Gurpinder Singh

STAT 108

1/26/2022

Load all the followin library

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(knitr)
library(broom)
```

Exercise 1: Load all the values with the filter carat equal to .5

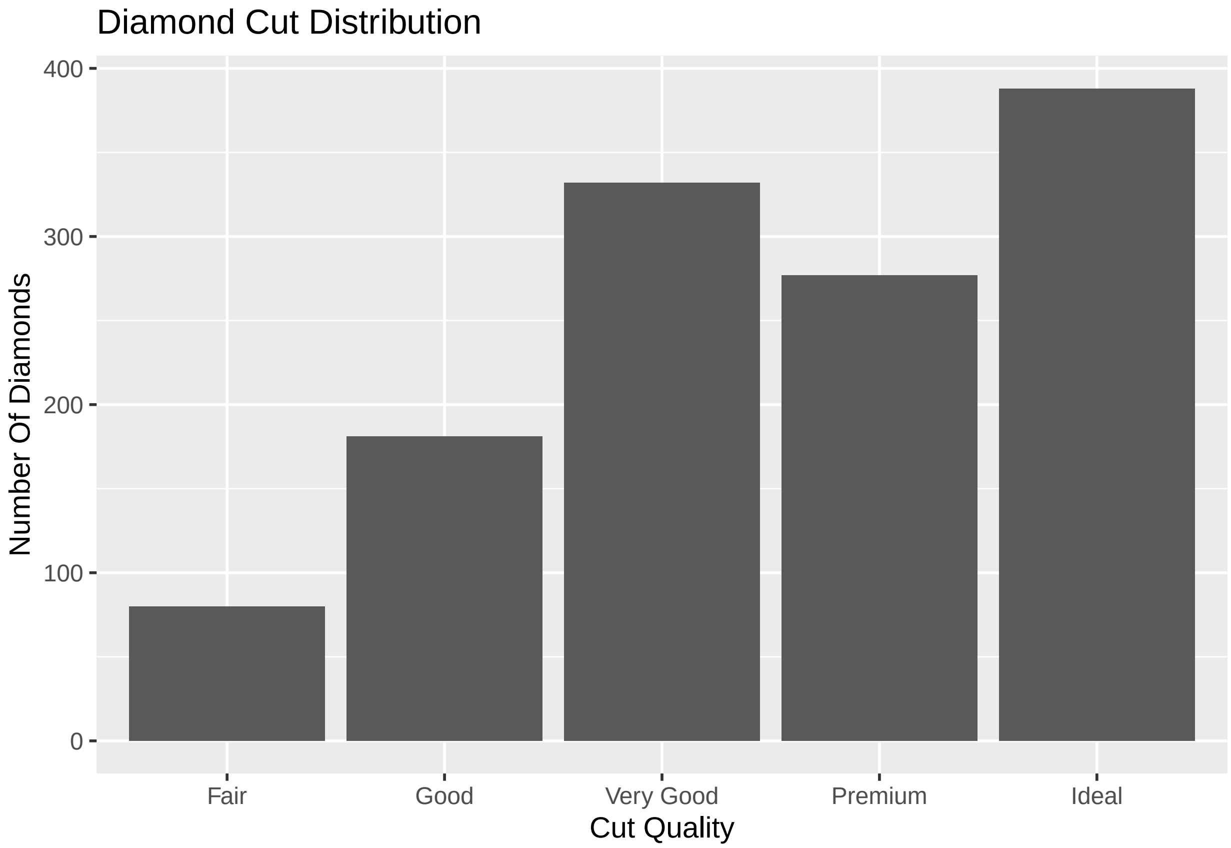
```
diamondData <- diamonds %>%
  filter(carat == 0.5)
glimpse(diamondData)
```

```
## Rows: 1,258
## Columns: 10
## $ carat   <dbl> 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.~
## $ cut     <ord> Ideal, Ideal, Good, Good, Very Good, Fair, Fair, Fair, Fair, F~
## $ color   <ord> E, E, D, D, D, F, F, F, F, F, G, F, E, G, G, F, F, E, E, F, E,~
## $ clarity <ord> VVS2, VVS2, VVS2, IF, IF, I1, I1, I1, I1, I1, I1, I1, I1, I1, ~
## $ depth   <dbl> 62.2, 62.2, 62.4, 63.2, 62.9, 69.8, 71.0, 68.4, 67.1, 68.3, 64~
## $ table   <dbl> 54, 54, 64, 59, 59, 55, 57, 54, 57, 58, 60, 58, 61, 57, 56, 60~
## $ price   <int> 2889, 2889, 3017, 3378, 3378, 584, 613, 613, 627, 627, 701, 71~
## $ x       <dbl> 5.08, 5.09, 5.03, 4.99, 4.99, 4.89, 4.87, 4.94, 4.92, 4.91, 5.~
## $ y       <dbl> 5.12, 5.11, 5.06, 5.04, 5.09, 4.80, 4.79, 4.82, 4.87, 4.78, 4.~
## $ z       <dbl> 3.17, 3.17, 3.14, 3.17, 3.17, 3.38, 3.43, 3.35, 3.28, 3.32, 3.~
```

There are 1258 observations There are 1,258 observations Exercise 2:

```
ggplot(data = diamondData, aes(x =cut)) +
  geom_histogram(stat = "count") +
  labs(x ="Cut Quality",
       y = "Number Of Diamonds",
       title = "Diamond Cut Distribution")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



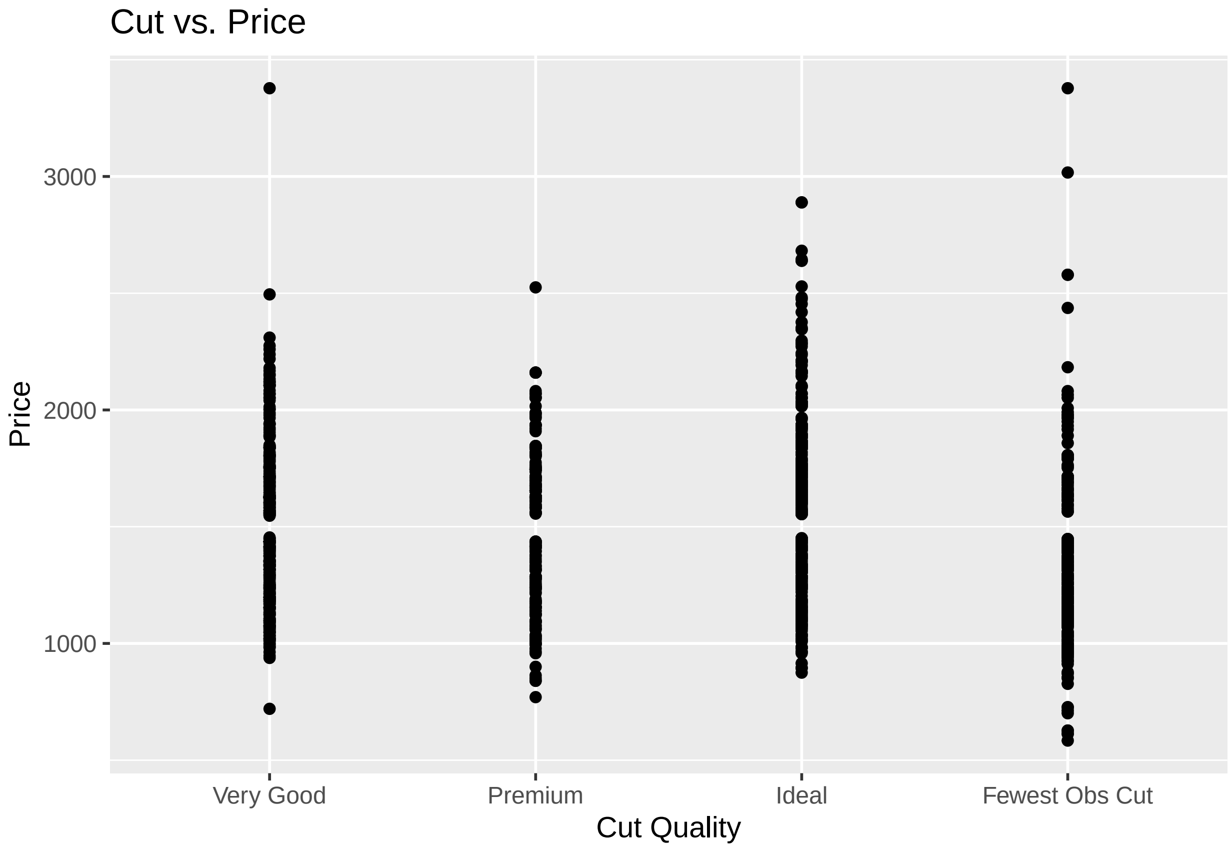
Exersize 3 As you can see there are fewest obs cut and no fair or good which have the fewest observations

```
combinedData <- diamondData %>%
  mutate(cut = fct_lump_n(cut, n=3,  other_level = "Fewest Obs Cut",))
glimpse(combinedData)
```

```
## Rows: 1,258
## Columns: 10
## $ carat   <dbl> 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.~
## $ cut     <ord> Ideal, Ideal, Fewest Obs Cut, Fewest Obs Cut, Very Good, Fewes~
## $ color   <ord> E, E, D, D, D, F, F, F, F, F, G, F, E, G, G, F, F, E, E, F, E,~
## $ clarity <ord> VVS2, VVS2, VVS2, IF, IF, I1, I1, I1, I1, I1, I1, I1, I1, I1, ~
## $ depth   <dbl> 62.2, 62.2, 62.4, 63.2, 62.9, 69.8, 71.0, 68.4, 67.1, 68.3, 64~
## $ table   <dbl> 54, 54, 64, 59, 59, 55, 57, 54, 57, 58, 60, 58, 61, 57, 56, 60~
## $ price   <int> 2889, 2889, 3017, 3378, 3378, 584, 613, 613, 627, 627, 701, 71~
## $ x       <dbl> 5.08, 5.09, 5.03, 4.99, 4.99, 4.89, 4.87, 4.94, 4.92, 4.91, 5.~
## $ y       <dbl> 5.12, 5.11, 5.06, 5.04, 5.09, 4.80, 4.79, 4.82, 4.87, 4.78, 4.~
## $ z       <dbl> 3.17, 3.17, 3.14, 3.17, 3.17, 3.38, 3.43, 3.35, 3.28, 3.32, 3.~
```

Exersize 4

```
ggplot(combinedData, aes(x =cut, y= price)) +
  geom_point() +
  labs(x="Cut Quality",
       y="Price",
       title="Cut vs. Price")
```



Exersize 5 The following was refrenced to do the next part: https://dplyr.tidyverse.org/reference/summarise.html

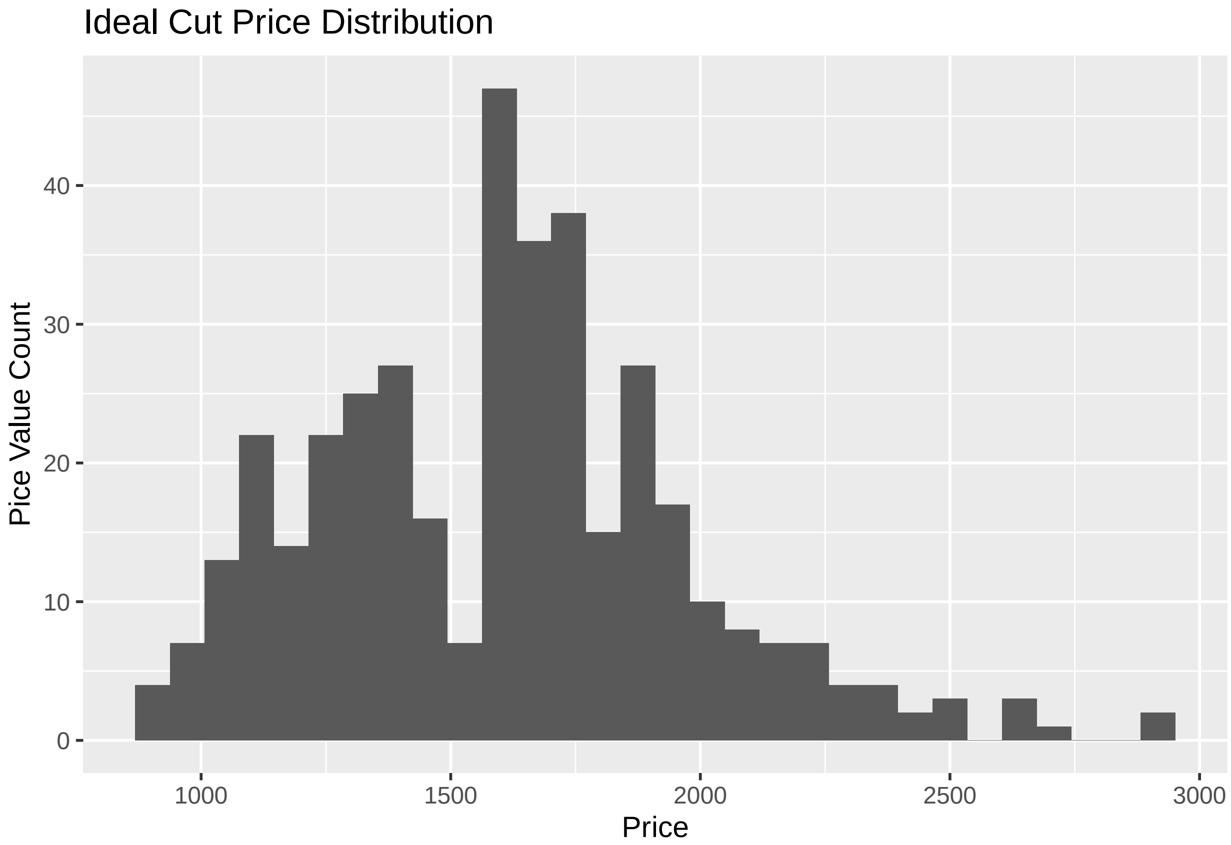
```
combinedData %>%
  group_by(cut) %>%
  summarise(mean = mean(price),sd = sd(price), obsvationCount = n())
```

```
## # A tibble: 4 x 4
##   cut             mean    sd obsvationCount
##   <ord>          <dbl> <dbl>          <int>
## 1 Very Good      1489.  339.            332
## 2 Premium        1532.  304.            277
## 3 Ideal          1609.  368.            388
## 4 Fewest Obs Cut 1341.  365.            261
```

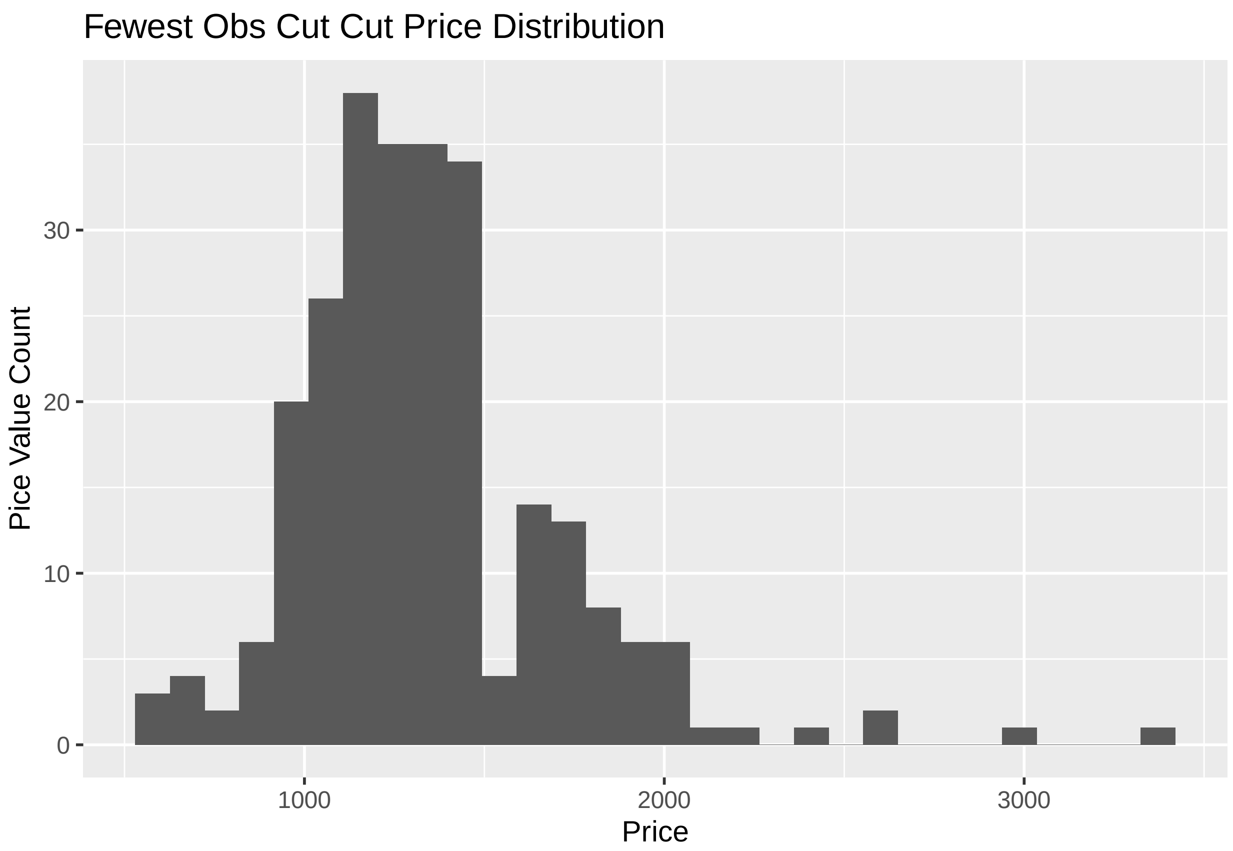
Exersize 6 There definetly is a correlation between the cut and the cost. The fewest observations are the worse cuts and have 1340 mean and ideal has 1608. Additionally as the cut gets betters the cost increases. There seems to be a linear correlation Exersize 7 The following is to showcase Normal assumption

```
for (i in unique(combinedData$cut)){
  plot<- ggplot(combinedData %>% filter(cut == i), aes(x = price)) +
  geom_histogram() +
  labs(x = "Price",
       y = "Pice Value Count",
       title = paste(i,"Cut Price Distribution"))
  print(plot)
}
```

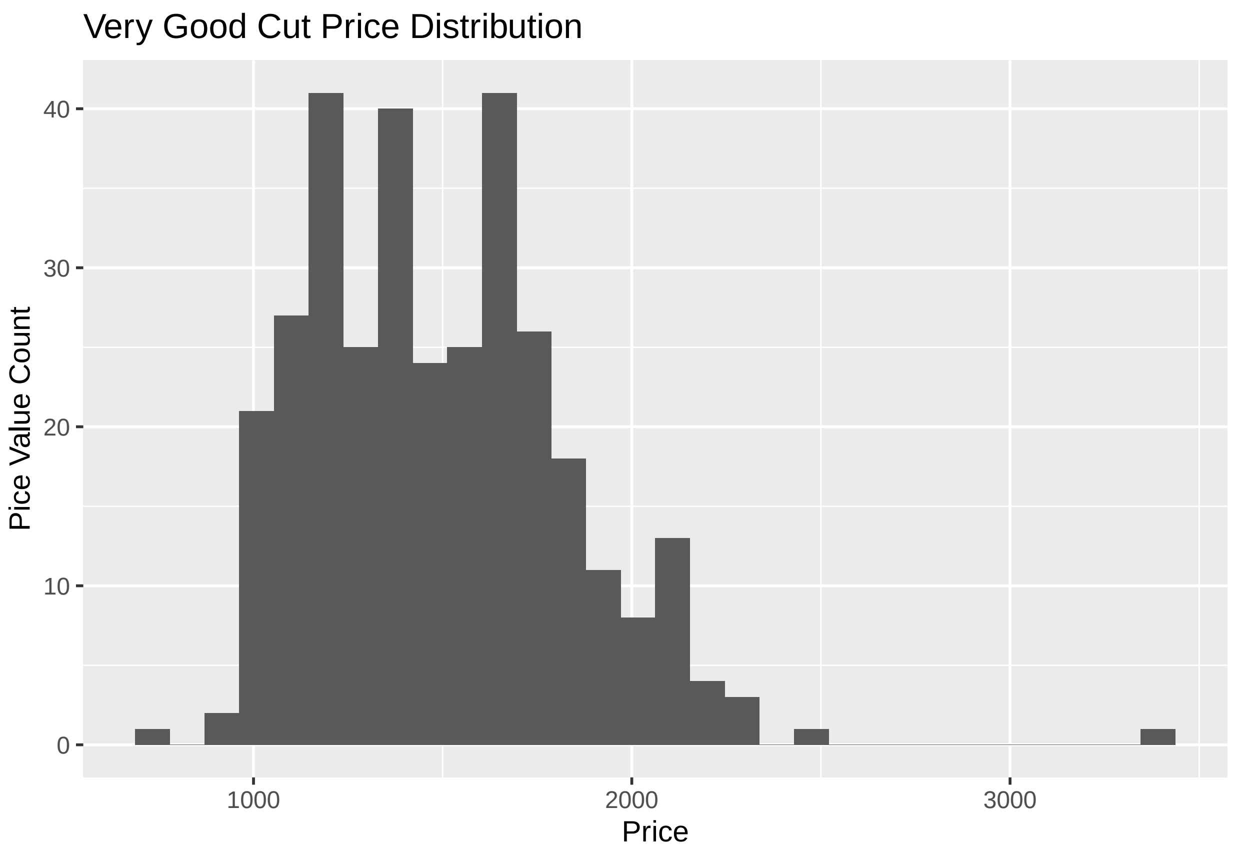
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



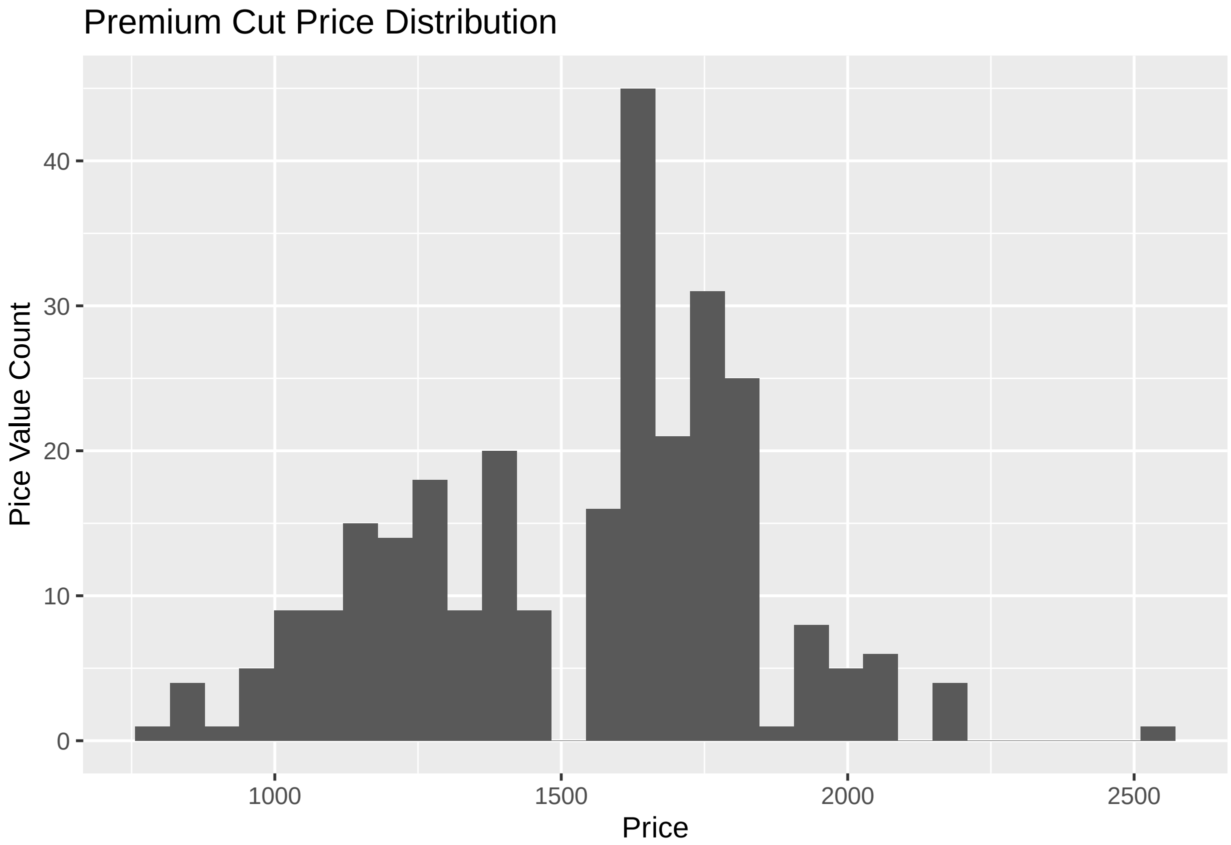
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



Because all the plots are normal distribution it is evident that Normal distribution is satified

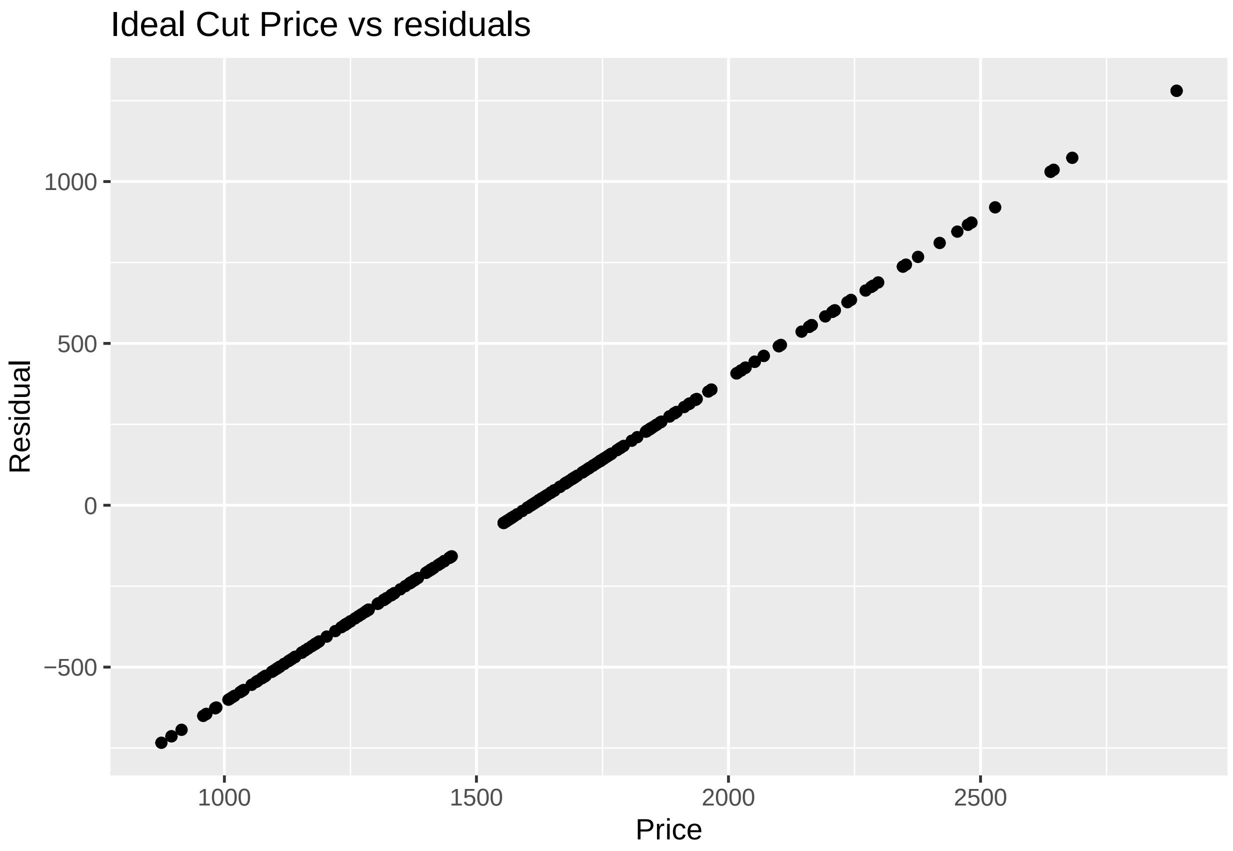
For Independece assumption it is verified because the data were collected with out any dependence to the last observation. Thus they were independent This code was taken partly by last assignment Constant varience is checked

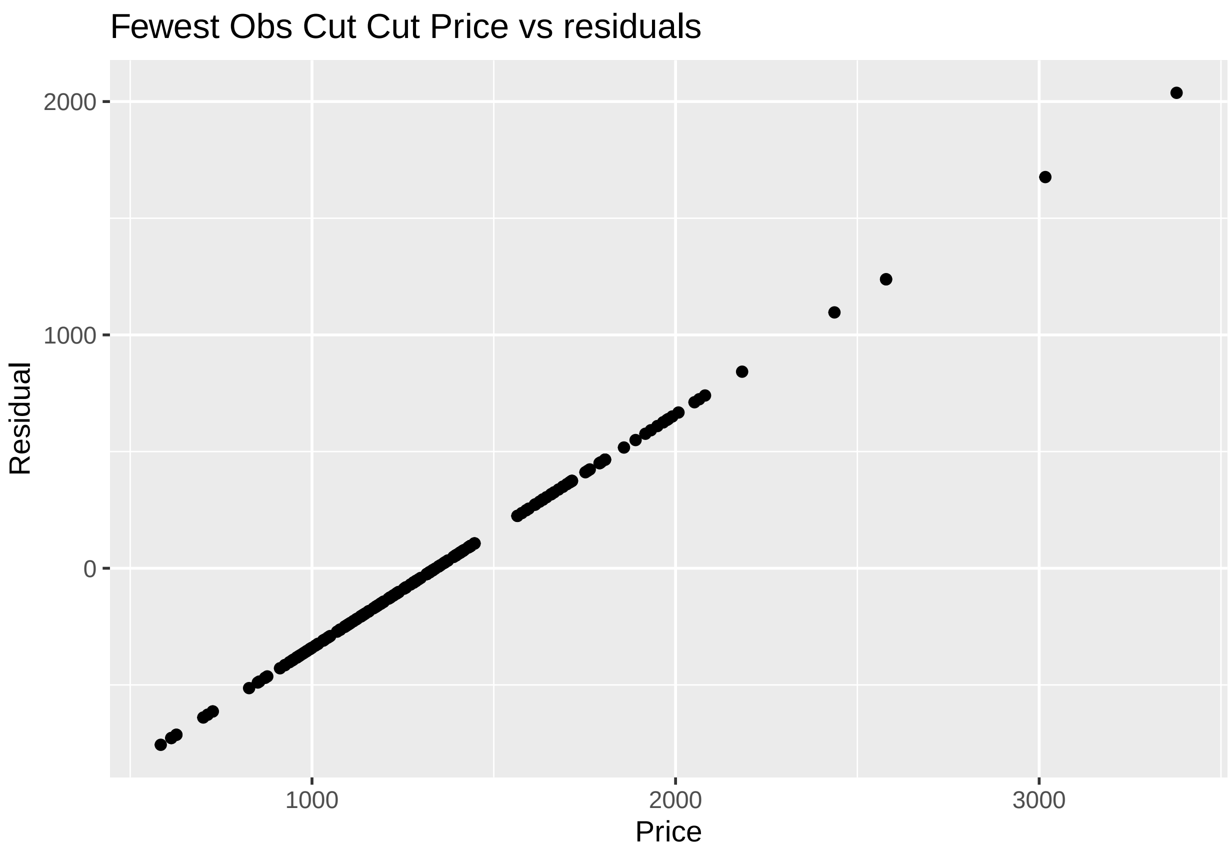
```
reg_model <- lm(price ~ cut, data = combinedData)
  tidy(reg_model) %>% # output model
  kable(digits = 3) # format model output
```

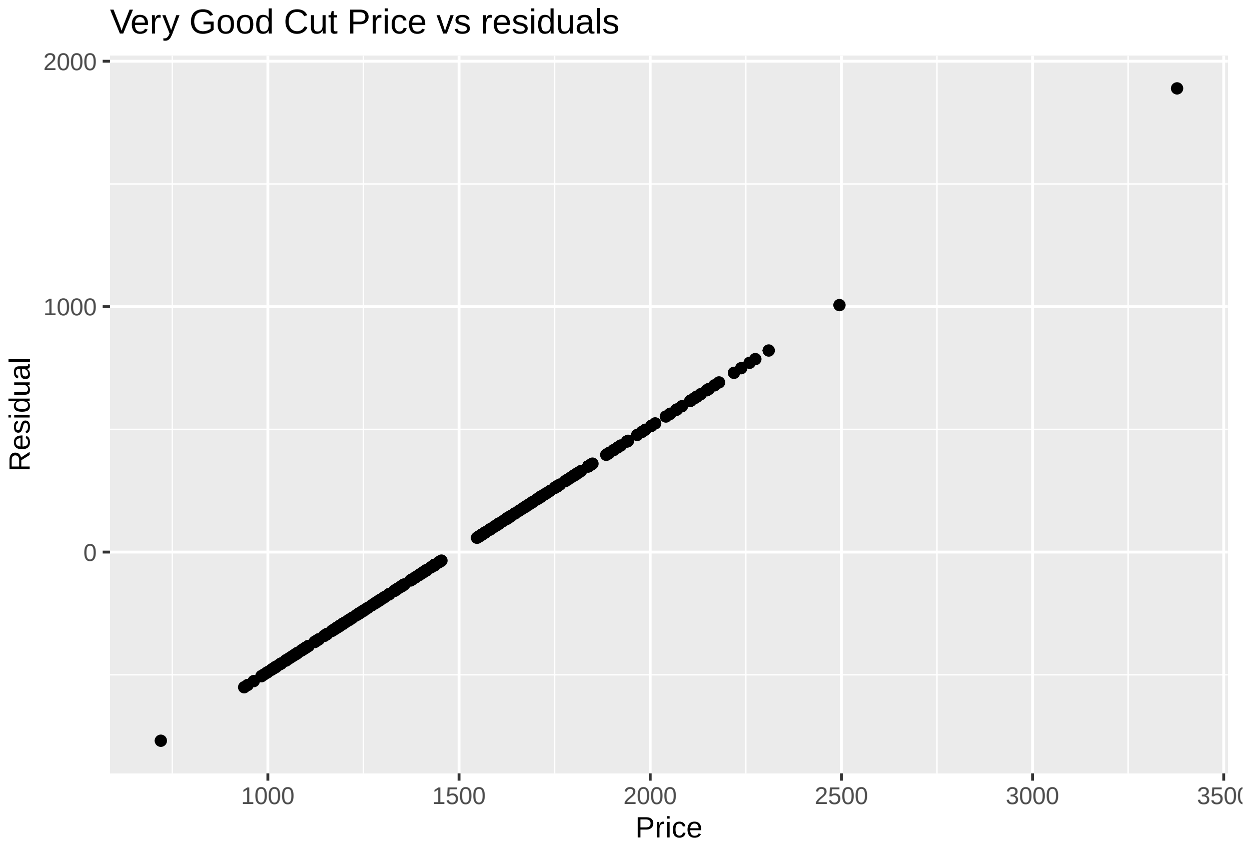
term	estimate	std.error	statistic	p.value
(Intercept)	1492.438	9.893	150.852	0
cut.L	-82.101	20.181	-4.068	0
cut.Q	-155.569	19.787	-7.862	0
cut.C	-84.678	19.384	-4.368	0

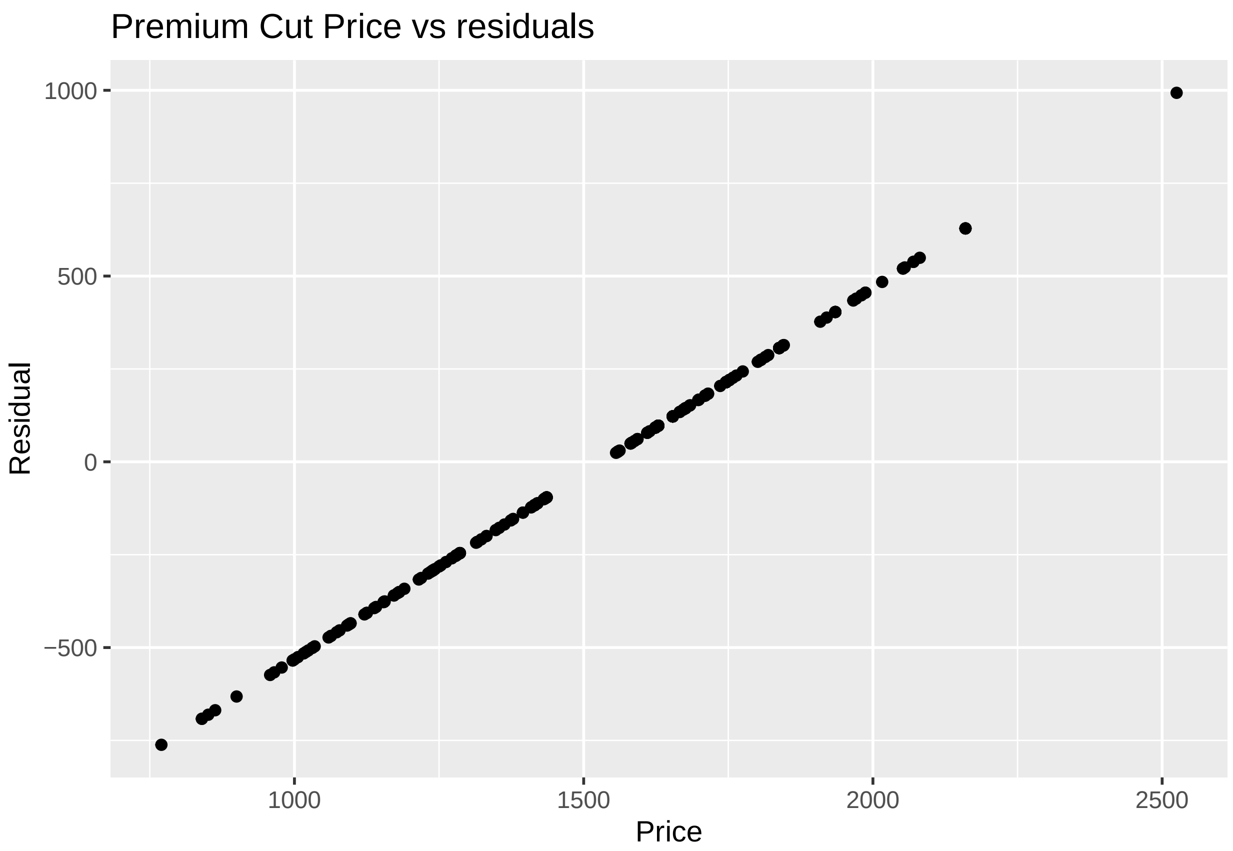
```
combinedData<- combinedData %>%
  mutate(resid = residuals(reg_model))

for (i in unique(combinedData$cut)){
  plot<- ggplot(combinedData %>% filter(cut == i), aes(x = price, y=resid)) +
  geom_point() +
  labs(x = "Price",
       y = "Residual",
       title = paste(i,"Cut Price vs residuals"))
  print(plot)
}
```


Ideal Cut Price vs residuals
Residual
1000
500
0
−500
Price
1000
1500
2000
2500


Fewest Obs Cut Cut Price vs residuals
Residual
2000
1000
0
Price
1000
2000
3000


Very Good Cut Price vs residuals
Residual
2000
1000
0
Price
1000
1500
2000
2500
3000
350



Given this knowledge of the graph you see the chart are linear and thus the constant variance predicamment can not be met. Moreover there is a clear line in each of these Exercise 8 Personal note: Ask teacher why adding tidy(reg_model) gives error

```
reg_model <- lm(price ~ cut, data = combinedData)
kable(anova(reg_model),digits = 3) # format model output
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
cut	3	11507056	3835685.3	31.916	0
Residuals	1254	150706506	120180.6	NA	NA

Exercise 9: The following is calculated by taking the residual sum and dividing by n-1 where n is number of observations, in other words: summ of (xi-xmean)^2/ (n-1). In this case the residual sum is 150,706,506 and n-1 is 1,257 which comes out to be 150,706,506/1,257= 119,893.799522673

Excersie 10:

```
tidy(reg_model)
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   1492.       9.89    151.   0
## 2 cut.L          -82.1     20.2      -4.07 5.03e- 5
## 3 cut.Q         -156.      19.8      -7.86 8.08e-15
## 4 cut.C          -84.7     19.4      -4.37 1.35e- 5
```

Exercise 11 Null hypothesis is that price and cut are not linearly related Alternative is that they are lienarly related

Exercise 12: We reject Null hypothesis meaning that There could be a linear relationship between price and cut.

Exercise 13: We see the difference shown to be in the hundreds but the variance is very little in comparison.