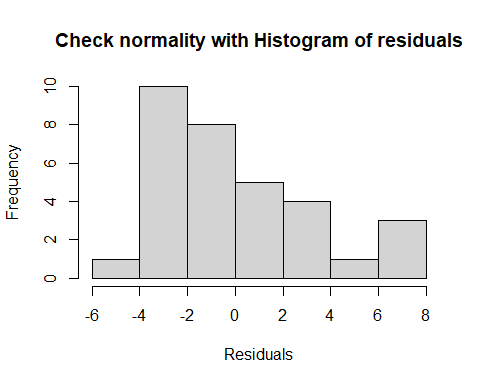


# 5a.) Q-Q residuals for Lin Model appear to be normally distributed with many points concentrated in center of 0



hist(model1$residuals, xlab="Residuals", main="Check normality with Histogram of residuals")

# 6a.) Histogram for Lin model is skewed to right, not really normally distributed



#Not a normally distributed histogram, skewed to right

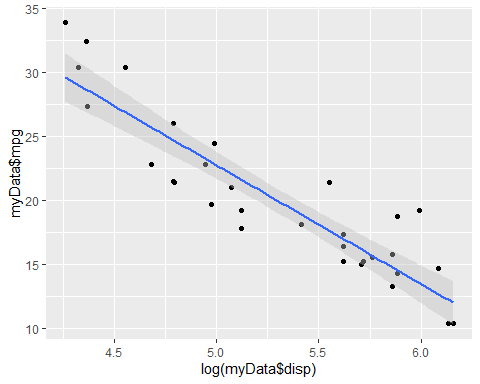
# Chunk 3

# 8.) Plots for steps 3-6, for Lin-Log Model, the transformation made it satisfy Normality, Linearity, and Homoskedasticity assumptions better

# 3b.) Lin-Log model and data plotted for visualization

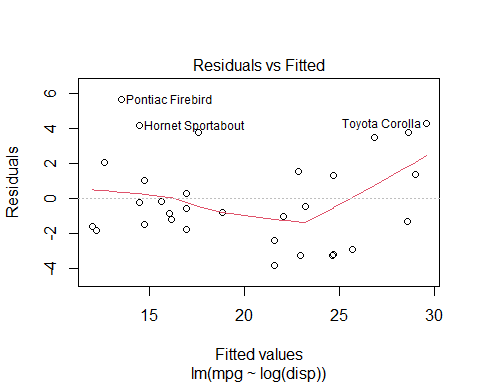
#Plot and see visuals  
library(ggplot2)  
ggplot(myData, aes(log(myData$disp), myData$mpg)) + geom\_point() + stat\_smooth(method="lm",fill="gray")

## `geom\_smooth()` using formula = 'y ~ x'



#ggplot(myData, aes(myData$disp, myData$mpg)) + geom\_point() + stat\_smooth(method="lm",formula=y~log(x),fill="gray")  
  
#Plot residuals vs fitted, and Q-Q  
plot(model2)

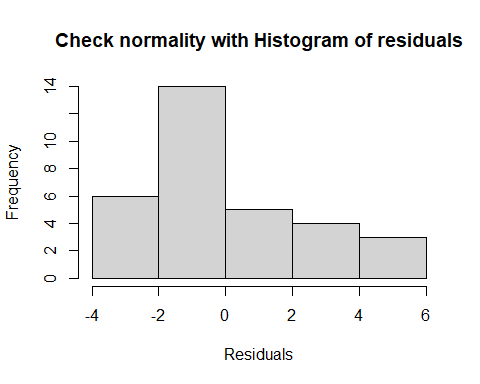
# 4b.) Residuals vs. fitted for Lin-Log Model, somewhat homoscedastic and linear if looking at initial plot transform



# 5b.) Q-Q residual plot for Lin-Log Model, appears to be more normally distributed than Lin Model

hist(model2$residuals, xlab="Residuals", main="Check normality with Histogram of residuals")

# 6b.) Histogram of residuals for Lin-Log Model, more normally distributed but still skewed



#Not nomrally distributed with histogram, but better than 1st model's histogram

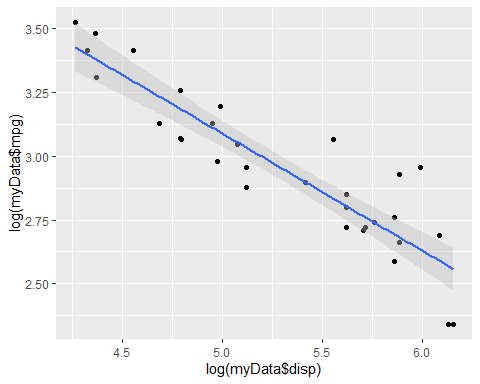
# Chunk 4

# 10.) Plots for steps 3-6, for Log-Log Model, the transformation made it satisfy Normality, Linearity, and Homoskedasticity assumptions best

# 3c.) Log-Log Model and data plot for visualization

#Plot to see relationship  
library(ggplot2)  
ggplot(myData, aes(log(myData$disp), log(myData$mpg))) + geom\_point() + stat\_smooth(method="lm",fill="gray")

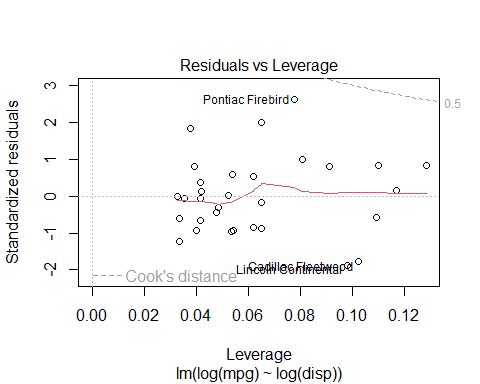
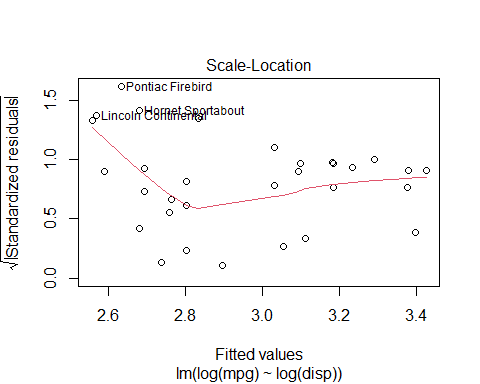
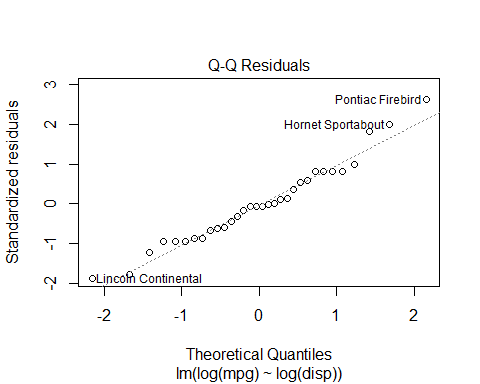
## `geom\_smooth()` using formula = 'y ~ x'



#Plot residuals vs. fitted, and Q-Q  
plot(model3)

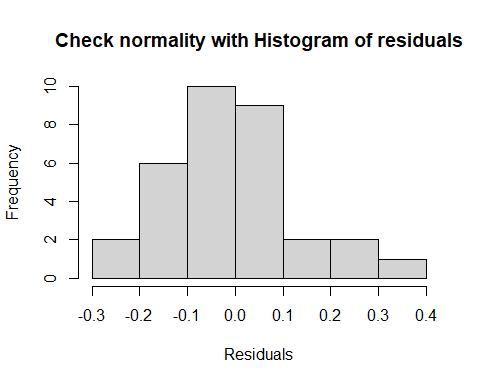
# 4c.) Residuals vs fitted for Log-Log model, better in terms of linearity and homoskedasticity

# 5c.) Q-Q residual plot for Log-Log Model, more normally distributed and center around 0



#Much more normally distributed  
hist(model3$residuals, xlab="Residuals", main="Check normality with Histogram of residuals")

# 6c.) Histogram of residuals for Log-Log Model, most normally distributed histogram of all models



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.