# STAT 6021: Project 1

Greg Madden, Maxwell Levinson, Andrew Setaro, and Trey Hamilton

3/3/2022

```
## Warning: package 'MASS' was built under R version 4.0.5
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

Reading in dataframe

```
Data <- read_csv("diamonds4.csv")
```

```
## Rows: 1214 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): clarity, color, cut
## dbl (2): carat, price
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

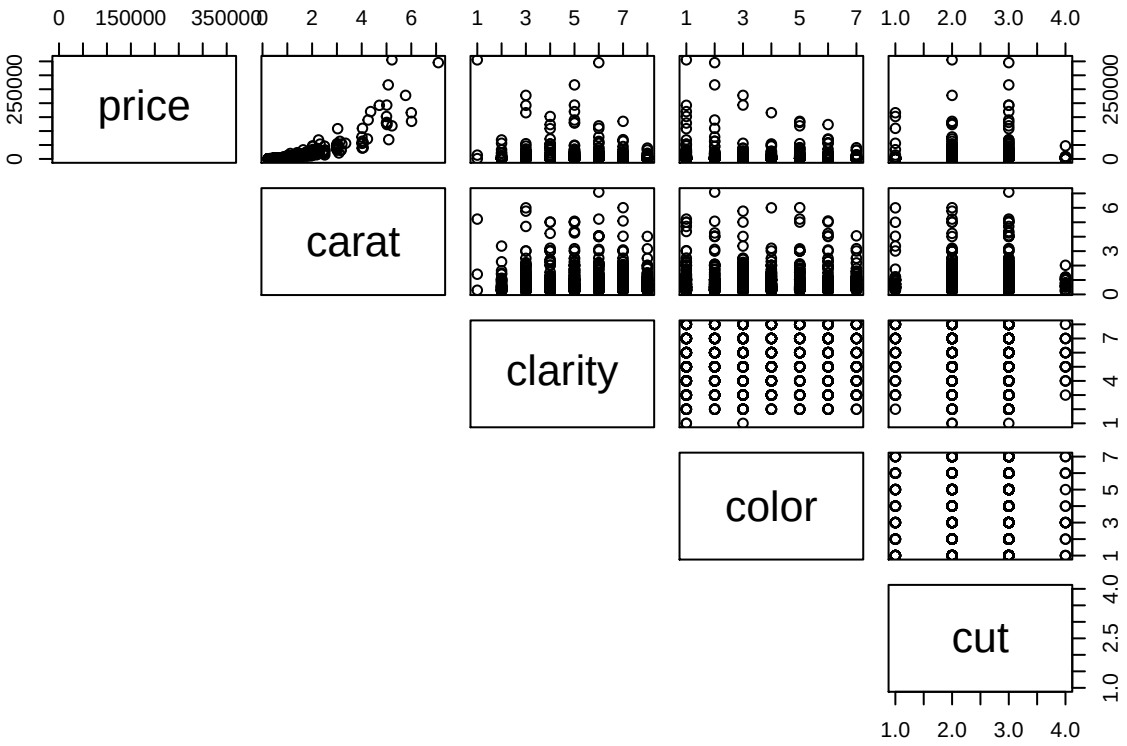
Refactoring categorical variables

```
#colnames(Data)
Data <- Data %>%
  #note VVS1 diamonds rank higher on the clarity scale than VVS2
  mutate(clarity = clarity%>%fct_relevel(c("FL", "IF", "VVS1","VVS2","VS1","VS2", "SI1", "SI2"))) %>%
  mutate(color = color%>%fct_relevel(c("D", "E", "F","G","H","I", "J"))) %>%
  mutate(cut = cut%>%fct_relevel(c("Good", "Very Good", "Ideal","Astor Ideal")))

# reordering so that price is first
Data <- Data[, c(5, 1, 2, 3, 4)]
```

Tasks: You have been approached by Blue Nile to perform the following tasks:

1. Use data visualizations to explore how price is related to the other variables (carat, clarity, color, cut), as well as how the other variables may relate to each other.

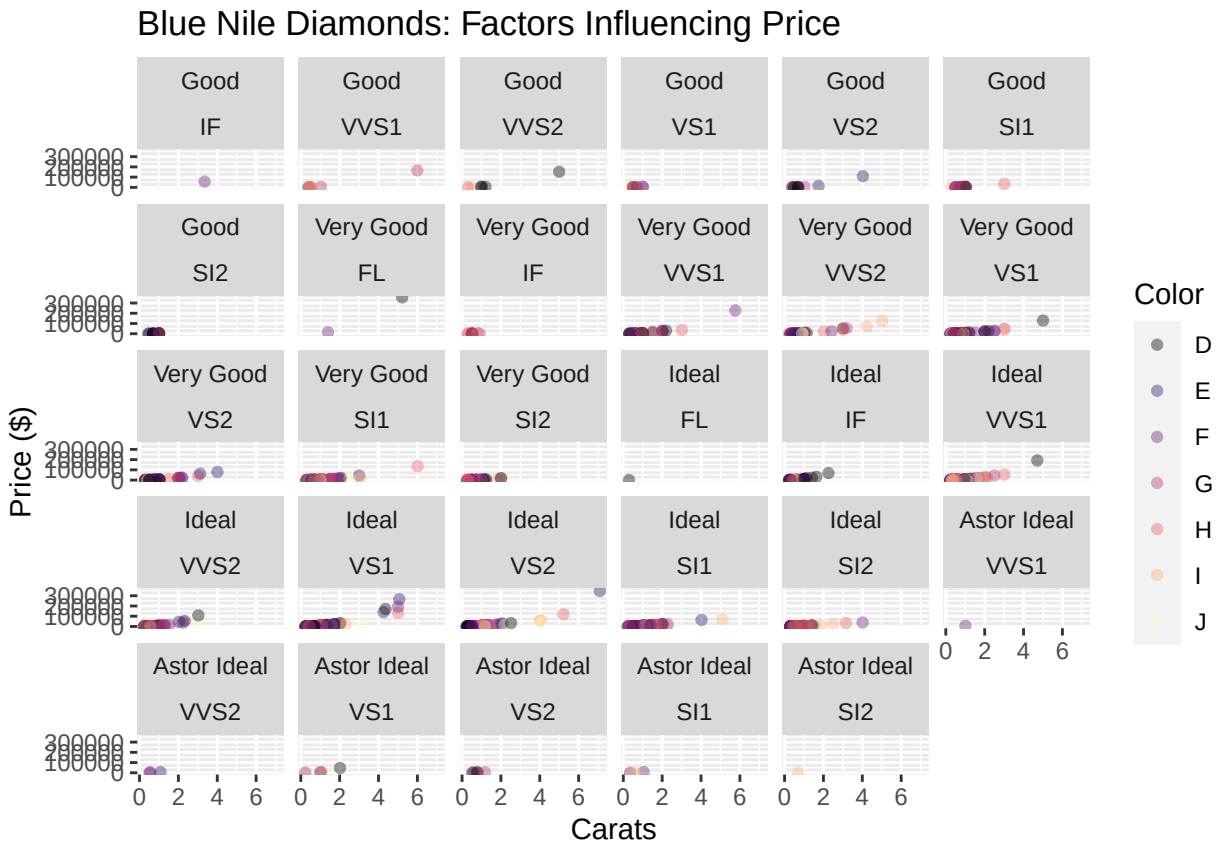
```
pairs(Data, lower.panel = NULL)
```




```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
Data %>%
  ggplot(aes(x=carat, y = price, color = color)) +
  geom_point(alpha = 0.4) +
  labs(x="Carats", y="Price ($)", title = "Blue Nile Diamonds: Factors Influencing Price") +
  guides(color = guide_legend(title = "Color"), label.position = "right") +
    scale_color_viridis(discrete = TRUE, option = "A") +
  scale_fill_viridis(discrete = TRUE) +
   facet_wrap(~ cut + clarity)
```




Address the various claims on the diamond education page on Blue Nile.

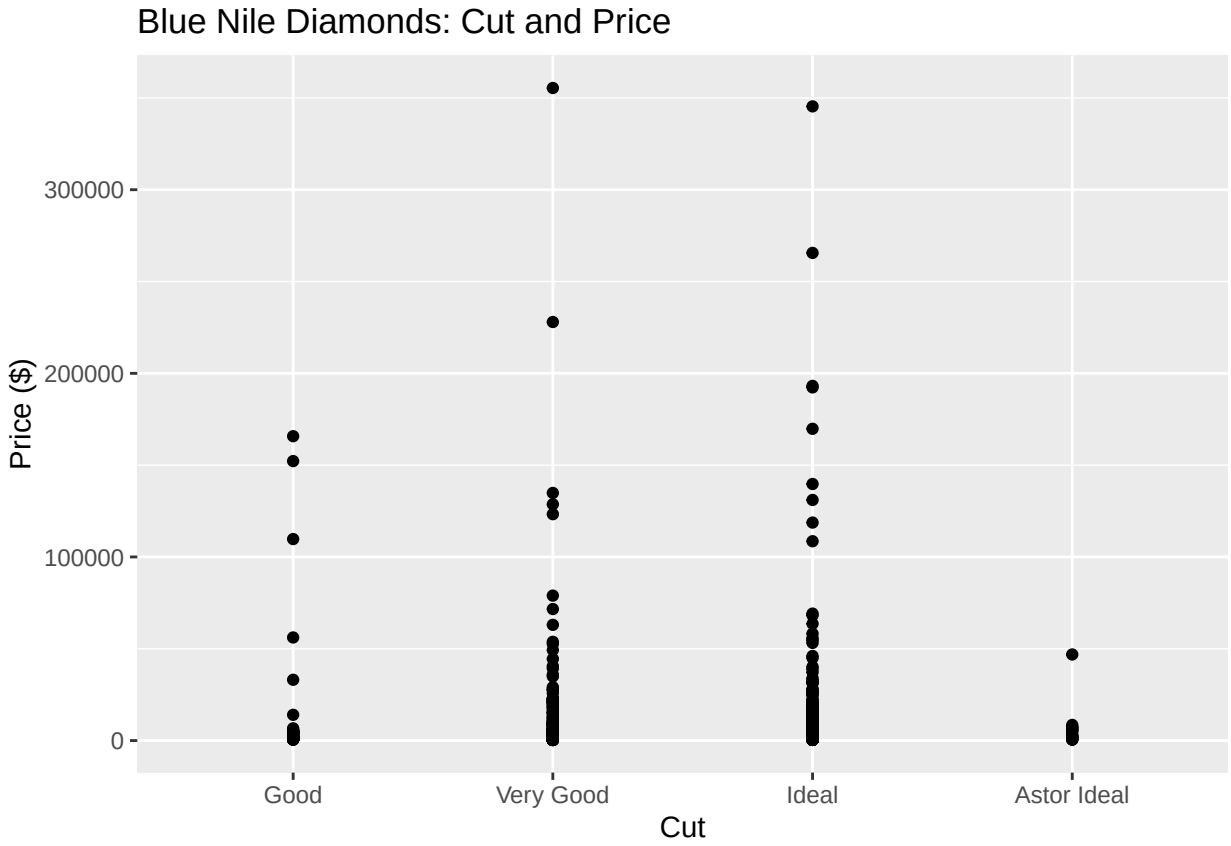
- Cut: https://www.bluenile.com/education/diamonds/cut +"A diamond's cut refers to how well-proportioned the dimensions of a diamond are, and how these surfaces, or facets, are positioned to create sparkle and brilliance. For example, what is the ratio of the diamond's diameter in comparison to its depth? These small, yet essential, factors determine the diamond's beauty and price."

Assertion above is that better cuts correlate with higher price. Let's check the scatterplot to see if that bears out in the data:

Increasing quality of diamond cut does not seem to have a linear relationship with price, contrary to the Blue Nile's claim.

```
Data %>%
  ggplot(aes(x=cut, y=price)) +
  geom_point() +
  geom_smooth(method="lm",se=F) +
  labs(x="Cut", y="Price ($)", title = "Blue Nile Diamonds: Cut and Price")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

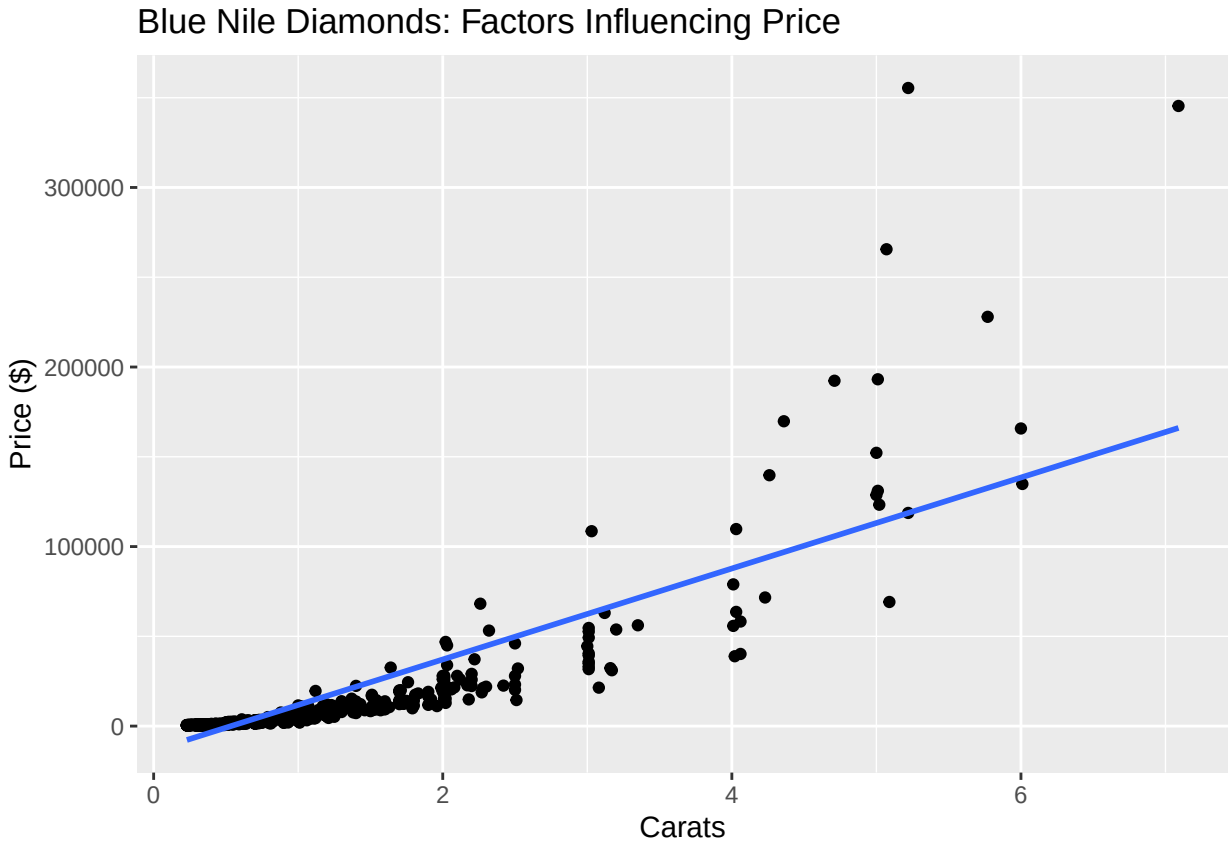



2. Fit an appropriate simple linear regression for price against carat.

First a scatterplot

```
Data %>%
  ggplot(aes(x=carat, y=price)) +
  geom_point() +
  geom_smooth(method="lm",se=F) +
  labs(x="Carats", y="Price ($)", title = "Blue Nile Diamonds: Factors Influencing Price")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




Fitting the model

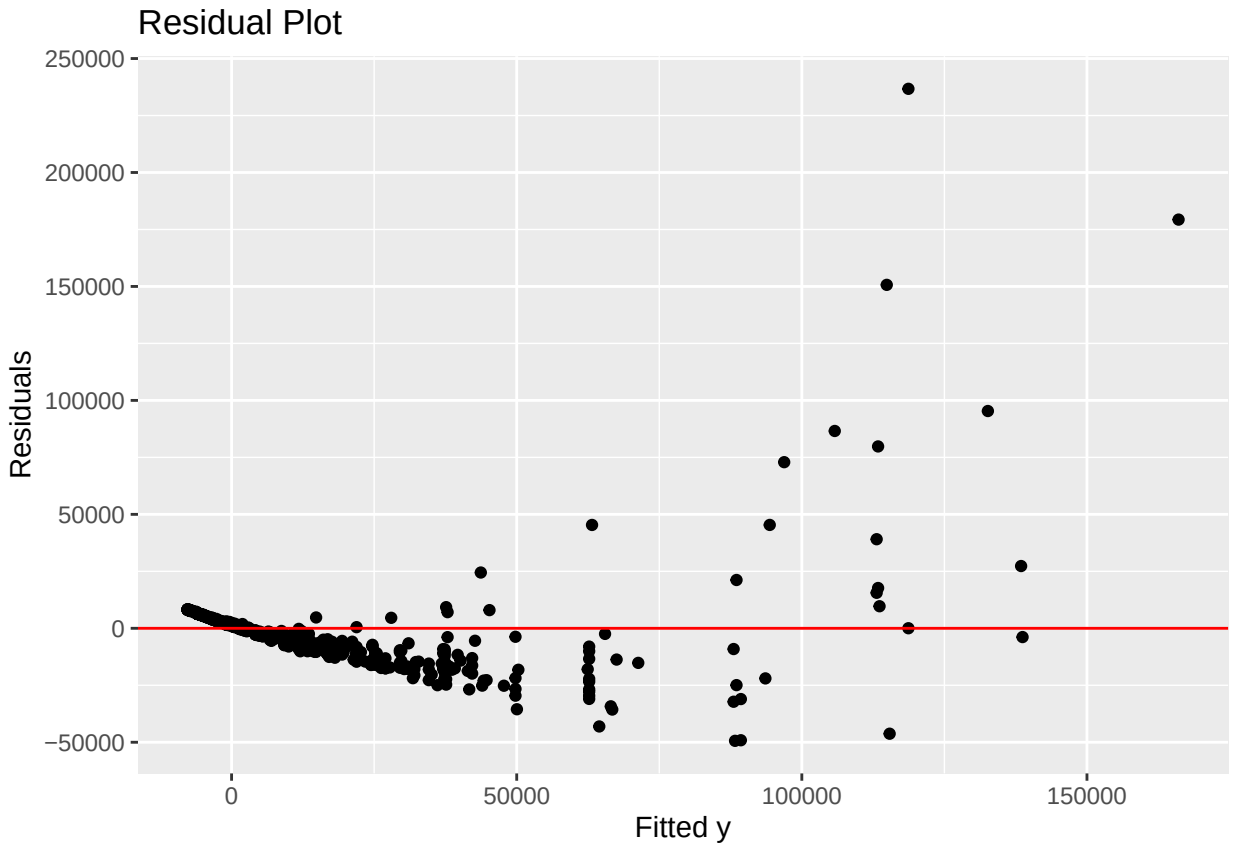
```
result<-lm(price~ carat, data=Data)
summary(result)
```

```
##
## Call:
## lm(formula = price ~ carat, data = Data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -49375  -5048   1867   4965 236711
##
## Coefficients:
##             Estimate Std. Error t value            Pr(>|t|)
## (Intercept) -13550.9      559.7  -24.21 <0.0000000000000002 ***
## carat        25333.9      494.4   51.24 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13560 on 1212 degrees of freedom
## Multiple R-squared:  0.6842, Adjusted R-squared:  0.6839
## F-statistic:  2625 on 1 and 1212 DF,  p-value: < 0.00000000000000022
```

Plotting the residuals.

Constant variance and mean of error = 0 assumptions do not appear to be met.

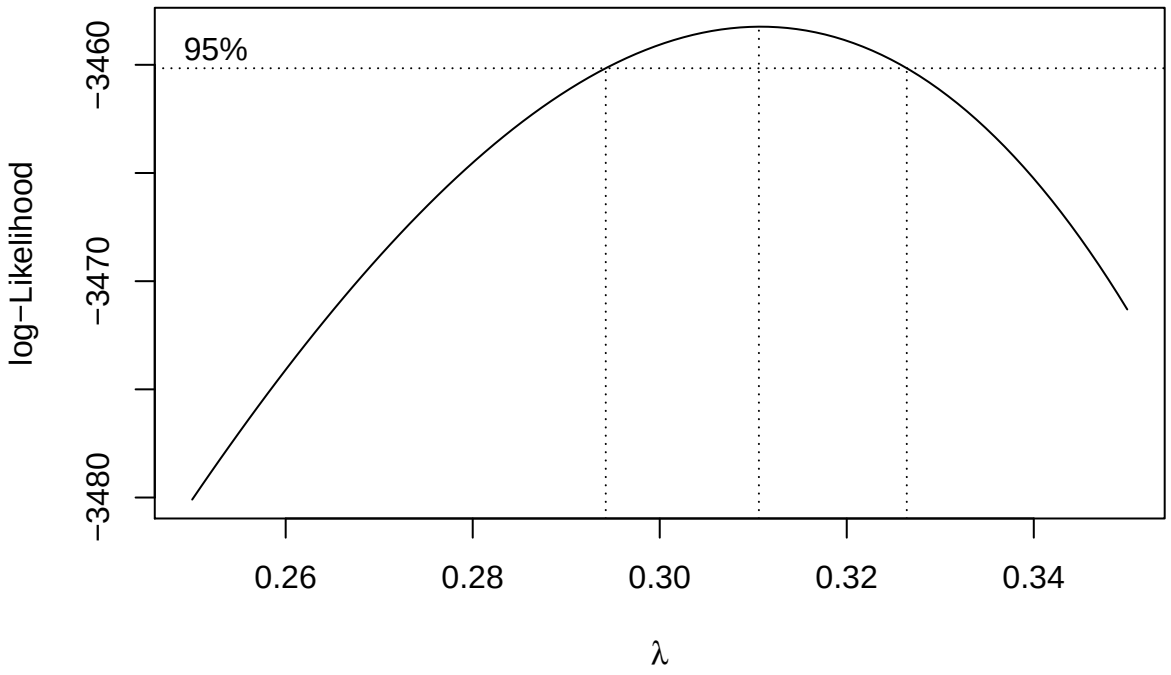
```
yhat<-result$fitted.values
res<-result$residuals
Data<-data.frame(Data,yhat,res)


ggplot(Data, aes(x=yhat,y=res))+
  geom_point()+
  geom_hline(yintercept=0, color="red")+
  labs(x="Fitted y", y="Residuals", title="Residual Plot")
```




Variance is definitely not contant so attempting to transform y first. Will start with boxcox plot to see what the optimal lambda may be.

Looks like a lambda of 0.31 would be appropriate.

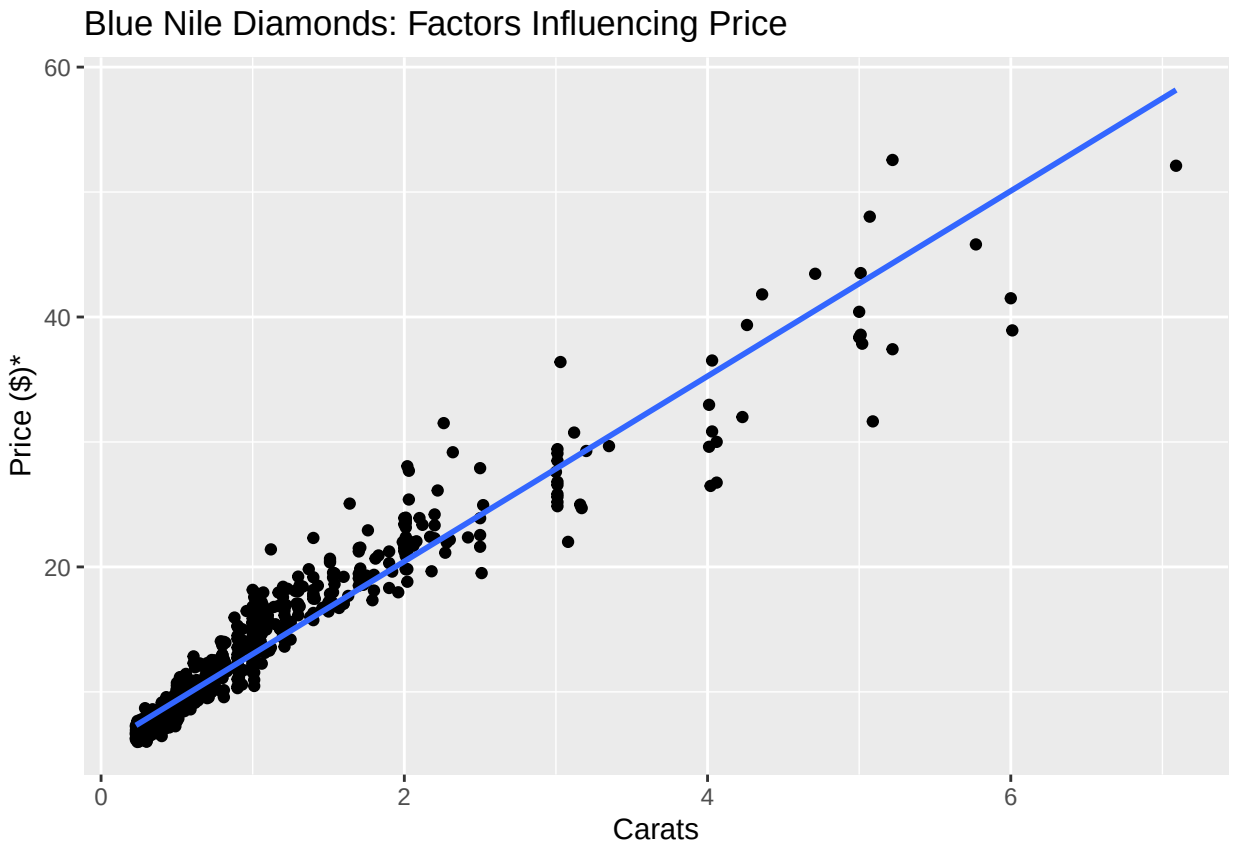
```
boxcox(result, lambda = seq(.25,0.35,1/100))
```




Transforming the y and plotting a new variables

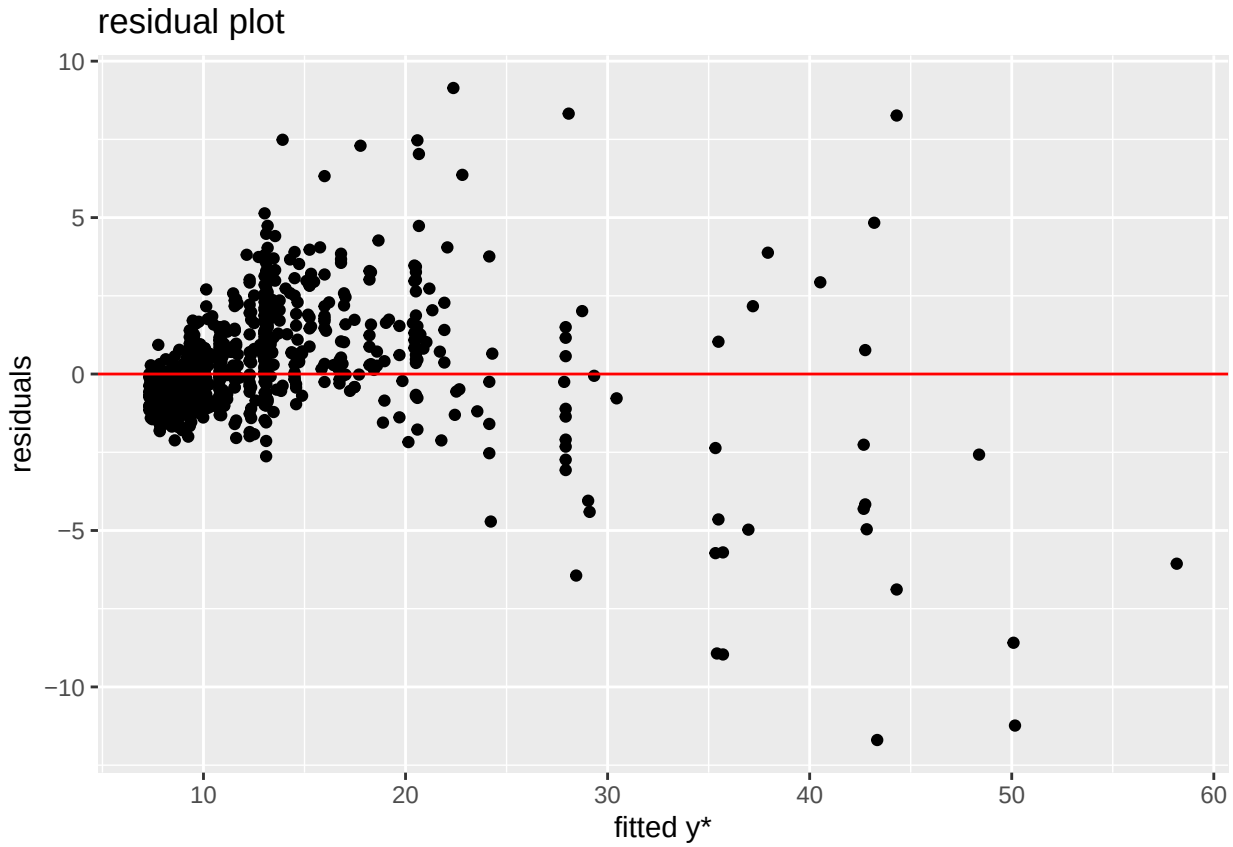
```
ystar <- (Data$price)^(0.31)
Data<-data.frame(Data,ystar)

Data %>%
  ggplot(aes(x=carat, y=ystar)) +
  geom_point() +
  geom_smooth(method="lm",se=F) +
  labs(x="Carats", y="Price ($)*", title = "Blue Nile Diamonds: Factors Influencing Price")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



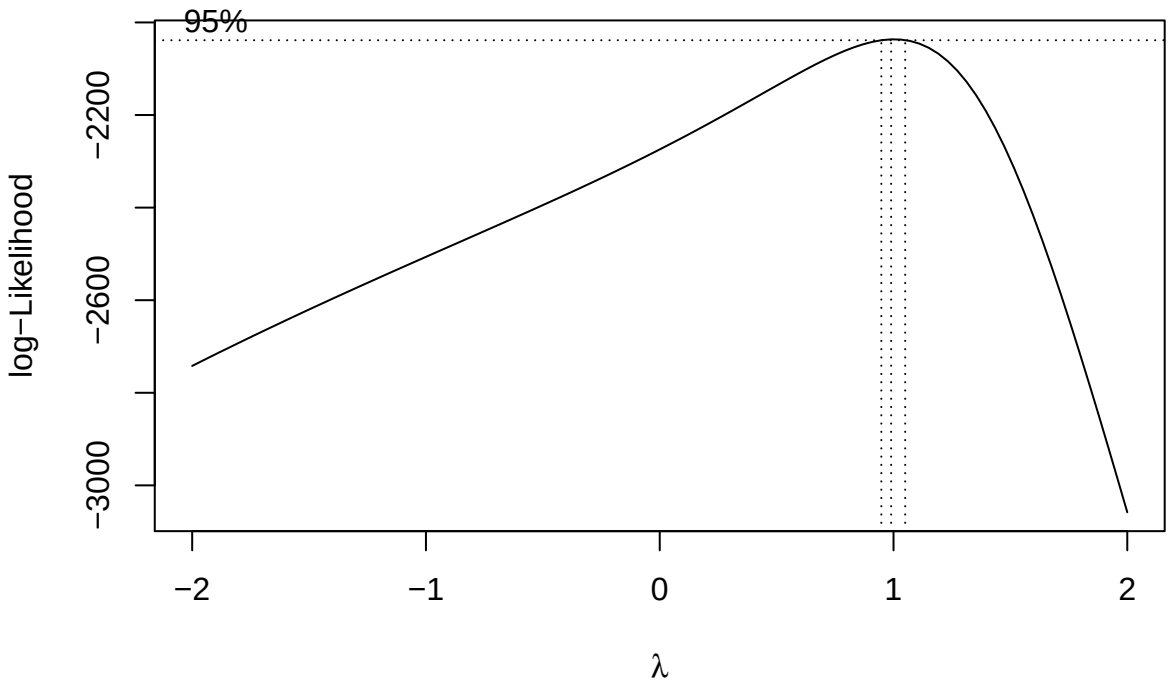

New residual plot:

```
result.ystar<- lm(ystar~carat, data=Data)

#storing fitted y residuals
yhat2 <- result.ystar$fitted.values
res2 <- result.ystar$residuals
#add to data frame

Data <-data.frame(Data,yhat2,res2)

#residual plot
Data %>%
  ggplot(aes(x=yhat2,y=res2)) +
  geom_point() +
  geom_hline(yintercept=0,color="red")+
  labs(x="fitted y*", y = "residuals", title="residual plot")
```




variance looks better but mean of errors still not equal to zero over x so will attempt to transform the x variable. Given the curved appearance, will try a square root transformation. Confirming that the box cox plot looks better. Now I see that confidence interval includes 1. so next step will be to consider transforming x.

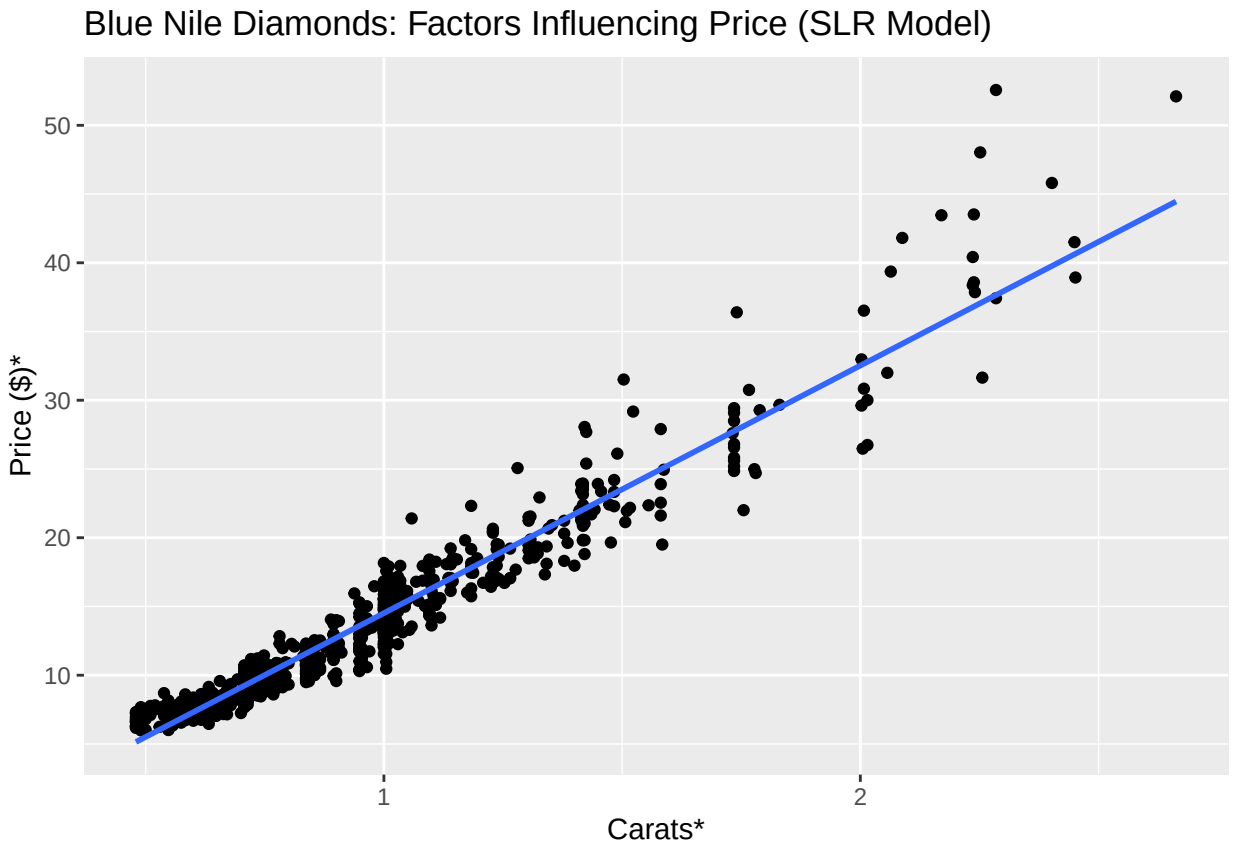
```
boxcox(result.ystar)
```




```
#boxcox(result.ystar, lambda = seq(-1,5.5,1/10))
```

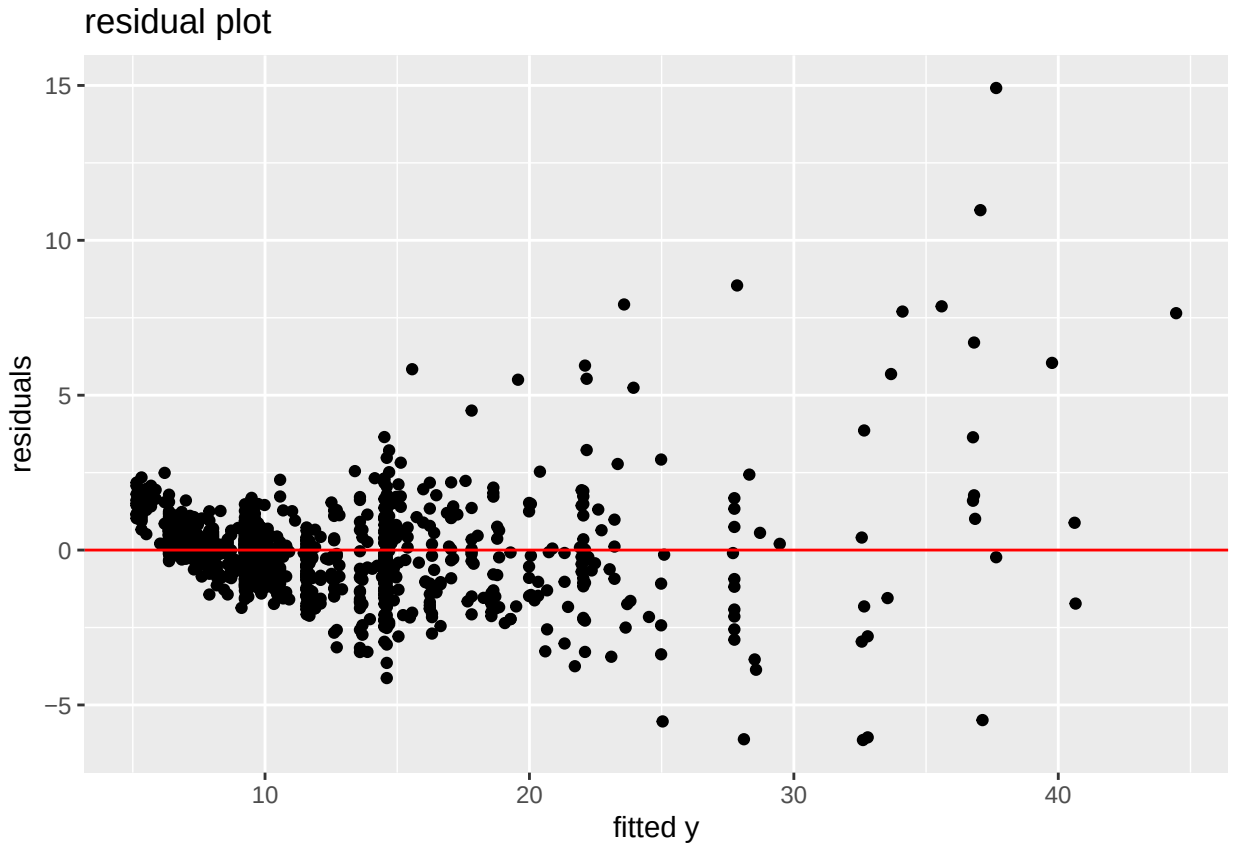
```
xstar <- sqrt(Data$carat)
Data<-data.frame(Data,xstar)

Data %>%
  ggplot(aes(x=xstar, y=ystar)) +
  geom_point() +
  geom_smooth(method="lm",se=F) +
  labs(x="Carats*", y="Price ($)*", title = "Blue Nile Diamonds: Factors Influencing Price (SLR Model)"
```

```
## `geom_smooth()` using formula 'y ~ x'
```




Fitting new model and creating yet another residual plot:

```
result.ystar.xstar<- lm(ystar~xstar, data=Data)

#storing fitted y residuals
yhat3 <- result.ystar.xstar$fitted.values
res3 <- result.ystar.xstar$residuals
#add to data frame

Data <-data.frame(Data,yhat3,res3)

#residual plot
Data %>%
  ggplot(aes(x=yhat3,y=res3)) +
  geom_point() +
  geom_hline(yintercept=0,color="red")+
  labs(x="fitted y", y = "residuals", title="residual plot")
```




Not perfect but overall better. Summarizing the model.

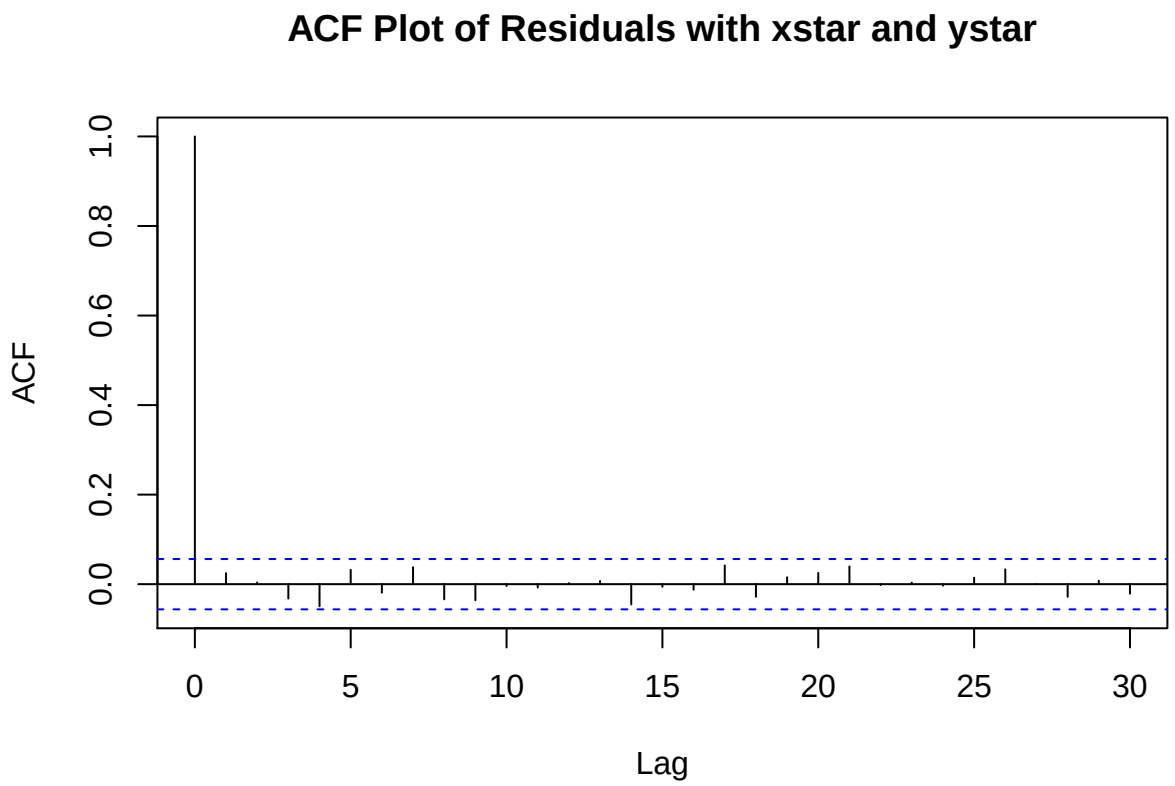
```
summary(result.ystar.xstar)
```

```
##
## Call:
## lm(formula = ystar ~ xstar, data = Data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1320 -0.6377  0.0373  0.5315 14.9172
##
## Coefficients:
##             Estimate Std. Error t value            Pr(>|t|)
## (Intercept)  -3.4936     0.1137  -30.73 <0.0000000000000002 ***
## xstar        18.0085     0.1261  142.87 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.436 on 1212 degrees of freedom
## Multiple R-squared:  0.9439, Adjusted R-squared:  0.9439
## F-statistic: 2.041e+04 on 1 and 1212 DF,  p-value: < 0.00000000000000022
```

ACF Plot

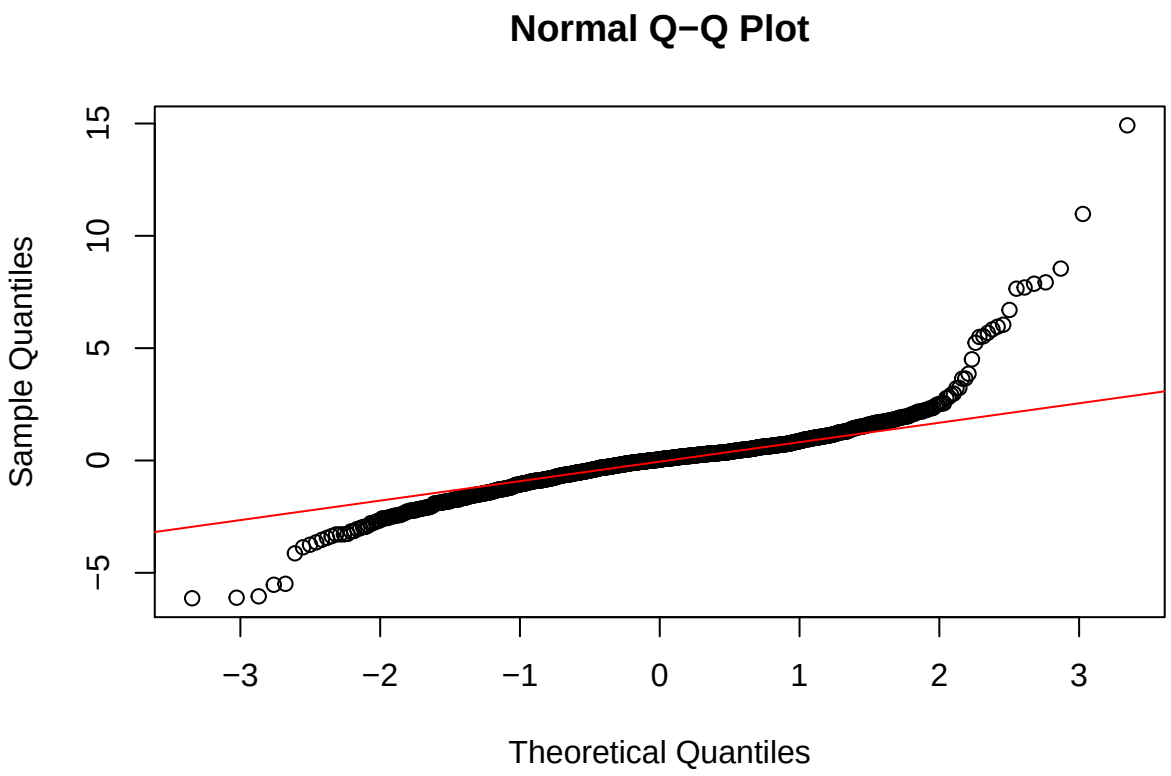
Assumption met.

```r
acf(res3,main="ACF Plot of Residuals with xstar and ystar")
```




Normality assumption is not met but this may be the least important.

```r
qqnorm(res3)
qqline(res3,col="red")
```


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
15
10
5
0
−5
−3
−2
−1
0
1
2
3