1. **Qual é a estrutura do conjunto de dados “diamantes”?**

```
[1]  install.packages('ggplot2')
     library(ggplot2)

     dados = diamonds

     Installing package into ‘/usr/local/lib/R/site-library’
     (as ‘lib’ is unspecified)

     str(diamonds)

     tibble [53,940 × 10] (S3: tbl_df/tbl/data.frame)
      $ carat  : num [1:53940] 0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
      $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
      $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
      $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...
      $ depth  : num [1:53940] 61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
      $ table  : num [1:53940] 55 61 65 58 58 57 57 55 61 61 ...
      $ price  : int [1:53940] 326 326 327 334 335 336 336 337 337 338 ...
      $ x      : num [1:53940] 3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
      $ y      : num [1:53940] 3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
      $ z      : num [1:53940] 2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

A estrutura deste conjunto de dados é heterogênea, sendo composta por variáveis qualitativas ordinais e variáveis quantitativas discretas e contínuas.

2. **Explore a parte inicial e a final do conjunto de dados.**

```
[4]  head(diamonds)
```

A tibble: 6 × 10

carat	cut	color	clarity	depth	table	price	x	y	z
<dbl>	<ord>	<ord>	<ord>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	2.48

```
tail(diamonds)
```

A tibble: 6 × 10

carat	cut	color	clarity	depth	table	price	x	y	z
<dbl>	<ord>	<ord>	<ord>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>
0.72	Premium	D	SI1	62.7	59	2757	5.69	5.73	3.58
0.72	Ideal	D	SI1	60.8	57	2757	5.75	5.76	3.50
0.72	Good	D	SI1	63.1	55	2757	5.69	5.75	3.61
0.70	Very Good	D	SI1	62.8	60	2757	5.66	5.68	3.56
0.86	Premium	H	SI2	61.0	58	2757	6.15	6.12	3.74
0.75	Ideal	D	SI2	62.2	55	2757	5.83	5.87	3.64

3. **Faça alguns sumários estatísticos para entender melhor a base de dados.**

```
dim(diamonds)
```

```
53940 · 10
```

4. **A saída da função summary() está de acordo com a descrição mostrada anteriormente?**

```
summary(diamonds)
```

```
     carat               cut        color        clarity          depth      
 Min.   :0.2000   Fair      : 1610   D: 6775   SI1     :13065   Min.   :43.00  
 1st Qu.:0.4000   Good      : 4906   E: 9797   VS2     :12258   1st Qu.:61.00  
 Median :0.7000   Very Good:12082   F: 9542   SI2     : 9194   Median :61.80  
 Mean   :0.7979   Premium   :13791   G:11292   VS1     : 8171   Mean   :61.75  
 3rd Qu.:1.0400   Ideal     :21551   H: 8304   VVS2    : 5066   3rd Qu.:62.50  
 Max.   :5.0100                     I: 5422   VVS1    : 3655   Max.   :79.00  
                                    J: 2808   (Other): 2531                  
     table           price             x                y         
 Min.   :43.00   Min.   :  326   Min.   : 0.000   Min.   : 0.000  
 1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710   1st Qu.: 4.720  
 Median :57.00   Median : 2401   Median : 5.700   Median : 5.710  
 Mean   :57.46   Mean   : 3933   Mean   : 5.731   Mean   : 5.735  
 3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540   3rd Qu.: 6.540  
 Max.   :95.00   Max.   :18823   Max.   :10.740   Max.   :58.900  

       z         
 Min.   : 0.000  
 1st Qu.: 2.910  
 Median : 3.530  
 Mean   : 3.539  
 3rd Qu.: 4.040  
 Max.   :31.800  
```

Sim, está de acordo, pois temos 10 colunas e 53940 linhas.

5. **Explore a variável price, seguindo o modelo de exploração.**

```
str(diamonds$price)
```

```
int [1:53940] 326 326 327 334 335 336 336 337 337 338 ...
```

```
head(diamonds$price)
tail(diamonds$price)
```

```
326 · 326 · 327 · 334 · 335 · 336
2757 · 2757 · 2757 · 2757 · 2757 · 2757
```

```
is.nan(diamonds$price)
```

```
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · ... · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE ·
FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE · FALSE
```

Por fim, uma exploração mais rebuscada, verificando sua integração com outras variáveis.

Diamantes com preço abaixo de $1000, rearranjado de maneira crescente, mostrando também as colunas cut, comprimento, largura e profundidade. Por fim, exibe-se a quantidade de linhas do dataframe

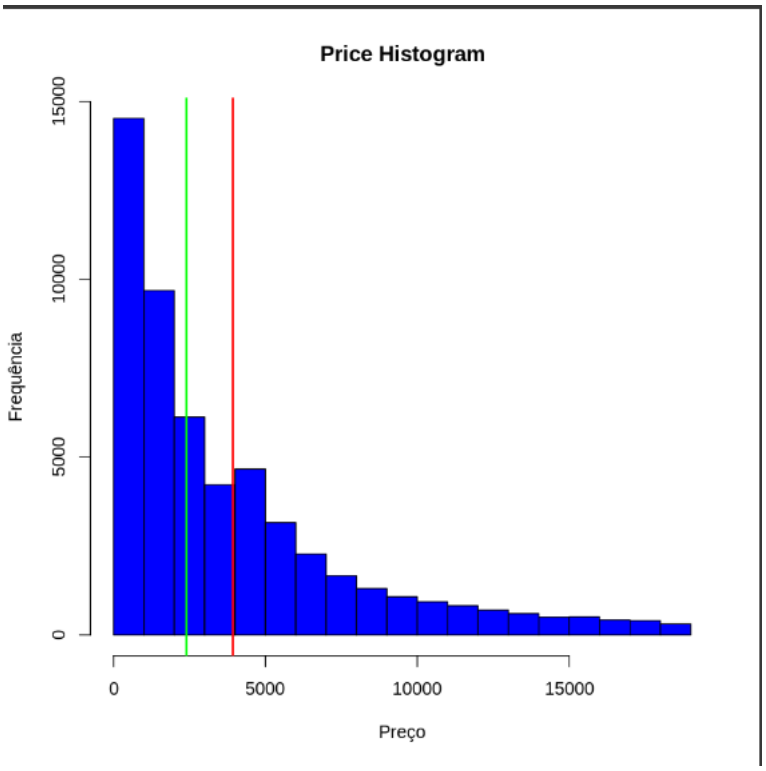
```
diamante = filter(diamonds, price < 1000) %>%
  select(price, cut, clarity, x, y, z)
diamante = arrange(diamante, price)
diamante
nrow(diamante)
```

998	Ideal	VVS2	4.57	4.61	2.84
998	Ideal	VS1	4.53	4.56	2.74
998	Ideal	VS1	4.68	4.64	2.91
998	Premium	VS1	4.71	4.68	2.86
998	Ideal	VS2	4.68	4.64	2.89
998	Ideal	VS1	4.64	4.59	2.90
998	Premium	VS1	4.63	4.57	2.87
998	Premium	VS2	4.66	4.62	2.86
998	Premium	VS2	4.67	4.63	2.83
998	Ideal	VS1	4.64	4.58	2.88
998	Premium	SI1	4.88	4.85	3.03
999	Very Good	IF	4.54	4.56	2.78
999	Ideal	SI1	5.20	5.23	3.21
999	Ideal	IF	4.49	4.51	2.78
999	Ideal	VS2	4.78	4.76	2.95
999	Ideal	VS2	4.79	4.76	2.95
999	Ideal	VS2	4.79	4.76	2.98
999	Premium	VS2	4.87	4.83	2.90
999	Premium	VS2	4.79	4.74	2.92
999	Ideal	VS2	4.79	4.74	2.96
999	Premium	VS2	4.78	4.75	2.96
999	Ideal	VS2	4.79	4.75	2.94
999	Ideal	VS2	4.79	4.72	2.95
999	Ideal	VS2	4.76	4.74	2.95
999	Ideal	VS2	4.77	4.74	2.96
999	Ideal	VS2	4.77	4.74	2.97
999	Ideal	VS2	4.78	4.74	2.94
999	Premium	VVS2	4.86	4.79	2.88
999	Premium	VS1	4.70	4.66	2.88
999	Ideal	VS1	4.68	4.65	2.86

14499

6. **Veja a distribuição da variável (histograma); observe a faixa de valores da variável e também**
7.

```
8.  hist(diamonds$price, col = 'blue', main = 'Price Histogram', xlab = 'Preço', ylab = 'Frequência'
    )
9.  abline(v=median(diamonds$price), col="green",lwd=2)
10. abline(v = mean(diamonds$price), col = "red", lwd = 2)
11.
```

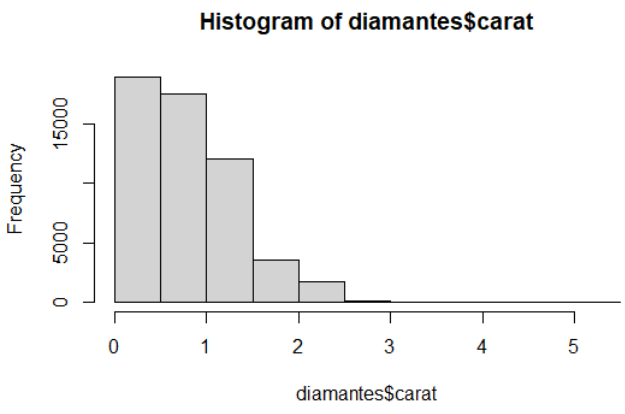


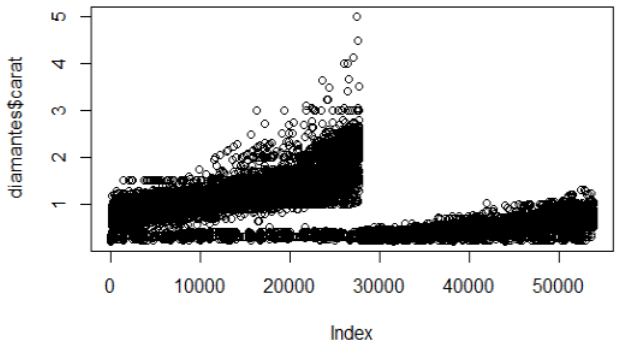
7. **Explore também as variáveis carat, cut, color, clarity, x, y, z, depth e table, seguindo o modelo de exploração.**

CARAT

```
head(diamonds$carat)
tail(diamonds$carat)
```

```
0.23 · 0.21 · 0.23 · 0.29 · 0.31 · 0.24
0.72 · 0.72 · 0.72 · 0.7 · 0.86 · 0.75
```

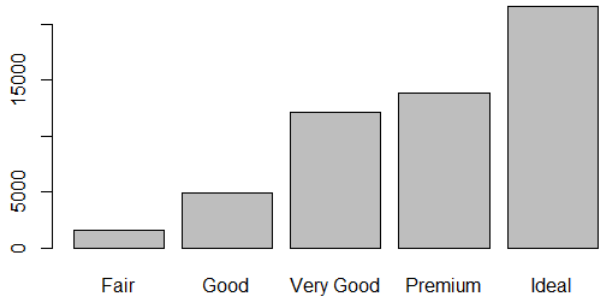




CUT

```
head(diamonds$cut)
tail(diamonds$cut)
```

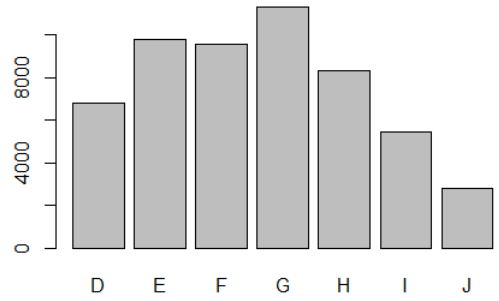
```
Ideal · Premium · Good · Premium · Good · Very Good
▶ Levels:
Premium · Ideal · Good · Very Good · Premium · Ideal
▶ Levels:
```



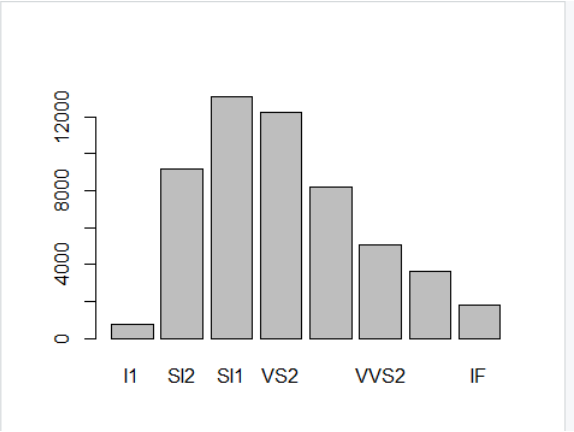
COLOR

```
head(diamonds$color)
tail(diamonds$color)
```

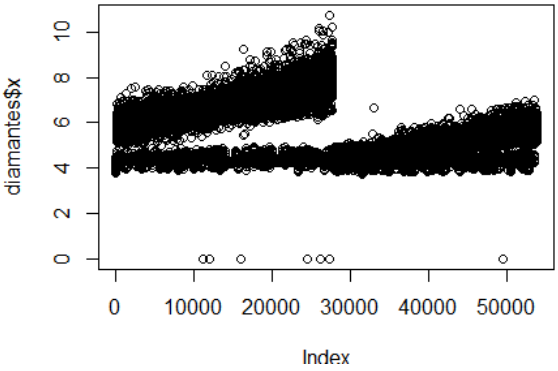
```
E · E · E · I · J · J
▶ Levels:
D · D · D · D · H · D
▶ Levels:
```

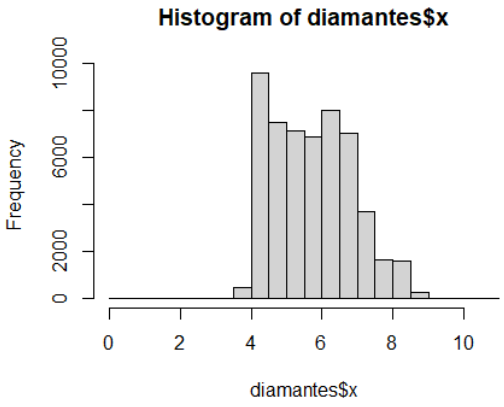


CLARITY

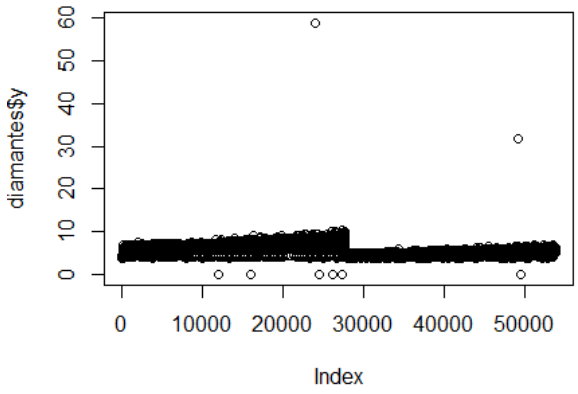

0
4000
8000
12000
I1
SI2
SI1
VS2
VVS2
IF

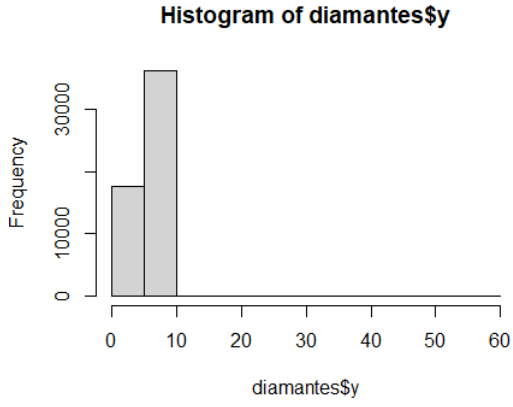
X


diamantes$x
0
2
4
6
8
10
0
10000
20000
30000
40000
50000
Index

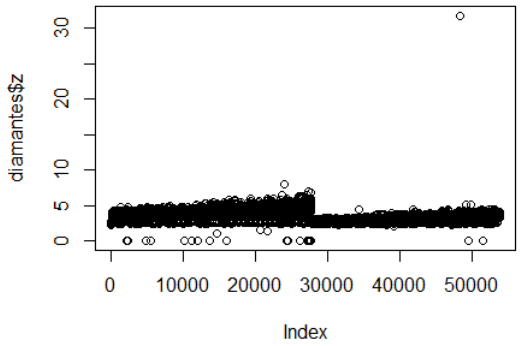

Histogram of diamantes$x
Frequency
0
2000
6000
10000
0
2
4
6
8
10
diamantes$x

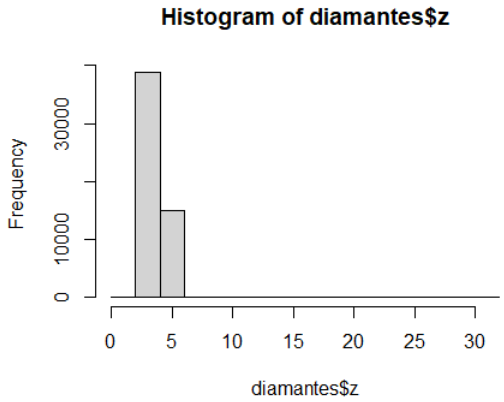
Y


diamantes$y
0
10
20
30
40
50
60
0
10000
20000
30000
40000
50000
Index

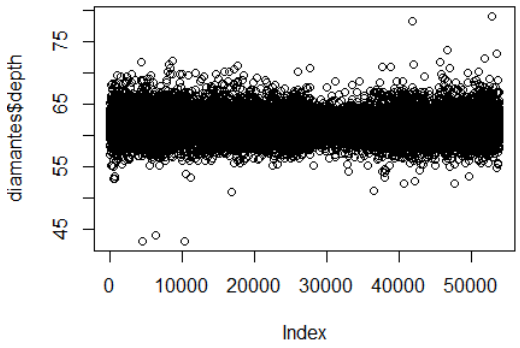

Histogram of diamantes$y
Frequency
0
10000
30000
0
10
20
30
40
50
60
diamantes$y

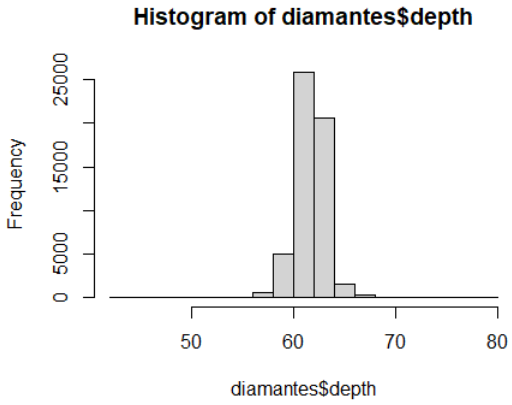
Z



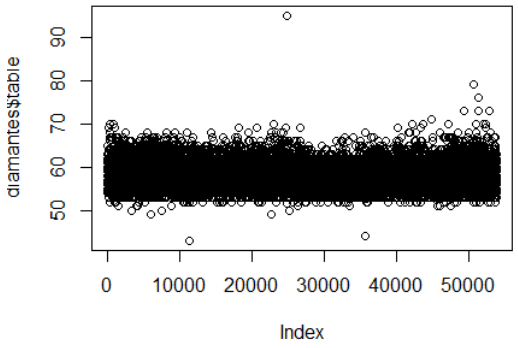


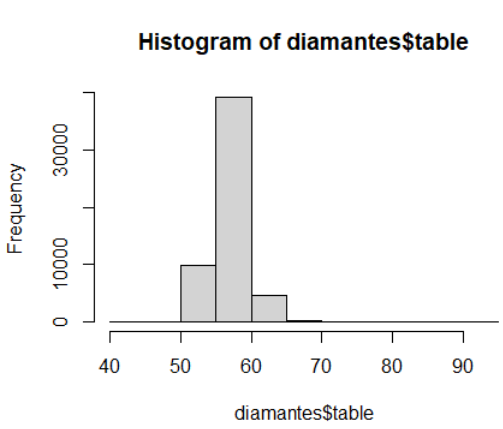
DEPTH





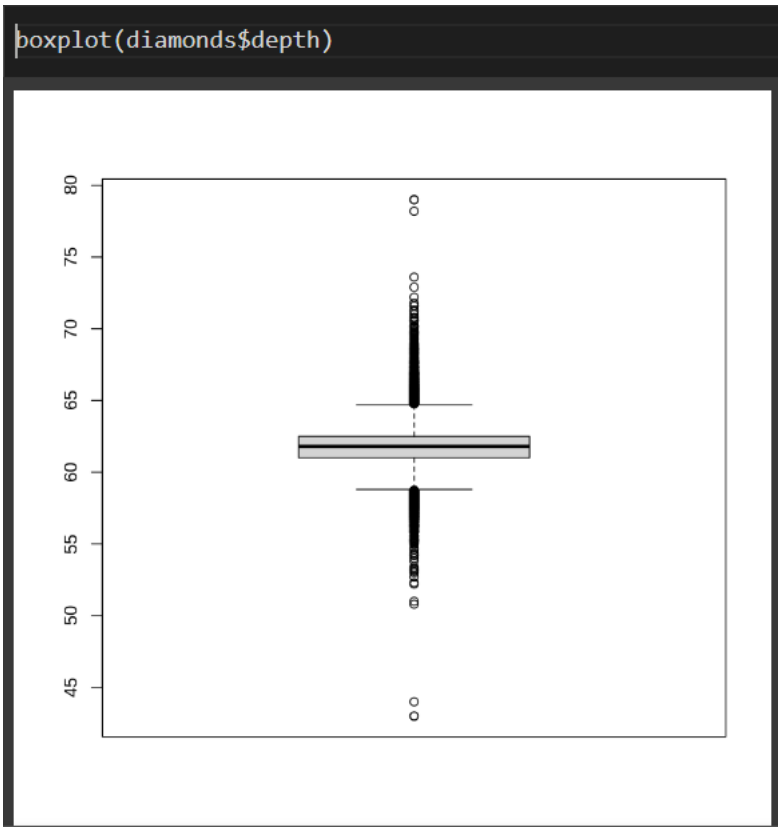
TABLE





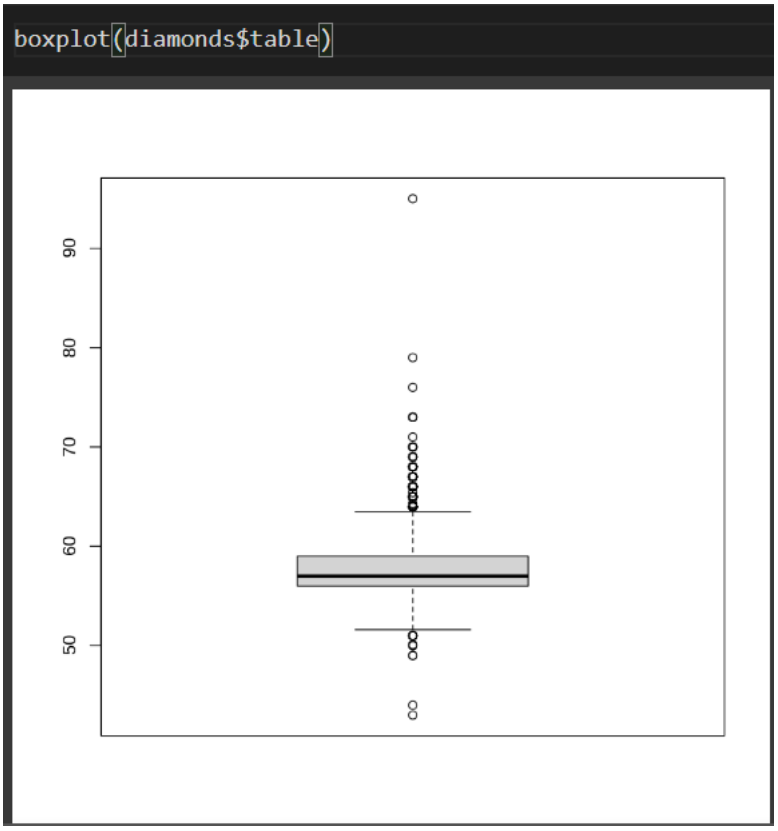
8. Crie boxplots para as variáveis numéricas; veja se existem dados anormais (outliers)

```
boxplot(diamonds$depth)
```



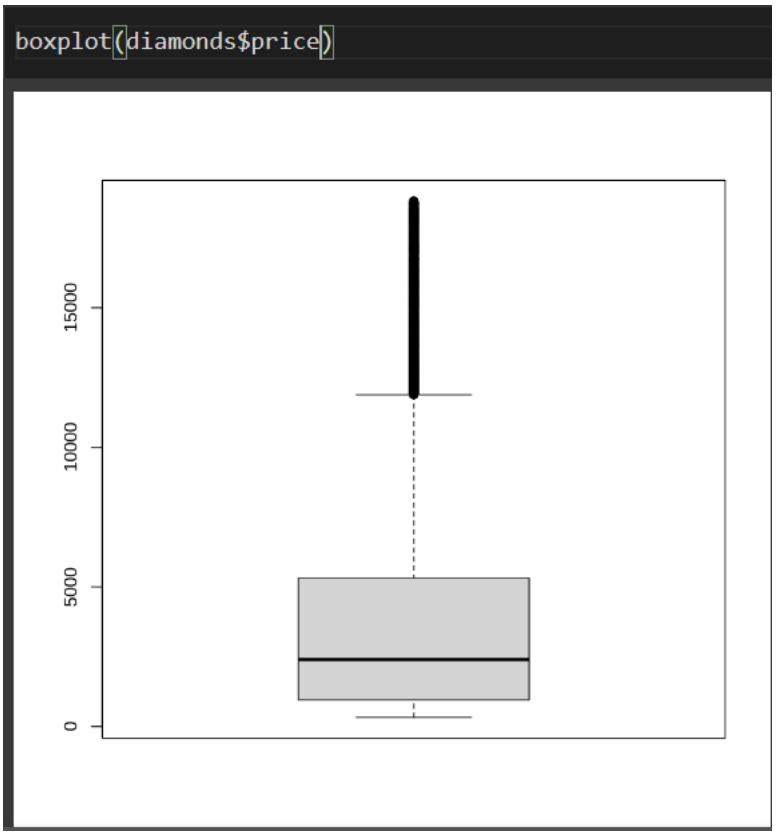
Outliers: abaixo de 45 e acima de 75

```
boxplot(diamonds$table)
```



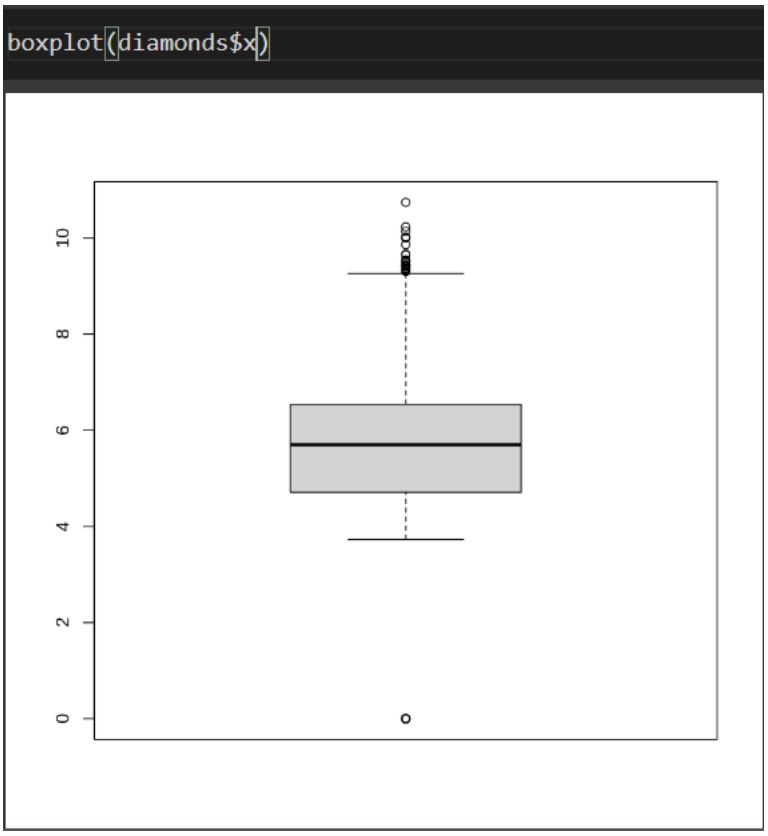
Outliers: abaixo de 50 e acima de 70

```
boxplot(diamonds$price)
```



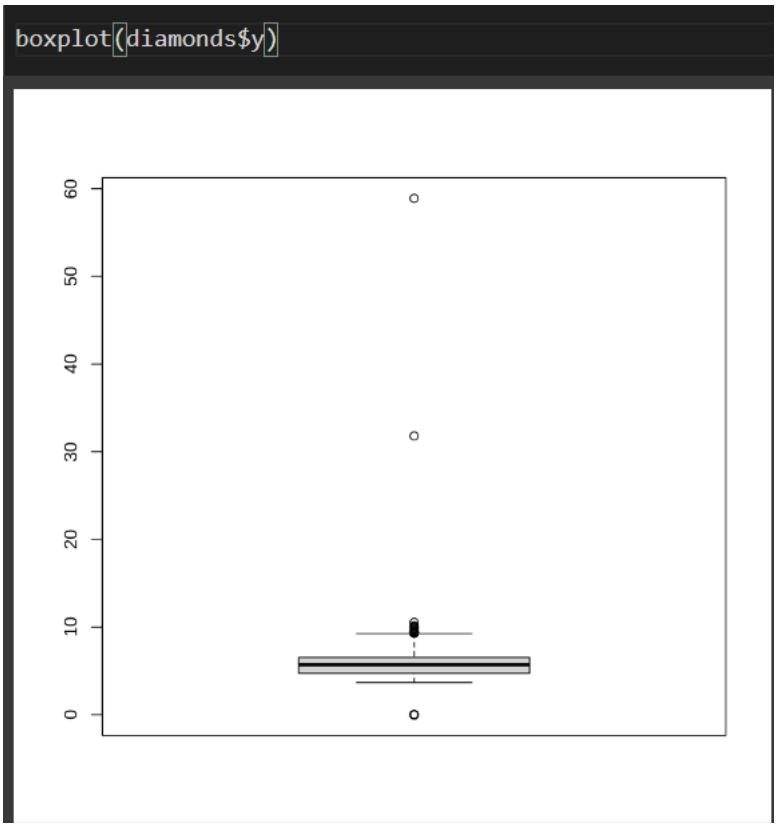
Não há dados anormais

```
boxplot(diamonds$x)
```



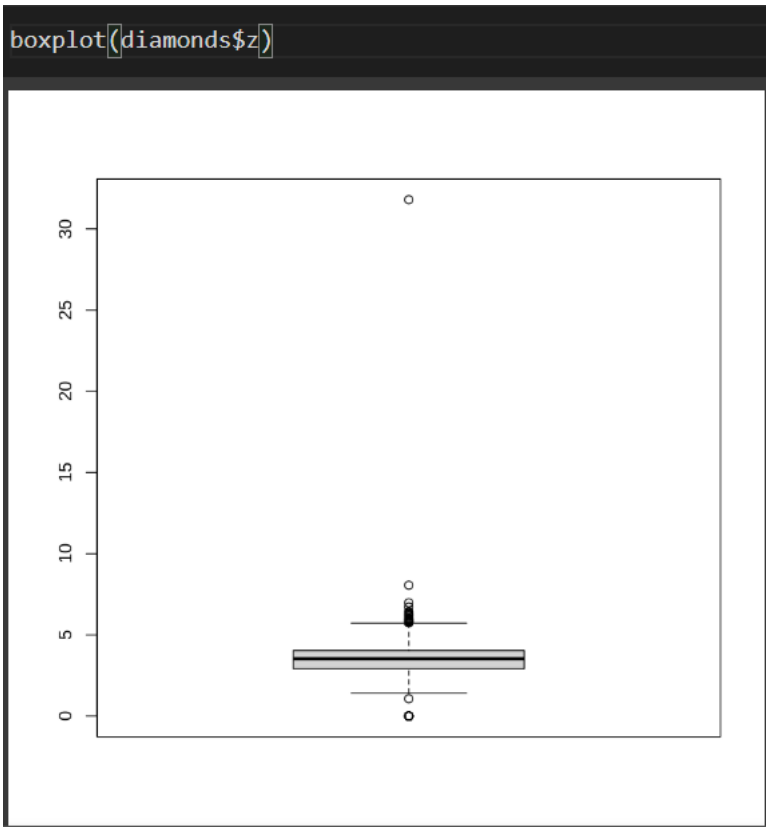
Outliers: abaixo de 2 e acima de 10

```
boxplot(diamonds$y)
```



Outliers: 0 e acima de 20

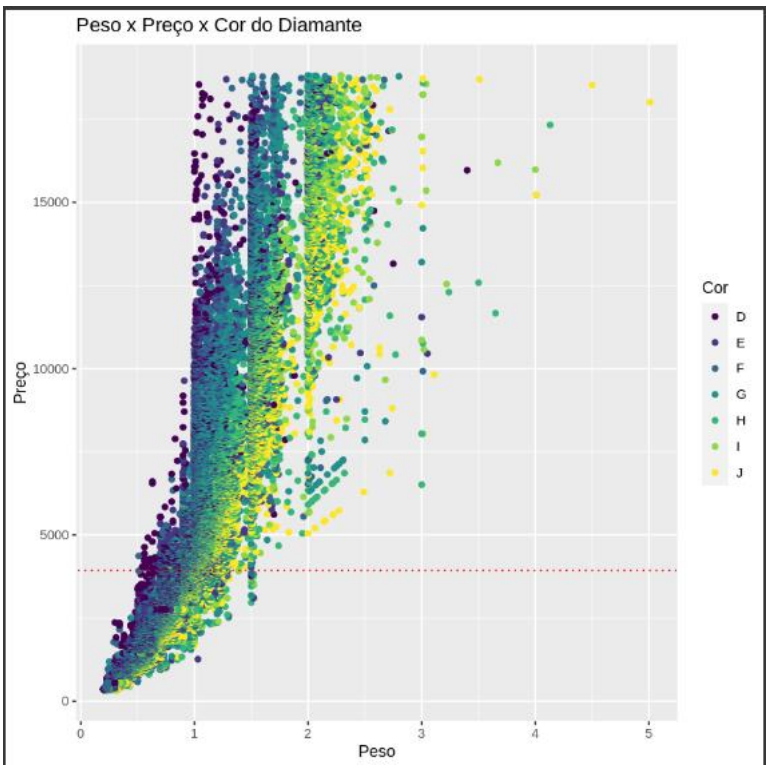
```
boxplot(diamonds$z)
```



Outliers: 0 e acima de 20

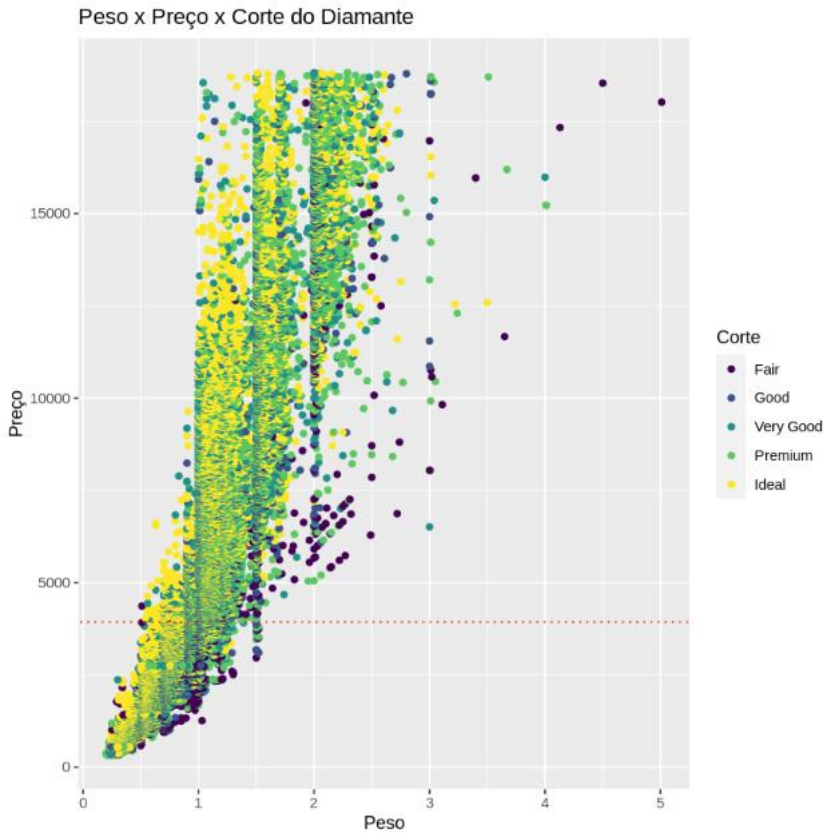
9. **Utilize as variáveis categóricas para fazer o facetamento dos dados, mostrando alguns gráficos com 2 ou mais variáveis contínuas lado a lado.**

```
ggplot(diamonds) +
  geom_point(aes(x = carat, y = price, color = color)) +
  geom_abline(slope = 0, intercept = mean(diamonds$price), color = 'red',
              linetype = 3) +
  labs(title = 'Peso x Preço x Cor do Diamante', y = 'Preço', x = 'Peso',
       col = 'Cor')
```



Com esse gráfico de dispersão de 3 variáveis (1 categórica e 2 contínuas) é possível identificar que quanto mais pesado, mais caro é o diamante, e, quanto mais pesado e mais caro, maior a probabilidade da cor deste diamante estar acima de “G”.

```
ggplot(diamonds) +
  geom_point(aes(x = carat, y = price, color = cut)) +
  geom_abline(slope = 0, intercept = mean(diamonds$price), color = 'red',
              linetype = 3) +
  labs(title = 'Peso x Preço x Corte do Diamante', y = 'Preço', x = 'Peso',
       col = 'Corte')
```



Com esse gráfico de dispersão de 3 variáveis (1 categórica e 2 contínuas) é possível identificar que diamantes pesados são vendidos por preços elevados independentemente da qualidade do corte. Em contrapartida, é possível observar diamantes quase que na média do peso (0,79) sendo vendidos por preços elevados por conta do corte “ideal”.