R_ggplot2

Ulises Jose Bustamante Mora

2023-10-29

Importing libraries

```
library(ggplot2)
library(tidyr)
library(dplyr)
library(cowplot)
library(GGally)
```

Getting into the datasert

```
str(diamonds)
```

```
## tibble [53,940 x 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 ...
```

```
nrow(diamonds)
```

```
## [1] 53940
```

```
ncol(diamonds)
```

```
## [1] 10
```

```
head(diamonds)
```

```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
## 3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

```
summary(diamonds)
```

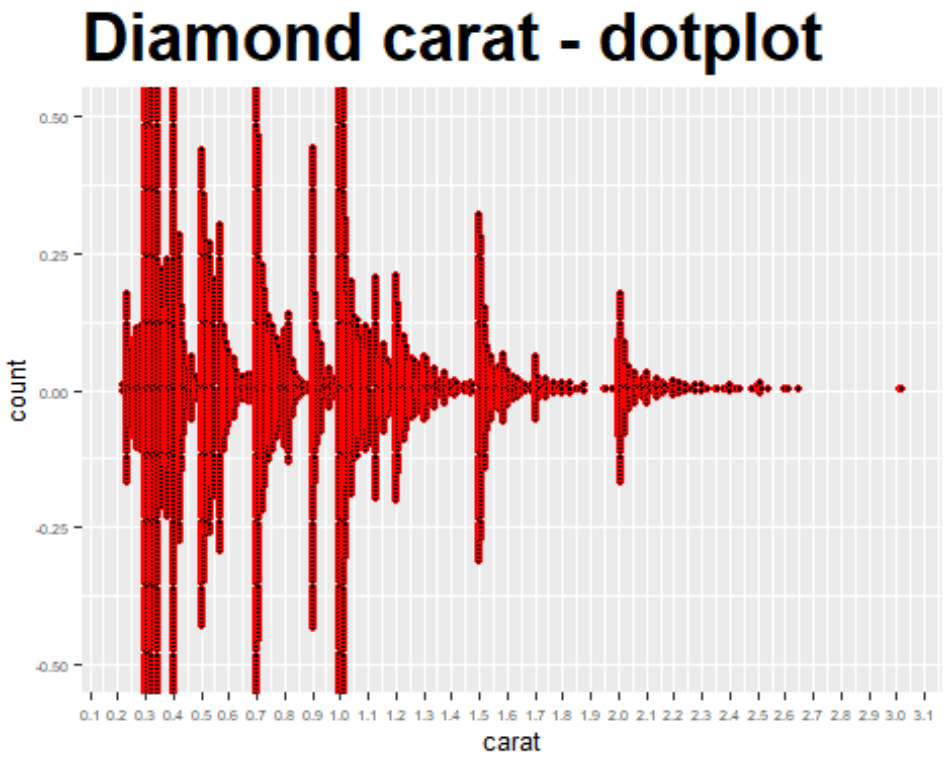
```
##      carat               cut        color        clarity
depth
##  Min.   :0.2000   Fair      : 1610   D: 6775   SI1    :13065   Min.
:43.00
##  1st Qu.:0.4000   Good      : 4906   E: 9797   VS2    :12258   1st
Qu.:61.00
##  Median :0.7000   Very Good:12082   F: 9542   SI2    : 9194   Median
:61.80
##  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
##
```

Getting a sample of the dataset

```
set.seed(123)
diamondsCopy = diamonds
diamondsSample = diamondsCopy %>% sample_n(size = 5000, replace = F)
```

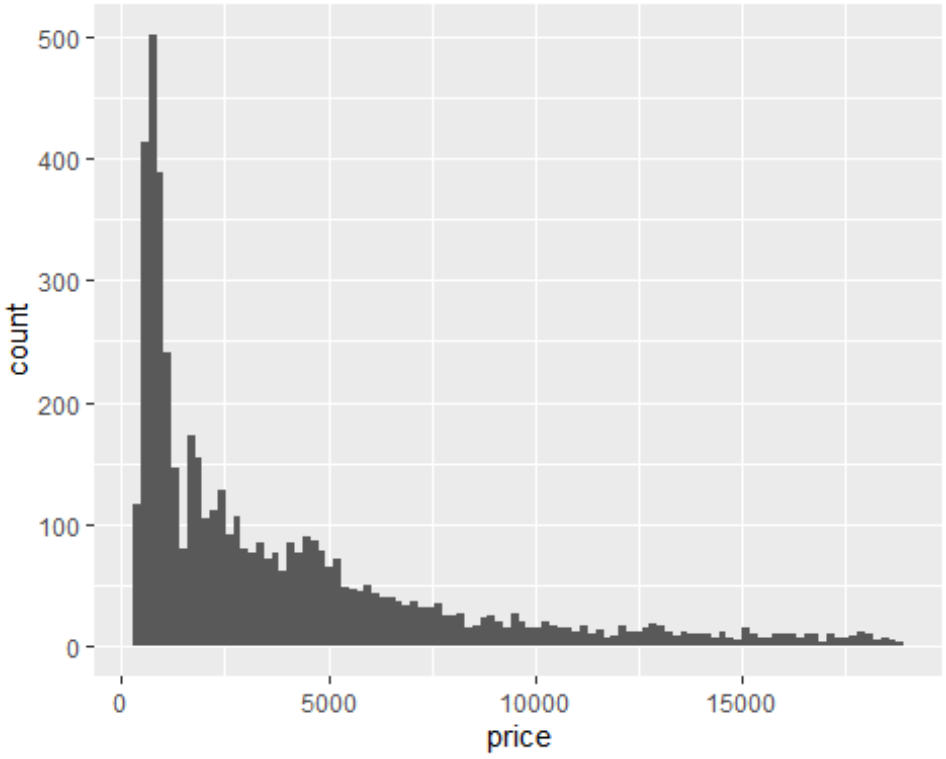
Dotplot

```
diamondsSample %>% ggplot(aes(x = carat)) + geom_dotplot(binwidth = 0.01,
stackdir = "center", color = "red", stackratio = 0.5, dotsize = 3) +
scale_x_continuous(breaks = seq(0,5,0.1)) + ggtitle("Diamond carat -
dotplot")+ theme(axis.text = element_text(size = 5), axis.title =
element_text(size = 10), plot.title = element_text(size = 25, face =
"bold"))
```

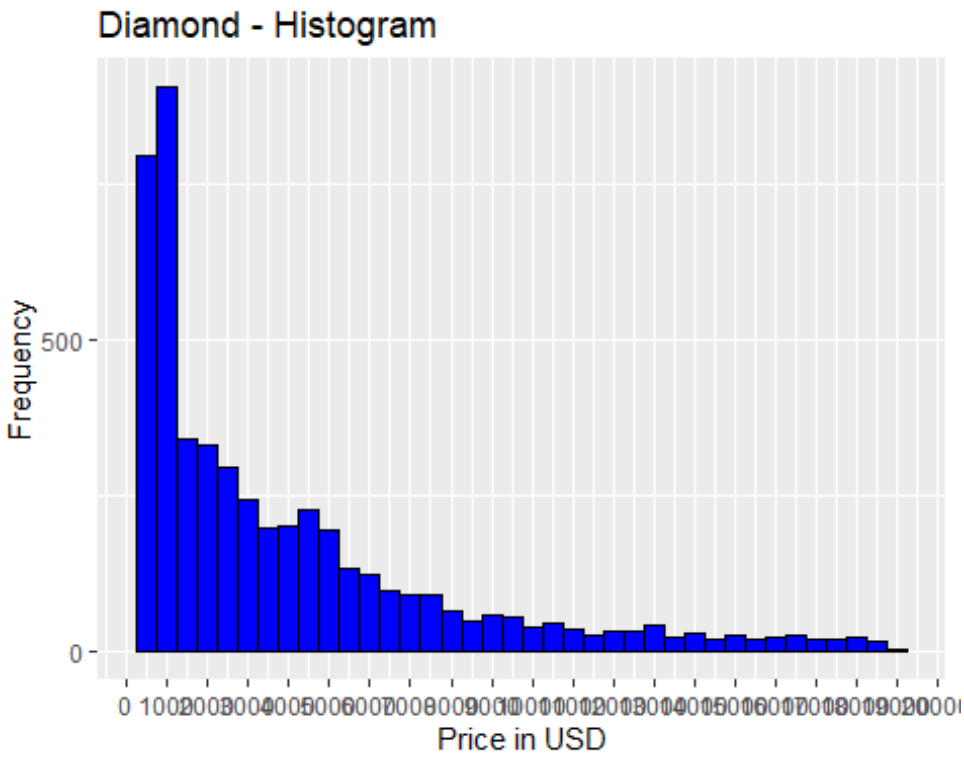


Histogram

```
diamondsSample %>% ggplot(aes(x=price)) + geom_histogram(bins = 100)
```

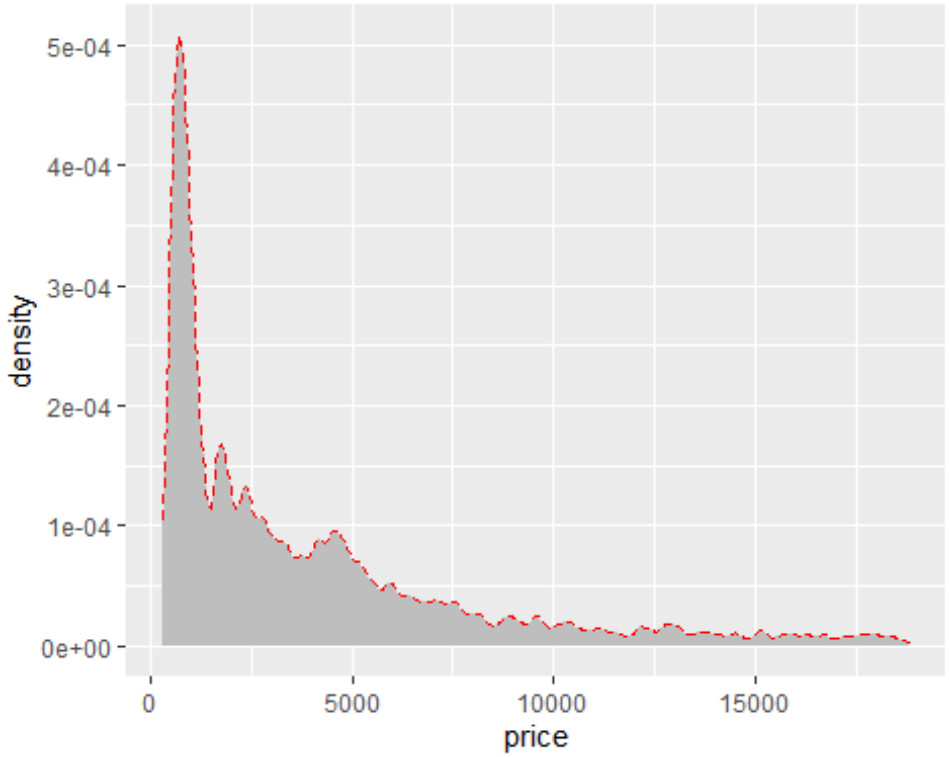


```
diamondsSample %>% ggplot(aes(x=price)) + geom_histogram(binwidth = 500,
color = "black", fill = "blue") + xlab("Price in USD") +
ylab("Frequency") + scale_x_continuous(breaks = seq(0,20000,1000)) +
scale_y_continuous(breaks = seq(0,3000,500)) + ggtitle("Diamond -
Histogram")
```



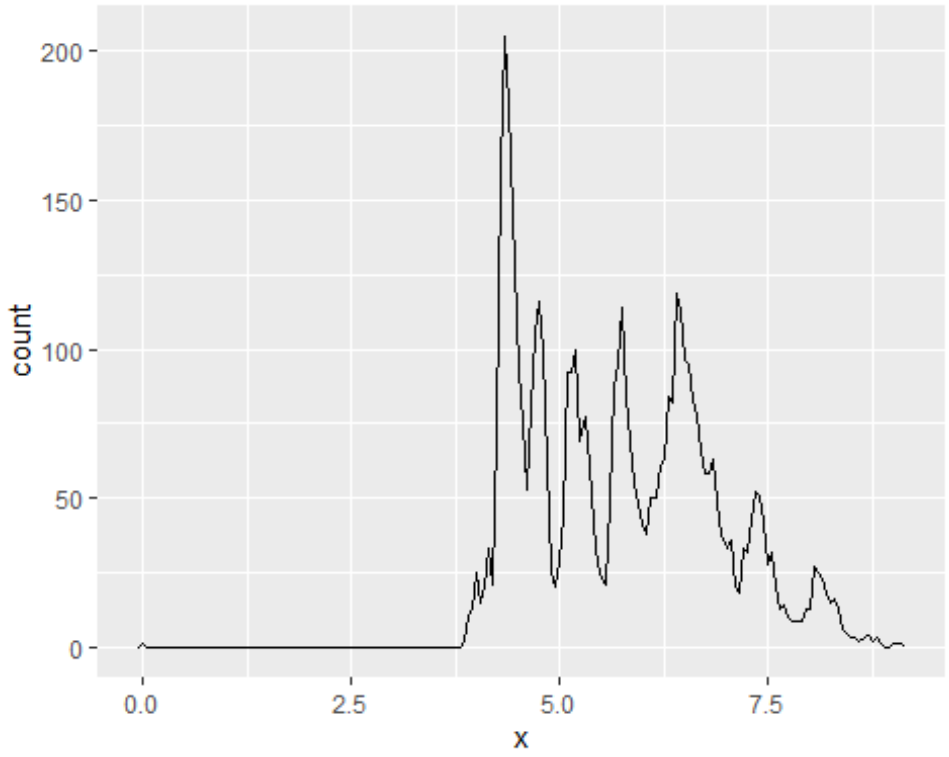
Density plot

```
diamondsSample %>% ggplot(aes(x = price)) + geom_density(adjust = 1/5,
linetype = "dashed", color = "red", fill = "gray", size = 0.5)
```

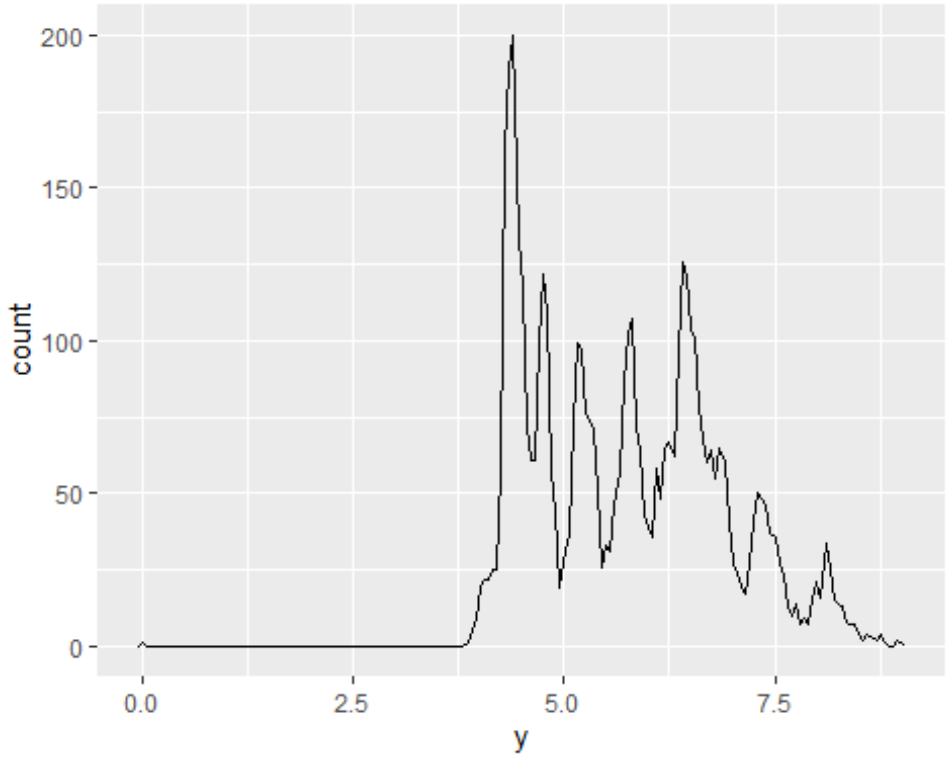


Frequency Plot

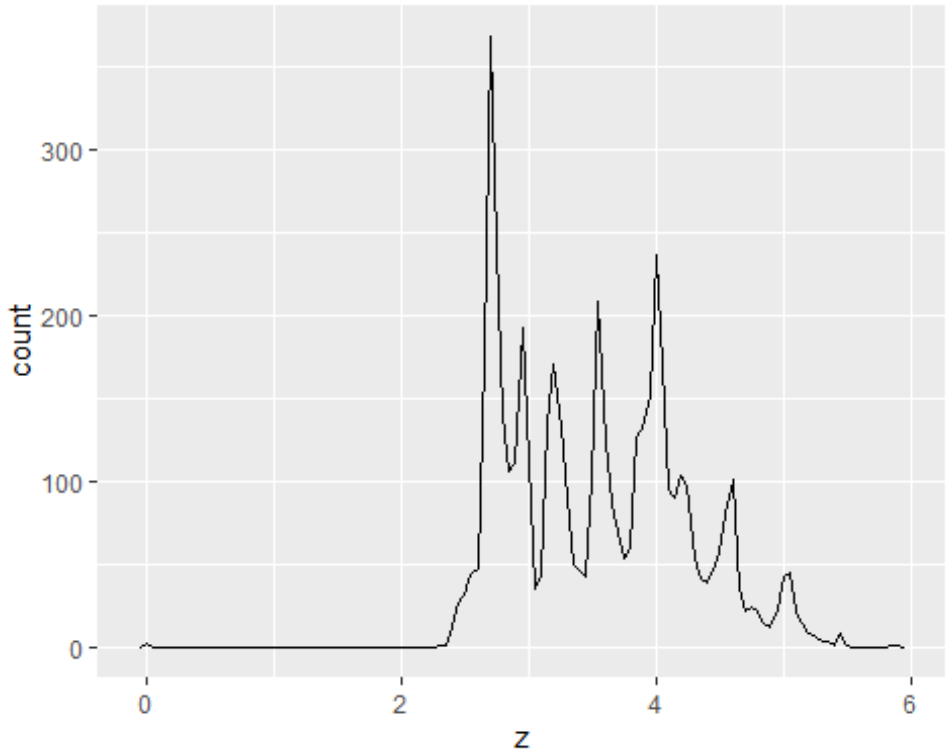
```
diamondsSample %>% select(x,y,z) %>% summary()

##        x                y                z         
##  Min.   :0.000   Min.   :0.000   Min.   :0.000  
##  1st Qu.:4.700   1st Qu.:4.710   1st Qu.:2.900  
##  Median :5.705   Median :5.720   Median :3.530  
##  Mean   :5.731   Mean   :5.734   Mean   :3.539  
##  3rd Qu.:6.543   3rd Qu.:6.540   3rd Qu.:4.030  
##  Max.   :9.110   Max.   :9.020   Max.   :5.910  

diamondsSample %>% ggplot(aes(x=x)) + geom_freqpoly(binwidth = 0.05)
```



```
diamondsSample %>% ggplot(aes(x=y)) + geom_freqpoly(binwidth = 0.05)
```

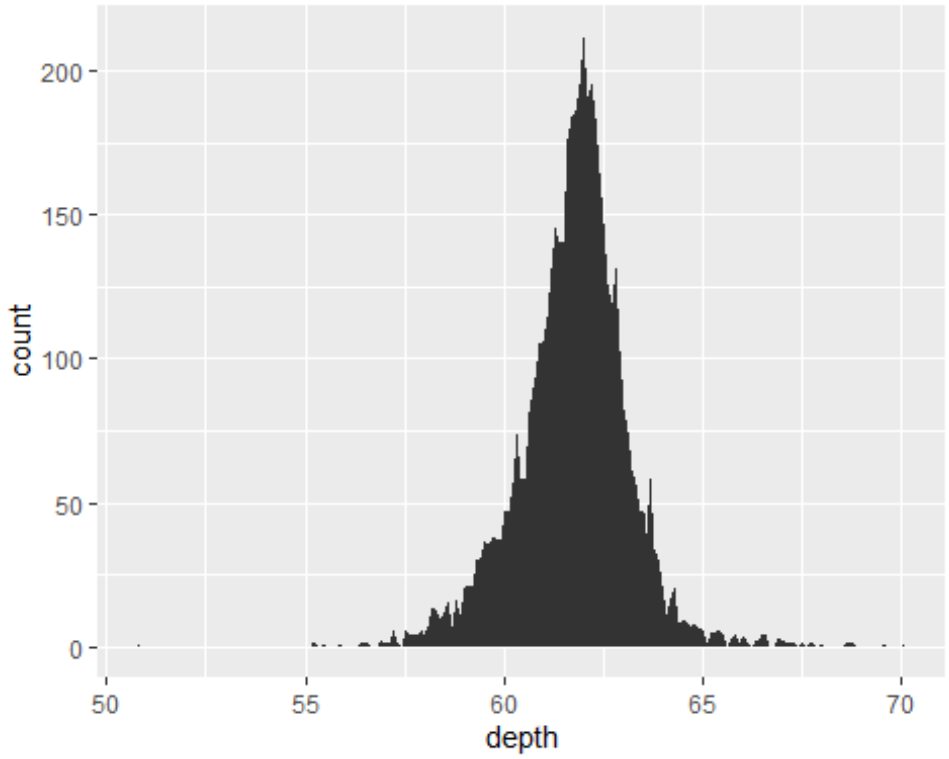



```
diamondsSample %>% ggplot(aes(x=z)) + geom_freqpoly(binwidth = 0.05)
```

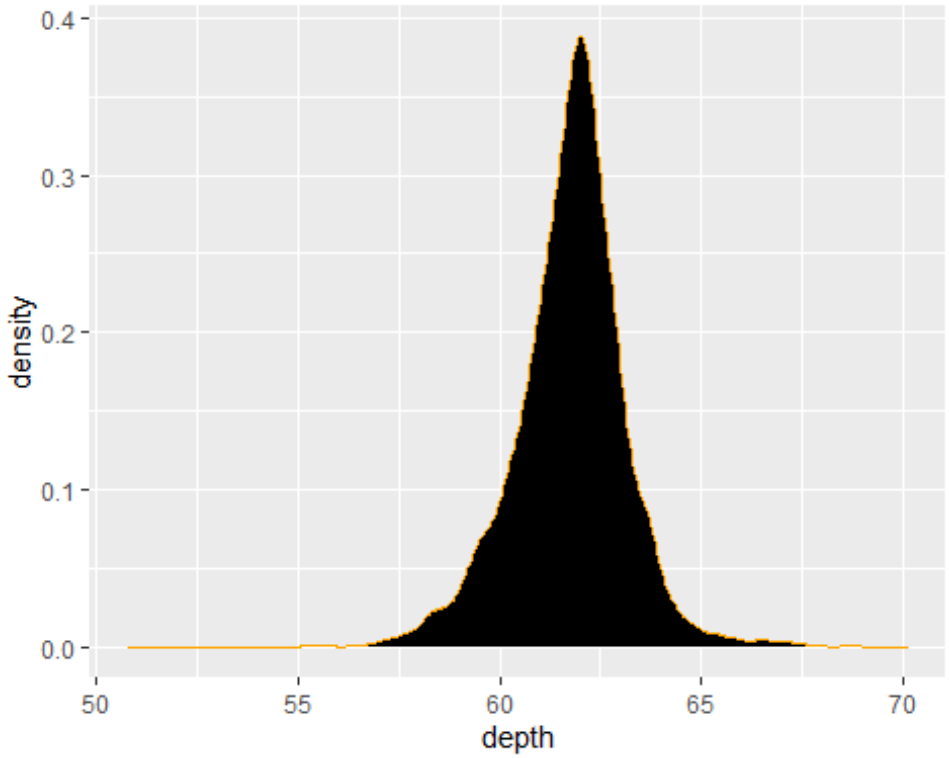


Area plot

```
diamondsSample %>% select(depth) %>% summary()

##      depth
##  Min.   :50.80
##  1st Qu.:61.00
##  Median :61.85
##  Mean   :61.75
##  3rd Qu.:62.50
##  Max.   :70.10

diamondsSample %>% ggplot(aes(x=depth)) + geom_area(stat = "bin",
binwidth = 0.1)
```

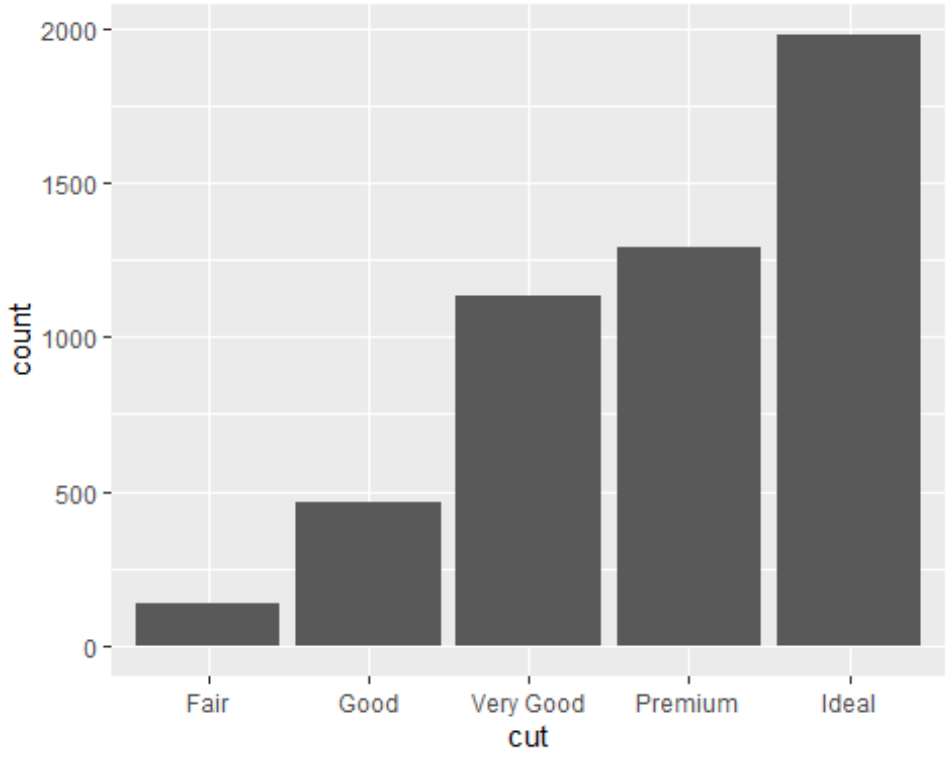


```
diamondsSample %>% ggplot(aes(x=depth)) + geom_area(stat = "density",
kernel = "gaussian", color ="orange", fill = "black")
```

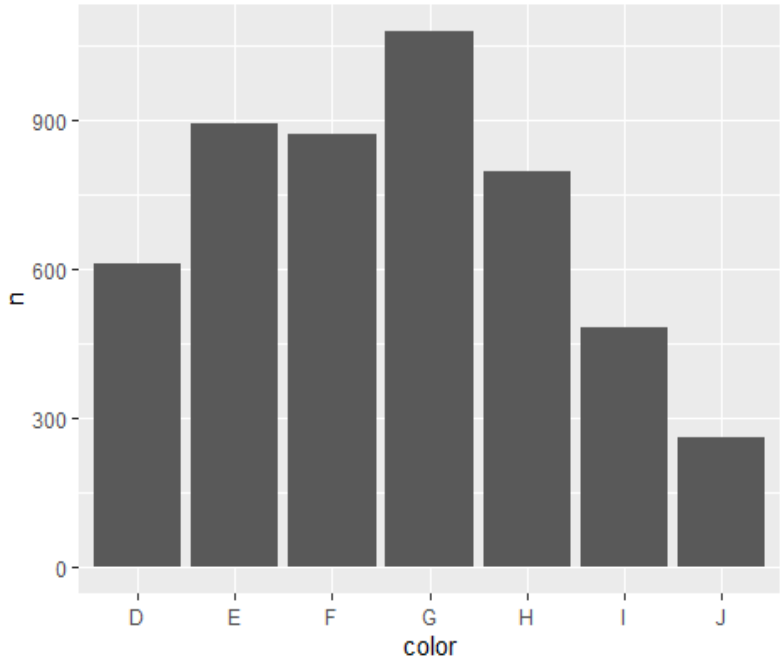


Bar plot

```
diamondsSample %>% ggplot(aes(x = cut)) + geom_bar(stat = "count")
```

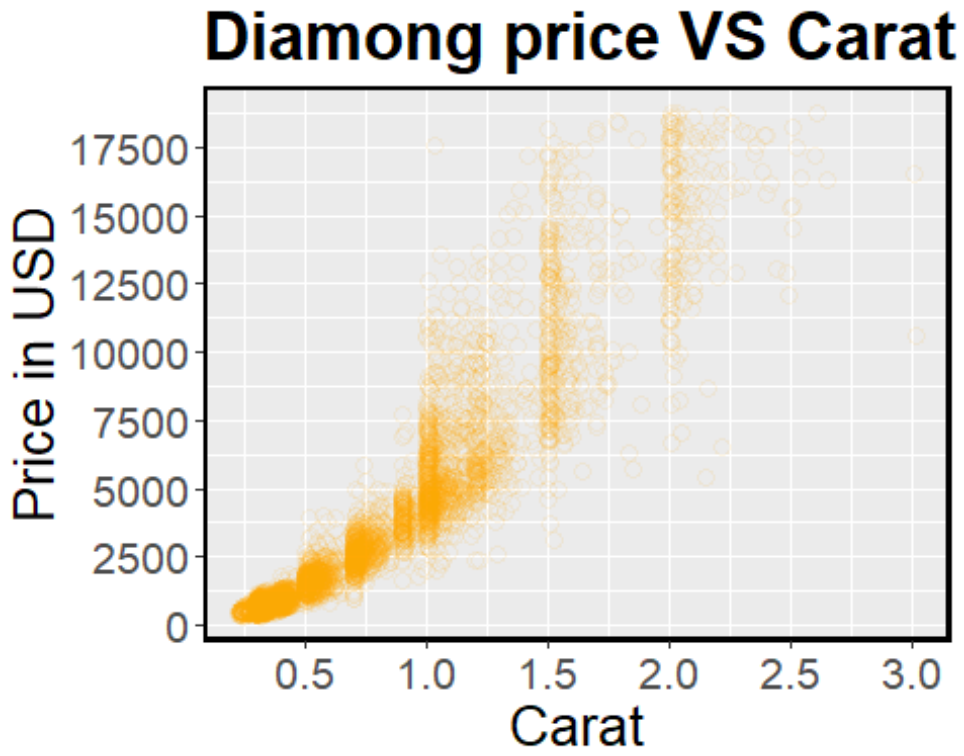



```
diamonds_color_count = diamondsSample %>% group_by(color) %>% count()
diamonds_color_count %>% ggplot(aes(x = color, y = n)) + geom_bar(stat =
"identity")
```



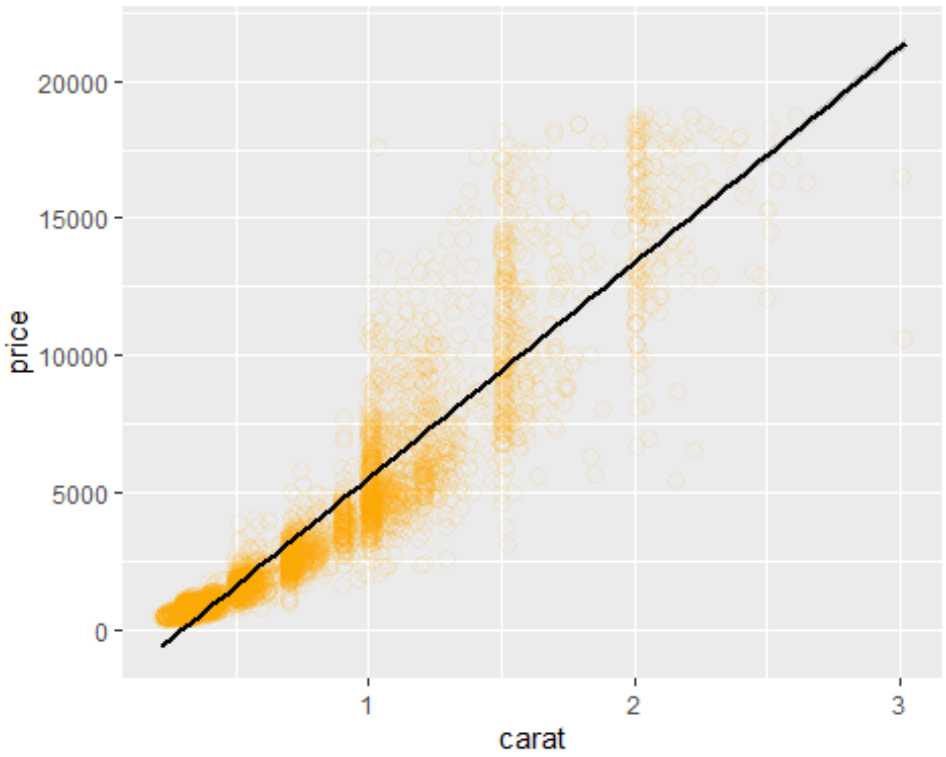
Scatter Plot

```
diamondsSample %>% ggplot(aes(x = carat, y = price)) +
  geom_point(alpha = 1/10, size = 3, color = "orange", shape = 1,
position = "jitter") +
  scale_x_continuous(breaks = seq(0,5,0.5)) + scale_y_continuous(breaks =
seq(0,20000,2500)) +
  xlab("Carat") +
  ylab("Price in USD") +
  ggtitle("Diamong price VS Carat - Scatterplot") +
  theme(axis.title = element_text(size = 20), axis.text =
element_text(size=16), plot.title = element_text(size = 25, face =
"bold"), panel.border = element_rect(color = "black", fill = NA, size =
1.5))
```



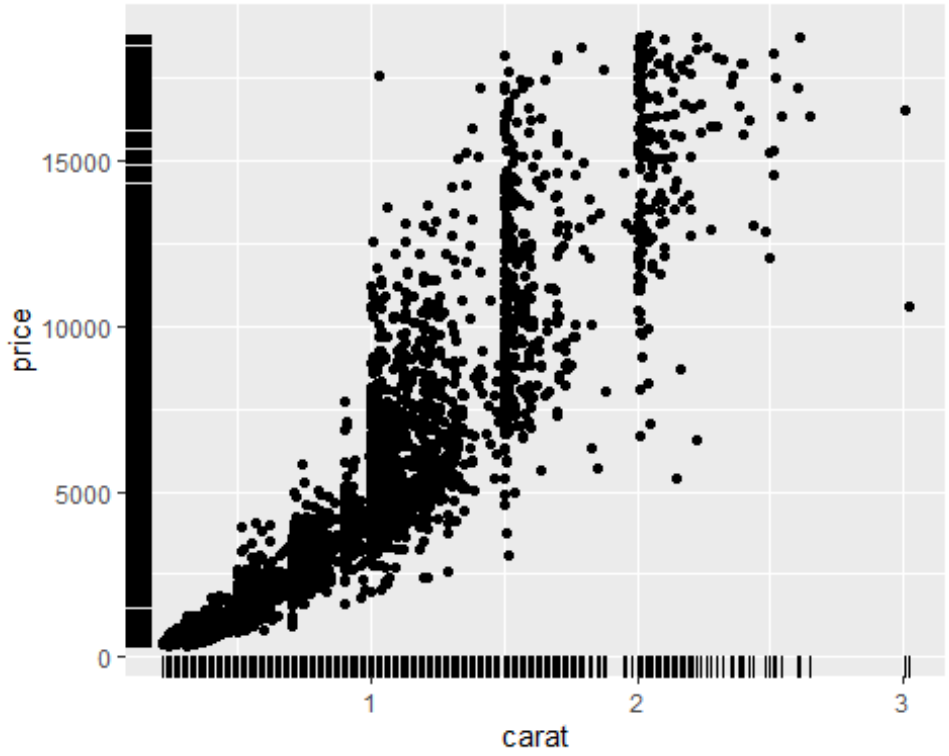
Smoothing Line

```
diamondsSample %>% ggplot(aes(x = carat, y = price)) +
  geom_point(alpha = 1/10, size = 3, color = "orange", shape = 1,
position = "jitter") + geom_smooth(method = "lm", formula = "y ~ x", se =
T, color = "black")
```



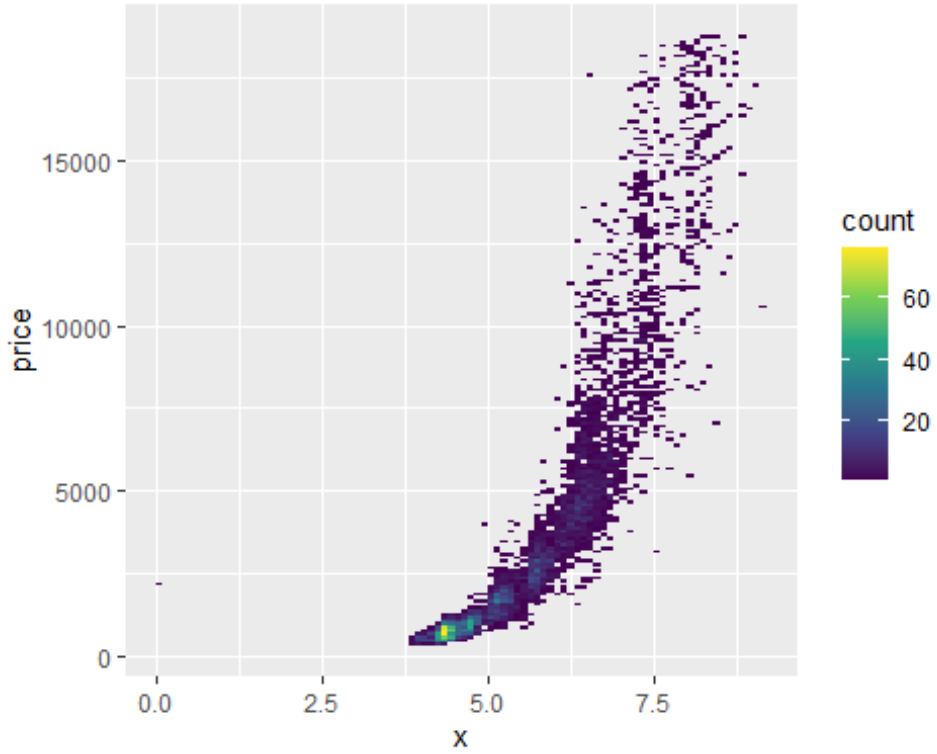
Rug plot

```
diamondsSample %>% ggplot(aes(x = carat, y = price)) + geom_point() +
geom_rug()
```



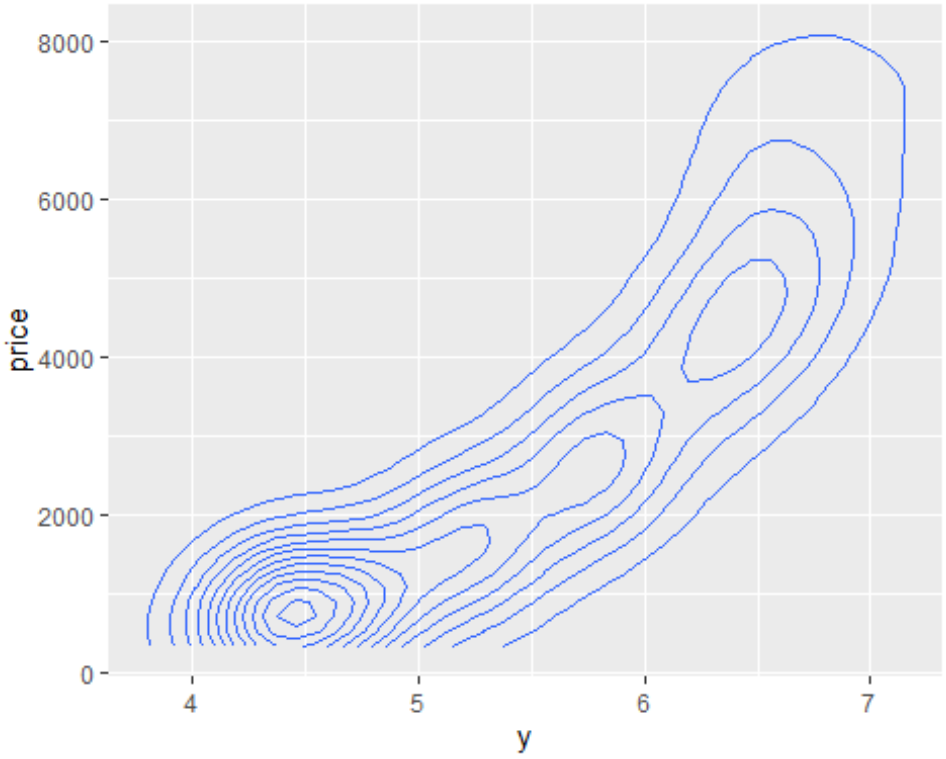
Heatmap

```
diamondsSample %>% ggplot(aes(x = x, y = price)) + geom_bin2d(binwidth =
c(0.1, 100)) +
  scale_fill_viridis_c()
```



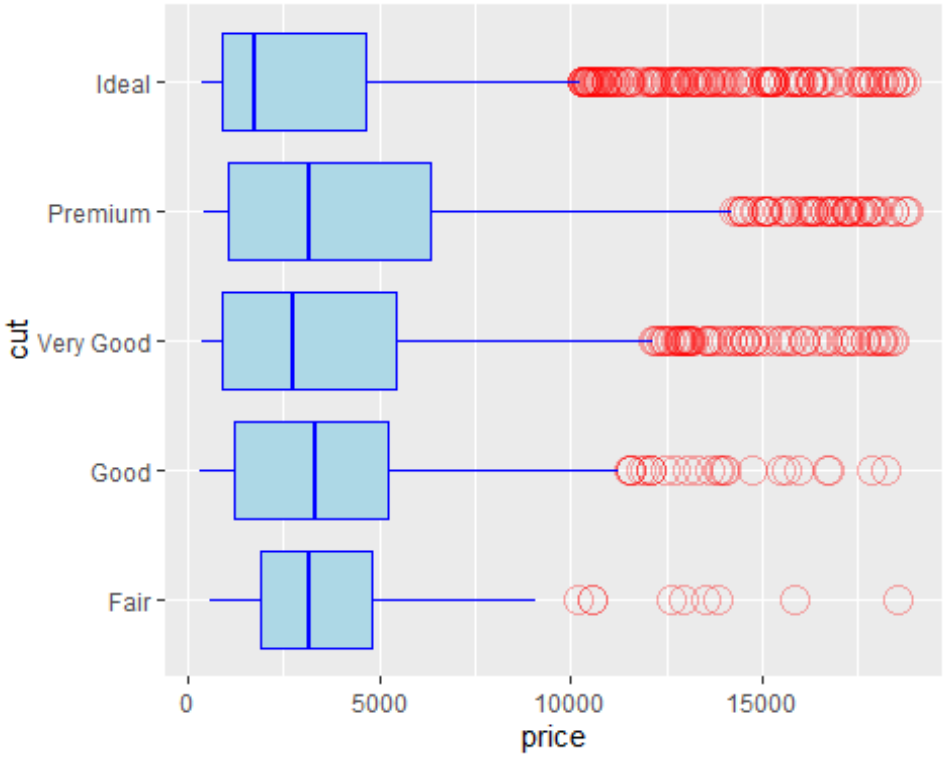
Density

```
diamondsSample  %>% ggplot(aes(x = y, y = price)) + geom_density2d()
```

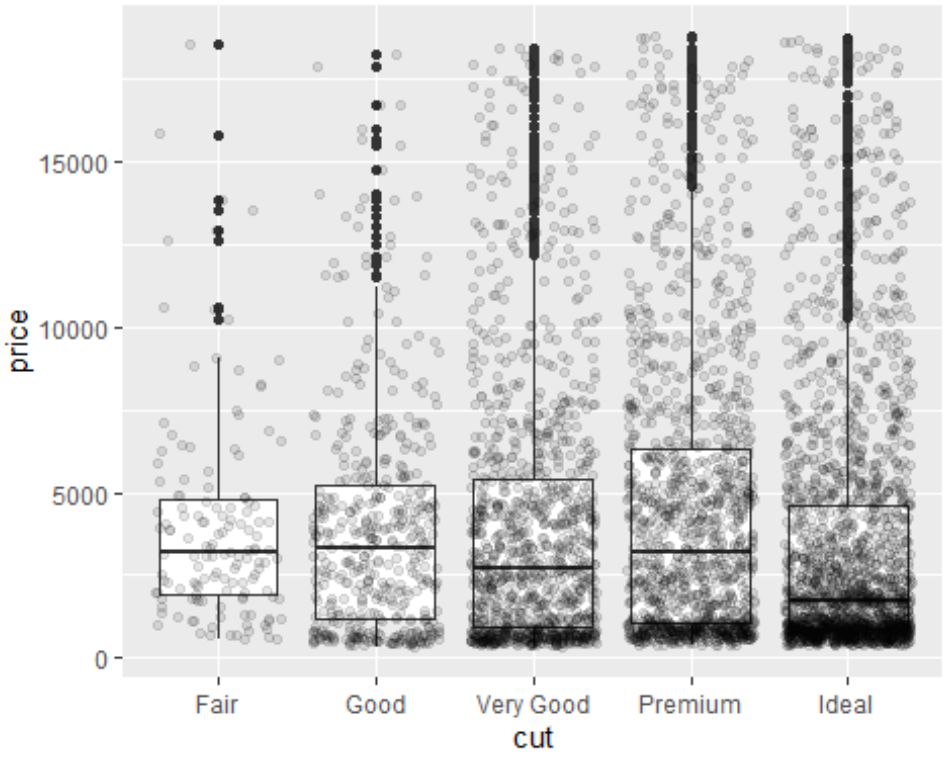


Boxplot

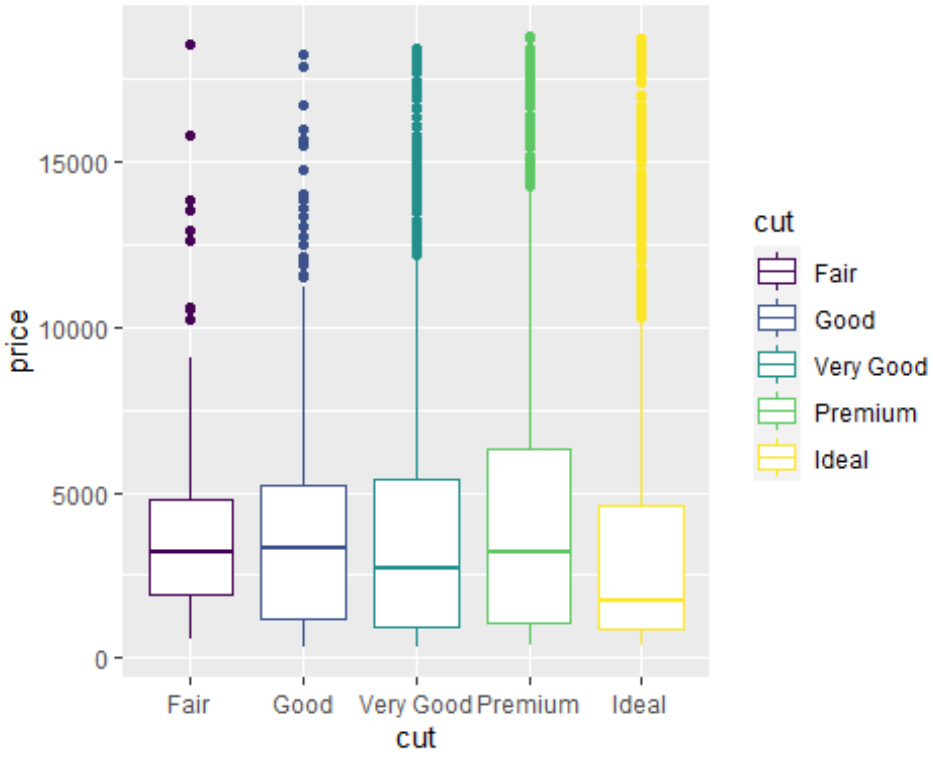
```
diamondsSample %>% ggplot(aes(x = cut, y = price)) +
geom_boxplot(outlier.colour = "red", outlier.alpha = 1/3, outlier.shape =
1, outlier.size = 5, color = "blue", fill = "lightblue")  + coord_flip()
```



```r
diamondsSample %>% ggplot(aes(x = cut, y = price)) + geom_boxplot() +
geom_jitter(alpha = 1/10)
```

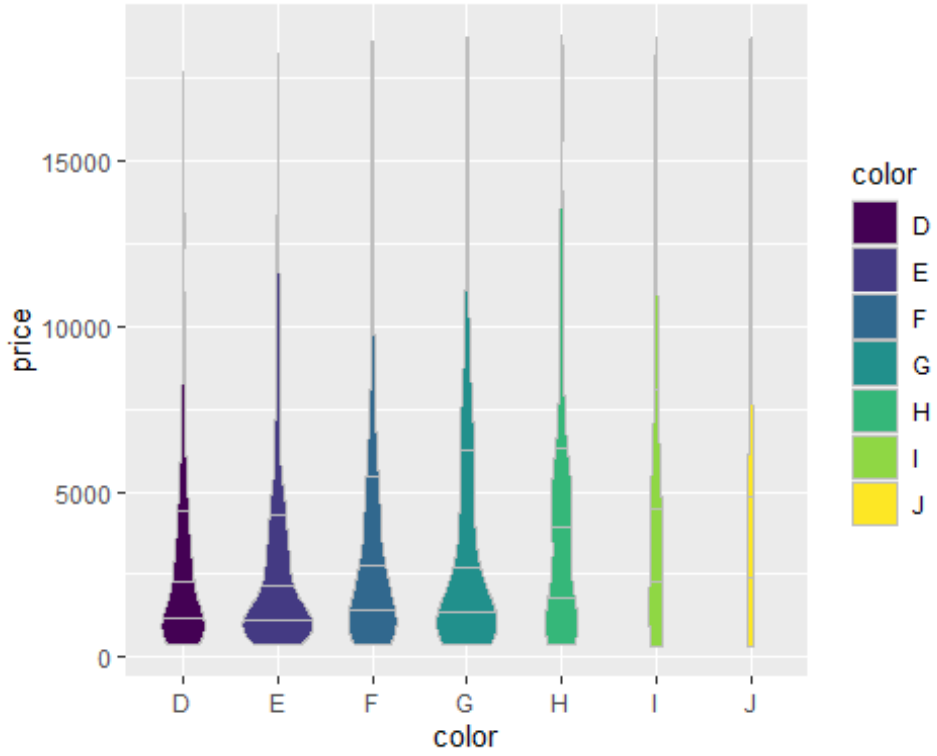


```
diamondsSample %>% ggplot(aes(x = cut, y = price, color = cut)) +
geom_boxplot()
```



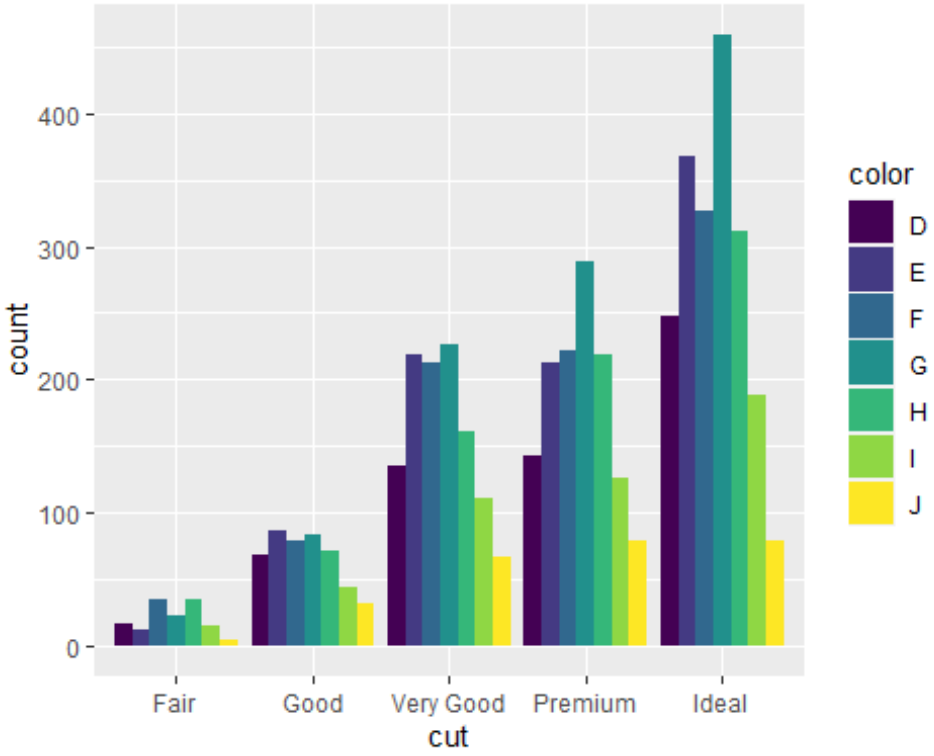
Violin plot

```
diamondsSample %>% ggplot(aes(y = price, x = color, fill = color))+
geom_violin(scale = "count", color = "gray", draw_quantiles =
c(0.25,0.5,0.75), kernel = "gaussian")
```

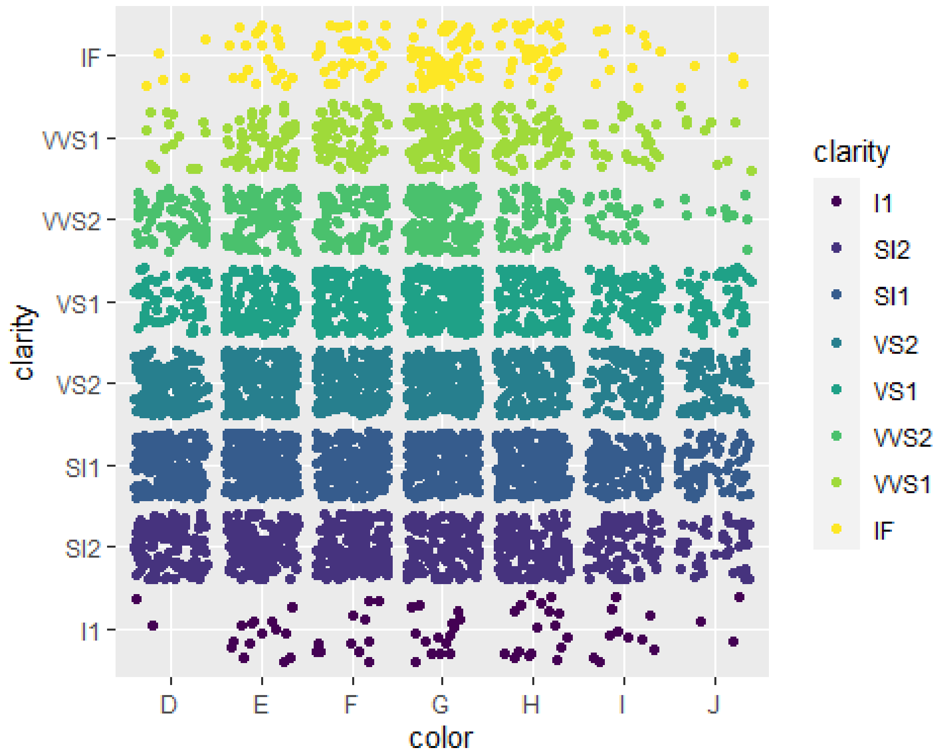


Counting plot

```
diamondsSample %>% ggplot(aes(x = cut, fill = color)) + geom_bar(position
= "dodge") #fill for 100%
```



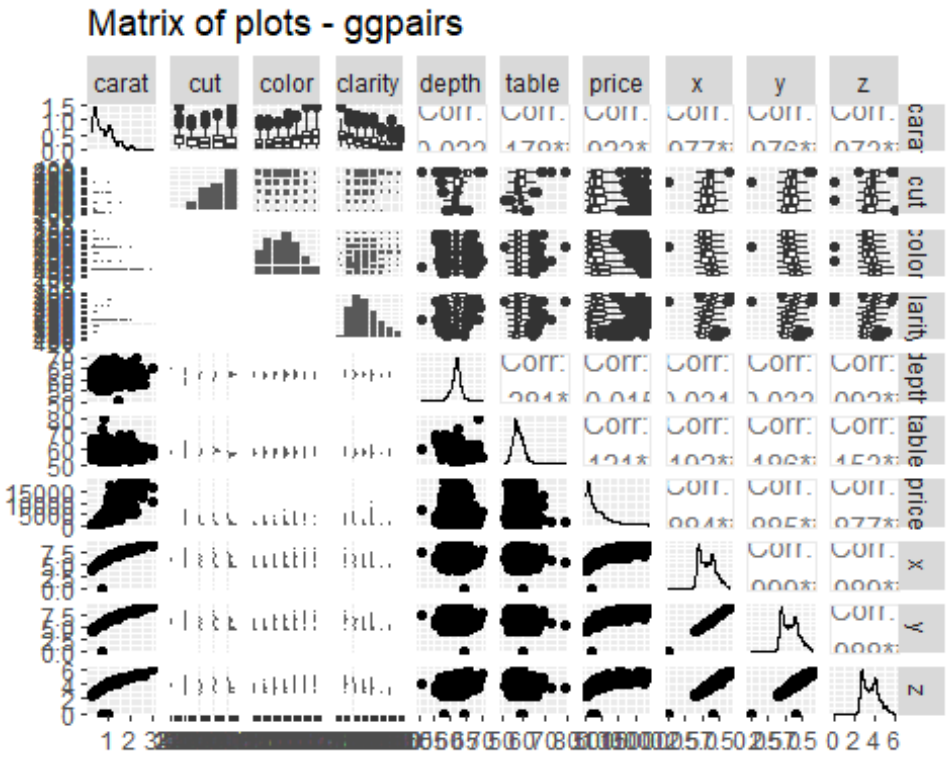
```r
diamondsSample %>% ggplot(aes(x = color, y = clarity, color = clarity)) +
geom_jitter()
```



Data Frame Matrix

```r
diamondsSample %>% GGally::ggpairs(title = "Matrix of plots - ggpairs")

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

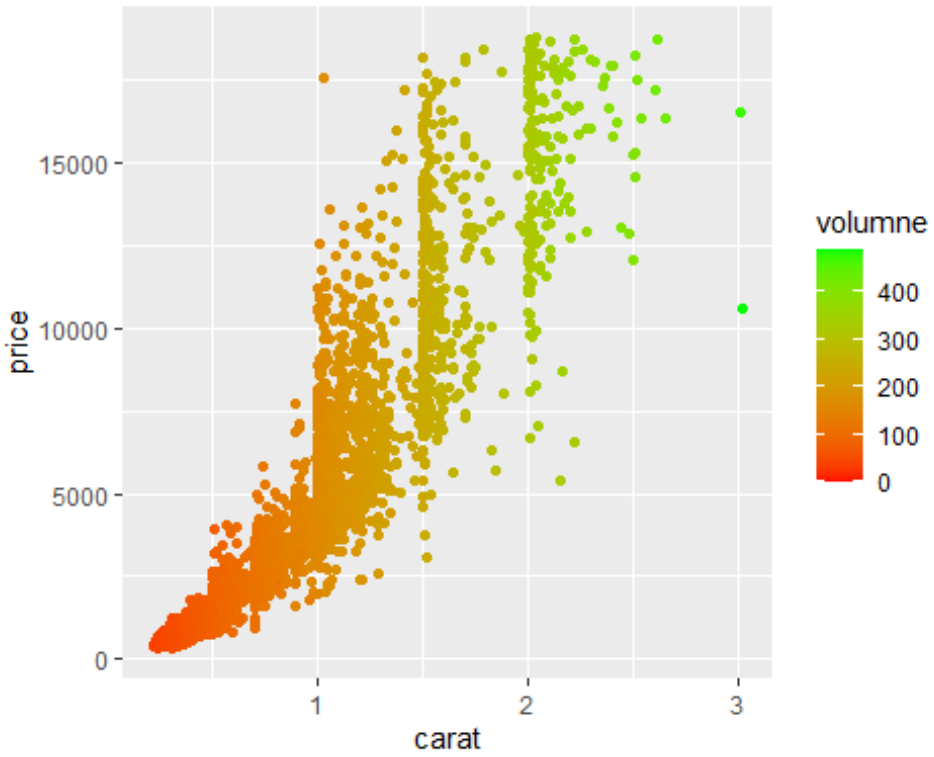


Creating a new variable

```
diamondsSample = diamondsSample %>% mutate(volumne = x*y*z)
```

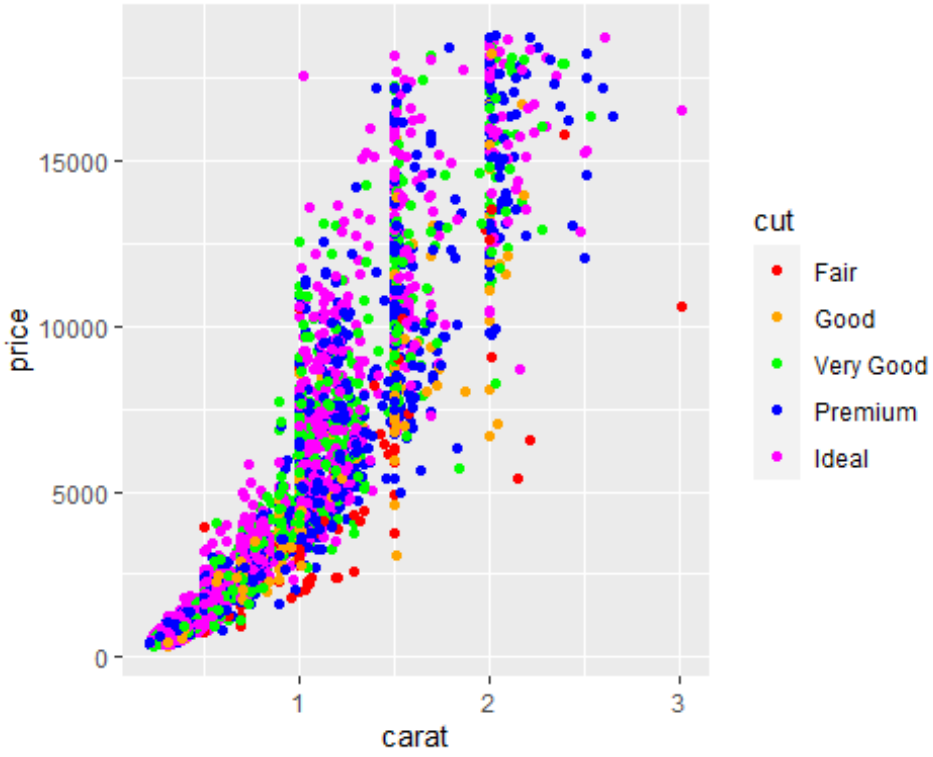
Scatter plot (continues)

```
diamondsSample %>%
  ggplot(aes(x = carat, y = price, color = volumne)) + geom_point() +
  scale_color_gradient(low = "red", high = "green")
#scale_colour_gradient(colours = c("red,"green", "blue"))
```

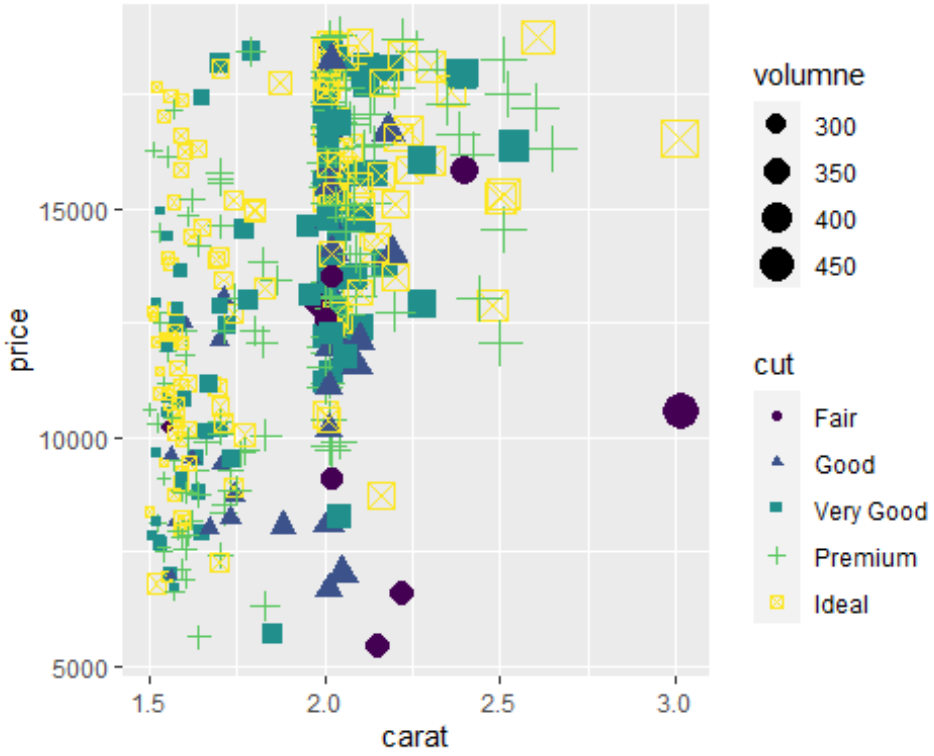


Scatter plot (discrete)

```
diamondsSample %>%
  ggplot(aes(x=carat, y = price, color = cut)) +
  geom_point() +
  scale_colour_manual(values = c("red", "orange", "green", "blue",
"magenta")) #scale-colour_brewer(palette = "reds")
```

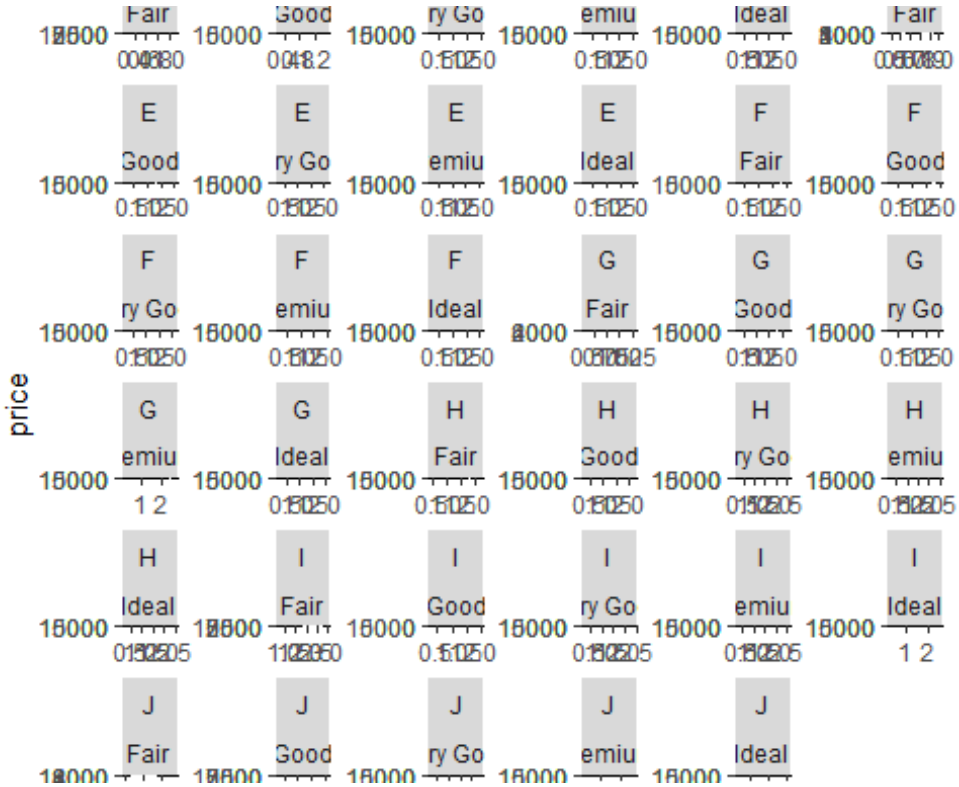


```
diamondsSample %>%
  filter(volumne >= 250 & volumne <=1000) %>%
  ggplot(aes(x = carat, y = price, size = volumne, shape = cut, color =
cut)) +
  geom_point()

## Warning: Using shapes for an ordinal variable is not advised
```



Facet Wrap

```
diamondsSample %>%
  ggplot(aes(x = carat, y = price)) +
  geom_point() +
  facet_wrap(vars(color,cut), scales = "free") #nrow = 10)
```



Facet Grid

```
diamondsSample %>%
  ggplot(aes(x = carat, y = price))+geom_point() +
  facet_grid(rows = vars(cut), cols = vars(clarity))
```