Exploratory Data Analysis On Diamonds Data Set From ggplot2 Package

Anik Das

August 25, 2022

Data Description :-

For this project, we have the '**diamonds**' dataset from ggplot2 package.This is a collection of the prices and other attributes of almost **54,000 diamonds**. The variables are as follows:

Usage :-

diamonds

Format :-

The following is the structure of the dataset :-

Column	***Description***
price	price in US dollars ($326 – $18,823)
carat	weight of the diamond (0.2–5.01)
cut	quality of the cut (Fair, Good, Very Good, Premium, Ideal)
clarity	a measurement of how clear the diamond is (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best))
x	length in mm (0–10.74)
y	width in mm (0–58.9)
z	depth in mm (0–31.8)
depth	total depth percentage = z / mean(x, y) = 2 * z / (x + y) (43–79)
table	width of top of diamond relative to widest point (43–95)

Objecive :-

Our objective is to perform dataset exploration using various types of data visualization.

Importing the Libraries :-

```
library(ggplot2)
library(tibble)
library(dplyr)
library(xtable)
```

Short Description of Our Data :-

After importing the libraries, the diamonds dataset that will be used will be loaded.

```
## Rows: 53,940
## Columns: 10
## $ carat   <dbl> 0.23, 0.21, 0.23, 0.29, 0.31, 0.24, 0.24, 0.26, 0.22, 0.23, 0.~
## $ cut     <ord> Ideal, Premium, Good, Premium, Good, Very Good, Very Good, Ver~
## $ color   <ord> E, E, E, I, J, J, I, H, E, H, J, J, F, J, E, E, I, J, J, J, I,~
## $ clarity <ord> SI2, SI1, VS1, VS2, SI2, VVS2, VVS1, SI1, VS2, VS1, SI1, VS1, ~
## $ depth   <dbl> 61.5, 59.8, 56.9, 62.4, 63.3, 62.8, 62.3, 61.9, 65.1, 59.4, 64~
## $ table   <dbl> 55, 61, 65, 58, 58, 57, 57, 55, 61, 61, 55, 56, 61, 54, 62, 58~
## $ price   <int> 326, 326, 327, 334, 335, 336, 336, 337, 337, 338, 339, 340, 34~
## $ x       <dbl> 3.95, 3.89, 4.05, 4.20, 4.34, 3.94, 3.95, 4.07, 3.87, 4.00, 4.~
## $ y       <dbl> 3.98, 3.84, 4.07, 4.23, 4.35, 3.96, 3.98, 4.11, 3.78, 4.05, 4.~
## $ z       <dbl> 2.43, 2.31, 2.31, 2.63, 2.75, 2.48, 2.47, 2.53, 2.49, 2.39, 2.~
```

So, from the above output, there are **53490 observations** and **10 variables**. Also, among the 10 variables, there are

- **7 continuous** variables, and
- **3 categorical**(ordered) variables.

Checking The Existance Of The Missing Values in The Data :-

So, using *is.na()* function, we get the number of missing values in our data set is

```
[1] 0
```

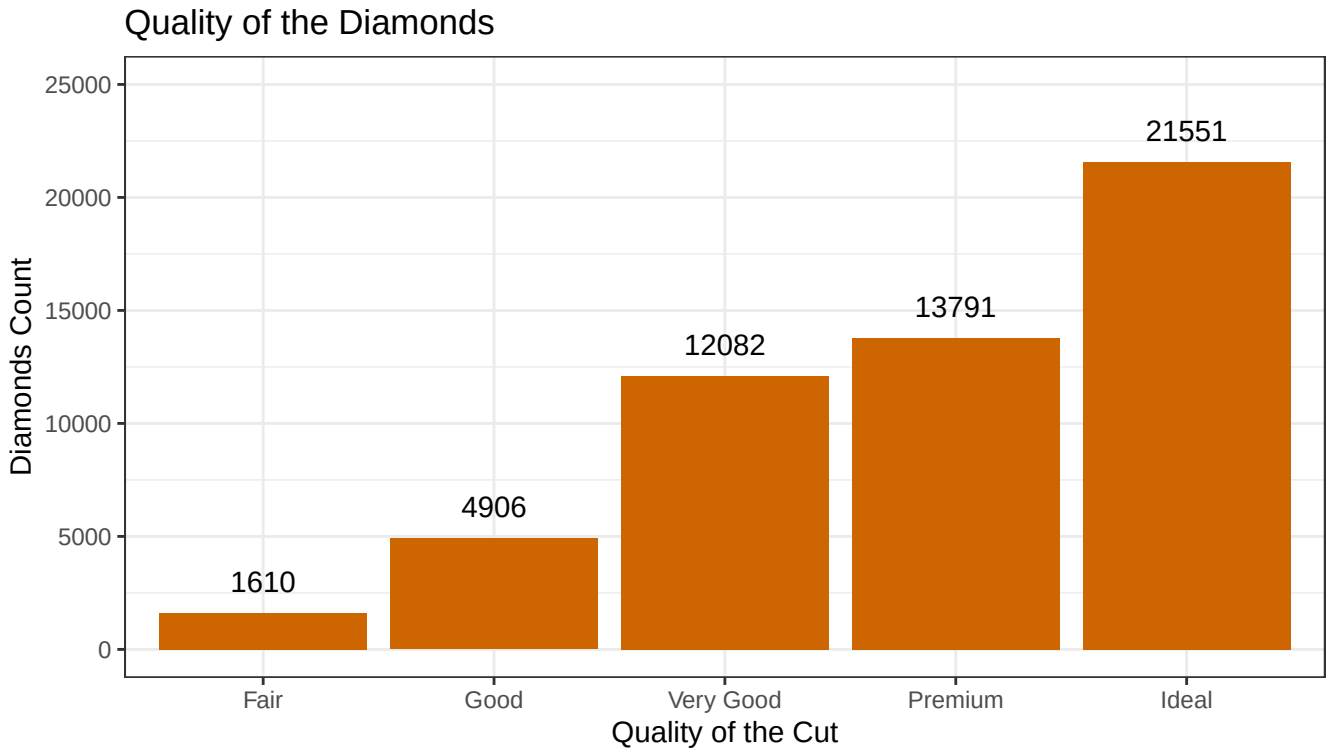
i.e., there are no null values in this dataset. However, there are some unusual data values for some columns. In the follwing sectons, we will fix those columns before analysis performed.

Visualize The Shape of The Distribution :-

- ***For Categrical Variables***

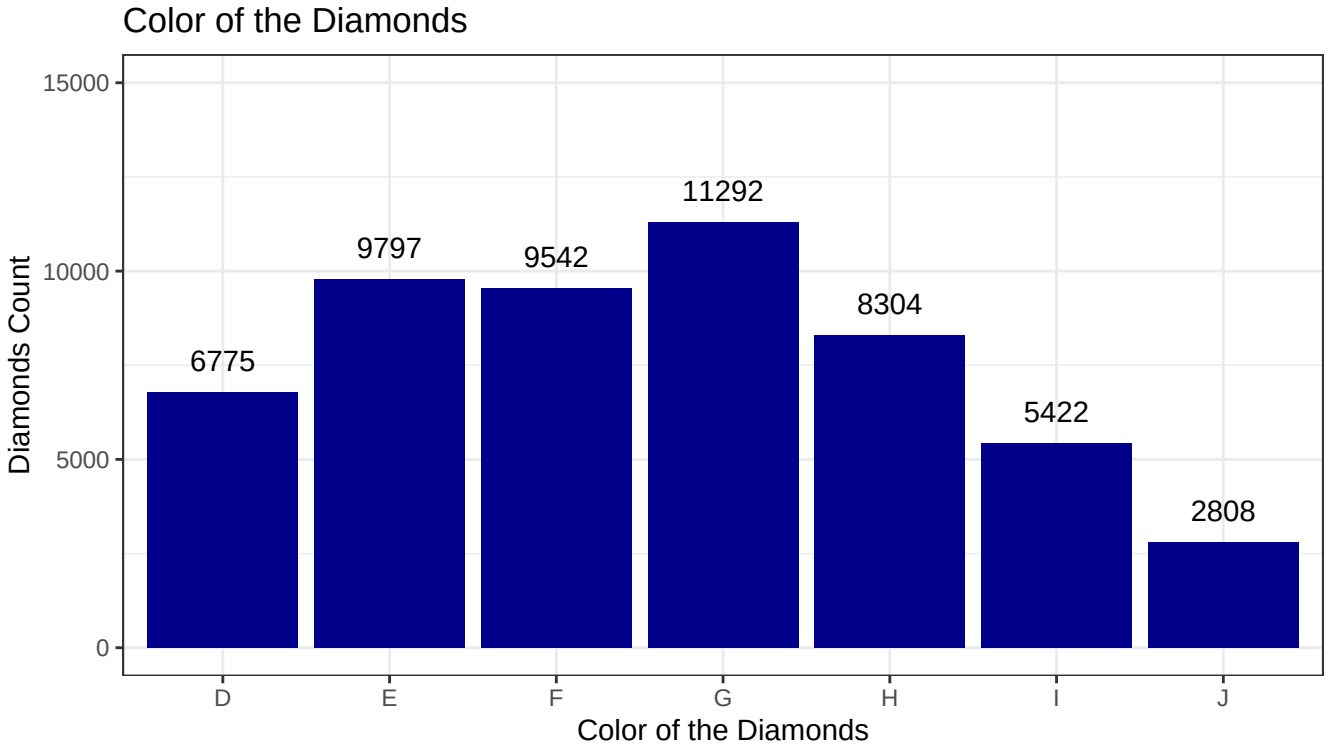
Bar Chart for the Cut of the Diamonds :-

Now, we might be curious to see how the counts of cut are distributed. So, to visualize it's distribution, we are to use bar chart. Bar charts with categorical variables on the x axis and in the fill are a common way to see a contingency table visually.

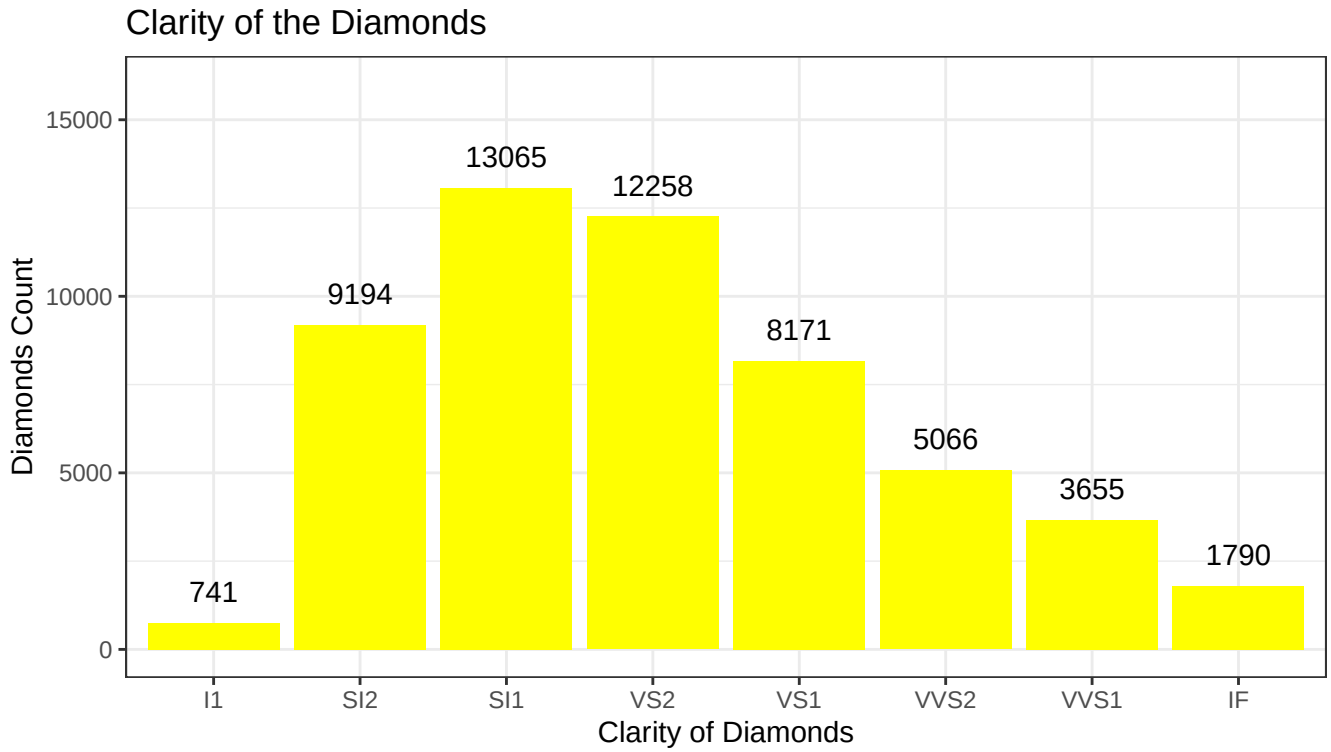


From the above bar chart, we can say that the **ideal** cut is the most common and the **fair** cut is less common one.

Bar Chart for the Color of the Diamonds :-



Bar Chart for the Clarity of the Diamonds :-



- **Different Levels of Cut**

```
[1] Ideal     Premium   Good      Very Good Fair
Levels: Fair < Good < Very Good < Premium < Ideal
```

So, the **fair** cut is the worst kind and the **ideal** cut is the best kind of diamond.

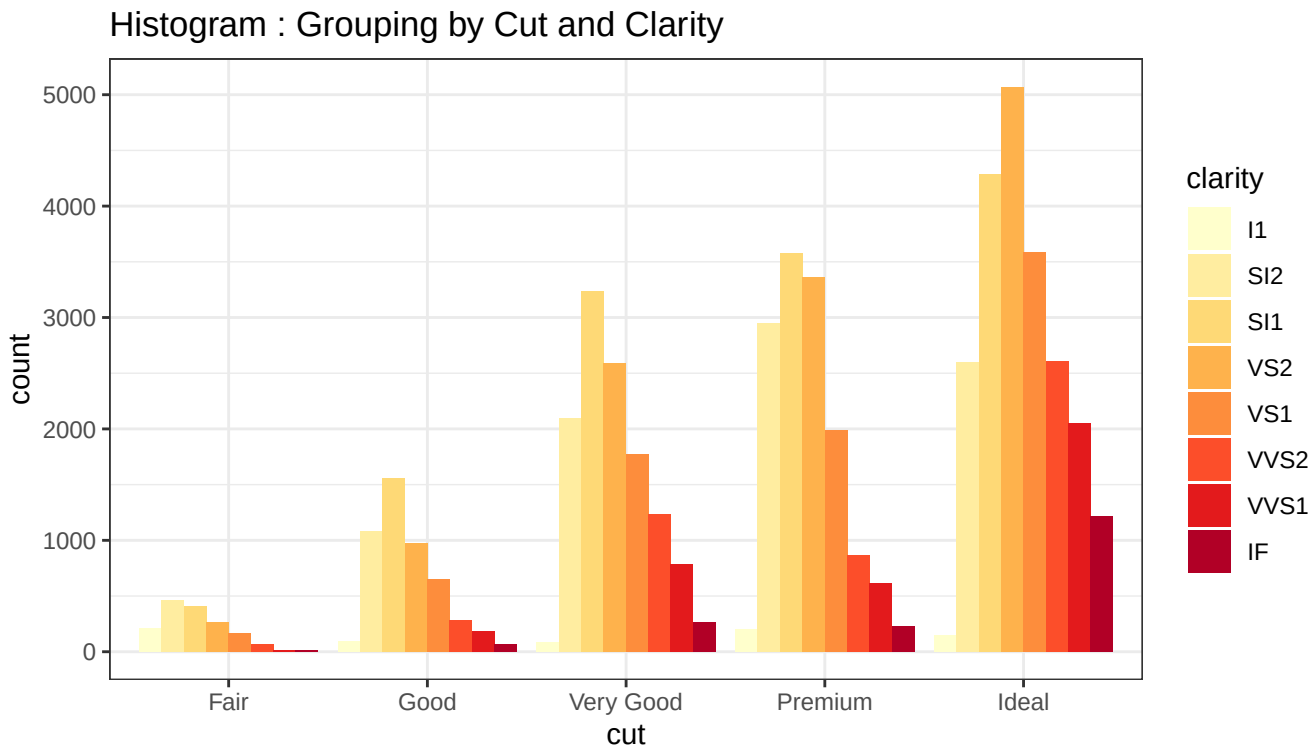
- **Different Levels of Clarity**

```
[1] "I1"   "SI2"  "SI1"  "VS2"  "VS1"  "VVS2" "VVS1" "IF"
```

Cut vs Clarity

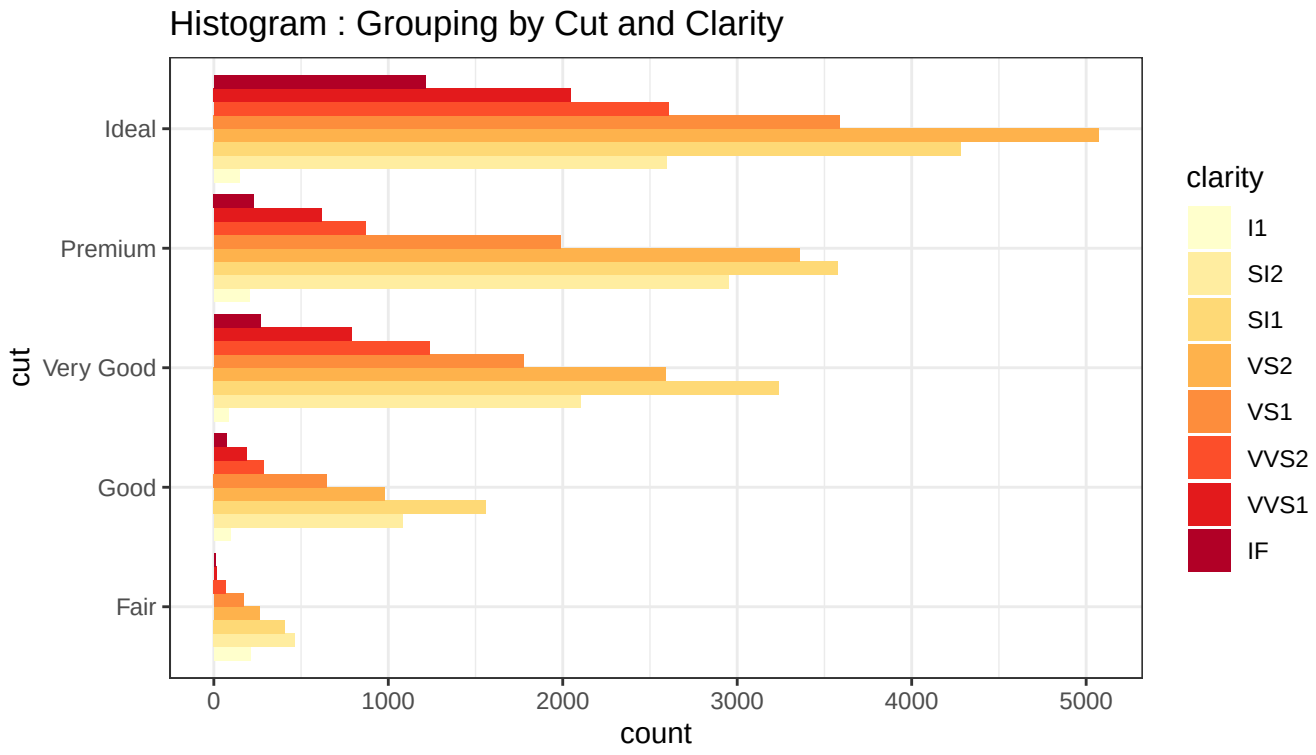
```
# A tibble: 40 x 3
   cut       clarity     n
   <ord>     <ord>   <int>
 1 Ideal     VS2      5071
 2 Ideal     SI1      4282
 3 Ideal     VS1      3589
 4 Premium   SI1      3575
 5 Premium   VS2      3357
 6 Very Good SI1      3240
 7 Premium   SI2      2949
 8 Ideal     VVS2     2606
 9 Ideal     SI2      2598
10 Very Good VS2      2591
# ... with 30 more rows
```

Histogram : Grouping by Cut and Clarity

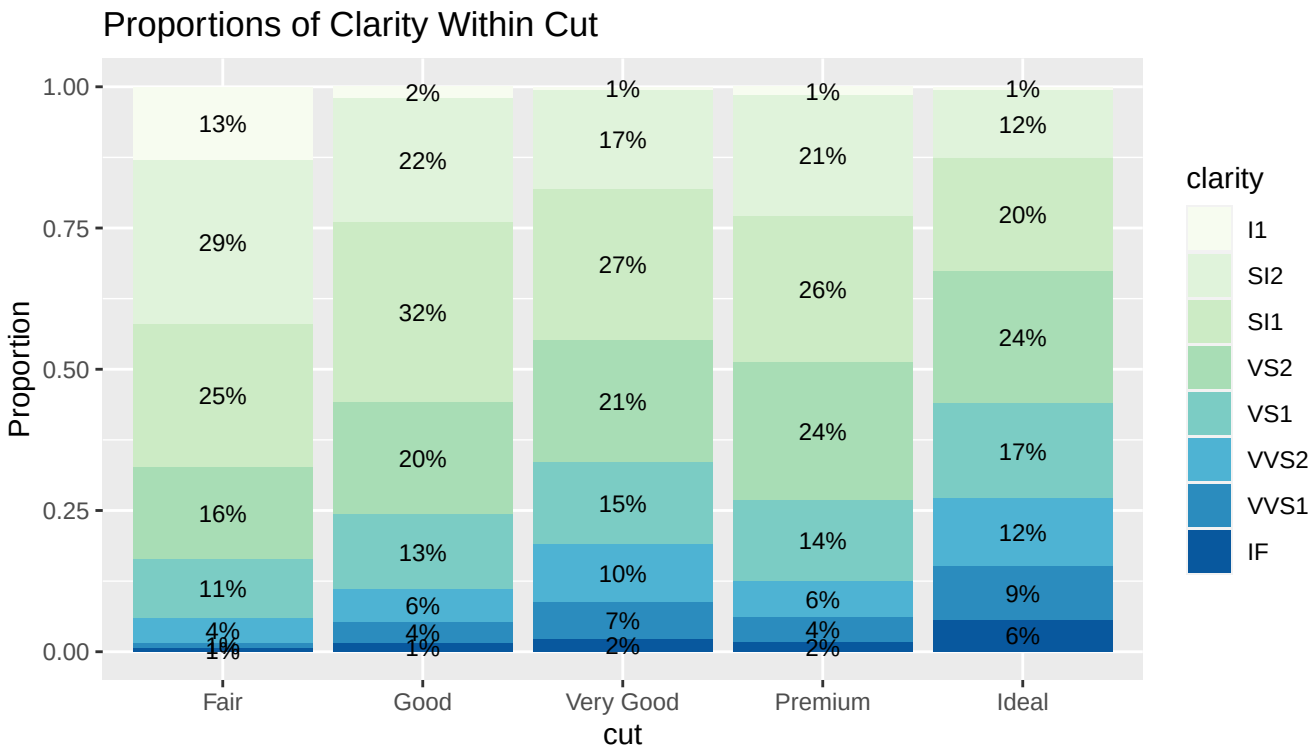


So, we have one categorical variable on the x-axis and other one within each of the previous categorical variable. Here, as diamonds become better in terms of cut, they, in general, become more numerous. However, within each cut, **SI1** and **VS2** are the most common and it does seem to be skewed towards lesser clarity. So, we can say that as the clarity inreases, the rarity within the each cut increases.

Flipping the Coordinates



Stacked bar plots : Cut vs Clarity

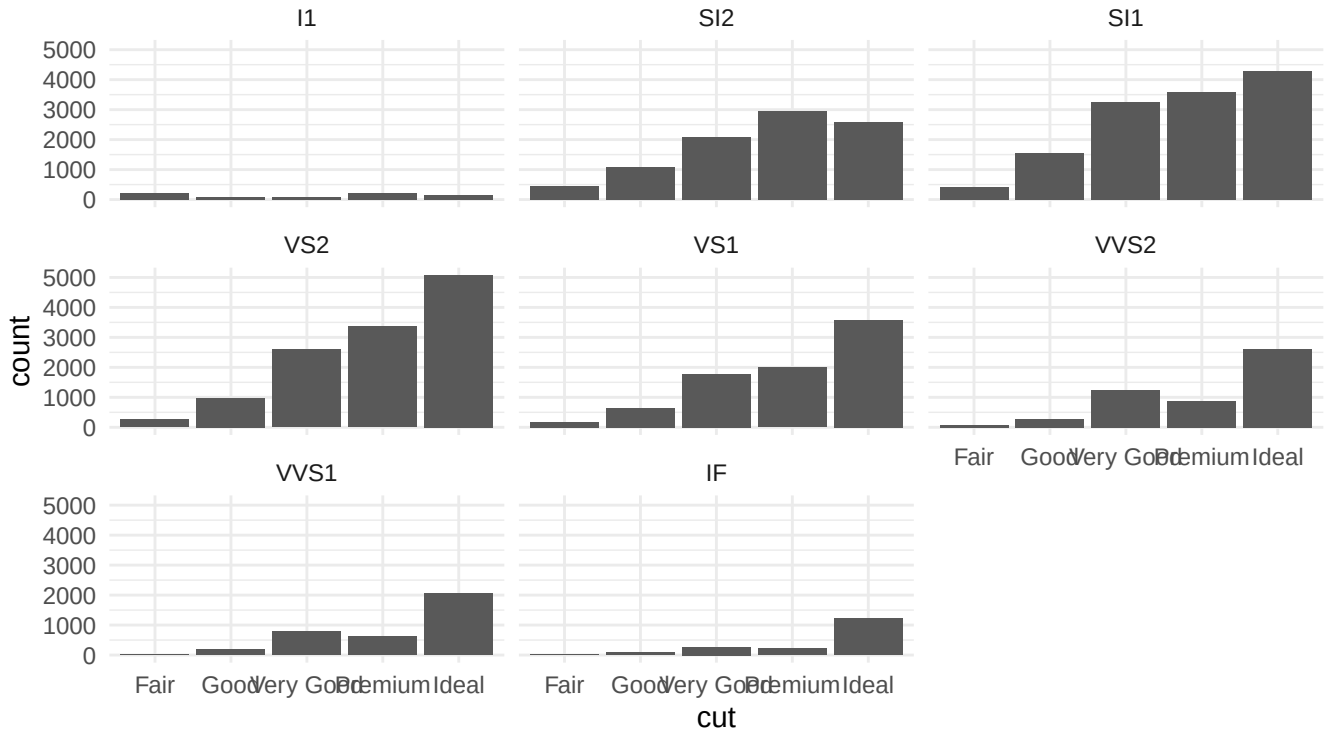


In general, **VS2** and **SI1** tends to have the highest percentages for each cut. But for the best clarity, **IF** has the smallest percentage for each cut.

Cross-tabulations : To observe percentage of each pair

```
             I1  SI2  SI1  VS2  VS1 VVS2 VVS1   IF
Fair       0.39 0.86 0.76 0.48 0.32 0.13 0.03 0.02
Good       0.18 2.00 2.89 1.81 1.20 0.53 0.34 0.13
Very Good  0.16 3.89 6.01 4.80 3.29 2.29 1.46 0.50
Premium    0.38 5.47 6.63 6.22 3.69 1.61 1.14 0.43
Ideal      0.27 4.82 7.94 9.40 6.65 4.83 3.79 2.25
```

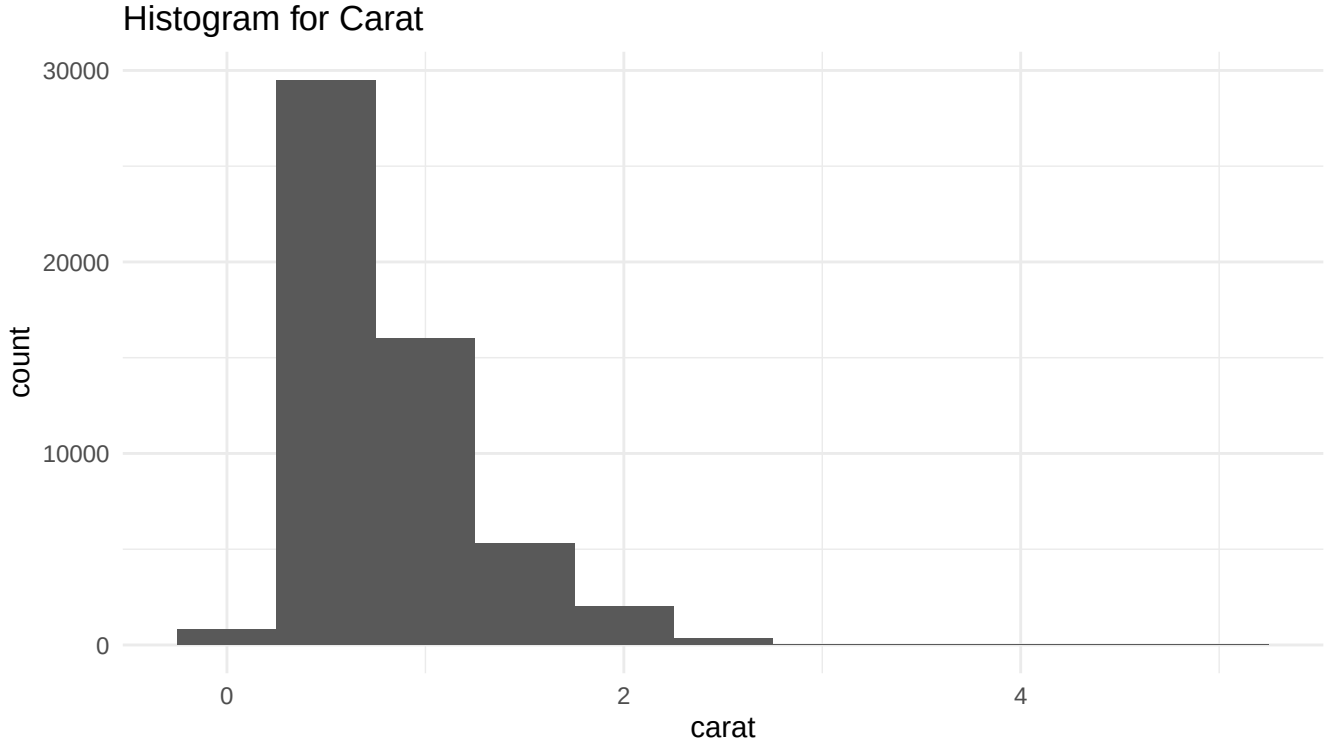
Bar Chart of Cut for each Clarity :



So, we notice that the lowest clarity **I1** has a uniform distribution of cuts. But for other levels of clarity, we can't such type of uniform distributions.

- ***For Continuous Variables***

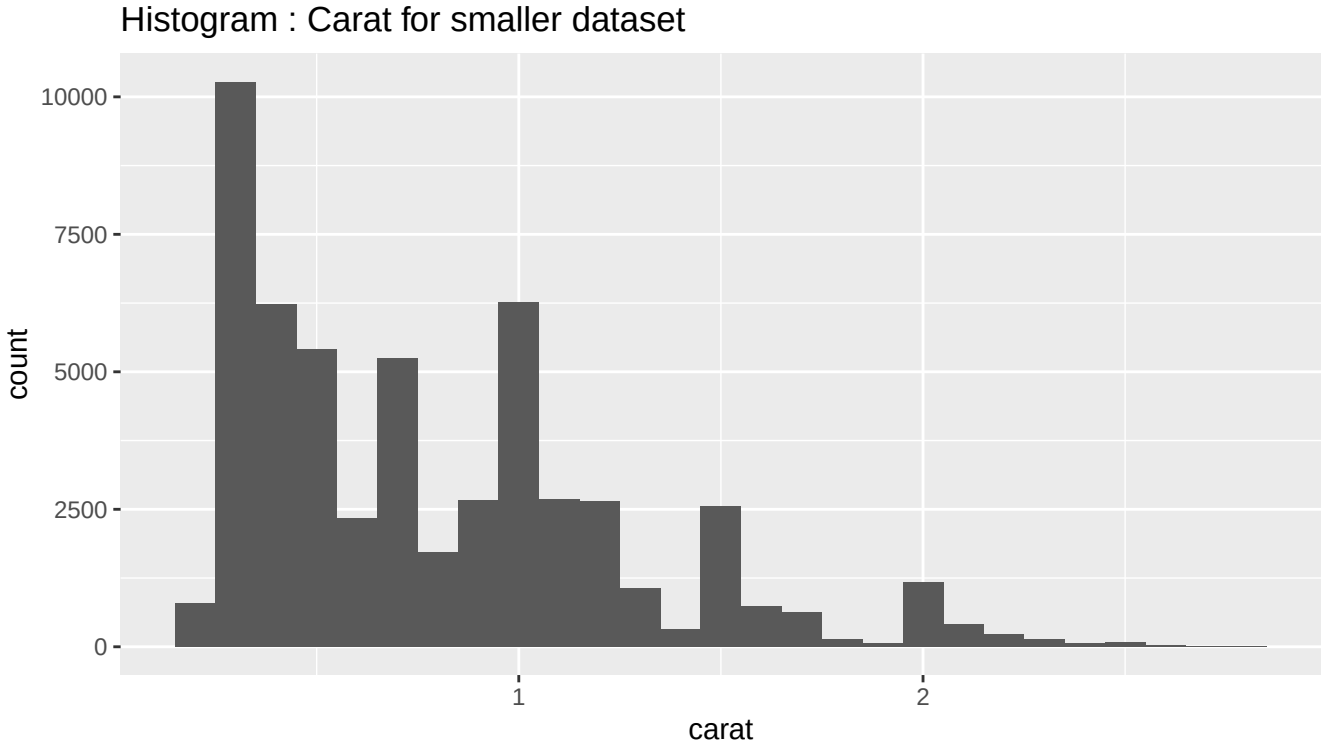
Histogram for Carat :



Diamond Frequency by Carat :

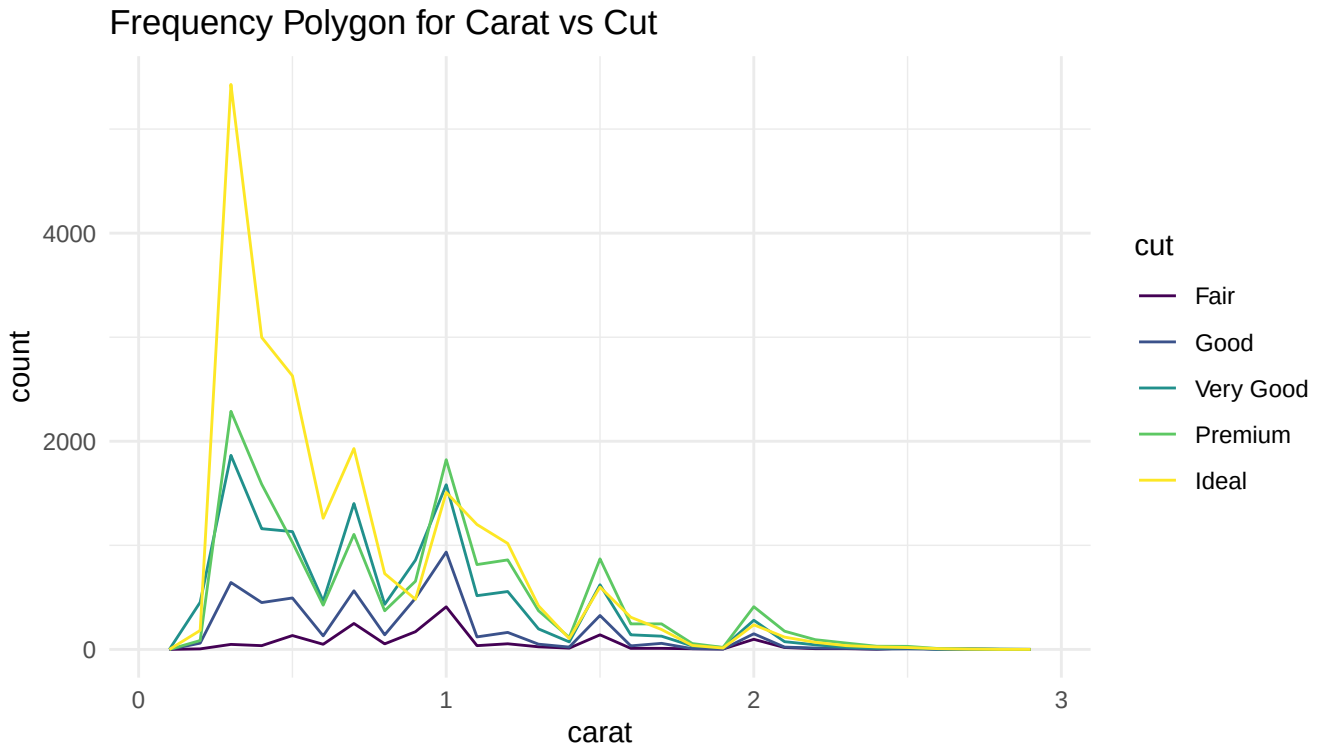
```
# A tibble: 11 x 2
   `cut_width(carat, 0.5)`     n
   <fct>                   <int>
 1 [-0.25,0.25]              785
 2 (0.25,0.75]             29498
 3 (0.75,1.25]             15977
 4 (1.25,1.75]              5313
 5 (1.75,2.25]              2002
 6 (2.25,2.75]               322
 7 (2.75,3.25]                32
 8 (3.25,3.75]                 5
 9 (3.75,4.25]                 4
10 (4.25,4.75]                 1
11 (4.75,5.25]                 1
```

Now, we filtering our dataset diamonds to visualize the shape of the distribution for carat.



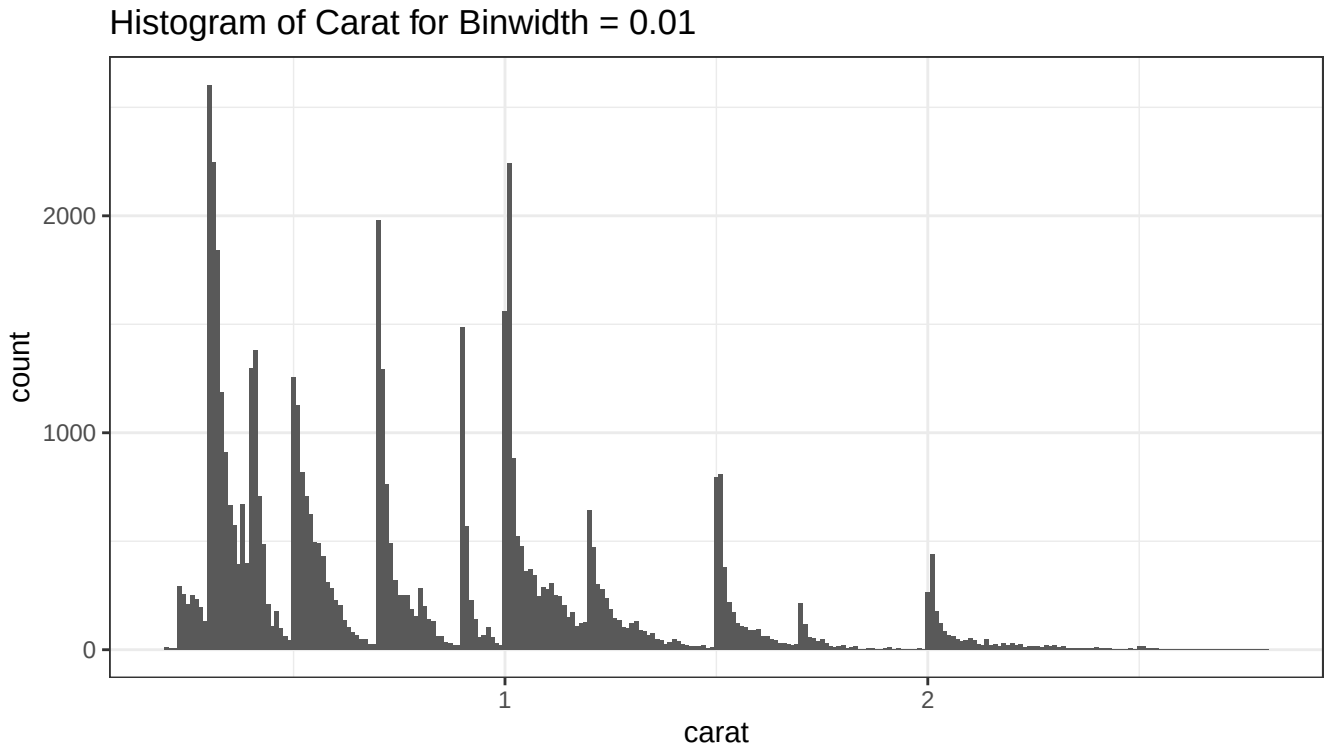
Here, smaller is the dataset where we filter diamonds that have carat values less than 3. Actually, this is a smaller subset of the dataset. Now, we have a more general distribution

Frequency Polygon : Carat vs Cut



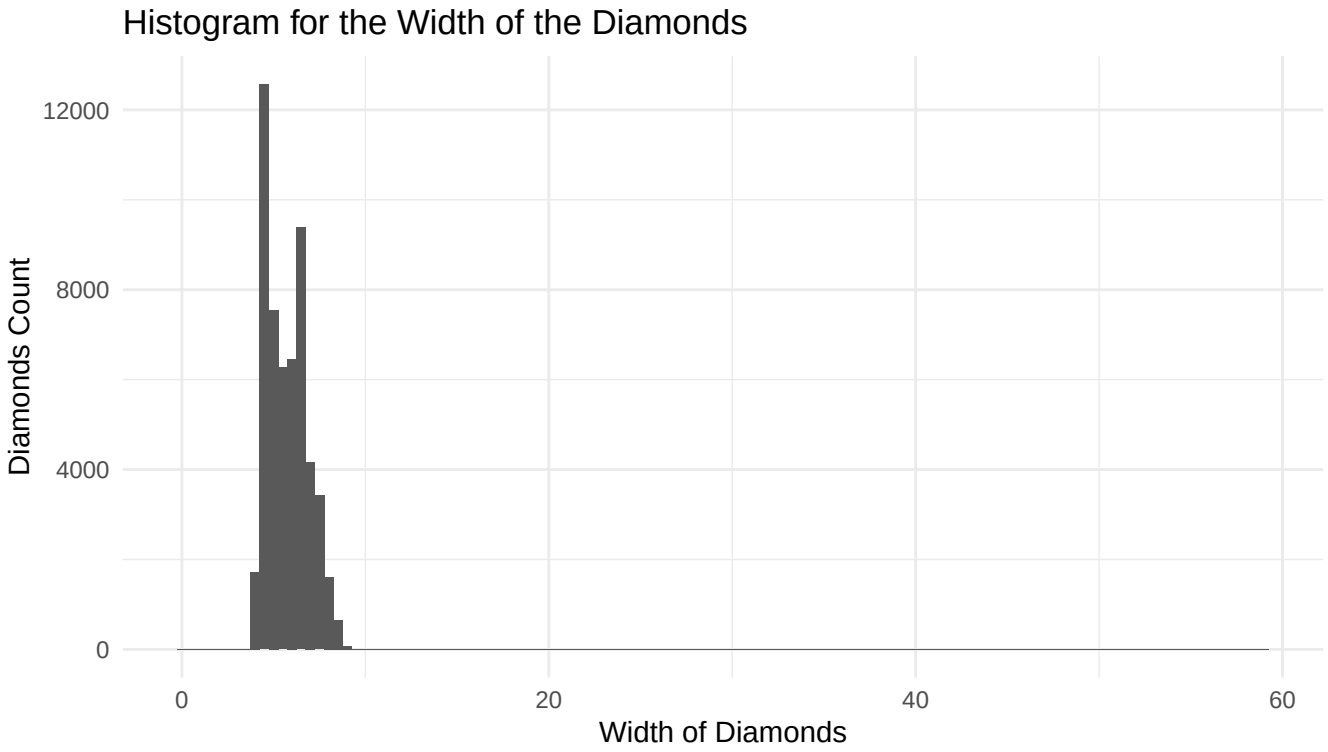
Here, as carat approaches to 1, the count for the good cut diaminds increases and then they drop off very quickly than the other cuts.

Detecting the Outliers Using Histogram of Carat :

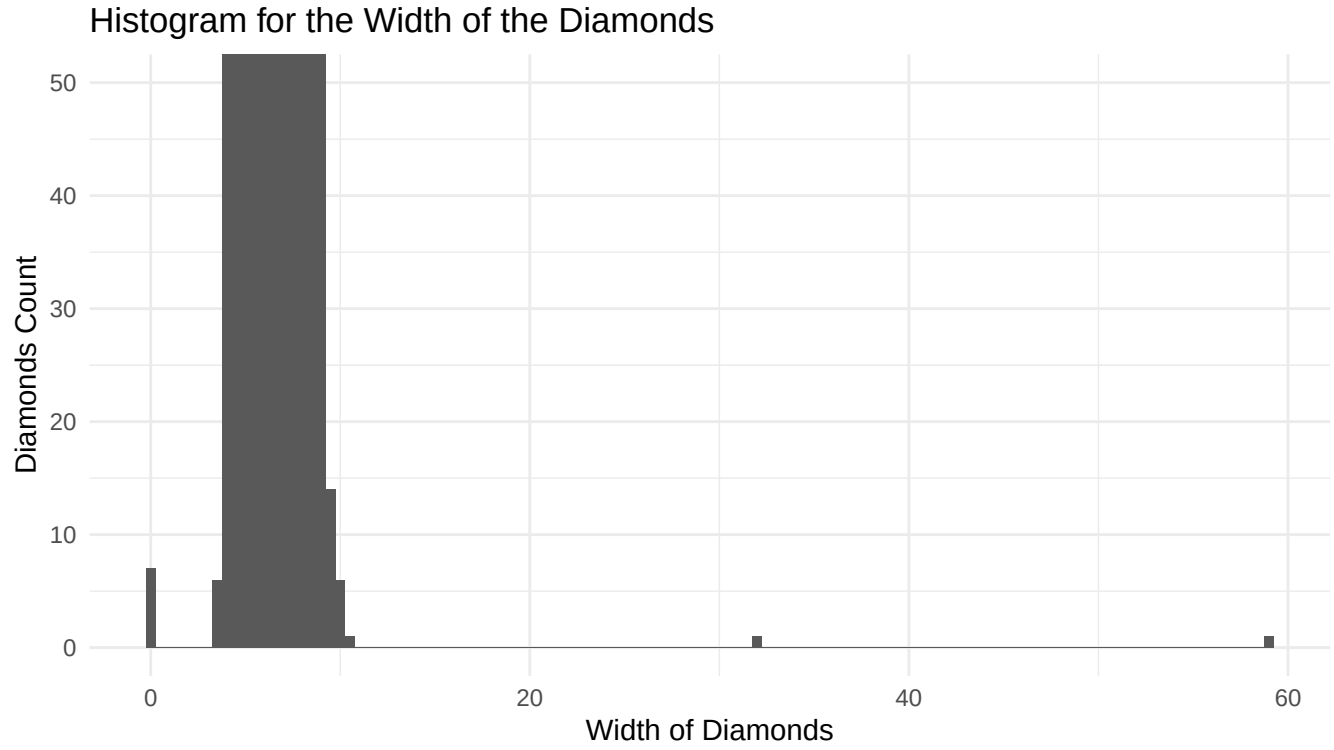


In the right tail, we see that there are some unusual values, which are actually the outliers for the dataset.

Histogram for the Width of diamonds :



This is a very good example of why it is very important to deal with unusual values, which are outliers. Because, when we look into this, we see that they are skewed towards the lower values. But if we remove those values, it would be spread out across the left tail and it would be a completely different distribution, maybe it is a actually skewed towards the left.



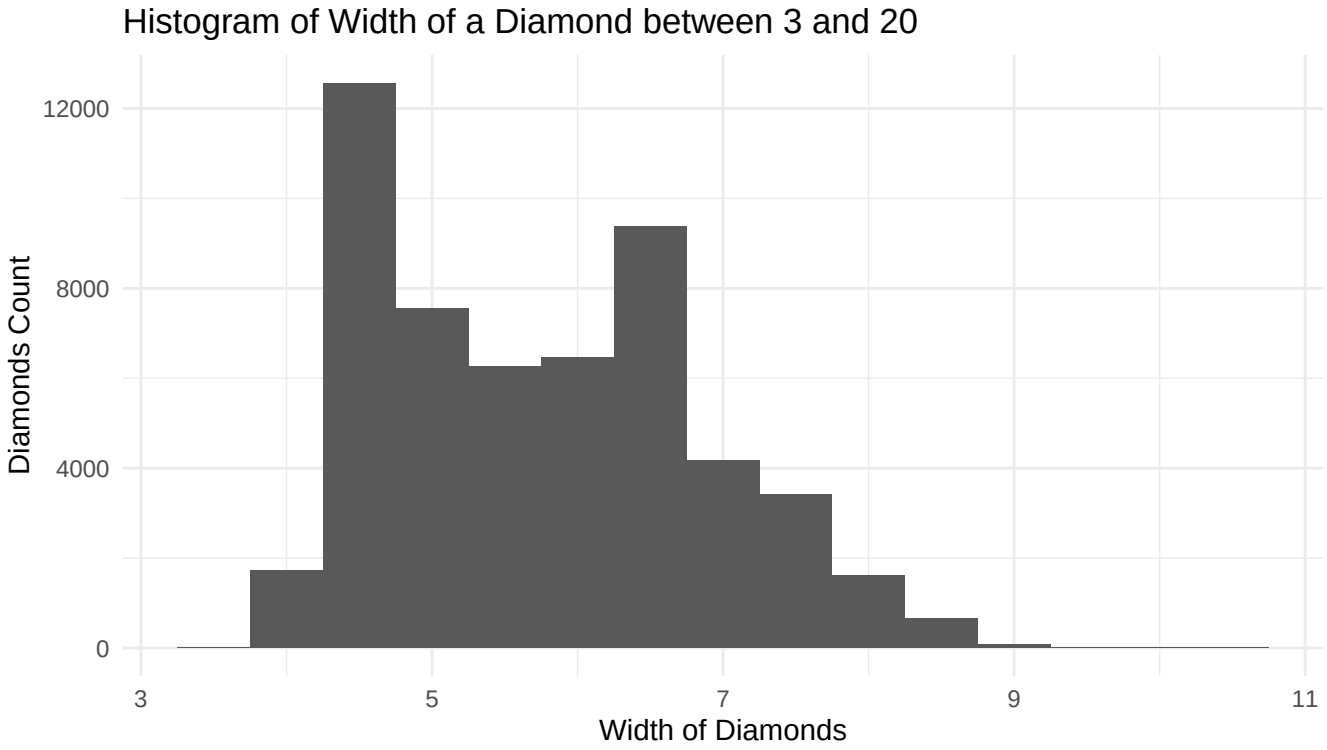
So, we want to zoom the y-axis to see if there are any value outside the range of y (**width** of a diamond), which is (0,50). Now, we notice that there are exactly 8 diaminds with 0 width, which is actually absurd. The reason is that they might either have not filled in or, they might be so small that it just rounded. So, for our analysis part, we have to decide what

to do with these values. Either we can remove those columns x, y and z for those rows or, we can use different types of imputation methods, like hot deck imputation, cold deck impuataion or, mean imputation etc.

Table : Showing the Unusual Values :

	price	x	y	z
1	5139	0.00	0.00	0.00
2	6381	0.00	0.00	0.00
3	12800	0.00	0.00	0.00
4	15686	0.00	0.00	0.00
5	18034	0.00	0.00	0.00
6	2130	0.00	0.00	0.00
7	2130	0.00	0.00	0.00
8	2075	5.15	31.80	5.12
9	12210	8.09	58.90	8.06

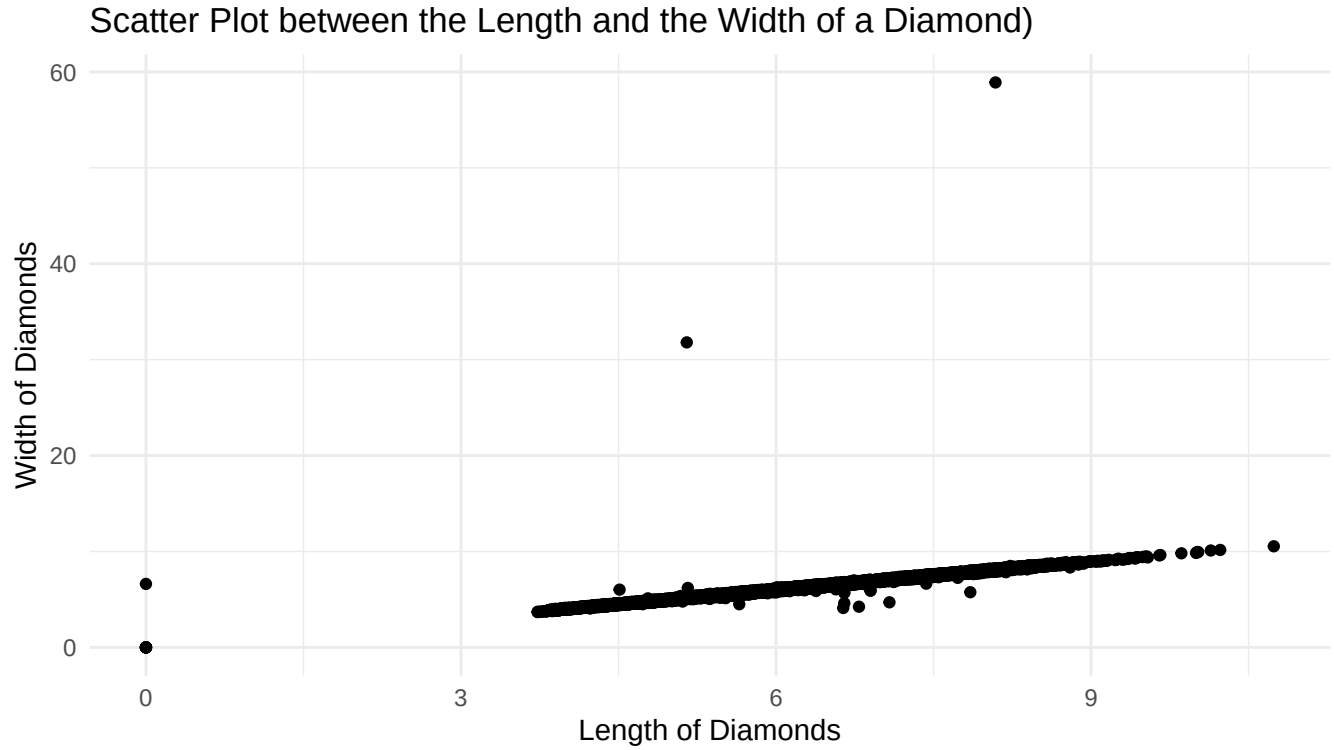
Filtering the dataset :



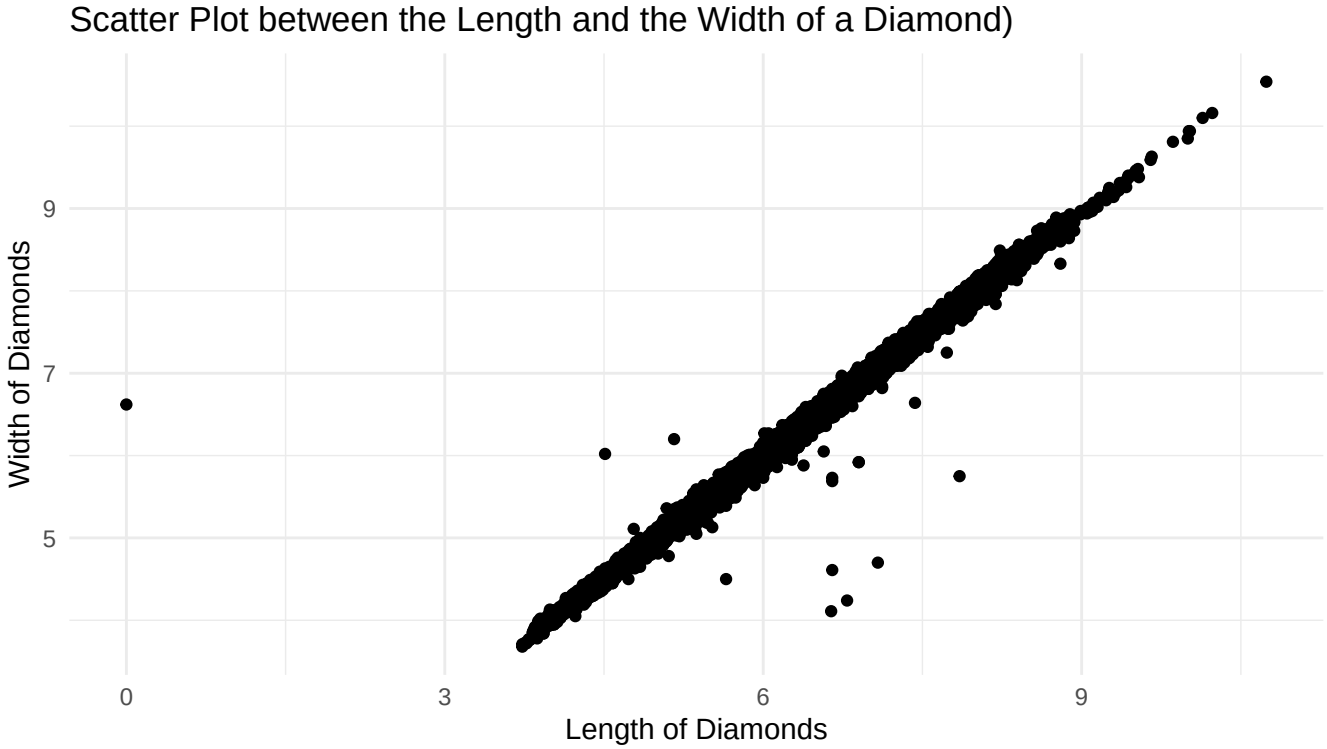
After removing those ouliers, now we are taking the diamonds, which have y values, (i.e., width of a diamond) between 3 and 20. We denote this new dataset by **diamonds2.** So, we get the actual distribution of the **width** of a diamond. That's why this is a good decision to remove those outliers.

Scatter Plot between x (Length of a Diamond) and y (Width of a Diamond) :

Now, we can replace values withmissing values



For the Modified Dataset :

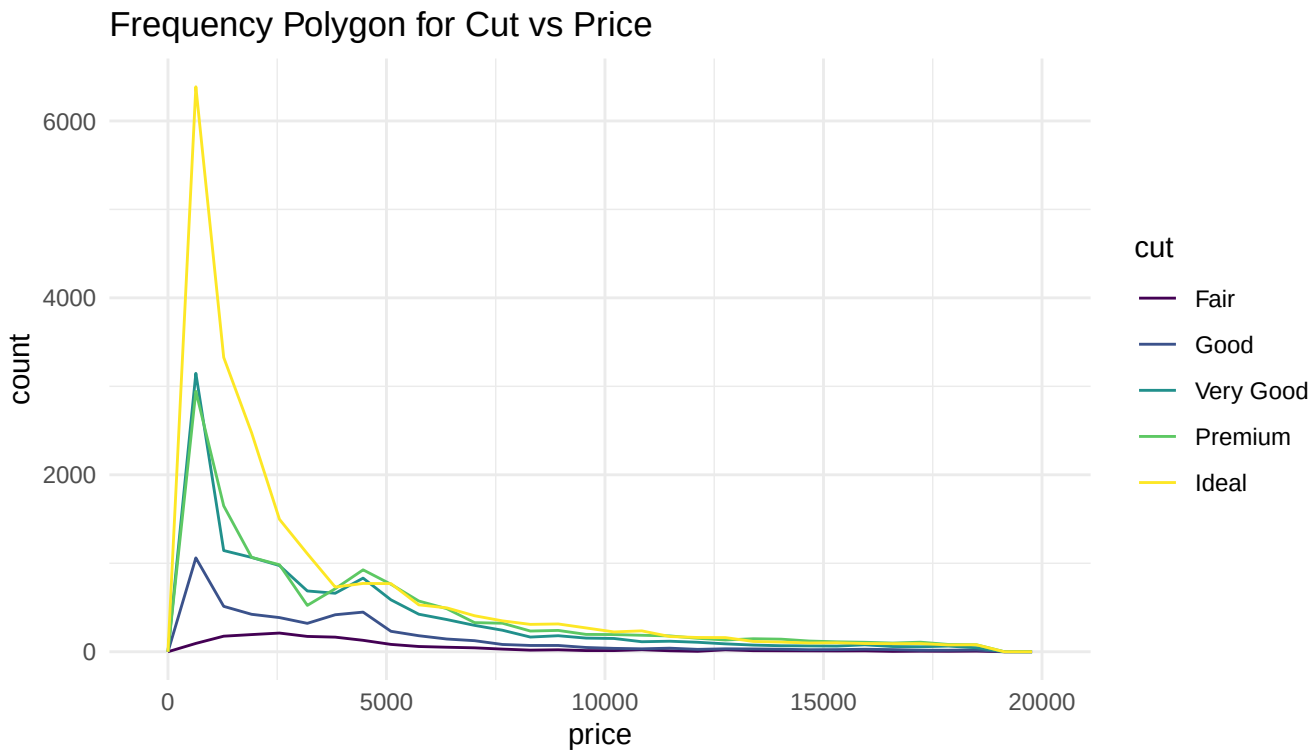


The whole shape of the plot has completely changed. In the previous plot, it's just looked like that there is not as deep of a relationship between **x** and **y**. In the cuurent plot, where we omit those missing/unusual values, there is almost direct correlation between **x** and **y**.

Covariation :-

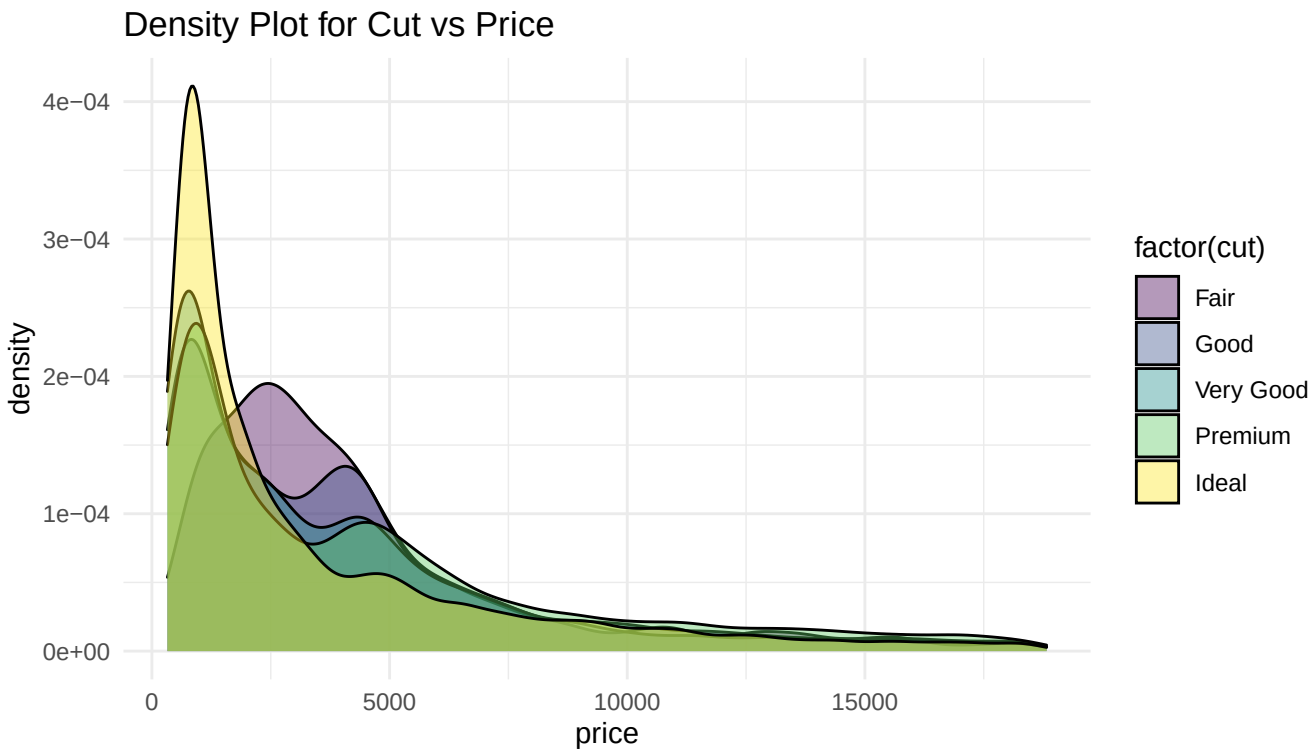
- ***Categorical vs Continuous***

Frequency Polygon : Cut vs Price



From the above plot, we can discuss about the correlation between cut and price. For each cut, as the price increases, the number of times, i.e., the count decreases. But it is hard to see the difference in distribution because, overall accounts differ so much. In the right tail of the plot, there might be some interesting patterns. But we can't see minor differeneces in this plot.

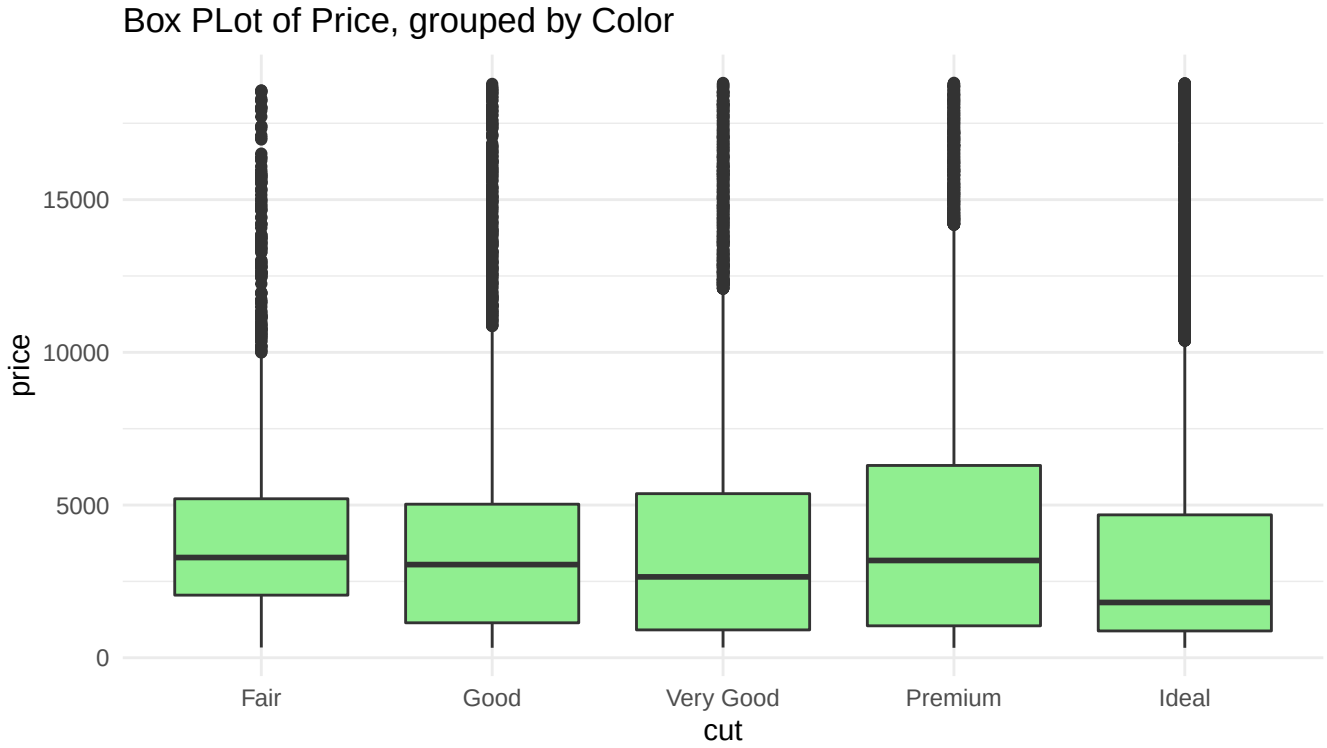
Density Plot : Cut vs Price



So, in the density plot, we basically standardizes the count so that the area under each frequency polygon is one. Here, among the top quality di-

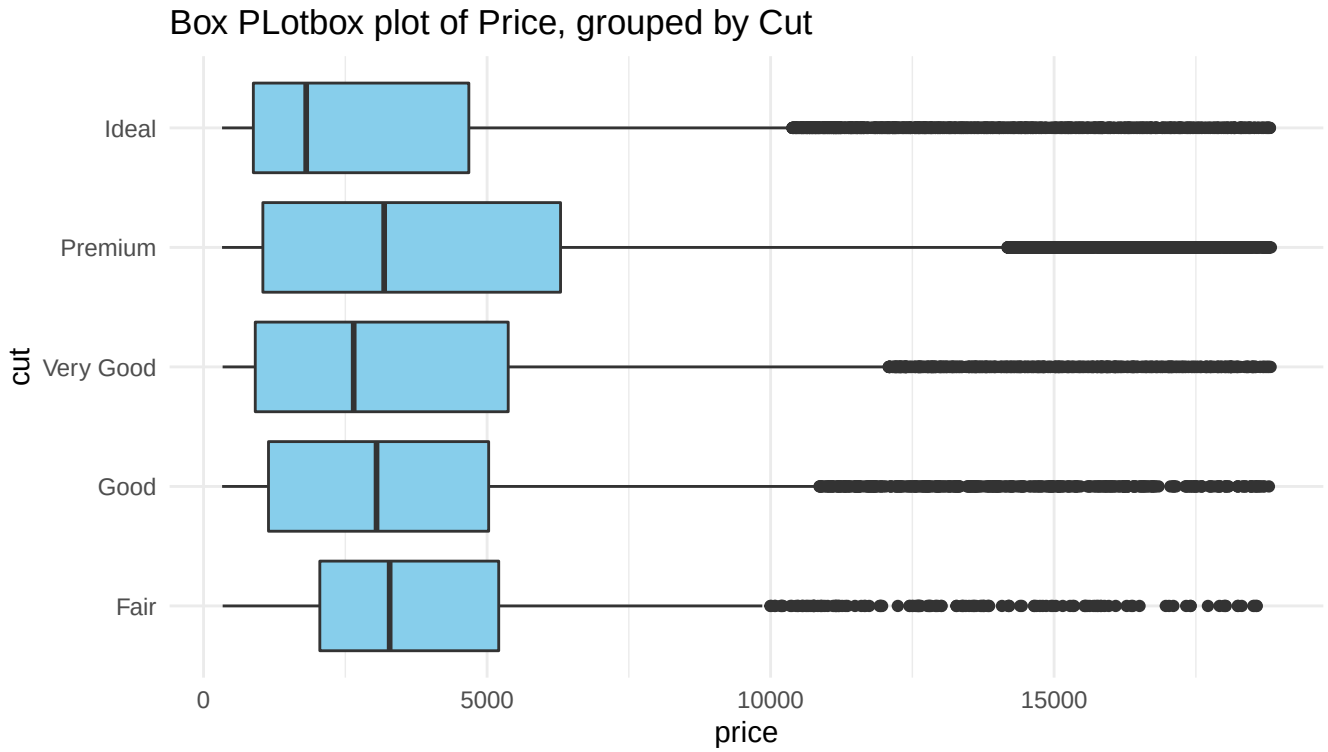
amonds, their average prices are around the 1000 dollar mark and the **fair** cut has the highest average price, although it's a low quality cut class. In reality, we expect that the higher the quality of the cut, the more expensive the diamond would be. But we notice that the lowest quality of cut in a diamond accounts for the highest average price.

Box Plot : Cut vs Price



We know that the box plot are basically an alternative to display the distribution of a continuous variable broken down by categorical variable. So, for the **ideal** cut, it is more skewed towards the lower values and it has a smaller spread. Again the **premium** cut has the higher spread in terms of price. So, there is a larger range of prices for a **premium** diamond than an **ideal** diamond. Also, the **ideal** diamonds also tend to be cheaper and the **fair** diamonds tend to be more expensive as it is further up the boxes. If we notice that for the **fair** diamonds, we may think that the distribution is skewed. But in our box plot, we notice that the distribution of the **fair** diamonds is far narrower.

Flipping the Coordinates



- ***Categorical vs Categorical***

Dot Plot : Cut vs Color

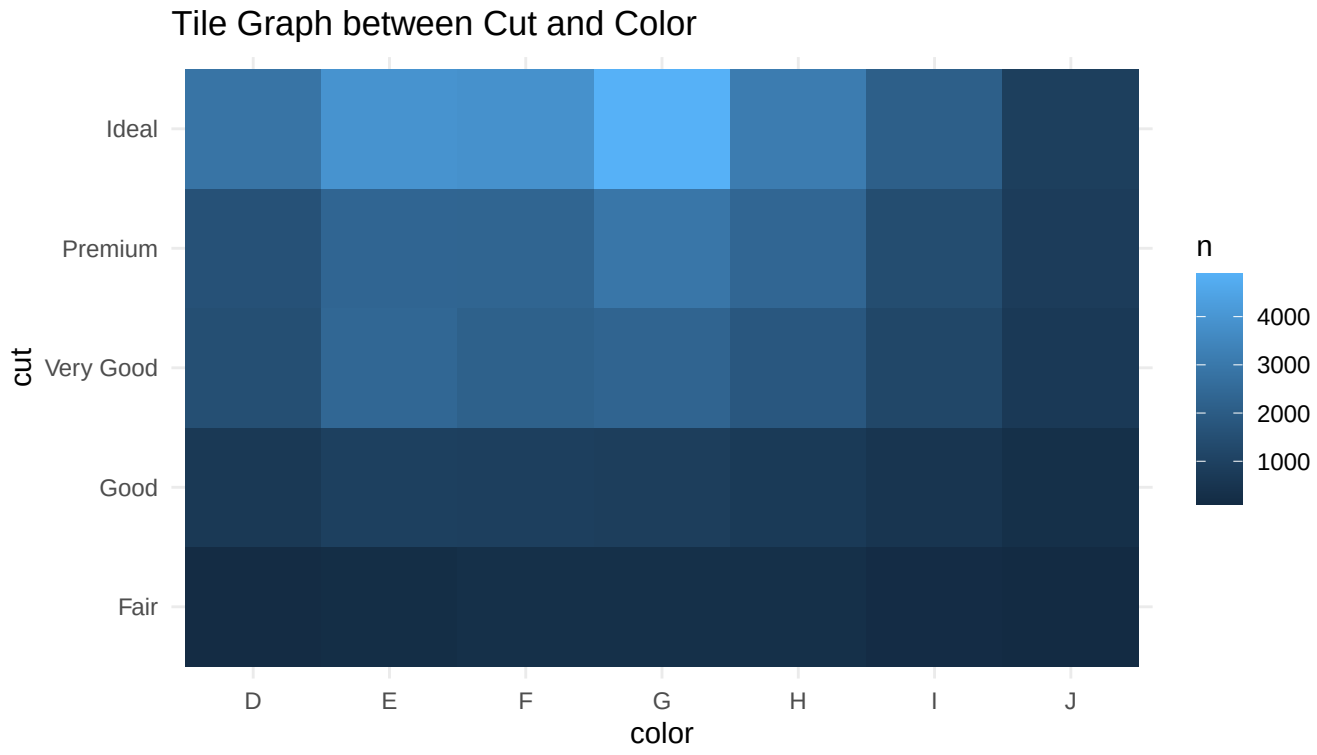
Now, we are interested in the relationship between two categroical variables **cut** and **color**. As these two variables are categorical, we can cross-tabulate the count of these two variables and visualise the correlation between them using the dot plot.

We notice that as the cut increases in quality, the count increases. But that does not imply any kind of a correlation as the color range is here

in the direction of top to bottom.

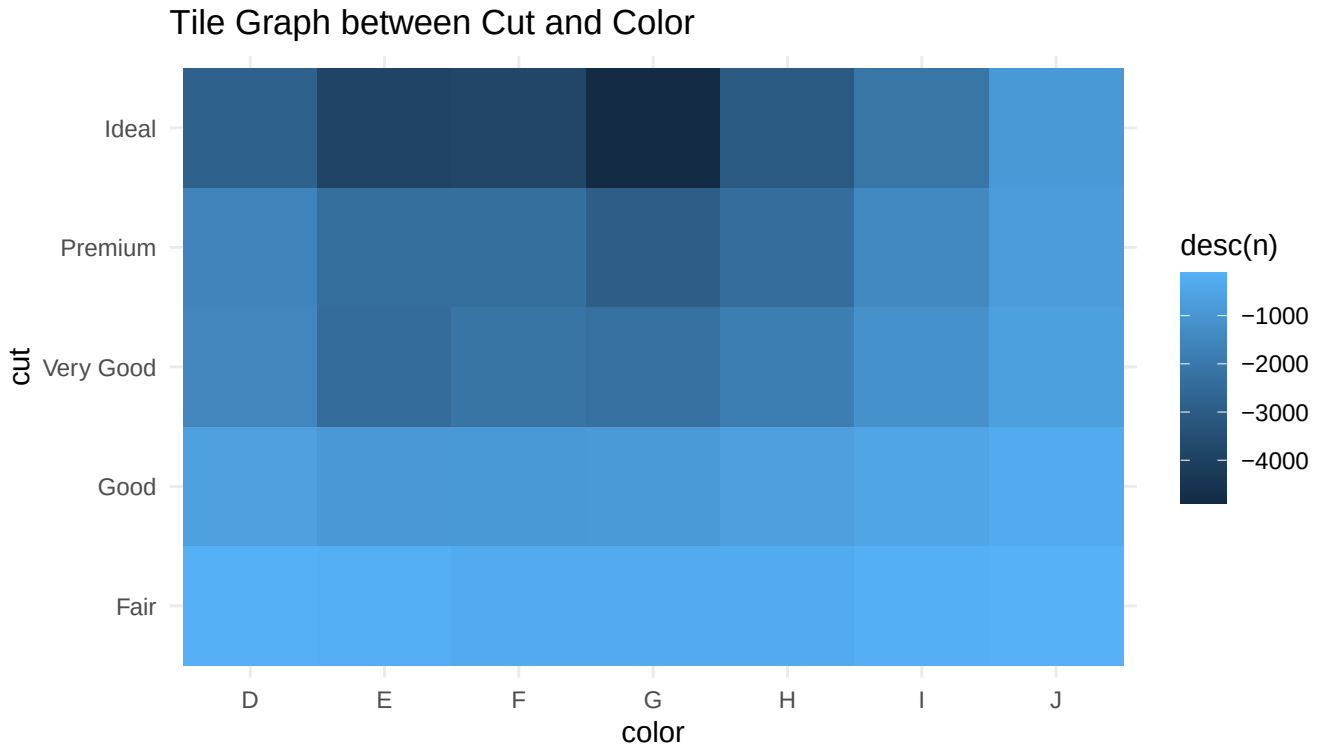
Tile Graph : Cut vs Color

Also, using a heatmap, we can visualise this information. Here, dark blue indicates lower count and light blue indicates higher count of the diamonds.



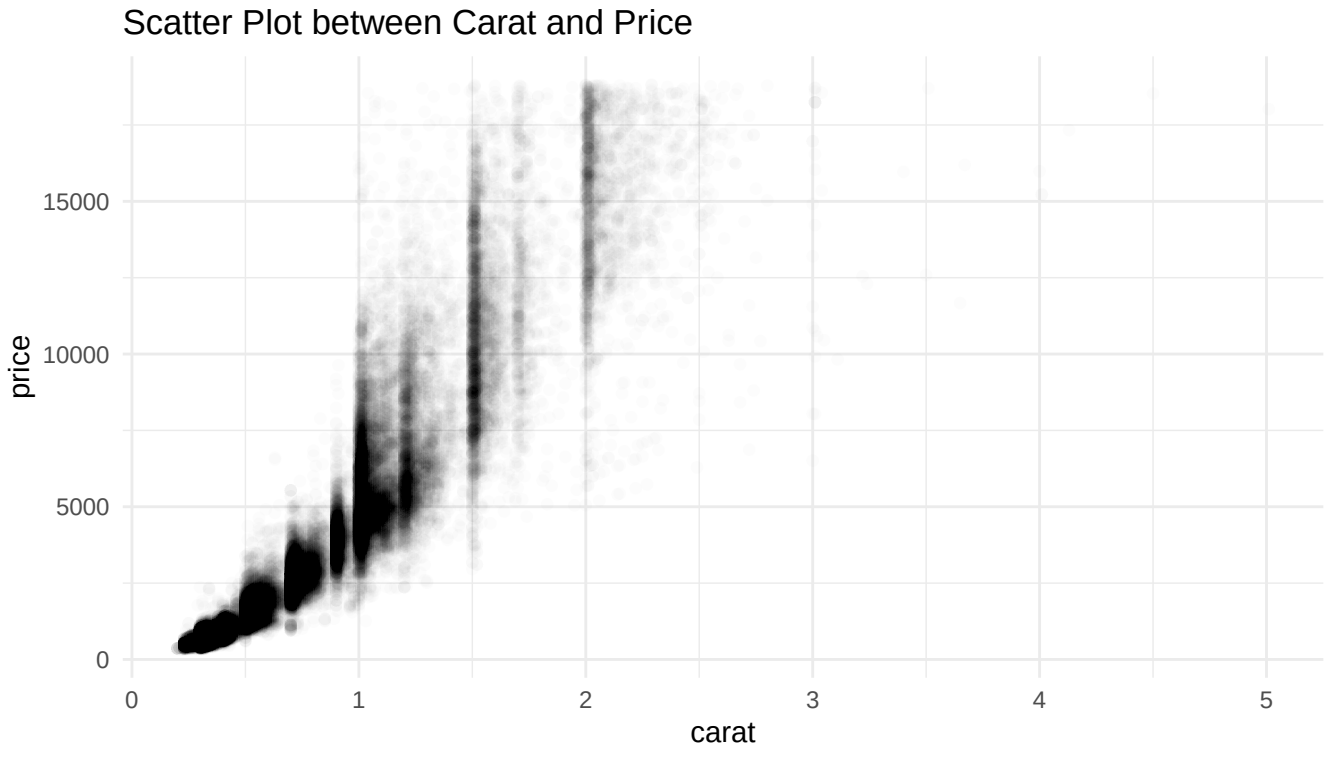
Now, we can clearly understand the covariance between two categorical variables. Because as the value of the count increases, the light gets light Here, we see that the correlation between the cut and the quality is pretty more balanced. But there is not really a very clear correlation between the colors directly with the cut. If the cut has a low quality color, it does not really play much of a role in the counts of a diamond. But for the higher quality diamonds, the color **G** and the surroundings have the more counts.

Tile Graph with the Descending Order of Count(n)



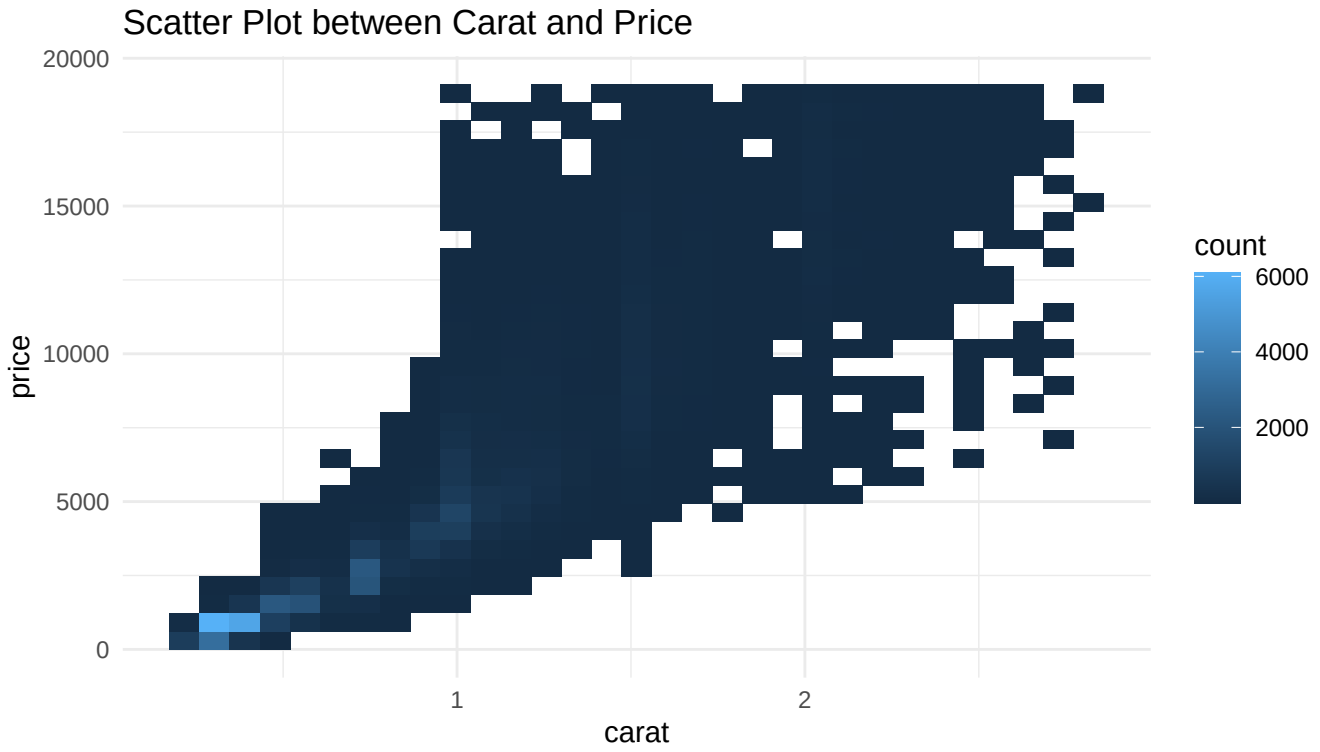
- ***Continuous vs Continuous***

Scatter Plot : Carat vs Price

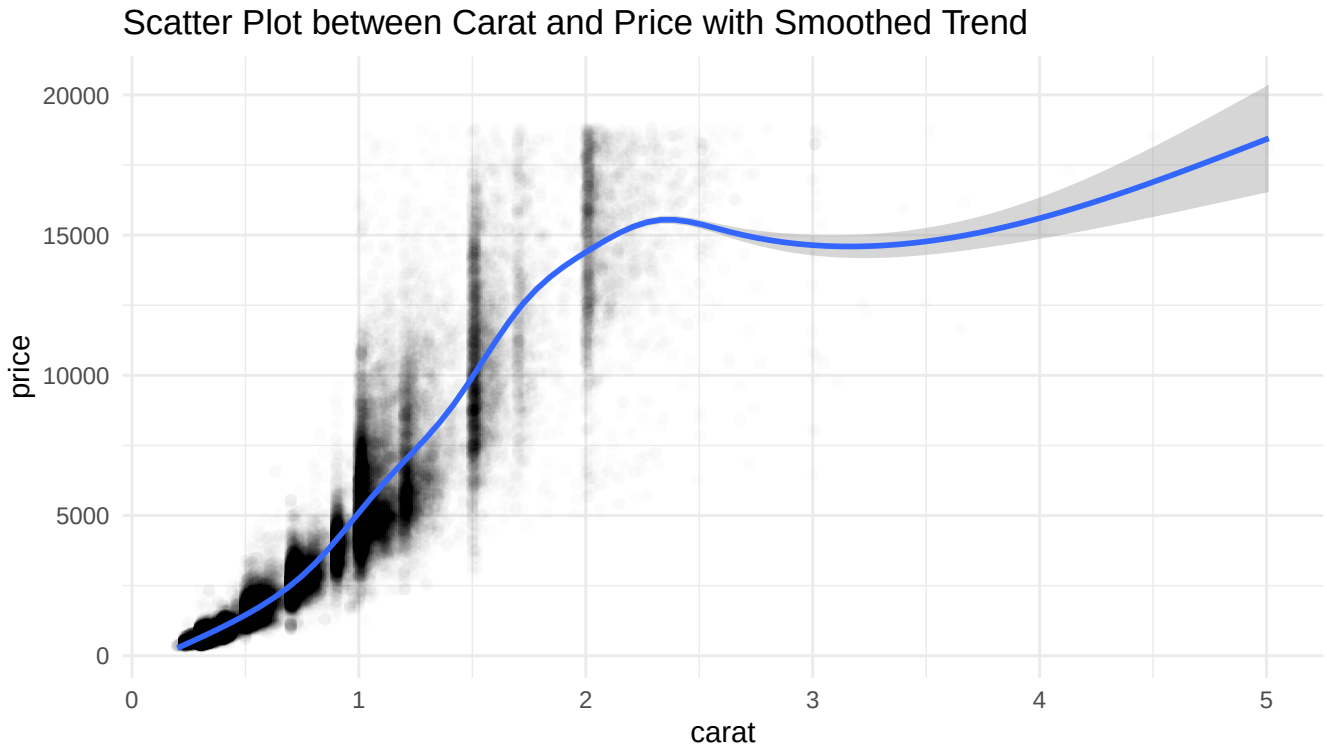


So,there is a clear correlation between the carat and the price of diamonds. We see that as the carat increases, the price does increase and it increases exponentially faster.

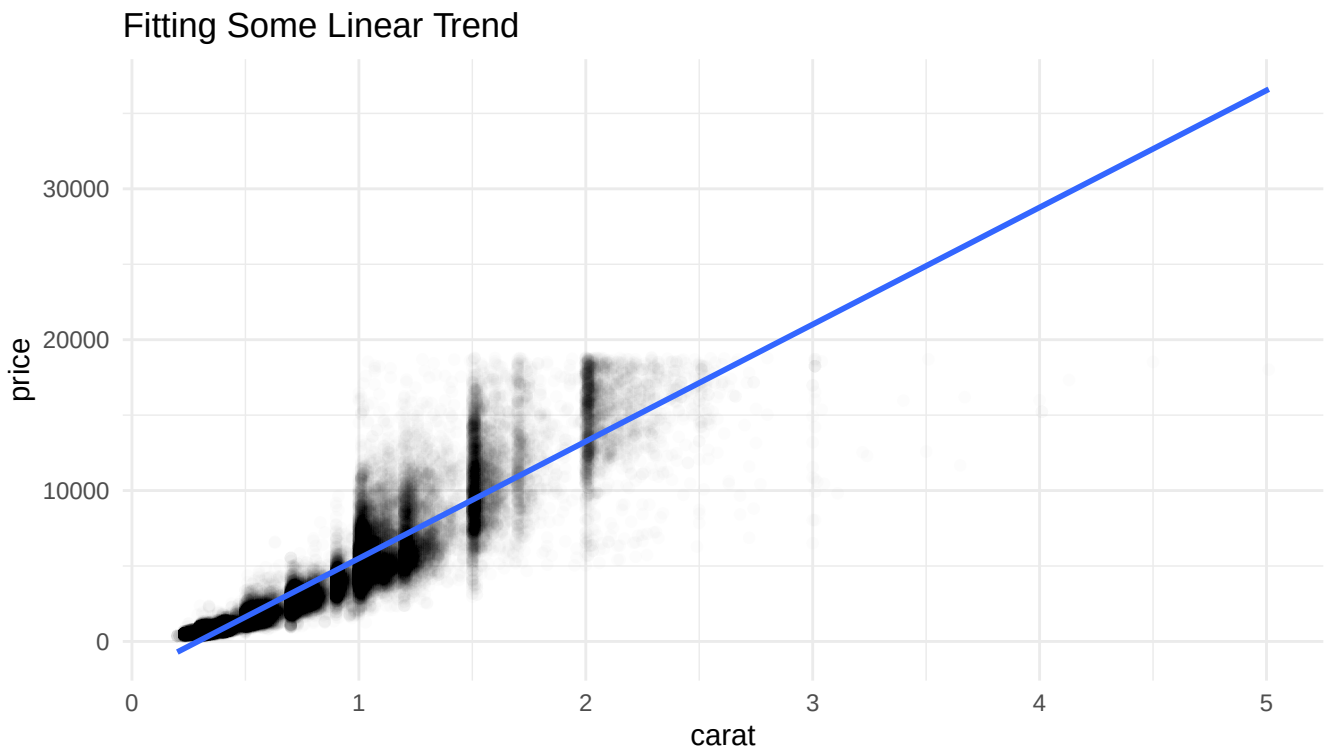
Scatter Plot : Carat vs Price (for the Smaller dataset)



- **Adding Smoothed Trend**

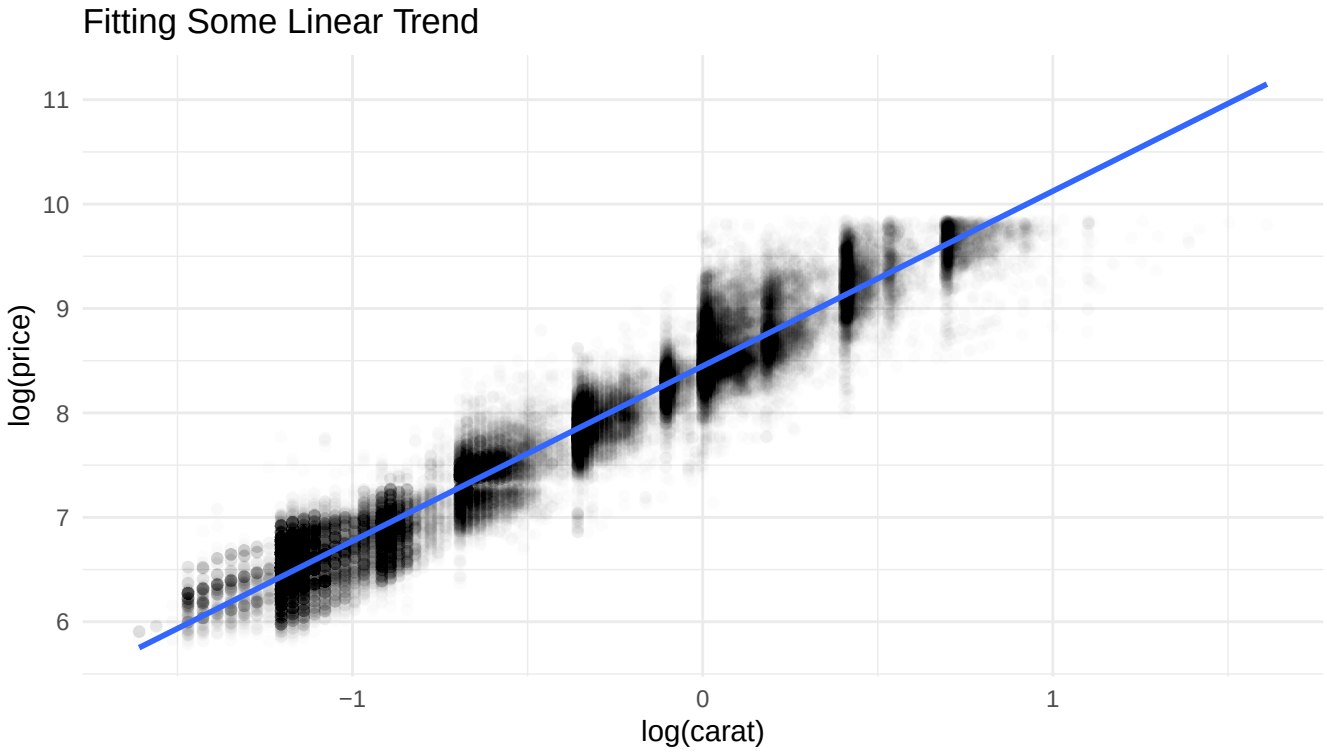


- **Fitting Some Linear Trend**



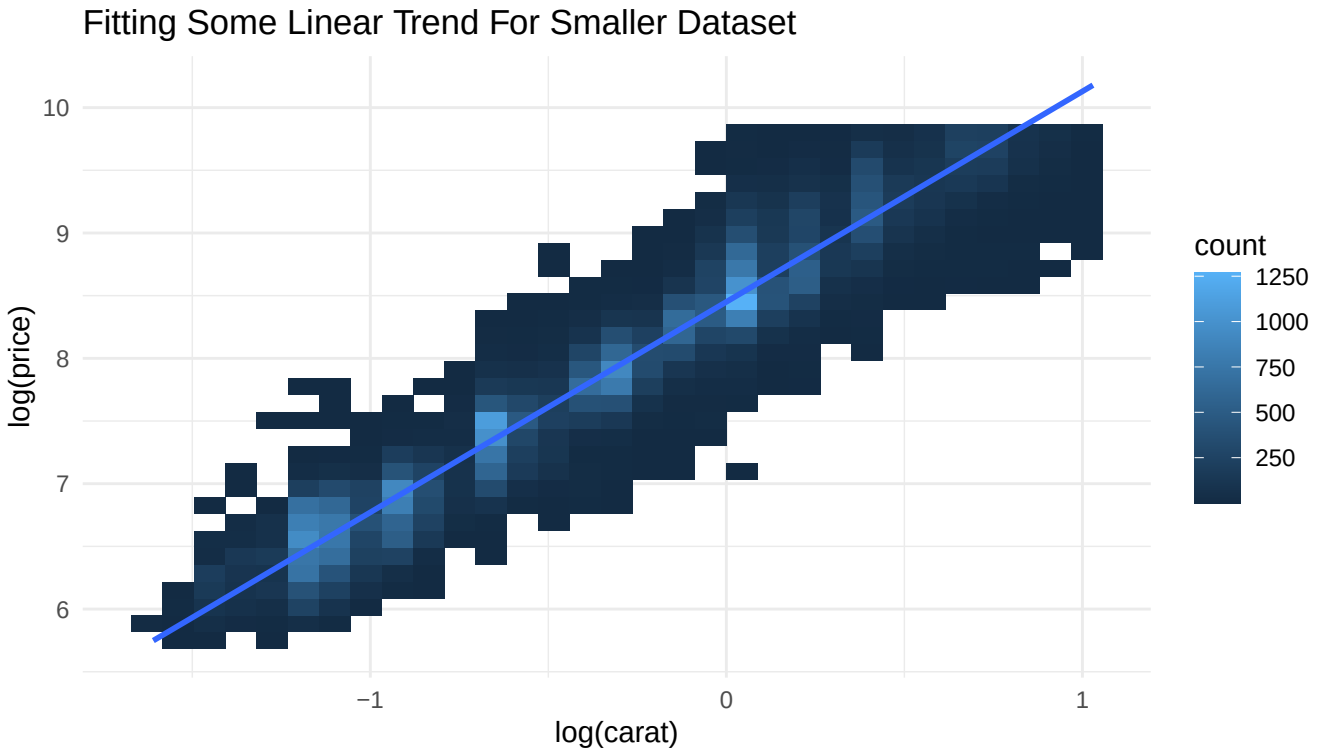
We can see that the fiiting is not good. Now, we are to transform both the variables carat and price to check whether this linear trend is a good assumption or, not.

- **Fitting Linear Trend After Transforming Carat and Price**



Now, this linear trend fitting is quite good.

- **For the Smaller Dataset**



Conclusion :-

- From the above exploratory data analysis, we can conclude that :-

1. The **ideal** cut is the most common and the **fair** cut is less common one.
2. Within each cut, **SI1** and **VS2** are the most common (highest percentage) clarity level.
3. The diamonds from the color **G** are the most ideal and premium quality diamonds.
4. There is a clear correlation between the carat and the price of diamonds.
5. The diamonds from the colors **F, G** and **H** are the fair cut diamonds.
6. Also, all cut group diamonds are rare in colour **J**.

R codes:-

```
library(ggplot2)
library(tibble)
library(dplyr)
library(xtable)

glimpse(diamonds)

sum(is.na(diamonds))

ggplot(data = diamonds,mapping = aes(x = cut)) +
  geom_bar() +
```

```
  geom_text(stat="count",aes(label=..count..), vjust = -1) +
  ylim(0, 25000) +
  theme(text = element_text(size=14)) +
  geom_bar(fill = 'darkorange3') +
  labs(x = "Quality of the Cut",
       y = "Diamonds Count",
       title = "Quality of the Diamonds") +
  theme_bw()

ggplot(data = diamonds,mapping = aes(x = color)) +
  geom_bar() +
  geom_text(stat="count",aes(label=..count..), vjust = -1) +
  ylim(0, 15000) +
  theme(text = element_text(size=14)) +
  geom_bar(fill = 'darkblue') +
  labs(x = "Color of the Diamonds",
       y = "Diamonds Count",
       title = "Color of the Diamonds") +
  theme_bw()

library(ggplot2)
ggplot(data = diamonds,mapping = aes(x = clarity)) +
  geom_bar() +
  geom_text(stat="count",aes(label=..count..), vjust = -1) +
  ylim(0, 16000) +
  theme(text = element_text(size=14)) +
  geom_bar(fill = 'yellow') +
  labs(x = "Clarity of Diamonds",
       y = "Diamonds Count",
       title = "Clarity of the Diamonds") +
  theme_bw()

unique(diamonds$cut)

levels(diamonds$clarity)

diamonds %>%
  count(cut, clarity) %>%
  arrange(desc(n))

ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = clarity), position = "dodge") +
  theme(axis.text.x = element_text(angle = 45)) +
  scale_fill_brewer(palette="YlOrRd") +
  labs(title = "Histogram : Grouping by Cut and Clarity") +
  theme_bw()

ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = clarity), position = "dodge") +
  labs(title = "Histogram : Grouping by Cut and Clarity") +
  scale_fill_brewer(palette="YlOrRd") +
  coord_flip() +
```

```
  theme_bw()

co_cl <- diamonds %>%
  group_by(cut) %>%
  count(clarity) %>%
  mutate(percent = (n/sum(n)) * 100,
         label = sprintf("%0.0f%%", percent))
ggplot(data = diamonds) +
  aes(x = cut, fill = clarity) +
  geom_bar(position = "fill") +
  geom_text(data=co_cl, aes(y=n,label=label),position=position_fill(vjust = 0
          size = 3) +
  scale_fill_brewer(palette="GnBu") +
  labs(y = "Proportion",
       title = "Proportions of Clarity Within Cut")

round(prop.table(table(diamonds$cut, diamonds$clarity))*100, 2)

ggplot(diamonds, aes(x = cut)) +
  geom_bar() +
  facet_wrap(~ clarity, strip.position = "top") +
  theme_minimal()

ggplot(data = diamonds) +
  geom_histogram(mapping = aes(x = carat), binwidth = 0.5) +
   labs(title = "Histogram for Carat") +
   theme_minimal()

diamonds %>%
  count(cut_width(carat, 0.5))

smaller <- diamonds %>%
  filter(carat < 3)
ggplot(data = smaller) +
  geom_histogram(mapping = aes(x= carat), binwidth = 0.1) +
  labs(title = "Histogram : Carat for smaller dataset")

ggplot(data = smaller, mapping = aes(x = carat, color = cut)) +
  geom_freqpoly(binwidth = 0.1) +
  labs(title = "Frequency Polygon for Carat vs Cut") +
  theme_minimal()

ggplot(data =  smaller, mapping = aes(x =  carat)) +
  geom_histogram(binwidth = 0.01) +
  labs(title = "Histogram of Carat for Binwidth = 0.01") +
  theme_bw()

ggplot(diamonds) +
  geom_histogram(mapping = aes(x = y), binwidth = 0.5) +
  labs(x = "Width of Diamonds",
       y = "Diamonds Count",
       title = "Histogram for the Width of the Diamonds") +
```

```
  theme_minimal()

diamonds %>%
  filter(y < 3 | y > 20) %>%
  select(price, x, y, z) %>%
  arrange(y) %>%
  xtable()

diamonds2 <- diamonds %>%
  filter(between(y, 3, 20))

ggplot(data = diamonds2) +
  geom_histogram(mapping = aes(x = y), binwidth = 0.5) +
  labs(x = "Width of Diamonds",
       y = "Diamonds Count",
       title = "Histogram of Width of a Diamond between 3 and 20") +
  theme_minimal()

diamonds %>%
  ggplot(aes(x = x, y = y)) +
  geom_point() +
  labs(x = "Length of Diamonds",
       y = "Width of Diamonds",
       title = "Scatter Plot between the Length and the Width of a Diamond)")
  theme_minimal()

diamonds %>%
  mutate(y = ifelse(y < 3 | y > 20, NA, y)) %>%
  ggplot(aes(x = x, y = y)) +
  geom_point() +
  labs(x = "Length of Diamonds",
       y = "Width of Diamonds",
       title = "Scatter Plot between the Length and the Width of a Diamond)")
  theme_minimal()

ggplot(data = diamonds, aes(x = price)) +
  geom_freqpoly(aes(color = cut), bindwidth = 500) +
  labs(title = "Frequency Polygon for Cut vs Price") +
  theme_minimal()

ggplot(data = diamonds, aes(x = price)) +
  geom_density(mapping = aes(fill = factor(cut)), binwidth = 500, alpha = 0.4
  labs(title = "Density Plot for Cut vs Price") +
  theme_minimal()

ggplot(data = diamonds, aes(x = cut, y = price)) +
  geom_boxplot(fill = "lightgreen") +
  labs(title = "Box PLot of Price, grouped by Color") +
  theme_minimal()

ggplot(data = diamonds, aes(x = cut, y = price)) +
  geom_boxplot(fill = "skyblue") +
```

```
  coord_flip() +
  labs(title = "Box PLotbox plot of Price, grouped by Cut") +
  theme_minimal()

ggplot(data = diamonds) +
  geom_count(mapping = aes(x = cut, y = color)) +
  labs(title = "Dot plot between Cut and Color") +
  theme_minimal()

diamonds %>%
  count(color, cut) %>%
  ggplot(aes(x = color, y = cut)) +
  geom_tile(aes(fill = n)) +
  labs(title = "Tile Graph between Cut and Color") +
  theme_minimal()

diamonds %>%
  count(color, cut) %>%
  ggplot(aes(x = color, y = cut)) +
  geom_tile(aes(fill = desc(n))) +
  labs(title = "Tile Graph between Cut and Color") +
  theme_minimal()

ggplot(data = diamonds) +
  geom_point(mapping = aes(x = carat, y = price), alpha = (1/100)) +
  labs(title = "Scatter Plot between Carat and Price") +
  theme_minimal()

ggplot(data = smaller) +
  geom_bin2d(mapping = aes(x = carat, y = price)) +
  labs(title = "Scatter Plot between Carat and Price")+
  theme_minimal()

ggplot(data = diamonds, aes(x = carat, y = price)) +
  geom_point(alpha = (1/100)) +
  geom_smooth() +
  labs(title = "Scatter Plot between Carat and Price with Smoothed Trend")+
  theme_minimal()

ggplot(data = diamonds, aes(x = carat, y = price)) +
  geom_point(alpha = (1/100)) +
  geom_smooth(method = "lm") +
  labs(title = "Fitting Some Linear Trend")+
  theme_minimal()

ggplot(data = diamonds, aes(x = log(carat), y = log(price))) +
  geom_point(alpha = (1/100)) +
  geom_smooth(method = "lm") +
  labs(title = "Fitting Some Linear Trend")+
  theme_minimal()

ggplot(data = smaller, aes(x = log(carat), y = log(price))) +
```

```
  geom_bin2d() +
  geom_smooth(method = "lm") +
  labs(title = "Fitting Some Linear Trend For Smaller Dataset")+
  theme_minimal()
```