Project1_Carat

Alanna Hazlett

2024-03-14

```
library(tidyverse)
```

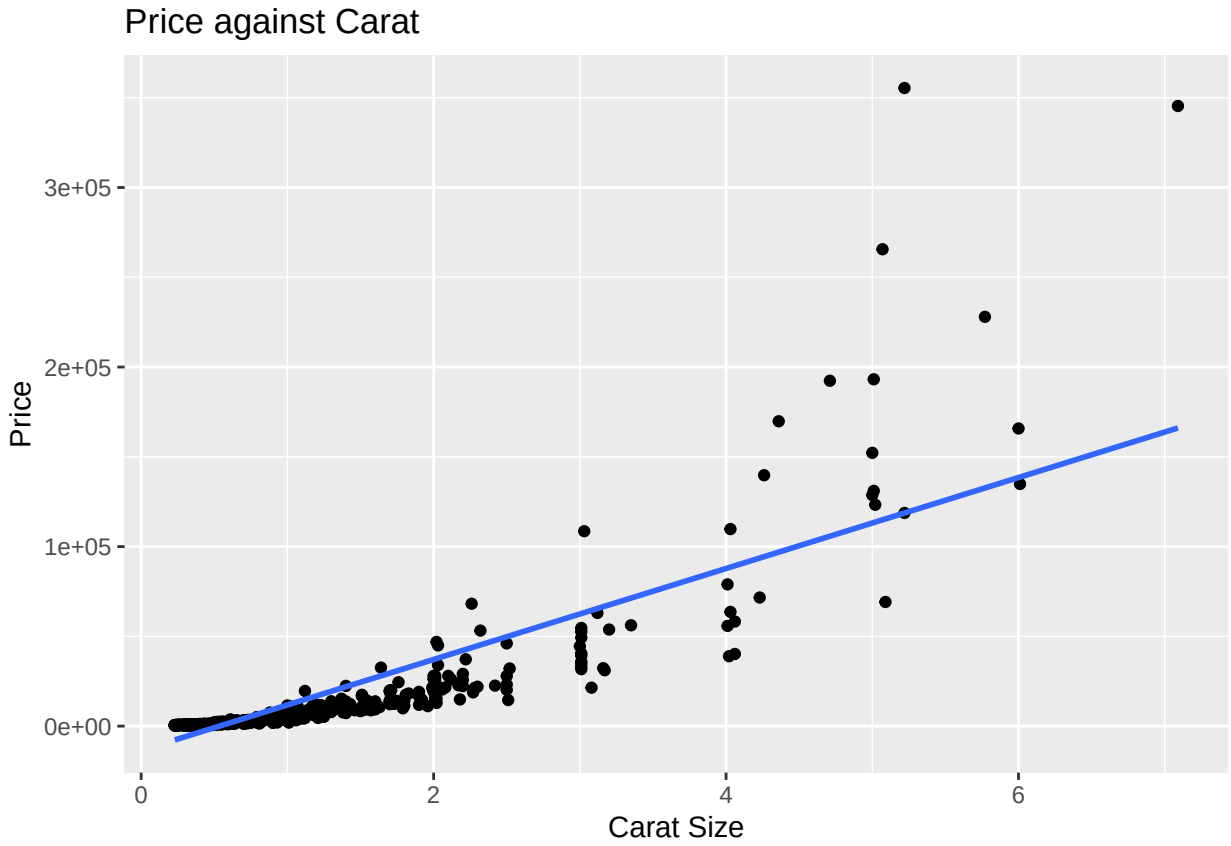
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
Data<-read.csv("diamonds4.csv", header=TRUE)
ggplot2::ggplot(Data, aes(x=carat,y=price))+
  geom_point()+
  geom_smooth(method = "lm", se=FALSE)+
  labs(x="Carat Size", y="Price", title="Price against Carat")
```

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

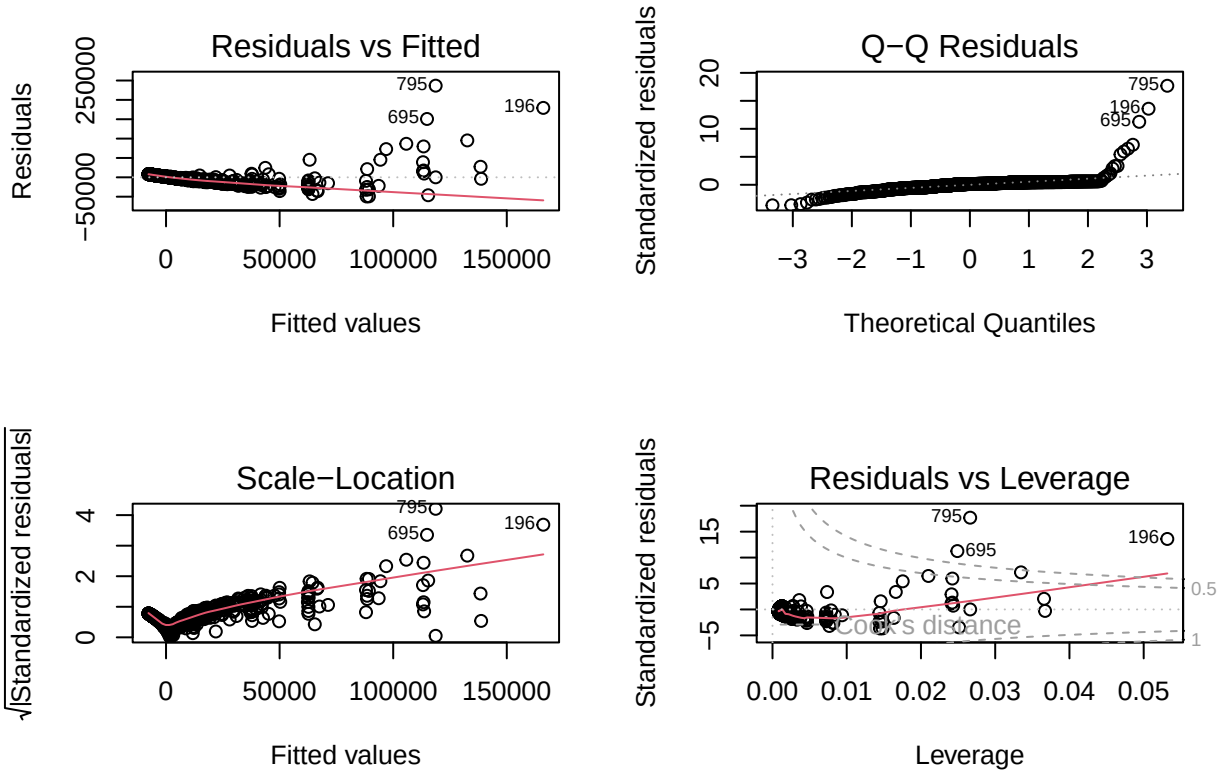


Comment on the appearance of the plot. Do any assumptions for simple linear regression appear to be violated? If so, which ones?

The data points do not visually appear to be in a linear pattern, they appear to be in an exponential pattern. Assumption 1 does not appear to be met. The data points are not evenly spread on either side of the regression line. From 0 to 1 the majority of the data points fall above the regression line, from 1 to about 4 the majority fall underneath the regression line, and from 4 to about 7 the majority of the data points fall above the regression line.

Assumption 2 does not appear to be met. The variance, the amount of vertical spread of the data for each value of the predictor, starts very small and grows larger as we increase in carat size. Because the variance is increasing from left to right we will transform y with a lambda value of less than one.

```
result_carat<-lm(price~carat,data=Data)
par(mfrow=c(2,2))
plot(result_carat)
```

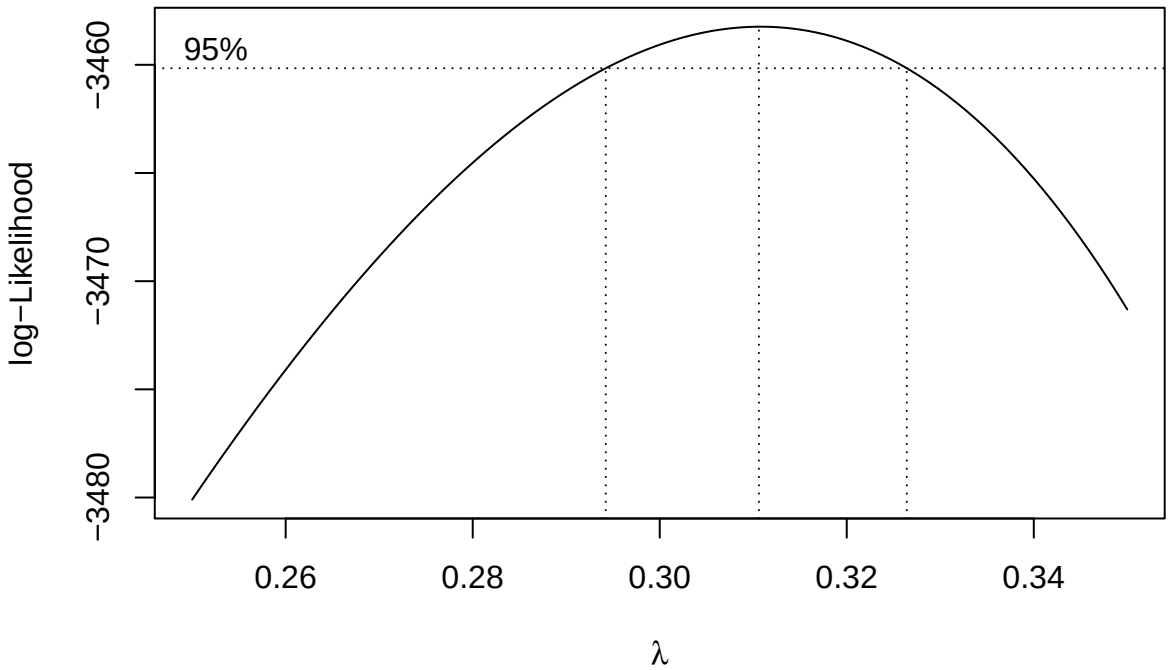


In our Residuals vs Fitted plot we can see that Assumption 1 is not met, as the errors do not have a mean of zero for each value of the predictor. We can also see that the errors do not have a constant variance for each value of the predictor, so Assumption 2 is not met. The variance of the errors is in-fact increasing from left to right. In the Q-Q Residuals plot we can see that the majority of the observations are normally distributed, but on the right had side we see that the observations stray further away from their theoretical residual value. Assumption 4 is somewhat met.

Based on our scatterplot and the residual plots we will want to transform our y variable first, as it will affect the outcome of Assumption 1 and Assumption 2.

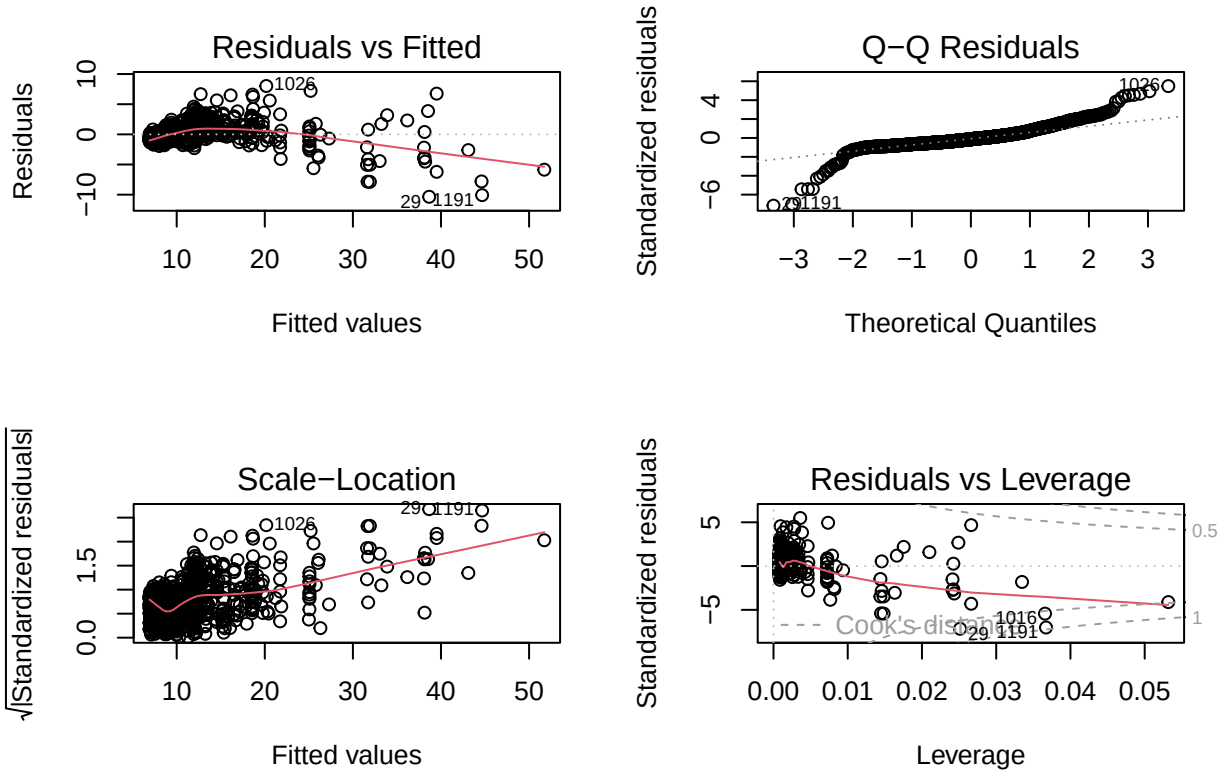
For the simple linear regression create a Box Cox plot. What transformation, if any, would you apply to the response variable? Briefly explain.

```
MASS::boxcox(result_carat, lambda = seq(0.25, 0.35, 0.01))
```



Our 95% confidence interval for lambda lies roughly between 0.295 and 0.325. This helps us to determine what value of lambda to use to transform our y variable.
We will transform our y variable with y^ 0.3, a value within our confidence interval.

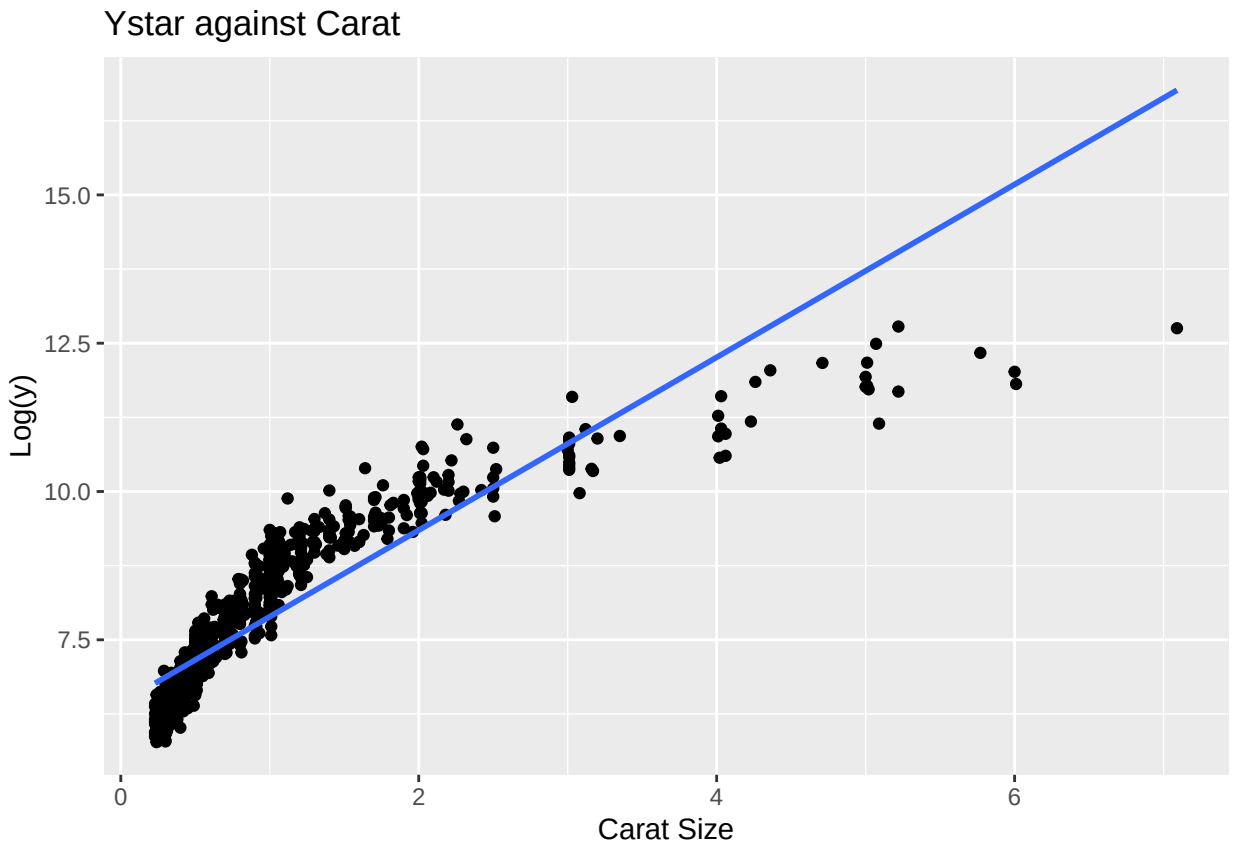
```
ystar2<-(Data$price) ** 0.3
Data<-data.frame(Data,ystar2)
ystar2_result<-lm(ystar2~carat,data=Data)
par(mfrow=c(2,2))
plot(ystar2_result)
```



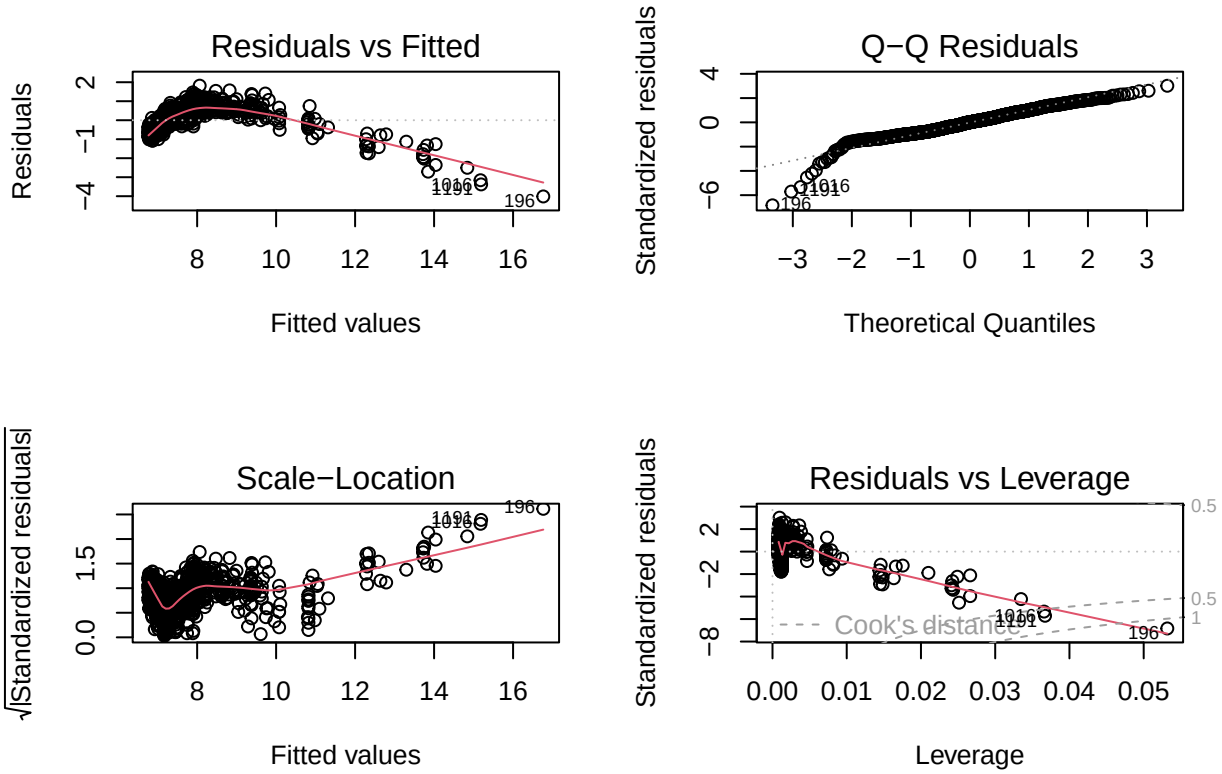
The first transformation completed is y^0.3, in an effort to stabilize the variance.
Our assumptions are still not met after this transformation. Let's try a value for lambda that is a close whole number to our 95% confidence interval, 0.

```
ystar<-log(Data$price)
Data<-data.frame(Data,ystar)
ystar_result<-lm(ystar~carat,data=Data)
ggplot2::ggplot(Data, aes(x=carat,y=ystar))+
  geom_point()+
  geom_smooth(method = "lm", se=FALSE)+
  labs(x="Carat Size", y="Log(y)", title="Ystar against Carat")
```

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



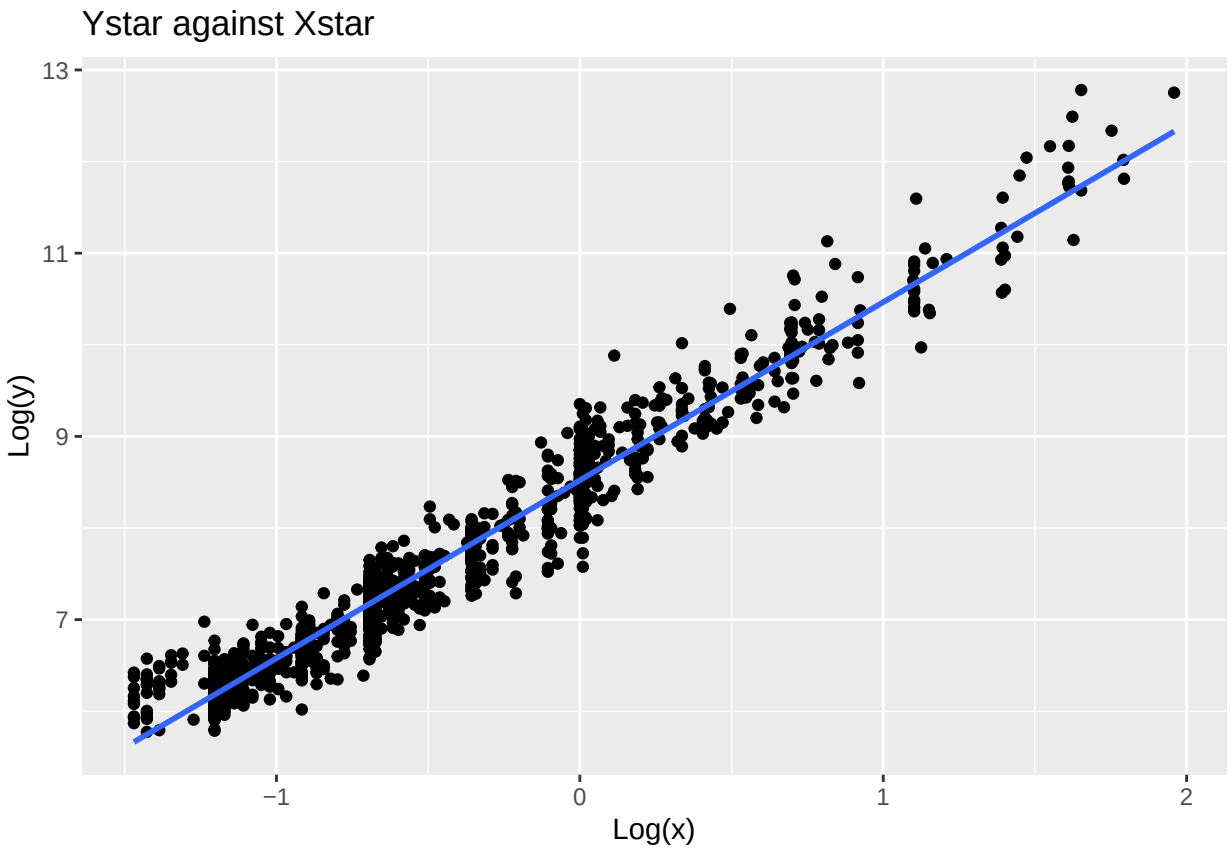

```
par(mfrow=c(2,2))
plot(ystar_result)
```



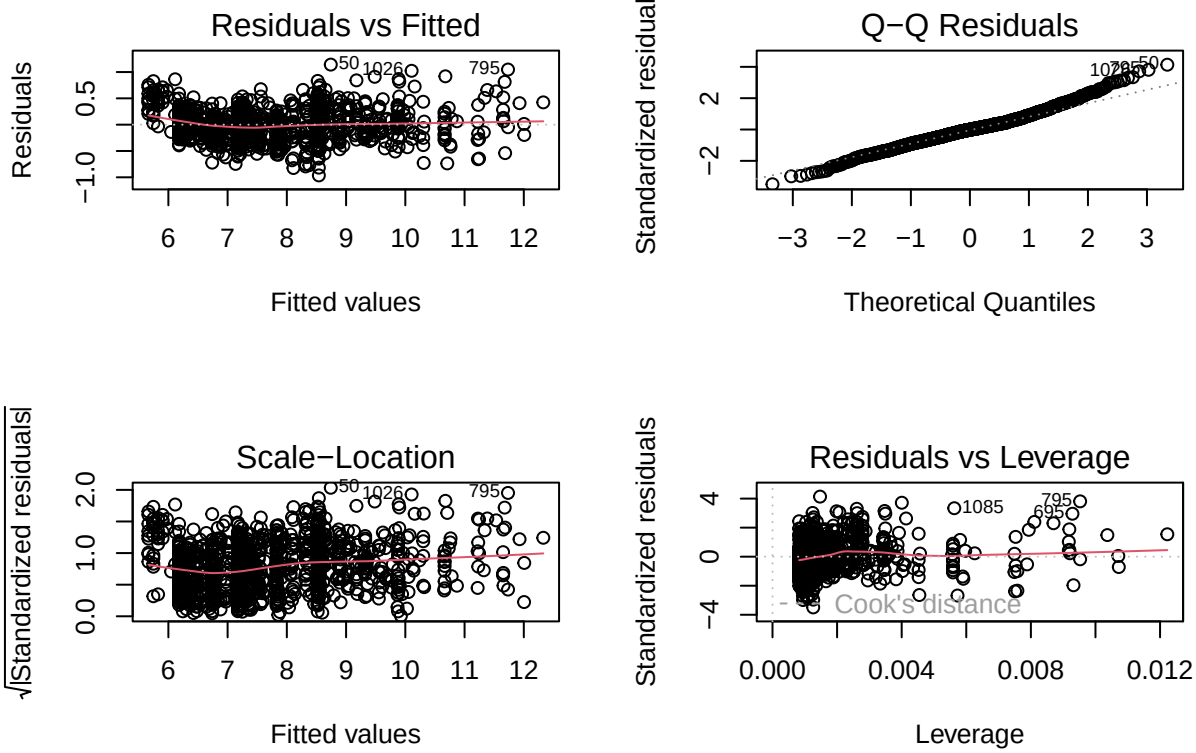
After the log transformation on the y variable we can see that the errors do not have a mean of zero for each predictor, so assumption 1 is not met. However, the variance of our errors is much improved and relatively constant from left to right, so assumption 2 is met. We can also see an improvement in the distribution of our observations in the Q-Q Residuals plot, so assumption 4 is met. In the scatterplot we can see that our data points still do not reflect a linear relationship, so we will transform our predictor variable, carat. Visually we can see that the relationship appears to be a log relationship, so we will perform a log transformation.

```
xstar<-log(Data$carat)
Data<-data.frame(Data,xstar)
xstar_result<-lm(ystar~xstar,data=Data)
ggplot2::ggplot(Data, aes(x=xstar,y=ystar))+
  geom_point()+
  geom_smooth(method = "lm", se=FALSE)+
  labs(x="Log(x)", y="Log(y)", title="Ystar against Xstar")
```

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




```
par(mfrow=c(2,2))
plot(xstar_result)
```

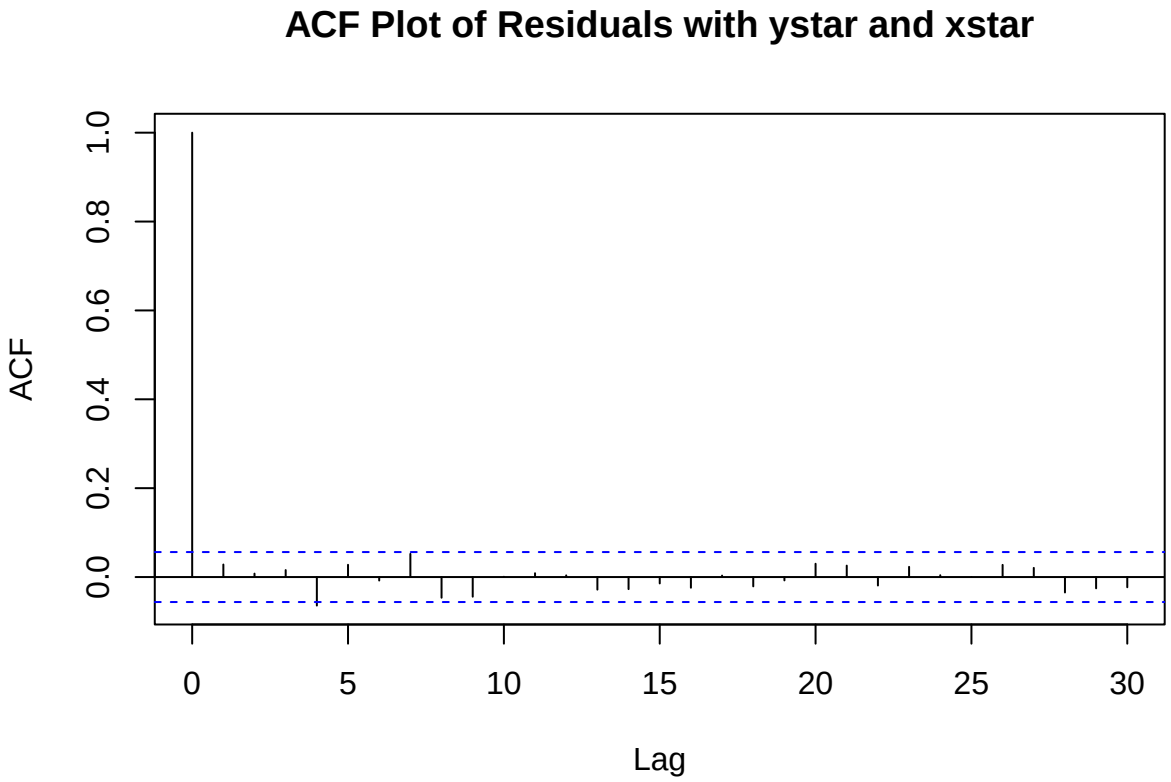


```r
summary(xstar_result)
```

```
##
## Call:
## lm(formula = ystar ~ xstar, data = Data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.96394 -0.17231 -0.00252  0.14742  1.14095
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.521208   0.009734   875.4   <2e-16 ***
## xstar       1.944020   0.012166   159.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2761 on 1212 degrees of freedom
## Multiple R-squared:  0.9547, Adjusted R-squared:  0.9546
## F-statistic: 2.553e+04 on 1 and 1212 DF,  p-value: < 2.2e-16
```

Now in our scatter plot and residual plots we can see that the errors have a mean of 0 and a constant variance for each value of the predictor. We can also see that the errors are normally distributed. Assumptions 1, 2, and 4 respectively met.
Contextual comments on how the SLR model inform us how price of diamonds are related to carat:
Since we performed a log transformation on both the predictor and response variable we are still able to interpret our regression coefficients. Beta1 = 1.944. This means that for a one percent increase in carat the prices increases by 1.944 percent.
We want to confirm our belief that the observations (diamonds) are independent of eachother and that this is truly a random sample.

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

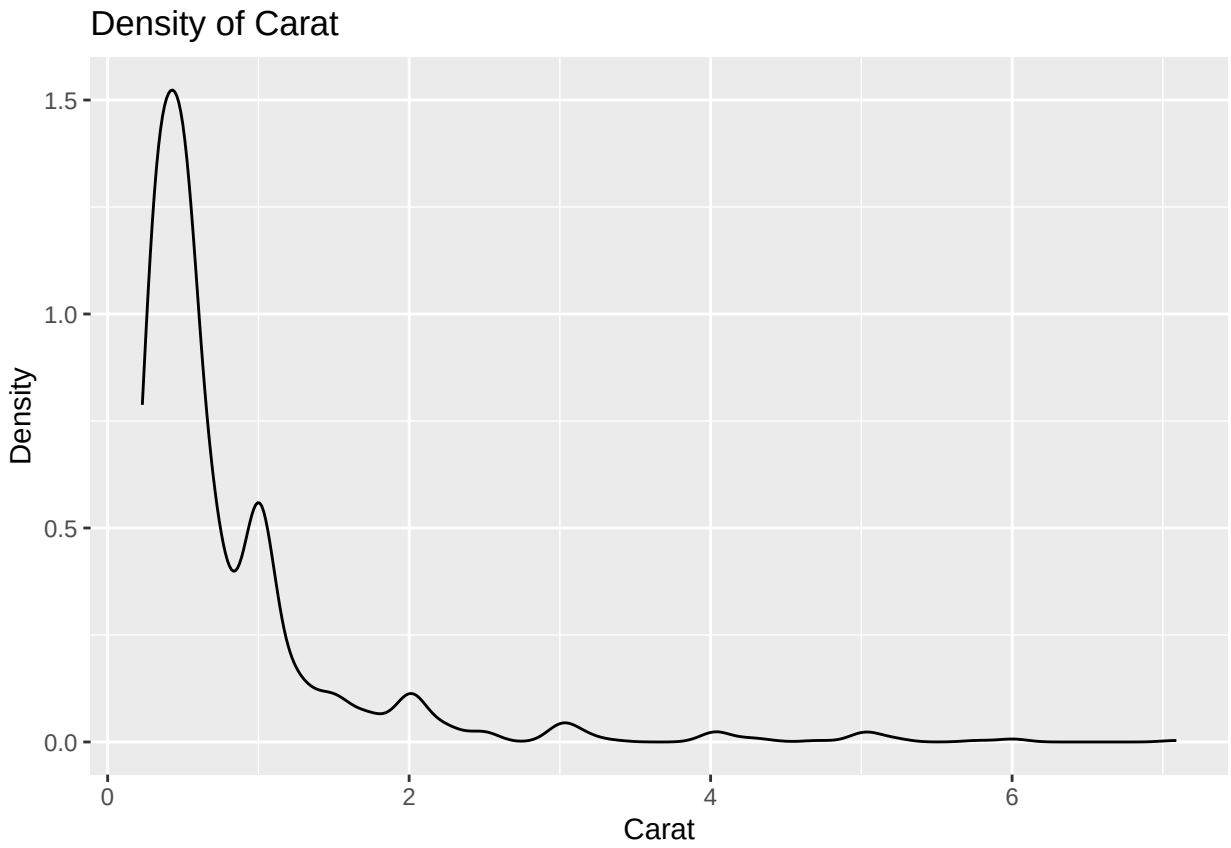


Our ACF Plot confirms our belief that the observations are uncorrelated and the correlations between the vector of observations and lagged versions of these observations are very near zero.

Claims to investigate regarding carat:
Carat has the biggest effect on price. Does buying shy of a carat save you money? Cut is the most imporant factor for appearance, even against carat.

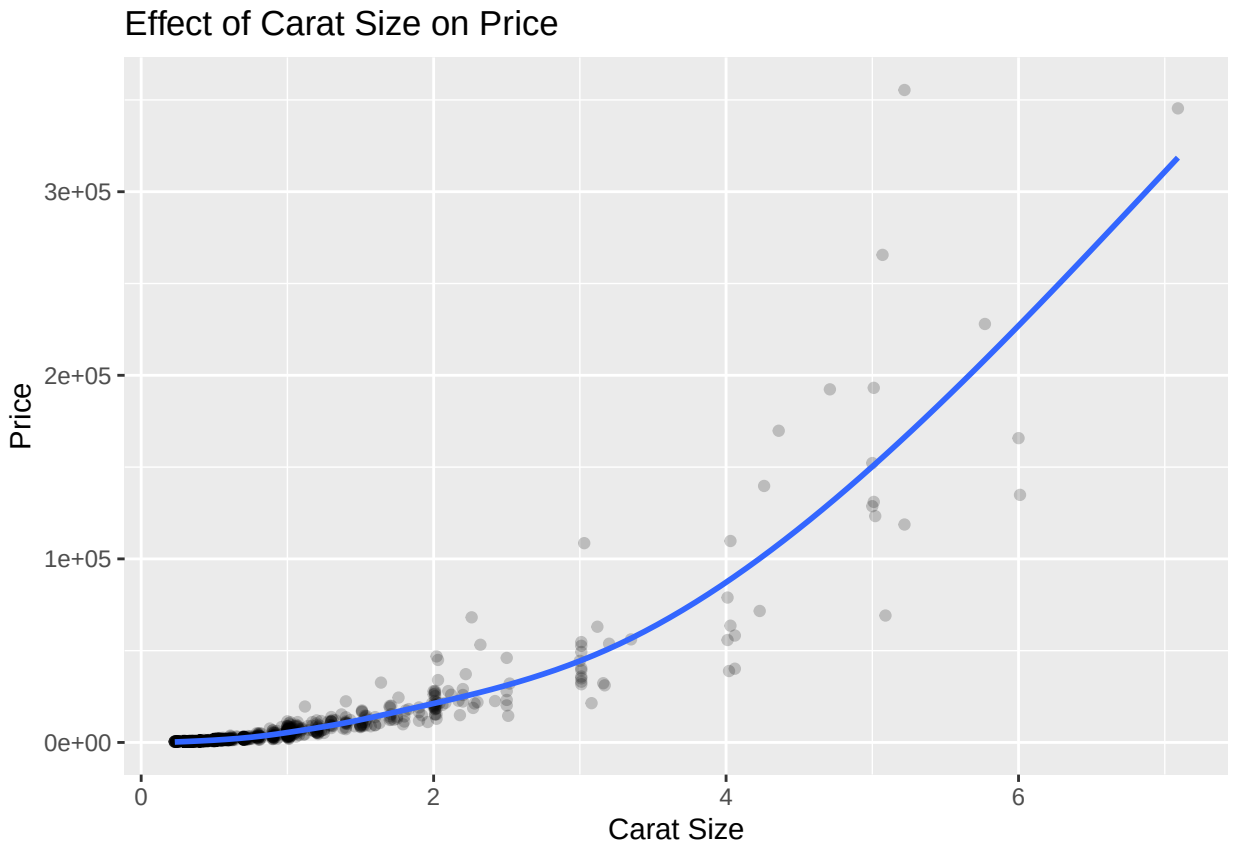
```
# Refactor the variables to be in categorical order
Data$color <- factor(Data$color, levels=c('D', 'E', 'F', 'G', 'H', 'I', 'J'))
Data$cut <- factor(Data$cut, levels=c('Astor Ideal', 'Ideal', 'Very Good', 'Good'))
Data$clarity <-factor(Data$clarity, levels=c('FL', 'IF', 'VVS1', 'VVS2', 'VS1', 'VS2', 'SI1', 'SI2'))
```

```
ggplot(Data,aes(x=carat))+ geom_density()+ labs(x="Carat",y="Density",title="Density of Carat")
```

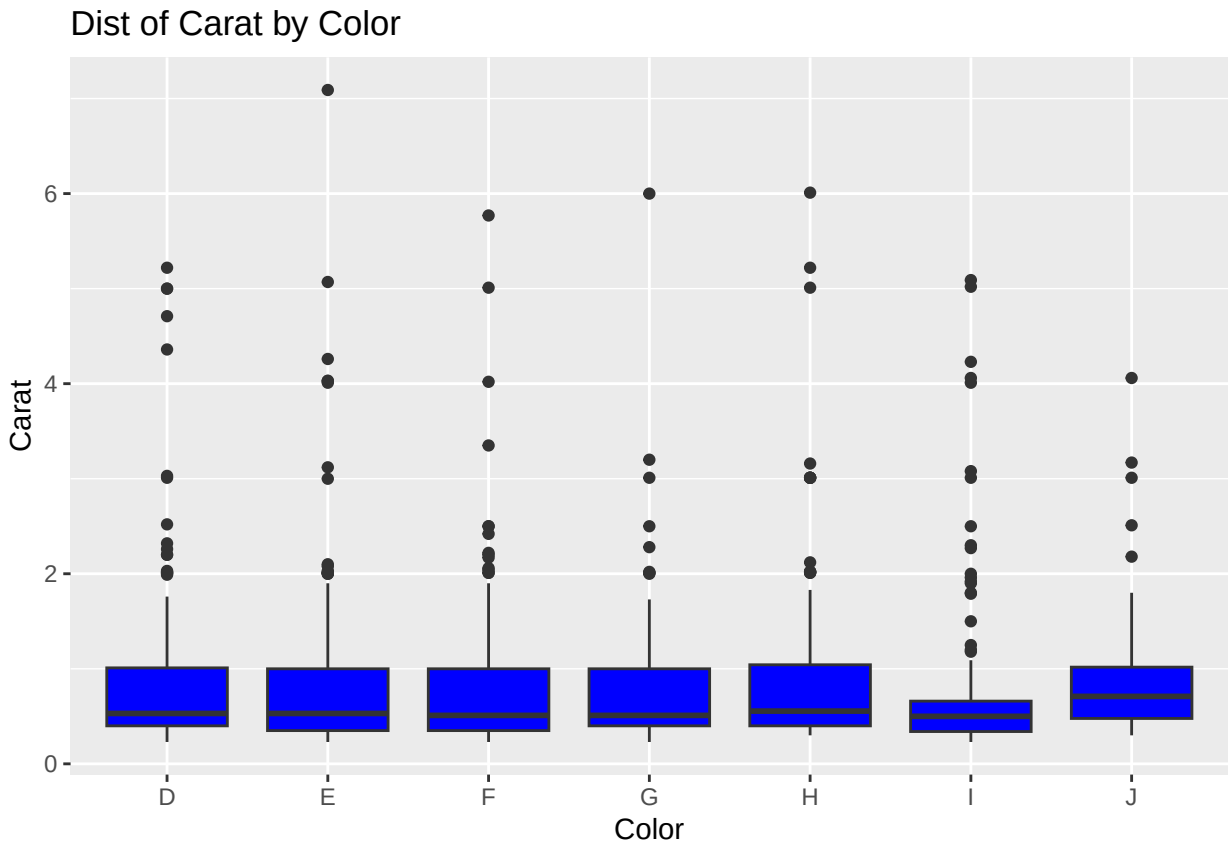



```
ggplot2::ggplot(Data, aes(x=carat,y=price))+
  geom_point(alpha=0.2)+
  geom_smooth(se=FALSE)+
  labs(x="Carat Size", y="Price", title="Effect of Carat Size on Price")
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

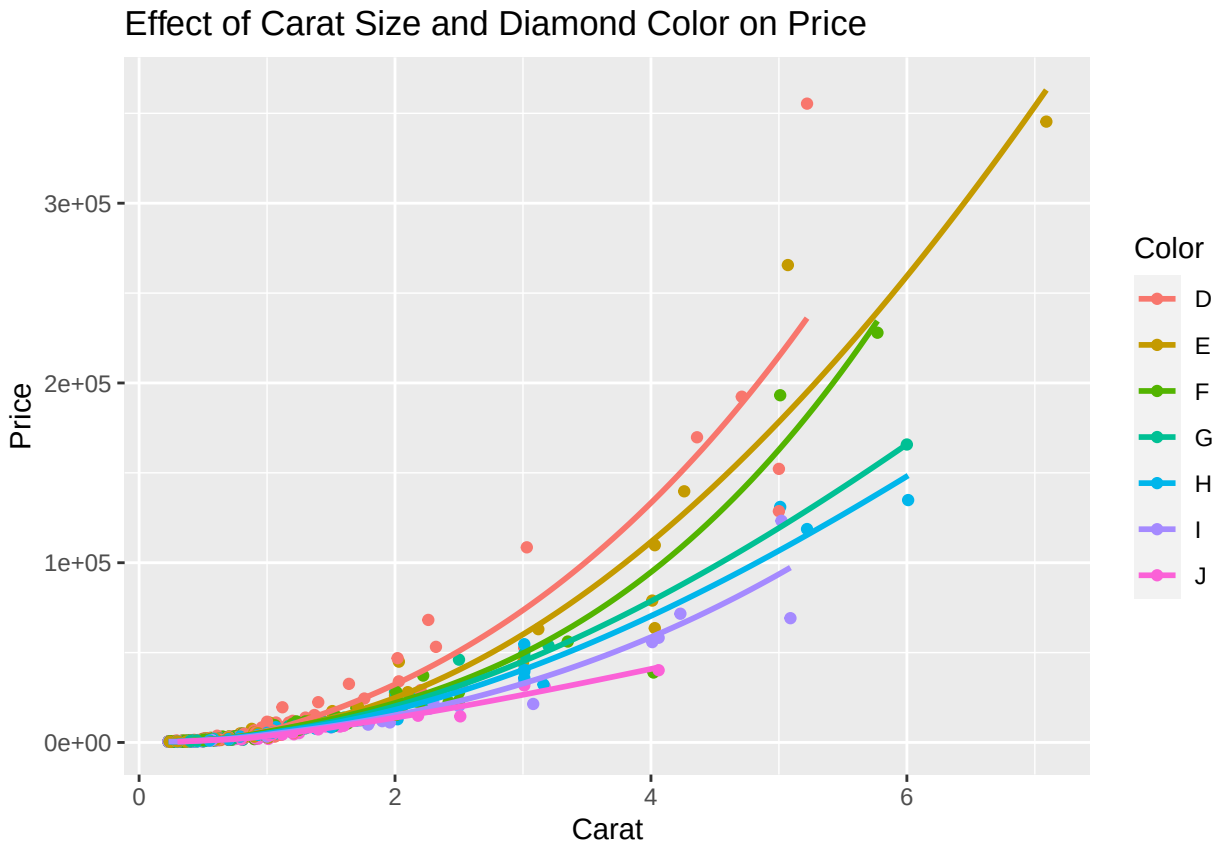



```
ggplot(Data, aes(x=color, y=carat))+
  geom_boxplot(fill="Blue")+
  labs(x="Color", y="Carat", title="Dist of Carat by Color")
```

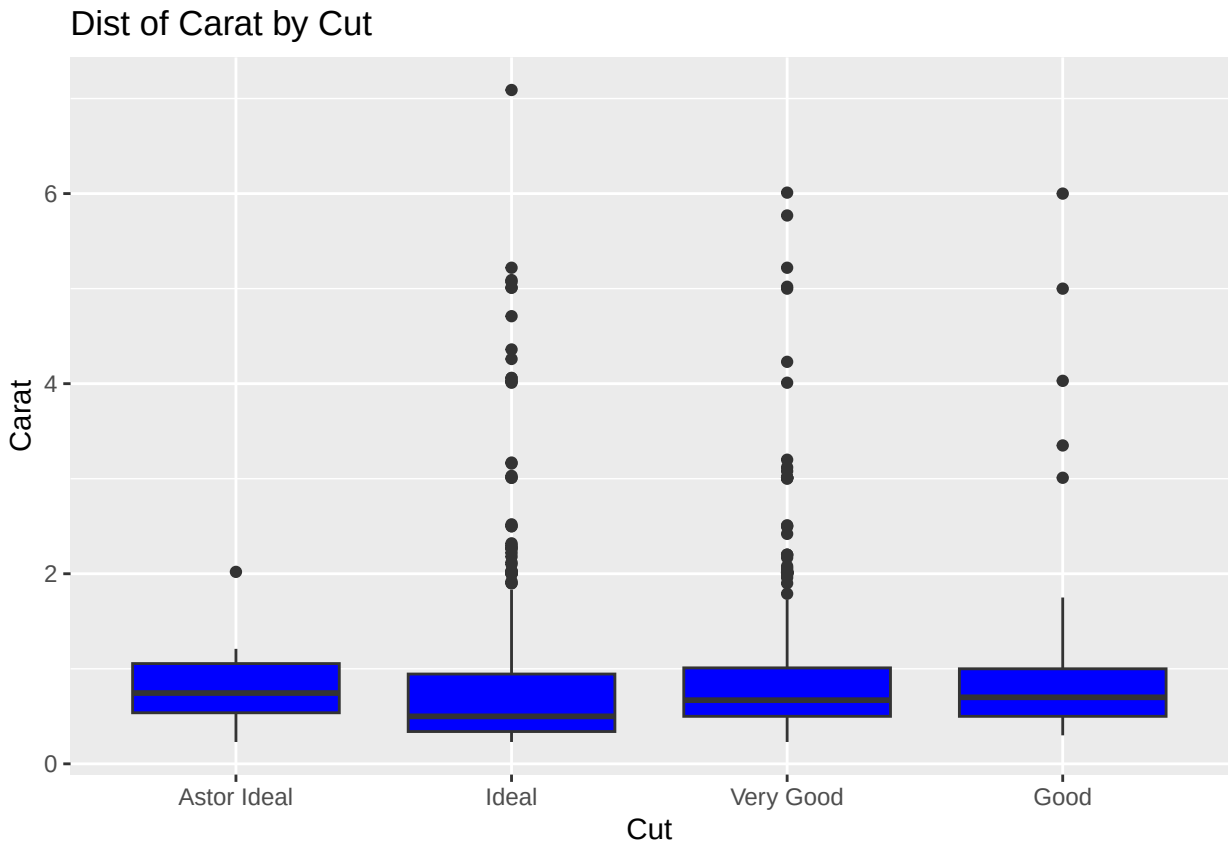



```
ggplot(Data, aes(x=carat, y=price, color=color)) +
geom_point() +
geom_smooth(se=FALSE)+
labs(x="Carat", y="Price", title="Effect of Carat Size and Diamond Color on Price", color = "Color")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

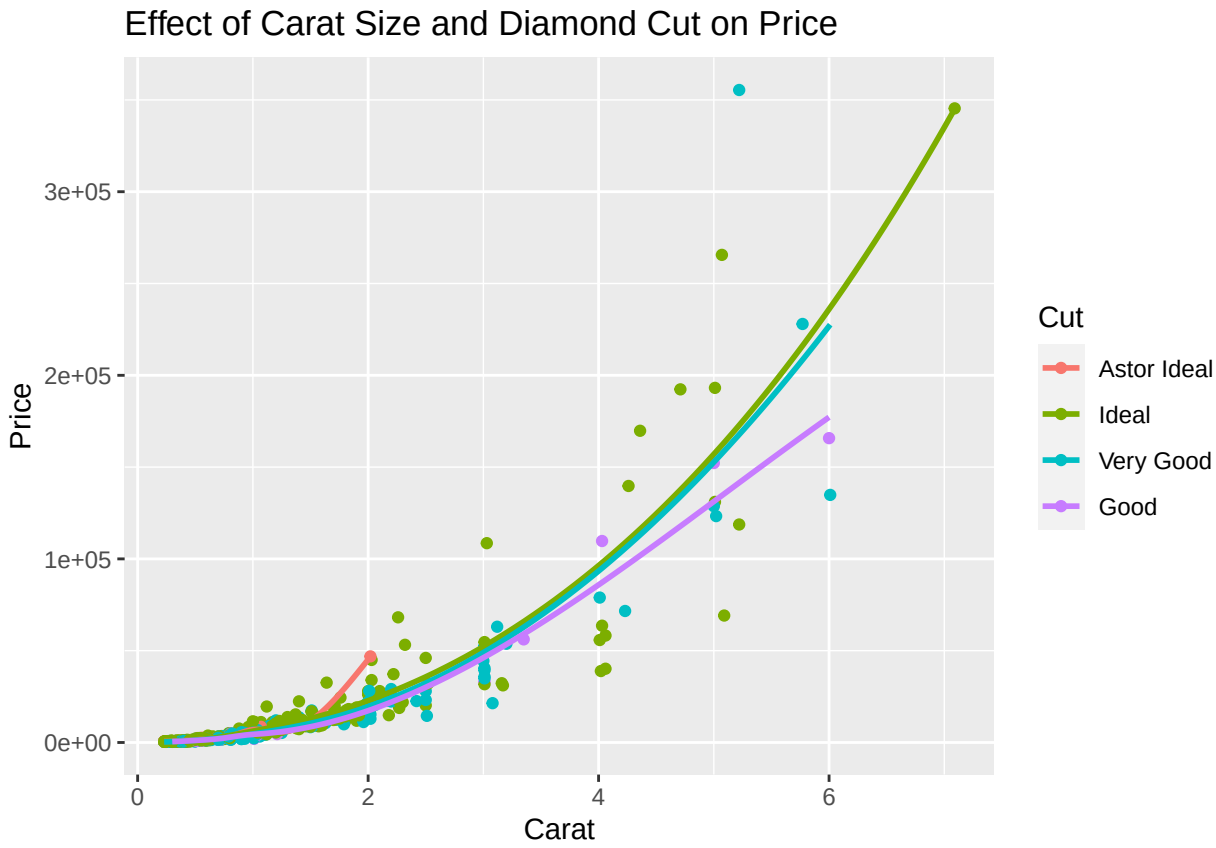



```
ggplot(Data, aes(x=cut, y=carat))+
  geom_boxplot(fill="Blue")+
  labs(x="Cut", y="Carat", title="Dist of Carat by Cut")
```

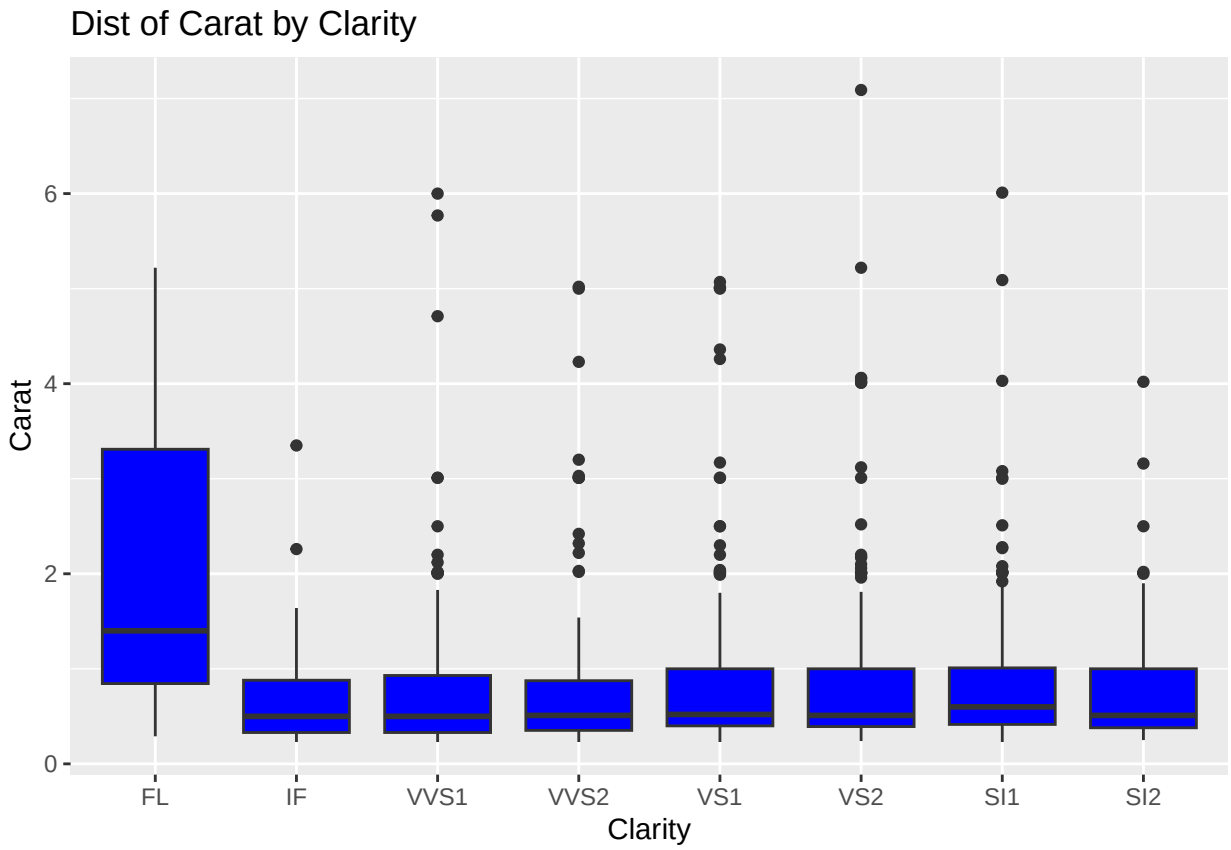



```
ggplot(Data, aes(x=carat, y=price, color=cut)) +
geom_point() +
geom_smooth(se=FALSE)+
labs(x="Carat", y="Price", title="Effect of Carat Size and Diamond Cut on Price", color = "Cut")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
ggplot(Data, aes(x=clarity, y=carat))+
  geom_boxplot(fill="Blue")+
  labs(x="Clarity", y="Carat", title="Dist of Carat by Clarity")
```




```
ggplot(Data, aes(x=carat, y=price, color=clarity)) +
geom_point() +
geom_smooth(se=FALSE)+
labs(x="Carat", y="Price", title="Effect of Carat Size and Diamond Clarity on Price", color = "Clarity")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

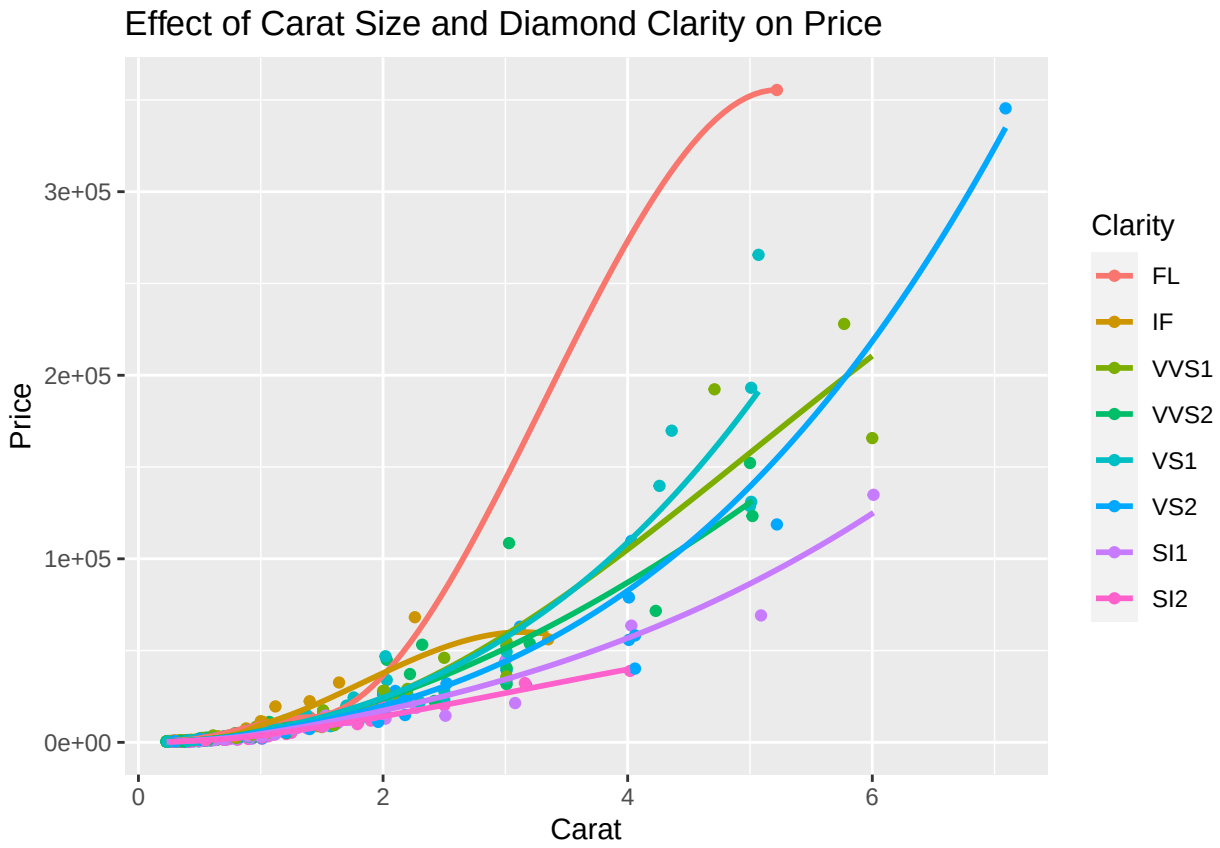
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : span too small.  fewer data values than degrees of freedom.
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at 0.26535
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 1.1346
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : There are other near singularities as well. 14.781
```



Thoughts: Most of clarity and carat against price is the trending the same, except for FL, the best clarity category, plays a much larger role in price and customers bought smaller carats for this category.
Based on the density of carats we can see that from our sample that despite the claim that buying just under a carat value will save you money that these customers still bought on the carat value. There is a slight uptick at each whole or half value of the carat.
We can see in carat and cut effect on price that getting the highest quality cut made a large impact on price compared to the other categories within the same carat weight.

1 carat is 200mg