Data Visualization with ggplot2

Pratik Gandhi

April 2, 2016

Exploring the functionality/applications of ggplot2 using diamonds-dataset

Prices of ~50,000 round cut diamonds

- Price - in $ (USD)
- Carat - weight of the diamond
- Cut - Quality of cutting
- Color - Color of Diamond
- Clairty - How clear the diamond is
- x - length
- y - width
- z - depth
- depth - total depth % --> z/mean(x,y)
- table - width of top of diamond relative to widest point

According to ggplot2:

Plot = data + Aesthetics + Geometry

Data --> data.frame

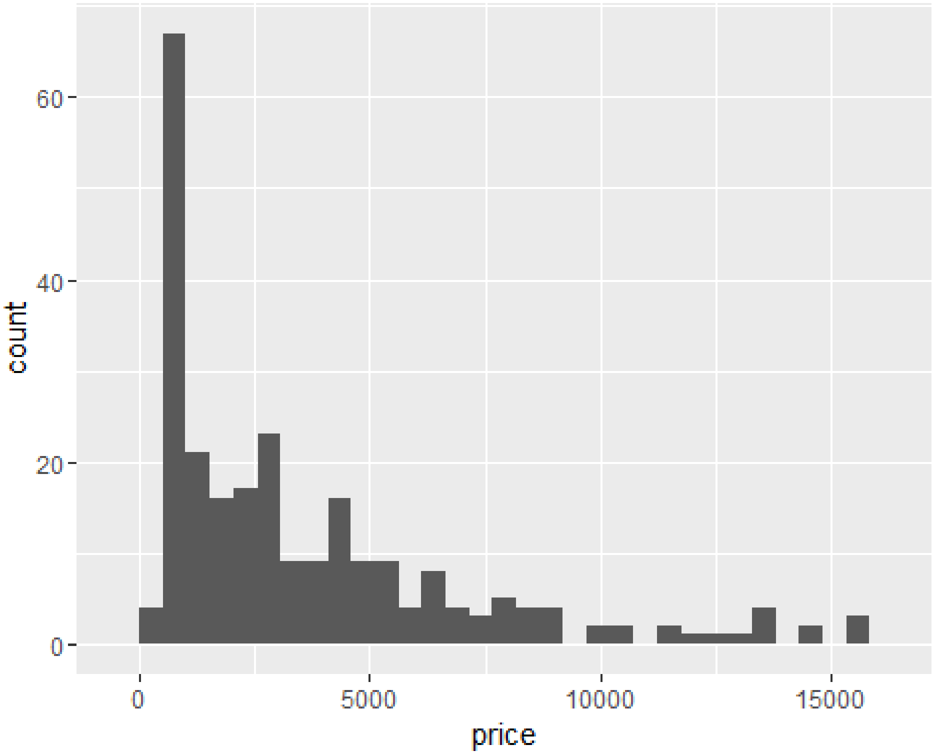
Aes. --> "x" and "y" variables. Also used to control color,size and shape of points, height of bars

Geo. --> Type of plot/graphics one wants - Bar, Box, Line, Density ....

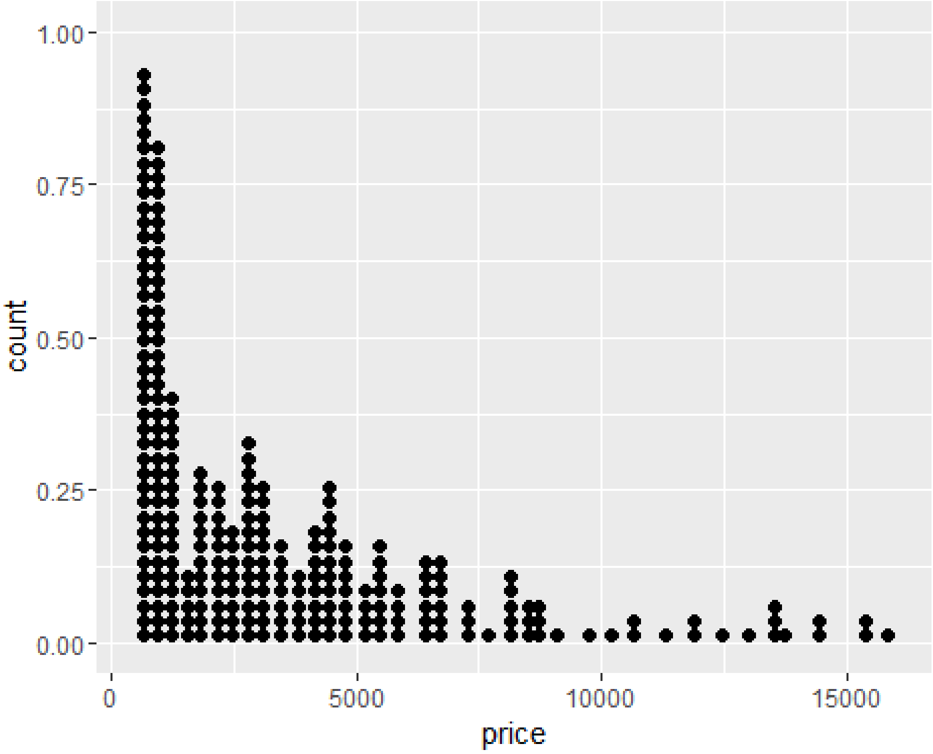
```r
library(ggplot2)
library(gridExtra)

data("diamonds") # Loading the dataset
head(diamonds,n=5) # Getting the first few rows of dataset

##   carat     cut color clarity depth table price    x    y    z
## 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 4.20 4.23 2.63
## 5  0.31    Good     J     SI2  63.3    58   335 4.34 4.35 2.75

str(diamonds) # Knowing the structure of the dataset

## Classes 'tbl_df', 'tbl' and 'data.frame':    53940 obs. of  10 variables:
##  $ carat  : num  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  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...

# Primarily exploring the levels of all variables
levels(diamonds$cut) # Levels of cut quality

## [1] "Fair"      "Good"      "Very Good" "Premium"   "Ideal"

levels(diamonds$color) # Levels of color

## [1] "D" "E" "F" "G" "H" "I" "J"

levels(diamonds$clarity) # Levels of clarity

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

# Taking only subset of the data:
#diam_ss <- diamonds[1:5000,]
diam_ss <- diamonds[sample(1:nrow(diamonds),250,replace = FALSE),]
```

Considering 1 variable:

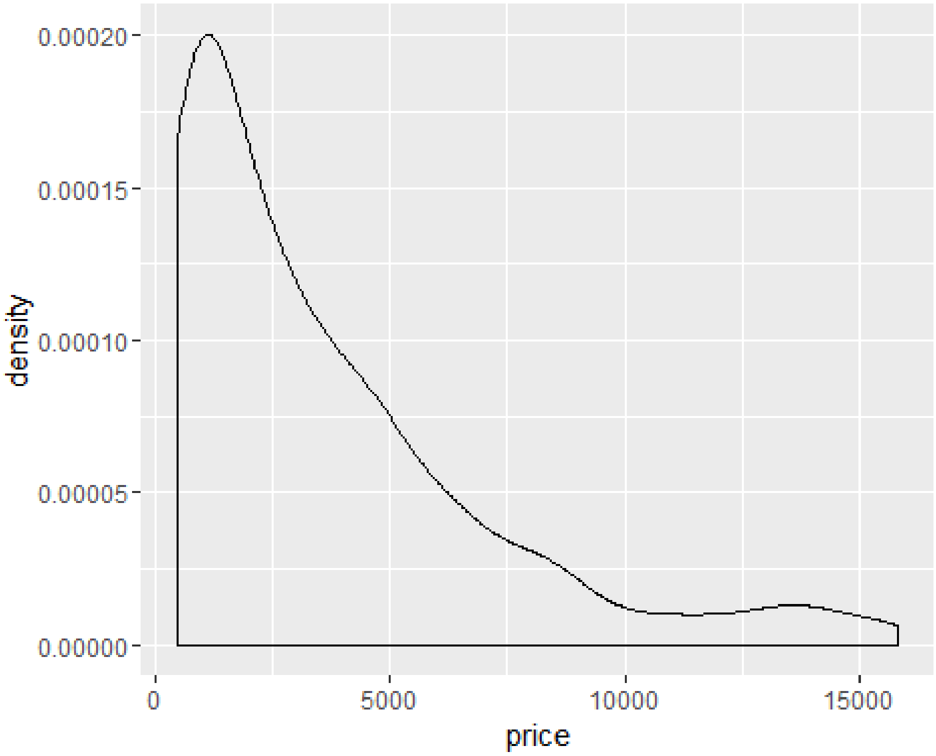
```
# CONTINUOUS
C1 <- ggplot(diam_ss,aes(price))

# Histogram:
C1 + geom_histogram()

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



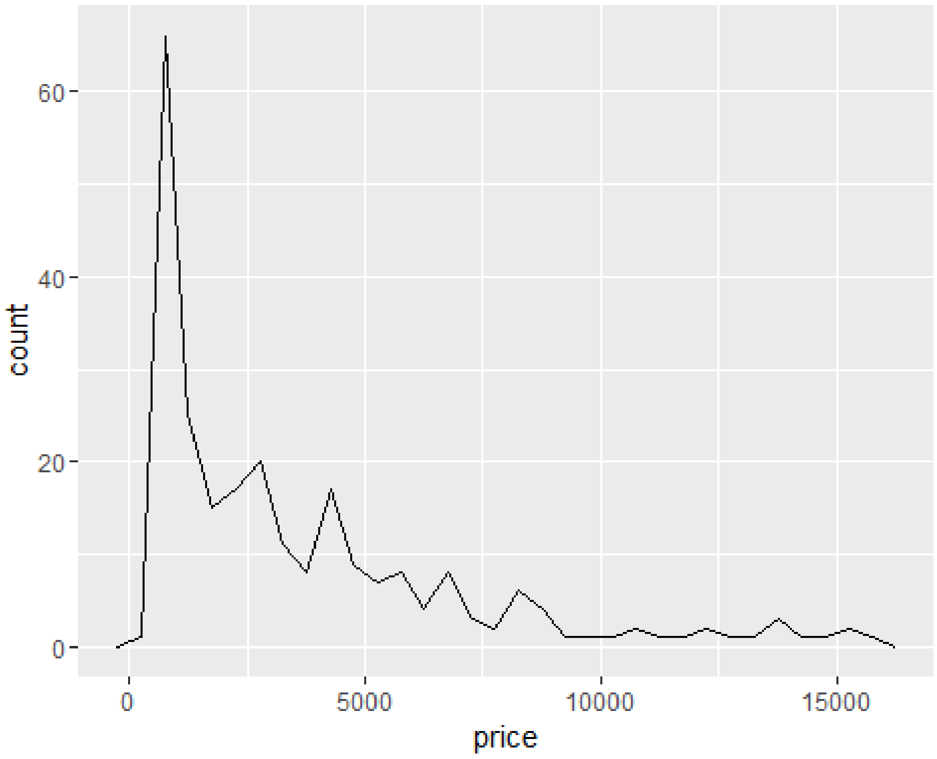
```
# Scatterplot:
C1 + geom_dotplot(binwidth = 300)
```



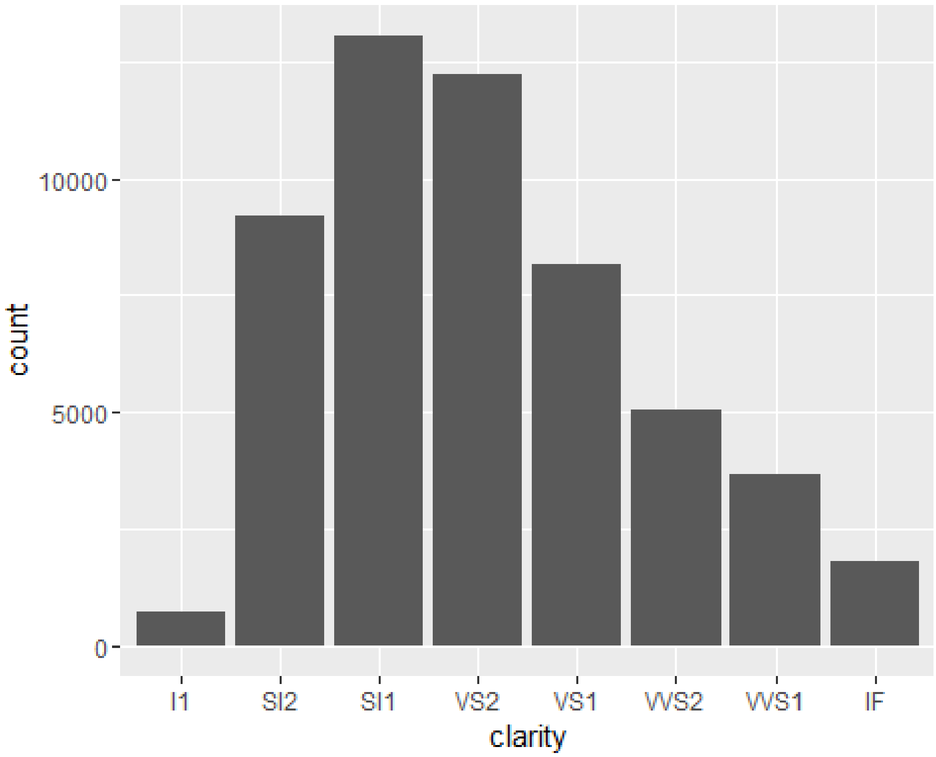
```
# Density:
C1 + geom_density()
```



```
# Frequency Polygon:
C1 + geom_freqpoly(binwidth = 500) # Getting warning w/o binwidth
```

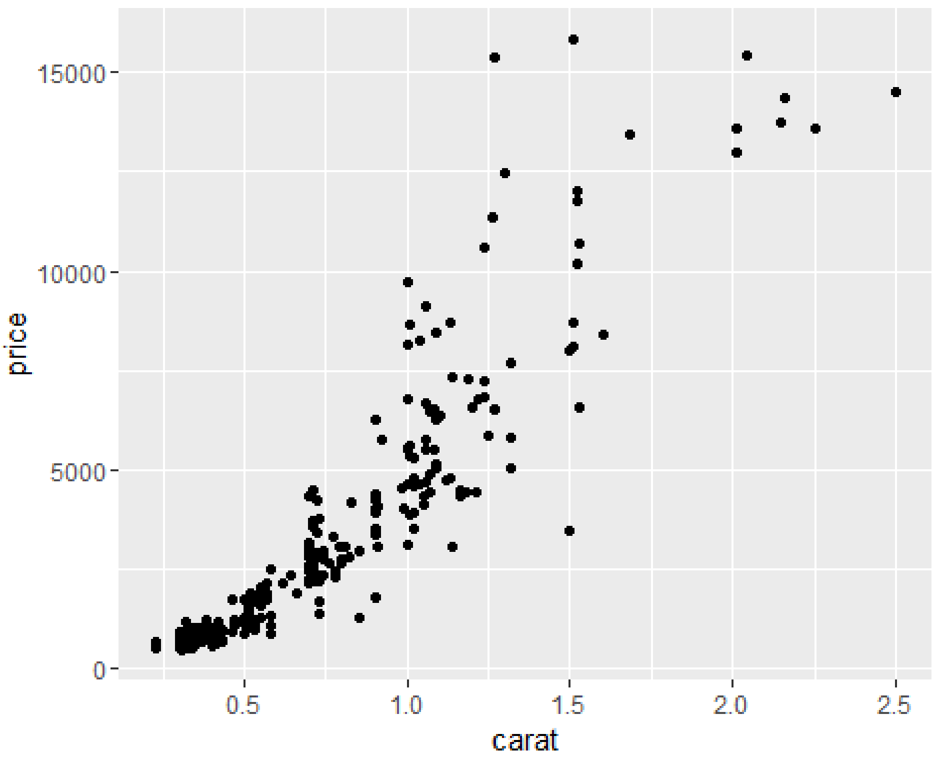


```
## Area plots?????
#C1 + geom_area(aes(y= ..density..), stat = "bin")
#C1 + geom_area(binwidth= 500, stat = "bin", color= "black", fill="#00AFBB")

# DISCRETE
D1 <- ggplot(diamonds,aes(clarity))

# Barplot:
D1 + geom_bar()
```

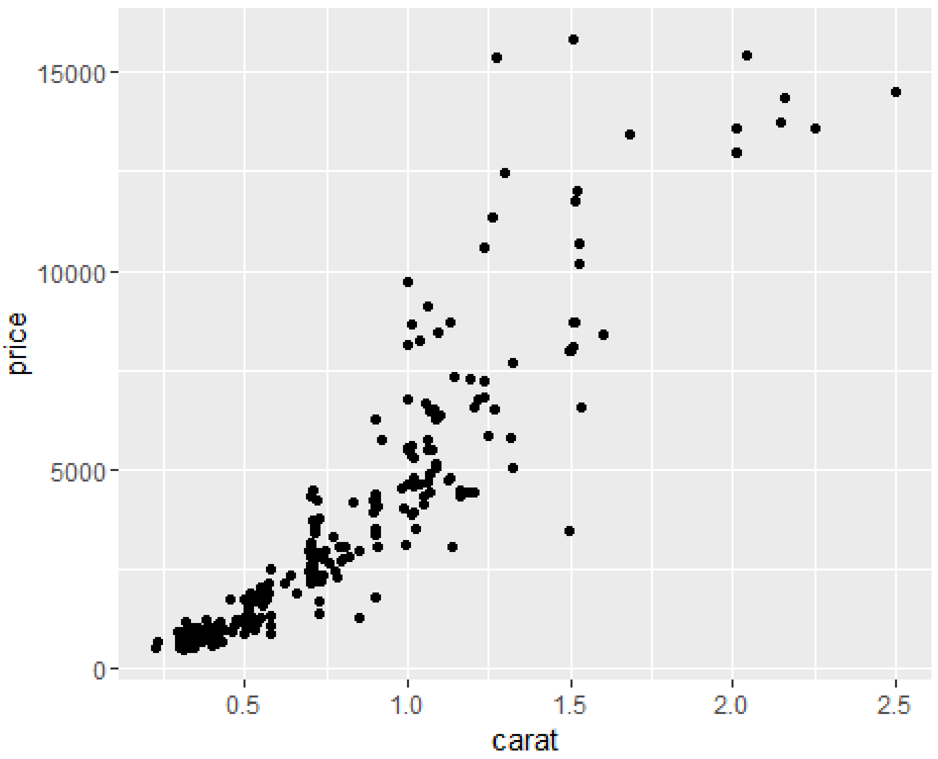


Considering 2 variables:

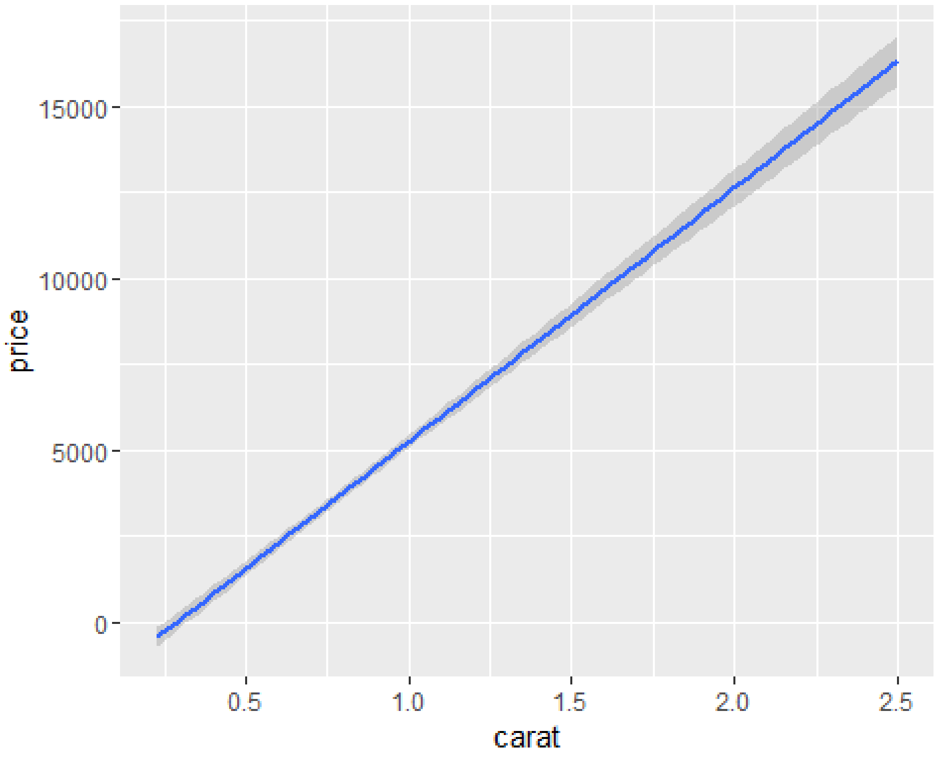
```
# CONTINUOUS X, CONTINUOUS Y
C2 <- ggplot(diam_ss,aes(carat,price))

# Plotting the scatter plot
C2 + geom_point()
```



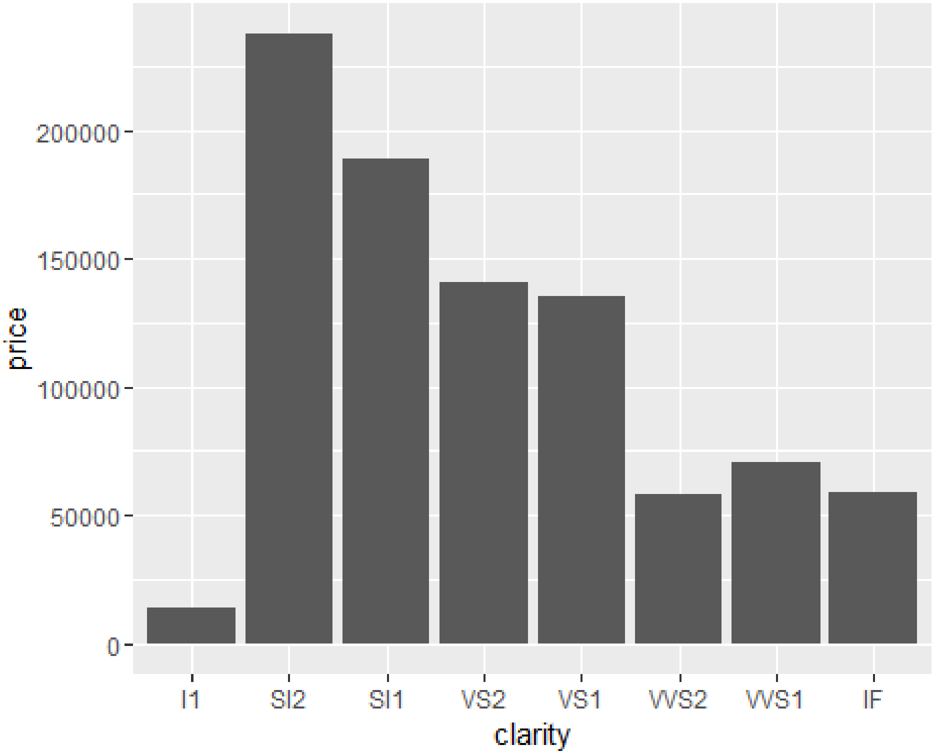
```
C2 + geom_jitter()
```



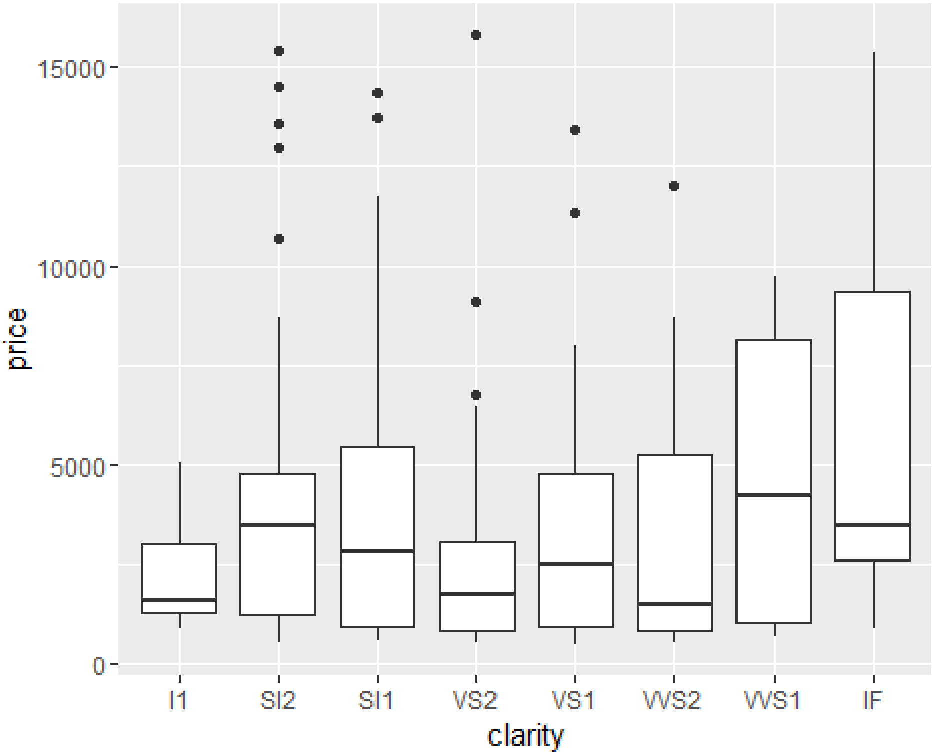
```
C2 + geom_smooth(method = lm)
```



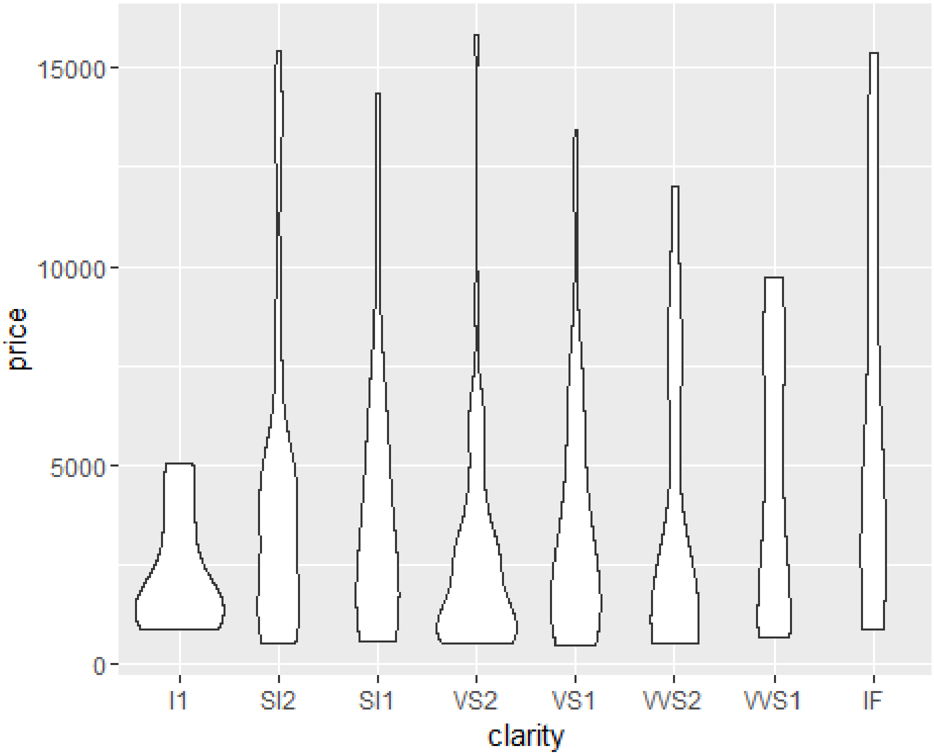
```
# DISCRETE X, CONTINUOUS Y
C1D1 <- ggplot(diam_ss,aes(clarity,price))

# Bar Plot
C1D1 + geom_bar(stat = "identity")
```



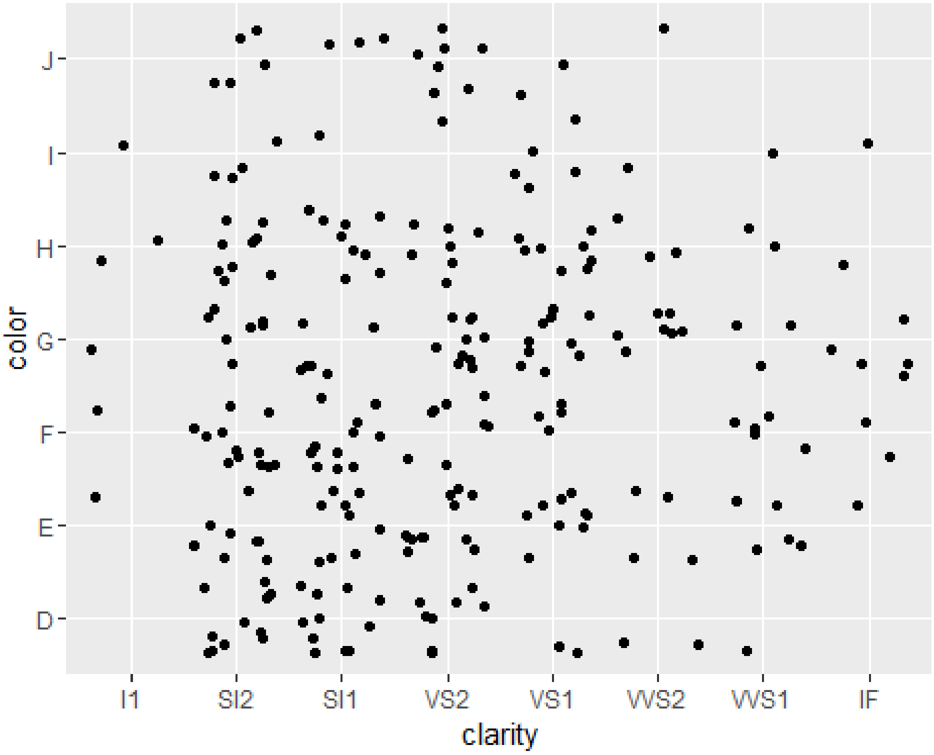
```
# Box Plot
C1D1 + geom_boxplot()
```



```
# Violin Plot
C1D1 + geom_violin()
```

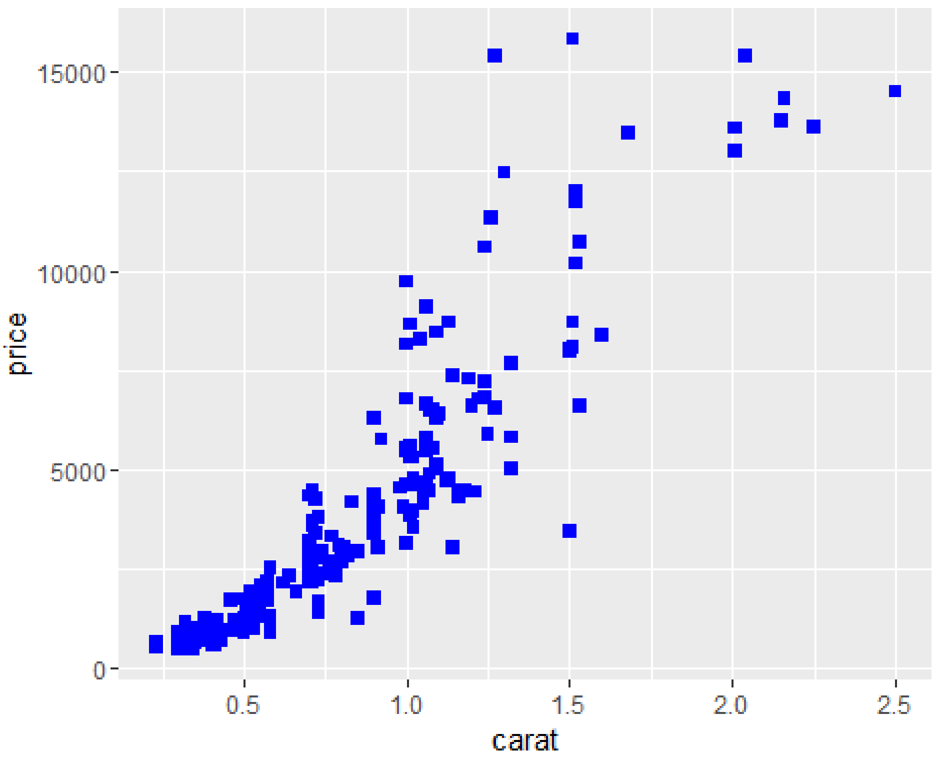


```
# DISCRETE X, DISCRETE Y
D2 <- ggplot(diam_ss,aes(clarity,color))

# Scatterplot
D2 + geom_jitter()
```

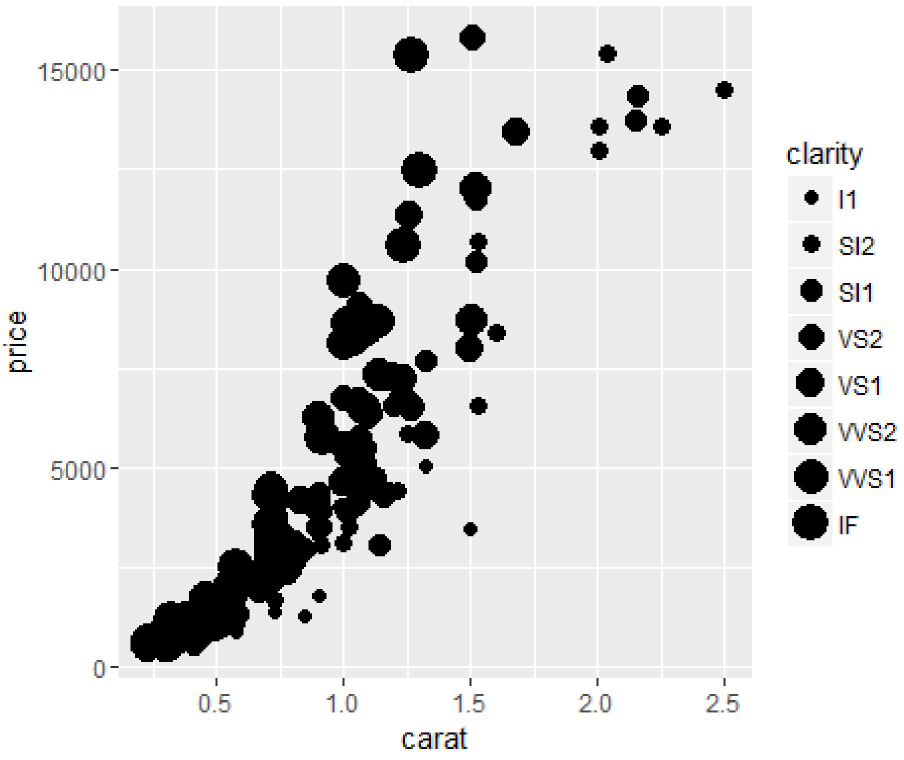


Considering size,color and shape - important part of Aesthetics

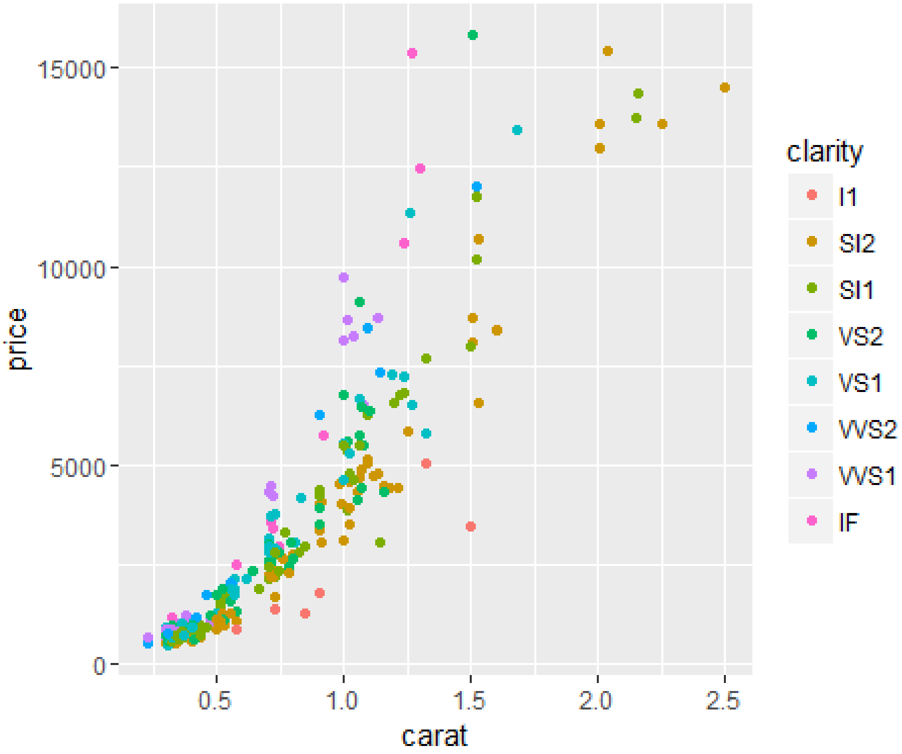
```
## Considering 2 CONTINUOUS var
C2 + geom_point(size=2, shape=15, color="blue")
```



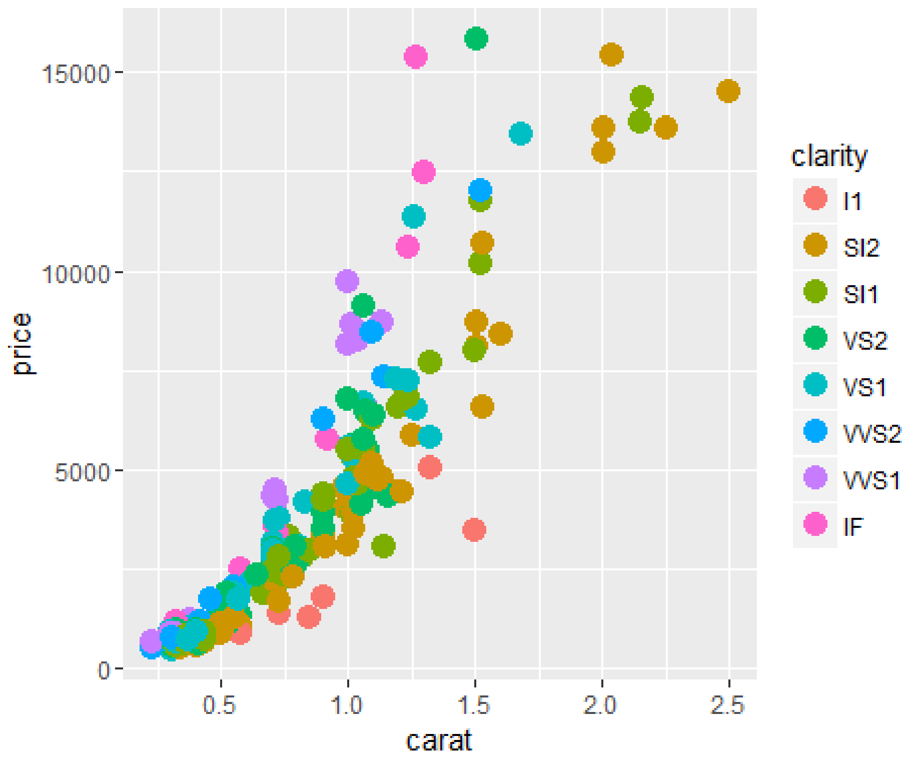
```
## Show by size:
C2 + geom_point(aes(size=clarity))
```



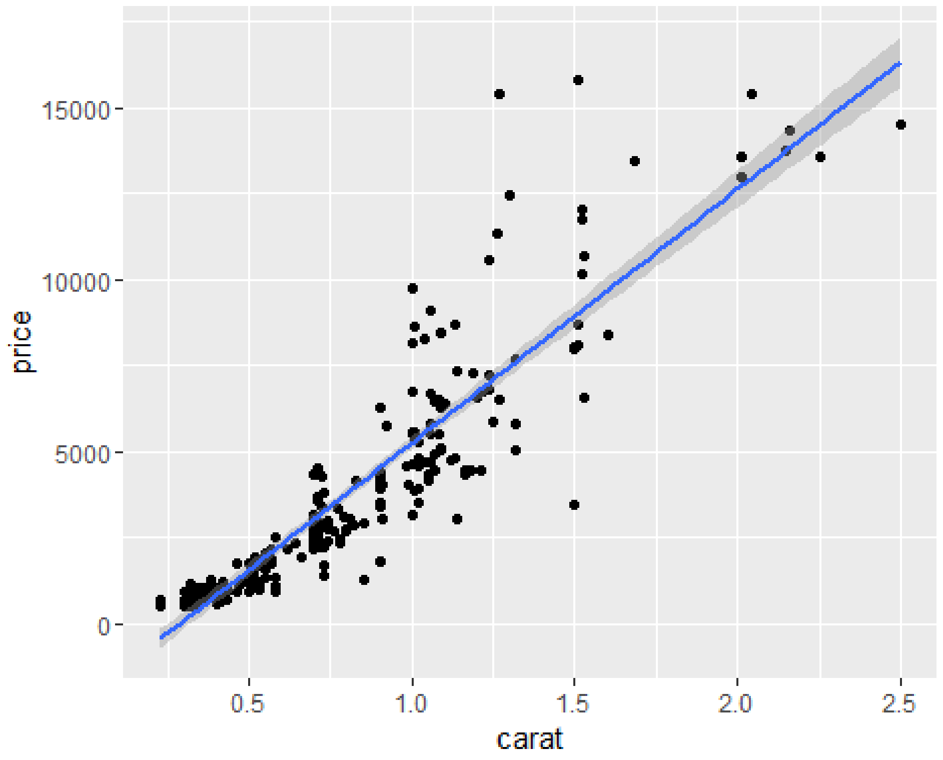
```
## Show by color:
C2 + geom_point(aes(color=clarity))
```



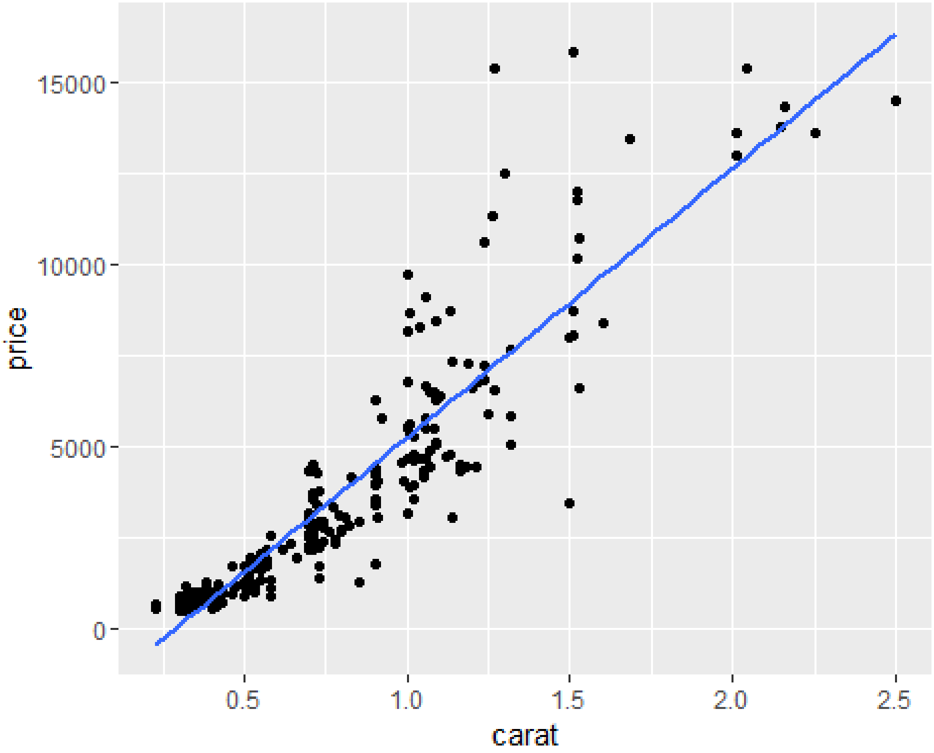
```
## Increase the size of dots
C2 + geom_point(size=4,aes(color=clarity))
```



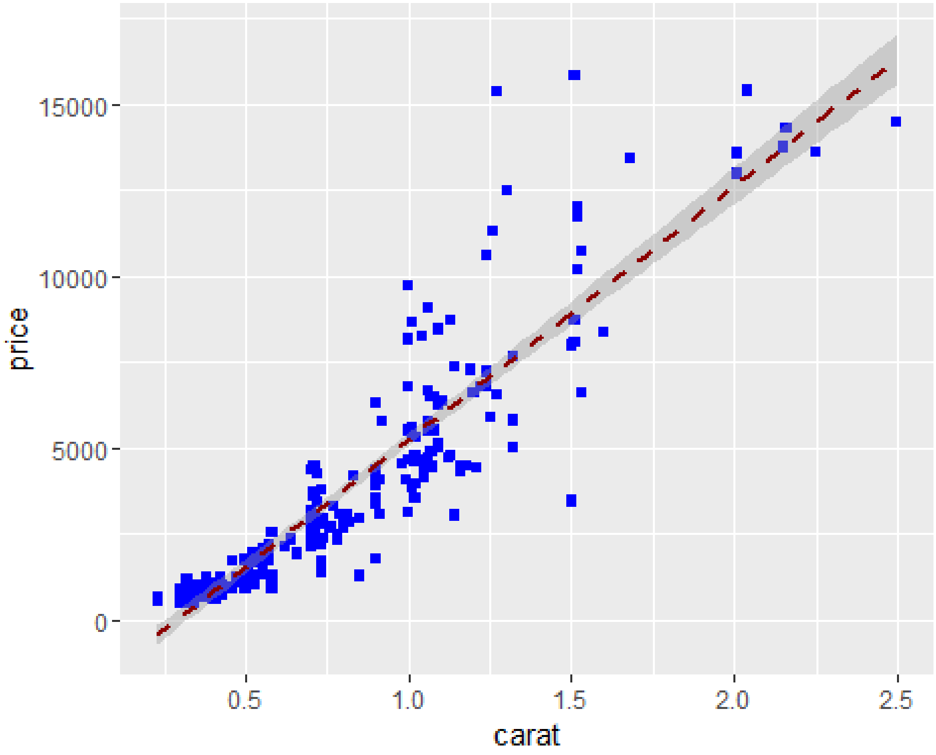
```
## Doing some Regression
C2 + geom_point() + geom_smooth(method = lm)
```



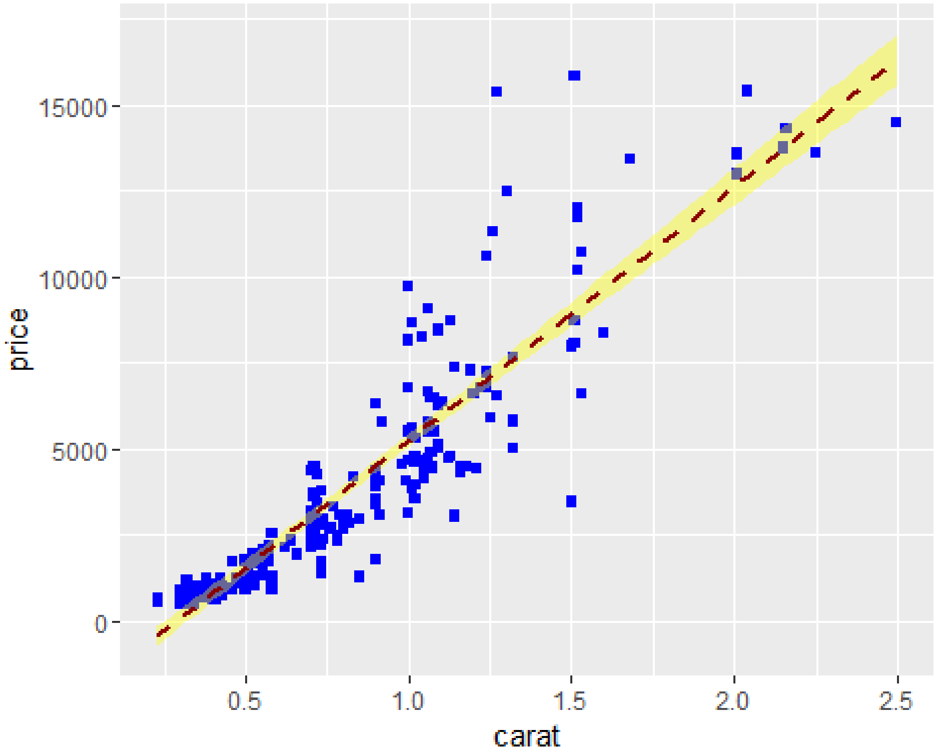
```
## No confidence interval
C2 + geom_point() + geom_smooth(method = lm, se = FALSE)
```



```
## Line type and color of regression line
C2 + geom_point(shape=15,color="blue") + geom_smooth(method = lm,
linetype="dashed", color="darkred")
```

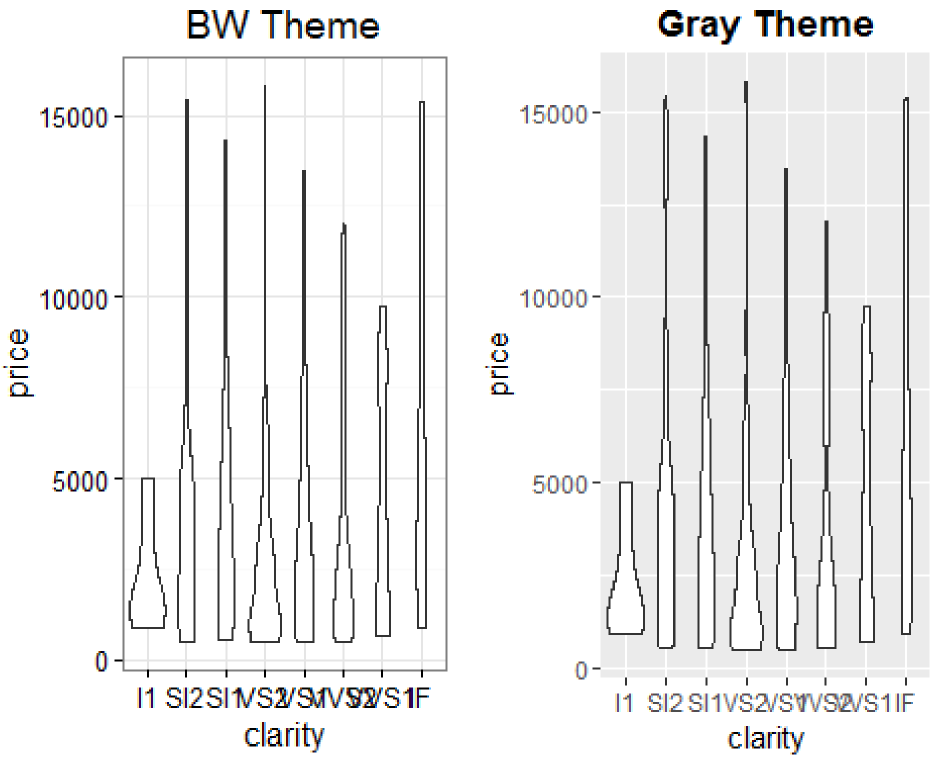


```
## Fill color of the confidence interval
C2 + geom_point(shape=15,color="blue") + geom_smooth(method = lm,
linetype="dashed", color="darkred",fill="yellow")
```

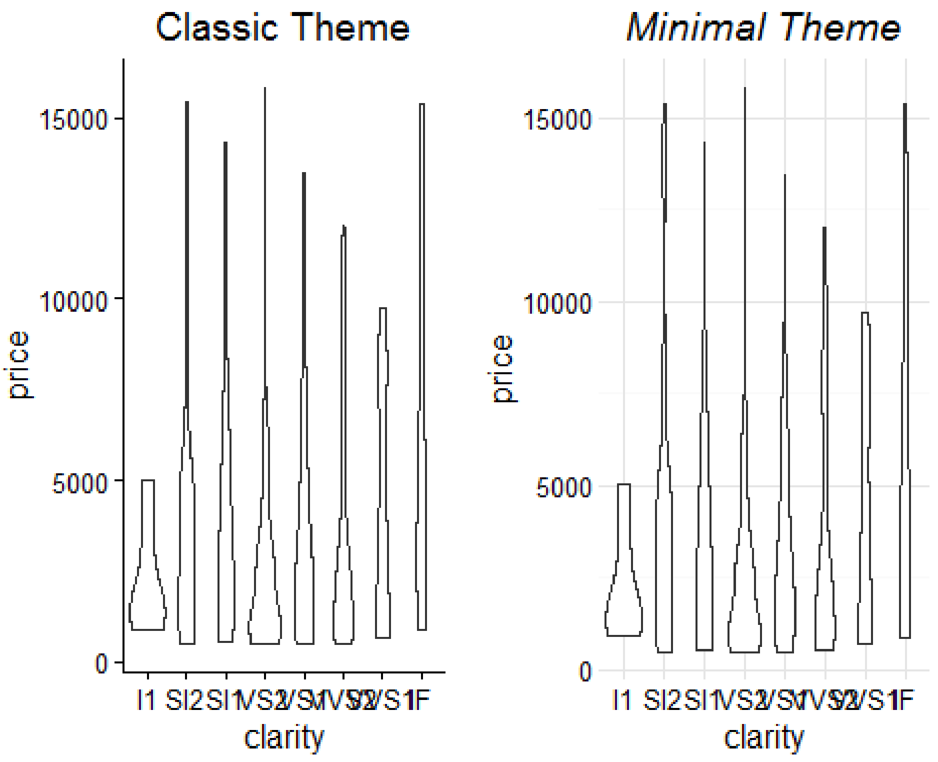


Considering the themes and titles

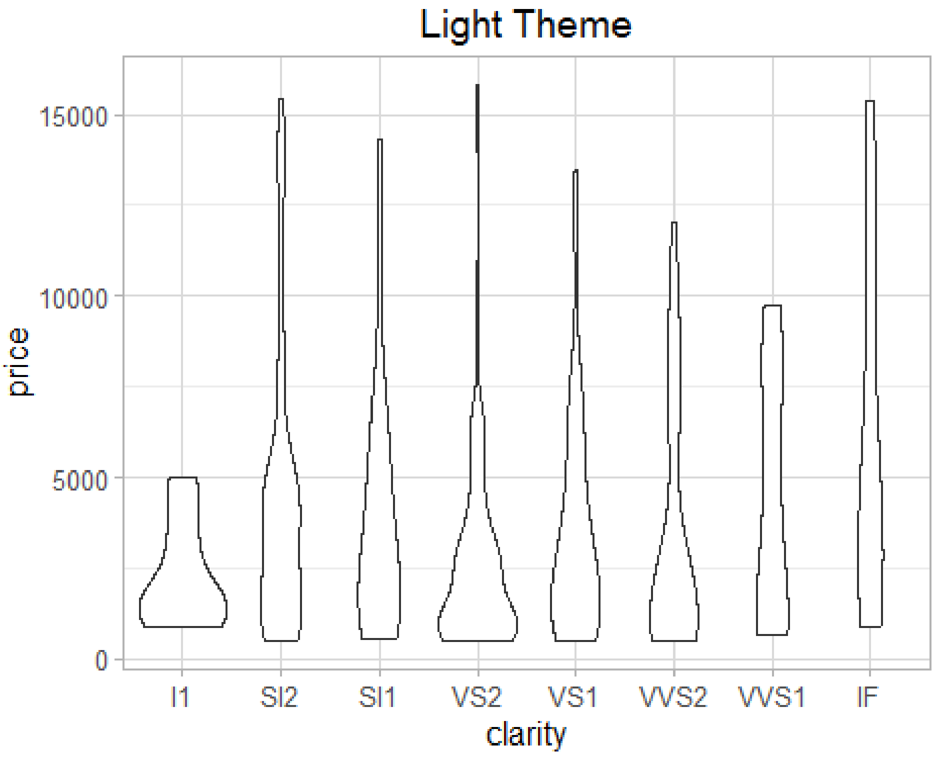
```
s1 <- C1D1 + geom_violin(scale = "area") + ggtitle("BW Theme") + theme_bw()
s2 <- C1D1 + geom_violin(scale = "area") + ggtitle("Gray Theme") +
theme_gray() + theme(plot.title = element_text(lineheight=.5, face="bold"))
s3 <- C1D1 + geom_violin(scale = "area") + ggtitle("Classic Theme") +
theme_classic()
s4 <- C1D1 + geom_violin(scale = "area") + ggtitle("Minimal Theme") +
theme_minimal() + theme(plot.title = element_text(lineheight=.9,
face="italic"))
s5 <- C1D1 + geom_violin(scale = "area") + ggtitle("Light Theme")  +
theme_light()

grid.arrange(s1,s2,ncol=2)
```



```
grid.arrange(s3,s4,ncol=2)
```



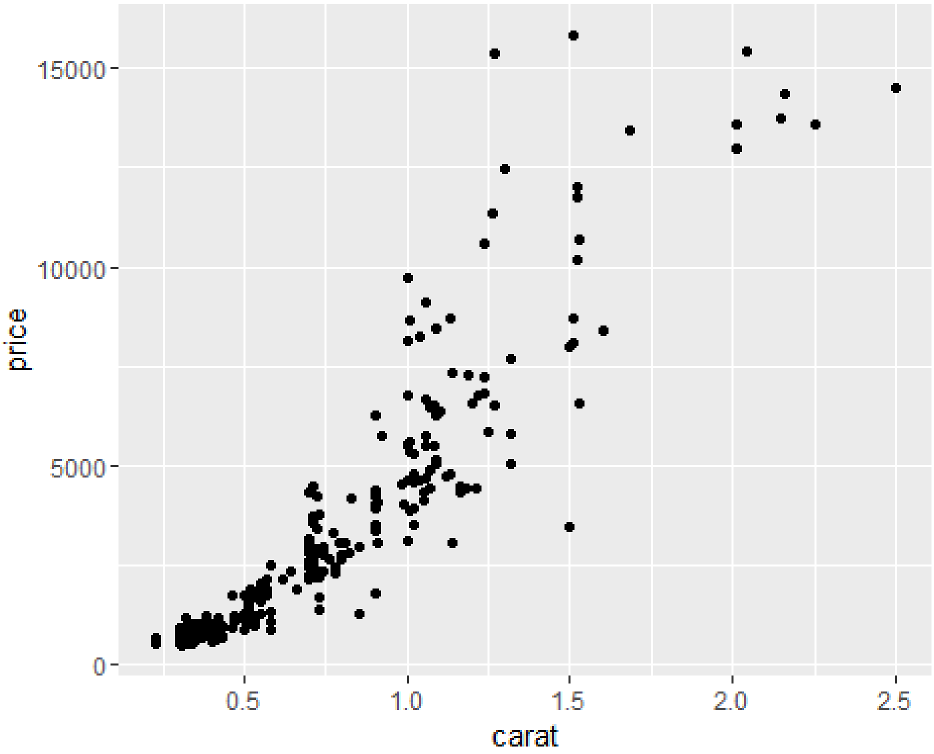
```
grid.arrange(s5,ncol=1)
```



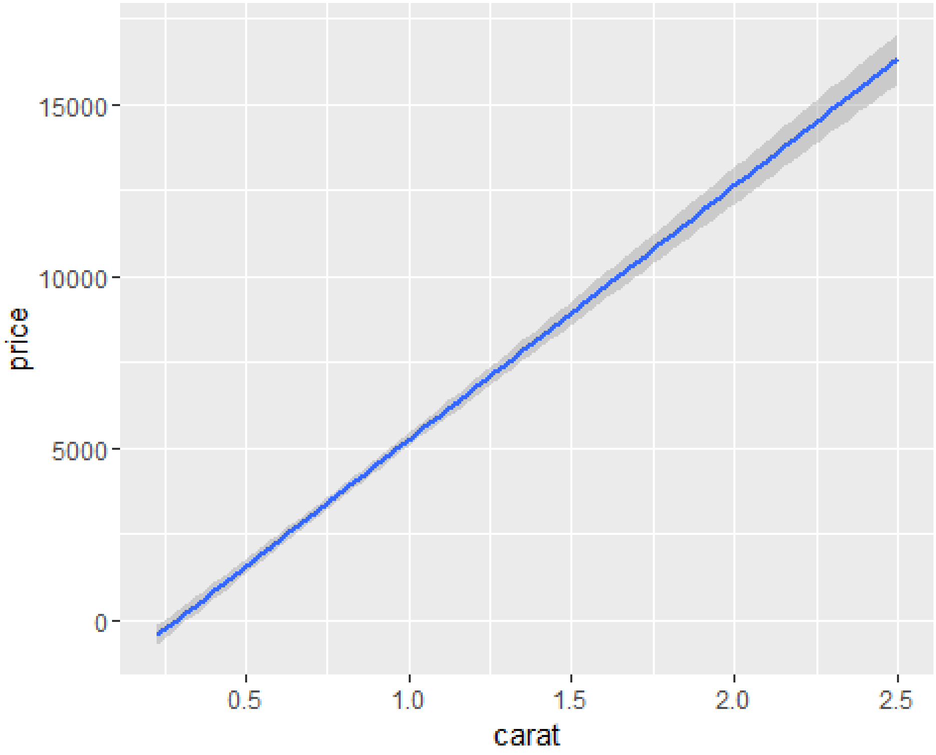
q-plot:

- Easy, quick and dirty
- No need of specifying the geom layer -- it assumes
- We can still add the geom if we want
- Advantage of grammar graphics
- Short-cut and not really flexible as compared to ggplot

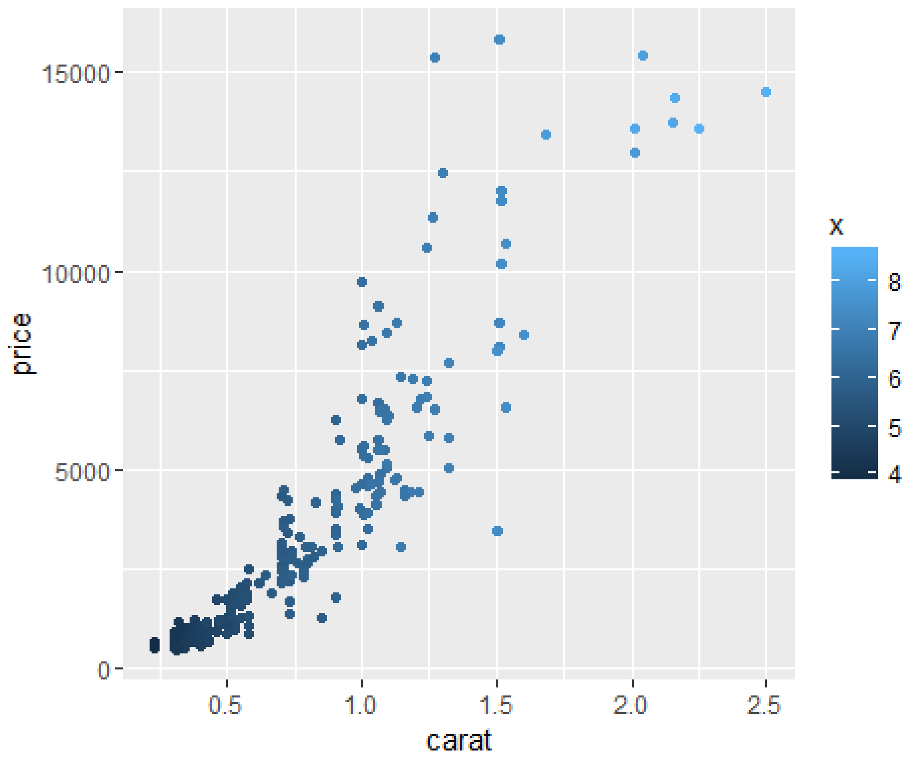
```
# Simple q-plot:
qplot(carat,price,data=diam_ss)
```



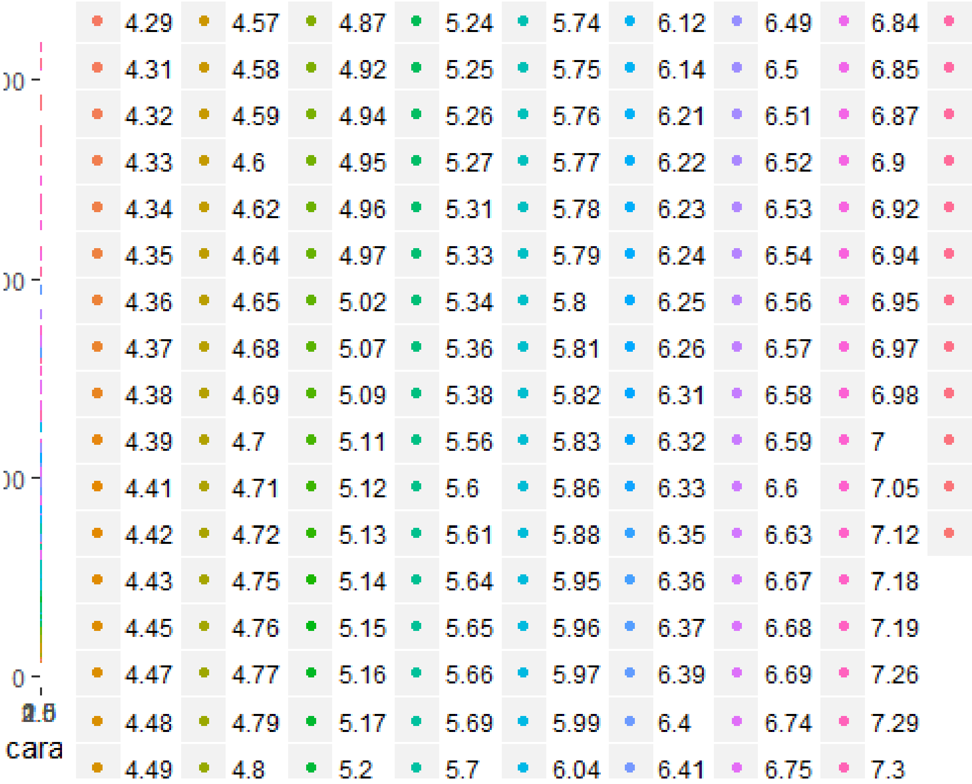
```
# Linear regression
qplot(carat,price,data=diam_ss, geom="smooth", method="lm")
```



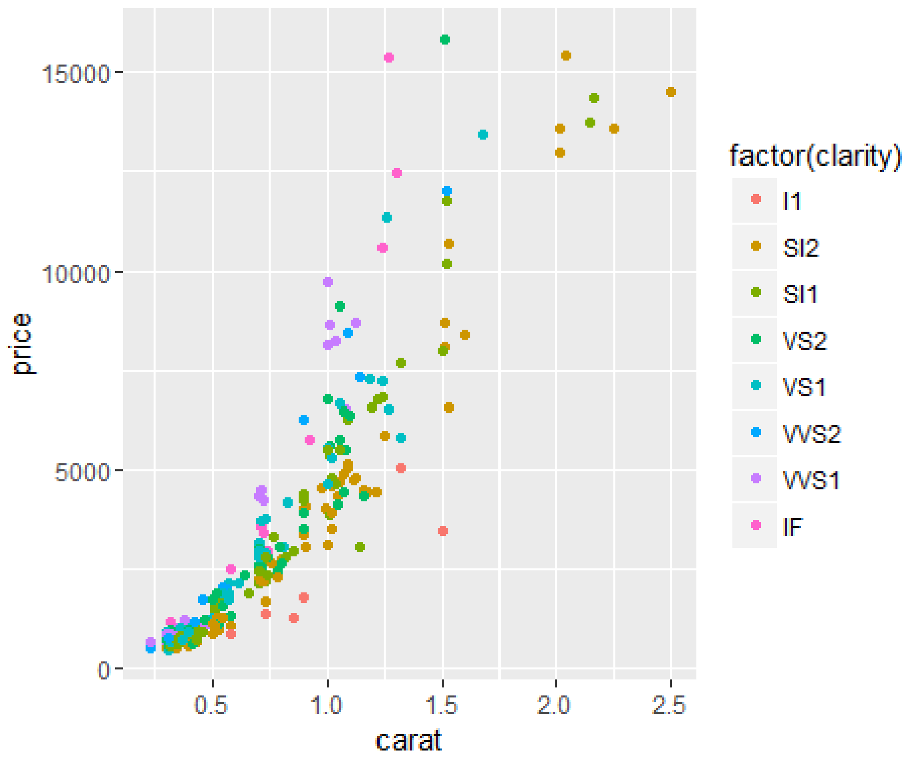
```
# Why is it not taking lm ?????????
#qplot(carat,price,data=diam_ss, geom=c("point","smooth"), method="lm")

# Changing color by continuous variable
qplot(carat,price,data=diam_ss,color=x)
```



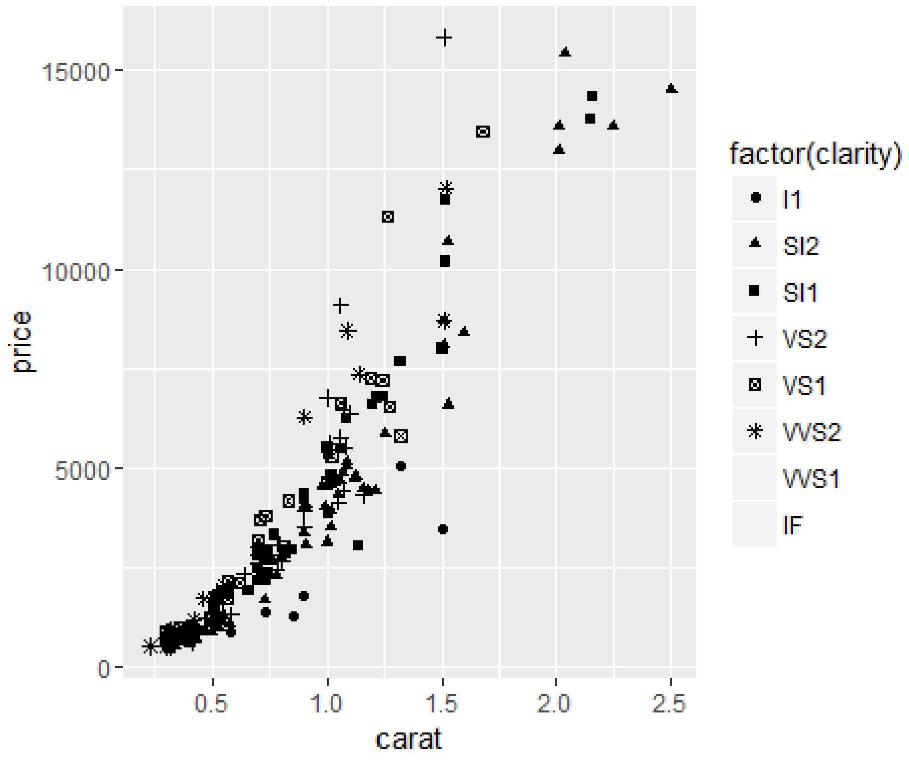
```
# Changing them by factors
qplot(carat,price,data=diam_ss,color=factor(x))
```



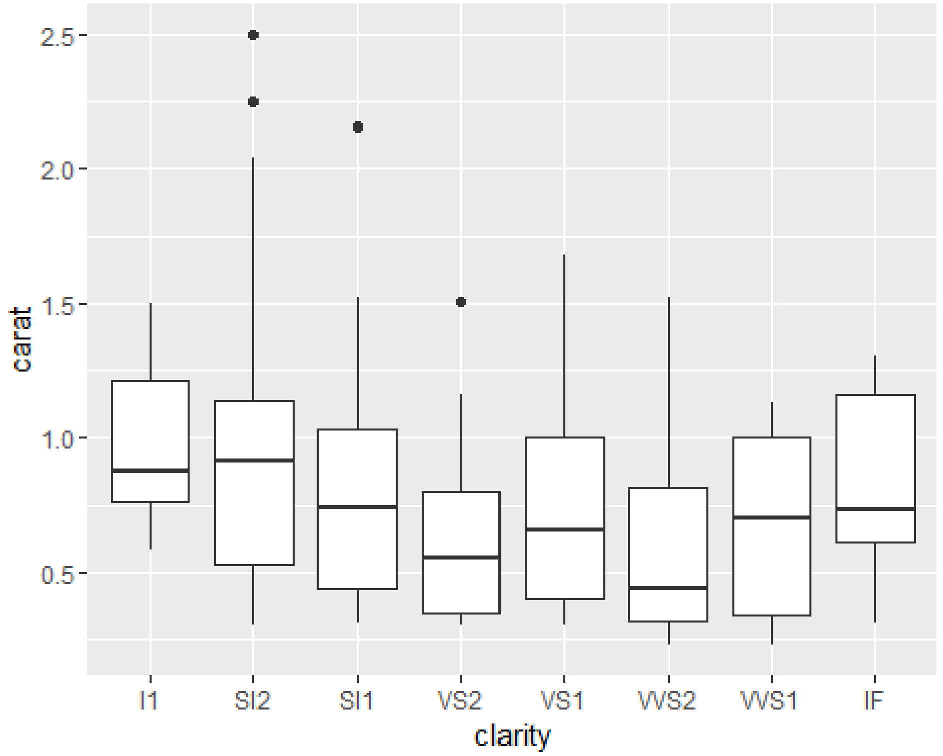
```
qplot(carat,price,data=diam_ss,color=factor(clarity))
```



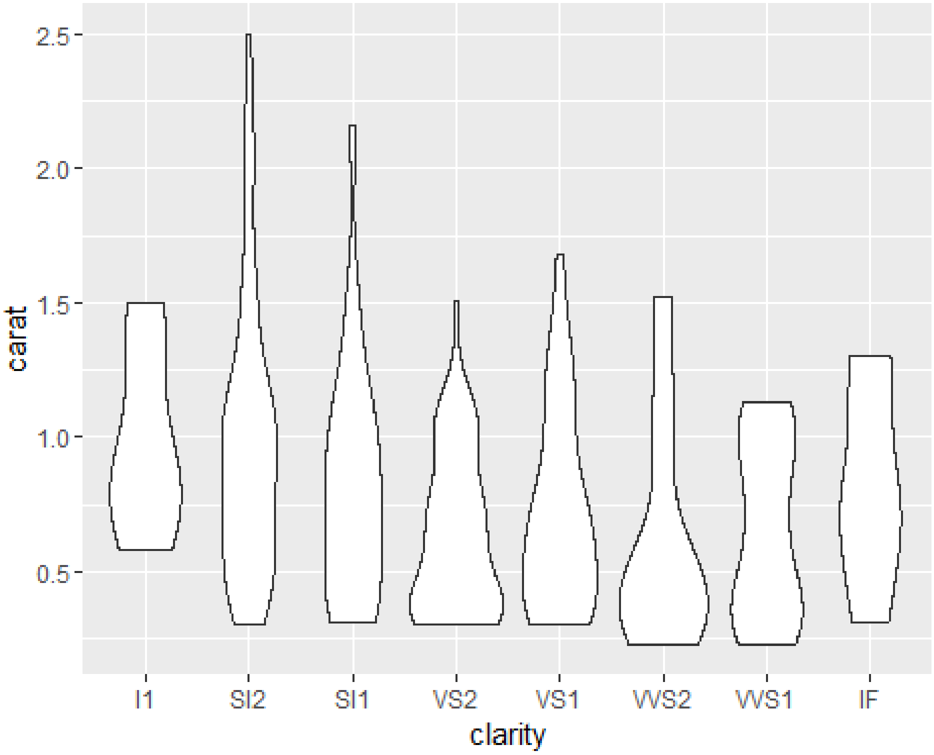
```r
# Change point shape by factors/groups:
qplot(carat,price,data=diam_ss,shape=factor(clarity))
```



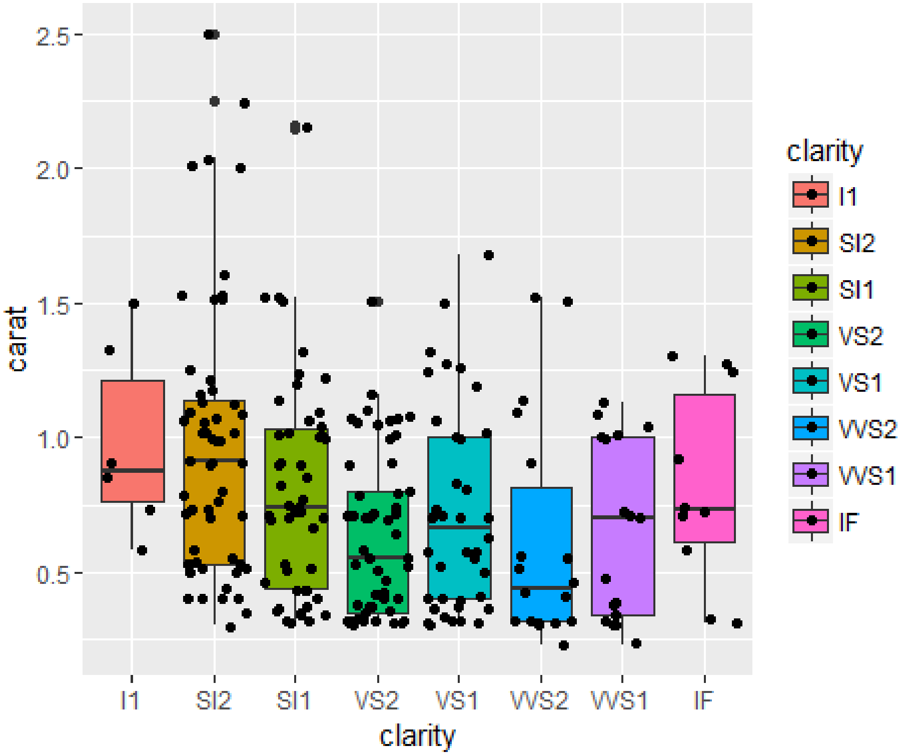
```r
# Boxplot:
qplot(clarity,carat,data=diam_ss,geom = c("boxplot"))
```



```r
# Violinplot:
qplot(clarity,carat,data=diam_ss,geom = c("violin"))
```



```r
# Change color by groups:
qplot(clarity,carat,data=diam_ss,geom = c("boxplot","jitter"), fill=clarity)
```


carat
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF
clarity
clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF