

Diamond Data Visualization with R

```
library(ggplot2)
data(diamonds)
names(diamonds)

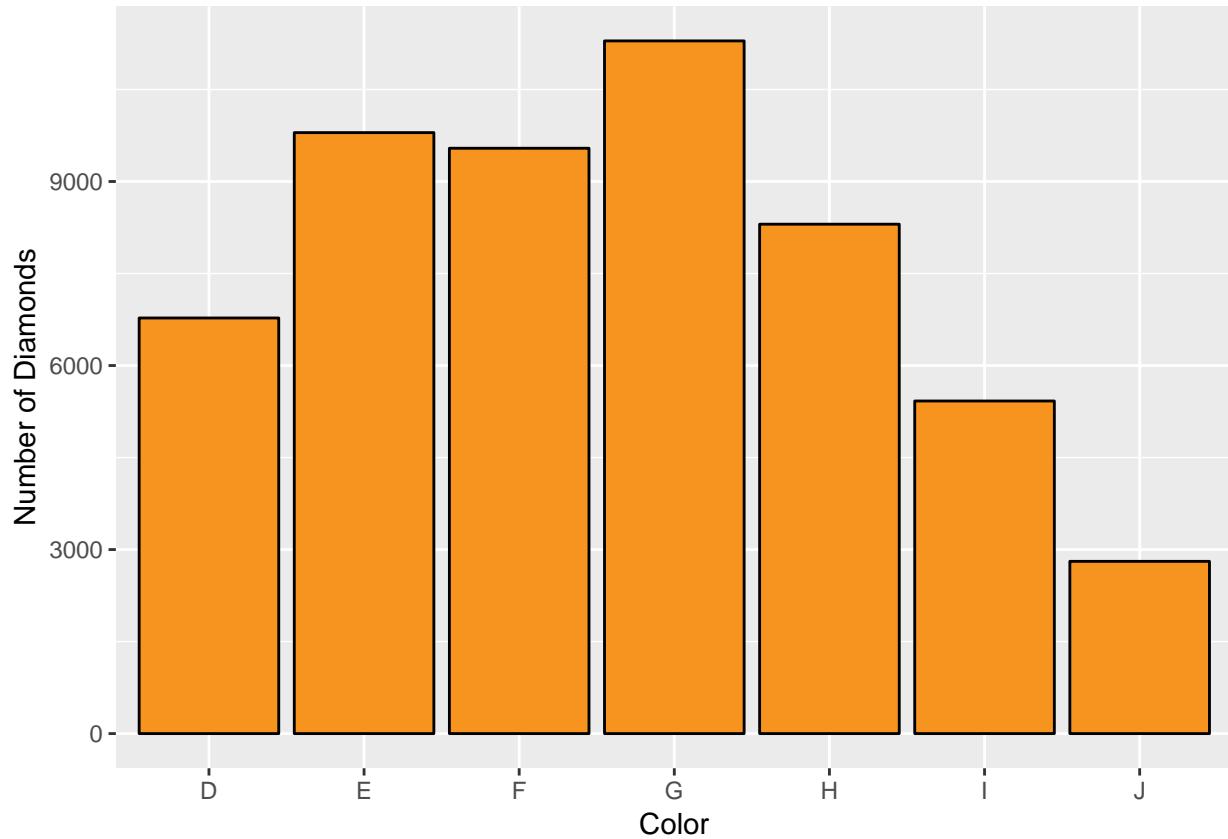
## [1] "carat"      "cut"        "color"       "clarity"     "depth"       "table"       "price"
## [8] "x"          "y"          "z"

summary(diamonds)

##      carat            cut           color          clarity
##  Min.   :0.2000   Fair      : 1610   D: 6775   SI1     :13065
##  1st Qu.:0.4000  Good     : 4906   E: 9797   VS2     :12258
##  Median :0.7000  Very Good:12082   F: 9542   SI2     : 9194
##  Mean   :0.7979  Premium  :13791   G:11292   VS1     : 8171
##  3rd Qu.:1.0400  Ideal    :21551   H: 8304   VVS2    : 5066
##  Max.   :5.0100                    I: 5422   VVS1    : 3655
##                               J: 2808   (Other): 2531
##      depth           table          price          x
##  Min.   :43.00   Min.   :43.00   Min.   : 326   Min.   : 0.000
##  1st Qu.:61.00  1st Qu.:56.00  1st Qu.: 950   1st Qu.: 4.710
##  Median :61.80  Median :57.00  Median :2401   Median : 5.700
##  Mean   :61.75  Mean   :57.46  Mean   :3933   Mean   : 5.731
##  3rd Qu.:62.50  3rd Qu.:59.00  3rd Qu.:5324   3rd Qu.: 6.540
##  Max.   :79.00  Max.   :95.00  Max.   :18823  Max.   :10.740
##
##      y                  z
##  Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 4.720   1st Qu.: 2.910
##  Median : 5.710   Median : 3.530
##  Mean   : 5.735   Mean   : 3.539
##  3rd Qu.: 6.540   3rd Qu.: 4.040
##  Max.   :58.900   Max.   :31.800
##
```

Representing the best Color

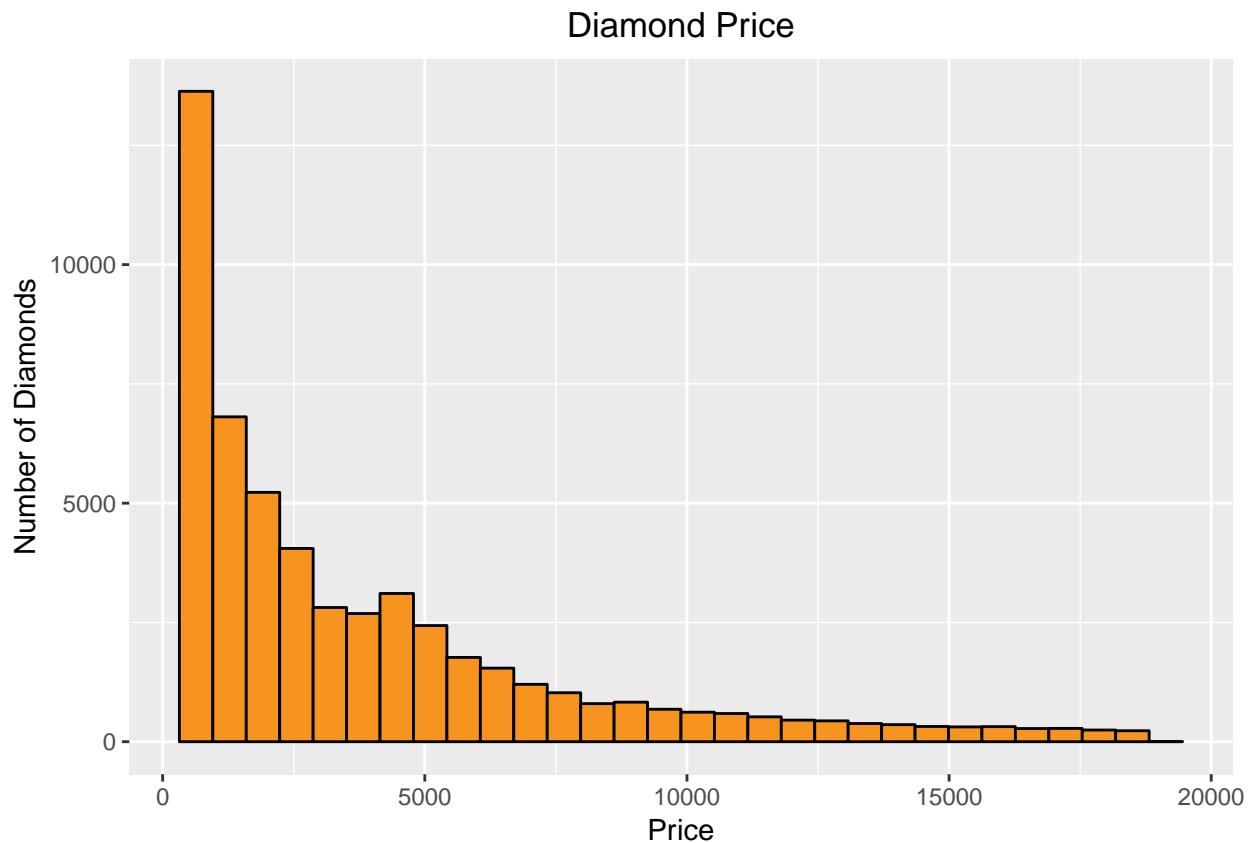
```
qplot(data = diamonds, x = color, color = I('black'), fill = I('#F79420'),
      xlab = 'Color', ylab = 'Number of Diamonds')
```



Creating a histogram of the price of all the diamonds in the diamond data set.

```
qplot(data = diamonds, x = price, color = I('black'), fill = I('#F79420'),
      xlab = 'Price', ylab = 'Number of Diamonds') +
  ggtitle('Diamond Price') +
  theme(plot.title = element_text(hjust = 0.5))
```

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



```
summary(diamonds$price)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##      326     950    2401     3933    5324   18823
```

How many diamonds cost less than \$500?

```
nrow(subset(diamonds, price < 500))

## [1] 1729
```

How many diamonds cost less than \$250?

```
nrow(subset(diamonds, price < 250))

## [1] 0
```

How many diamonds cost \$15000 or more?

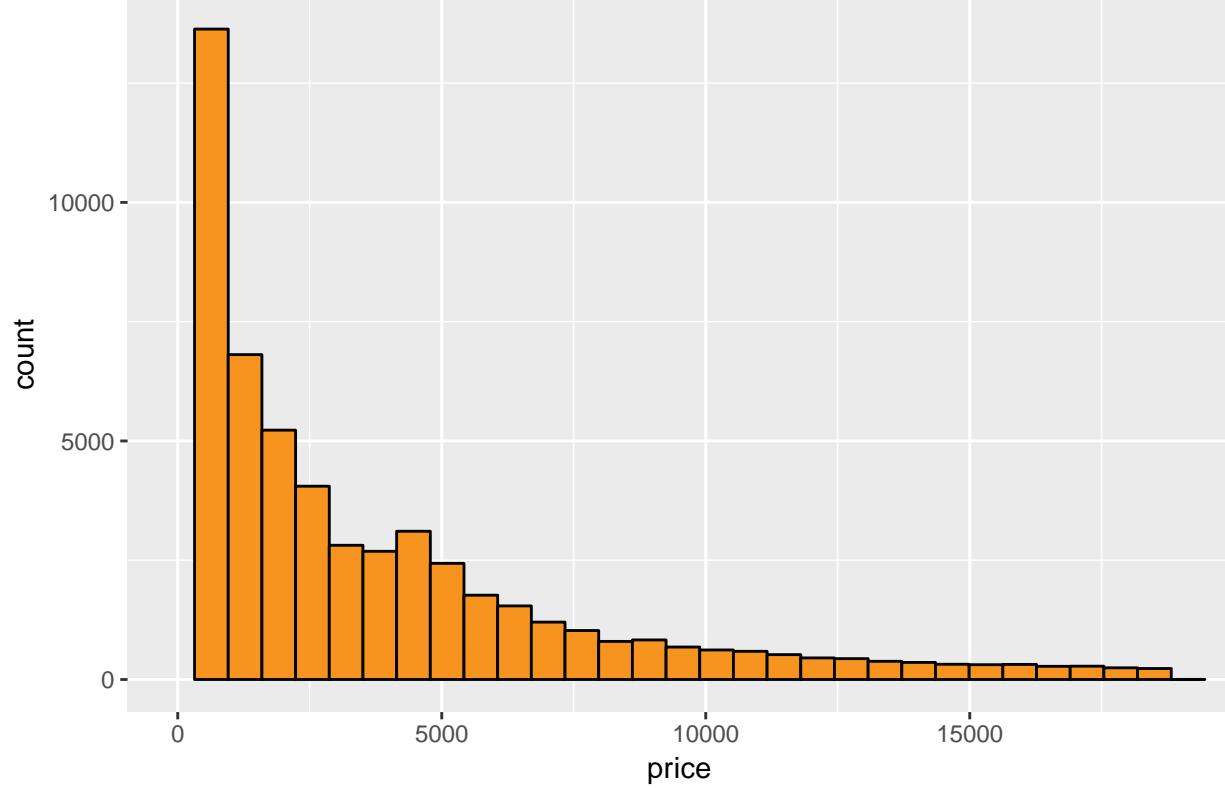
```
nrow(subset(diamonds, price <= 15000))

## [1] 52285
```

Changing the transparency of the points and making the x-axis every 2 units

```
ggplot(diamonds) +  
  geom_histogram( aes(x = price), color = I('black'), fill = I('#F79420'),  
    xlab = 'Price', ylab = 'Number of Diamonds') +  
  ggtitle('Diamond Price') +  
  theme(plot.title = element_text(hjust = 0.5)) +  
  coord_cartesian(xlim = c(0,19000))  
  
## Warning: Ignoring unknown parameters: xlab, ylab  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

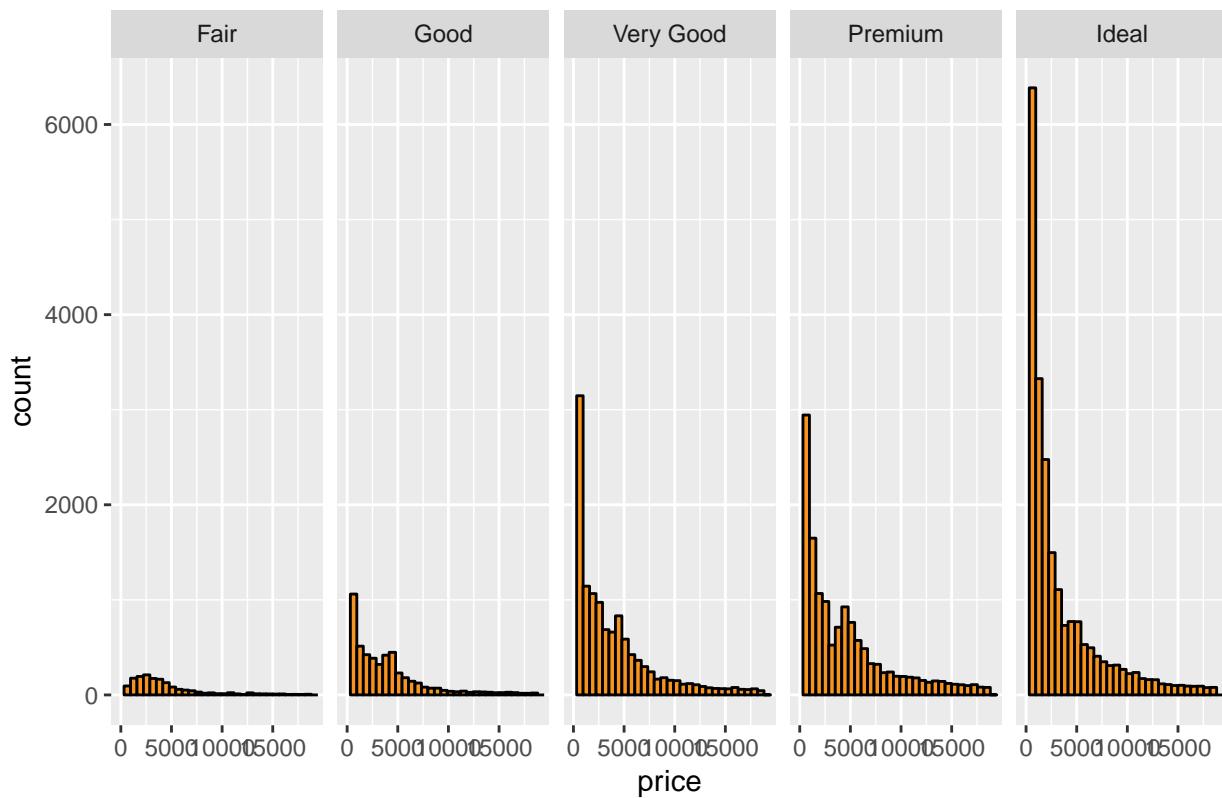
Diamond Price



Breaking down the main price histogram to five histograms in separate panels.

```
ggplot(diamonds) +  
  geom_histogram( aes(x = price), color = I('black'), fill = I('#F79420'),  
    xlab = 'Price', ylab = 'Number of Diamonds') +  
  ggtitle('Diamond Price') +  
  theme(plot.title = element_text(hjust = 0.5)) +  
  coord_cartesian(xlim = c(0,19000)) +  
  facet_grid(. ~ cut)  
  
## Warning: Ignoring unknown parameters: xlab, ylab  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

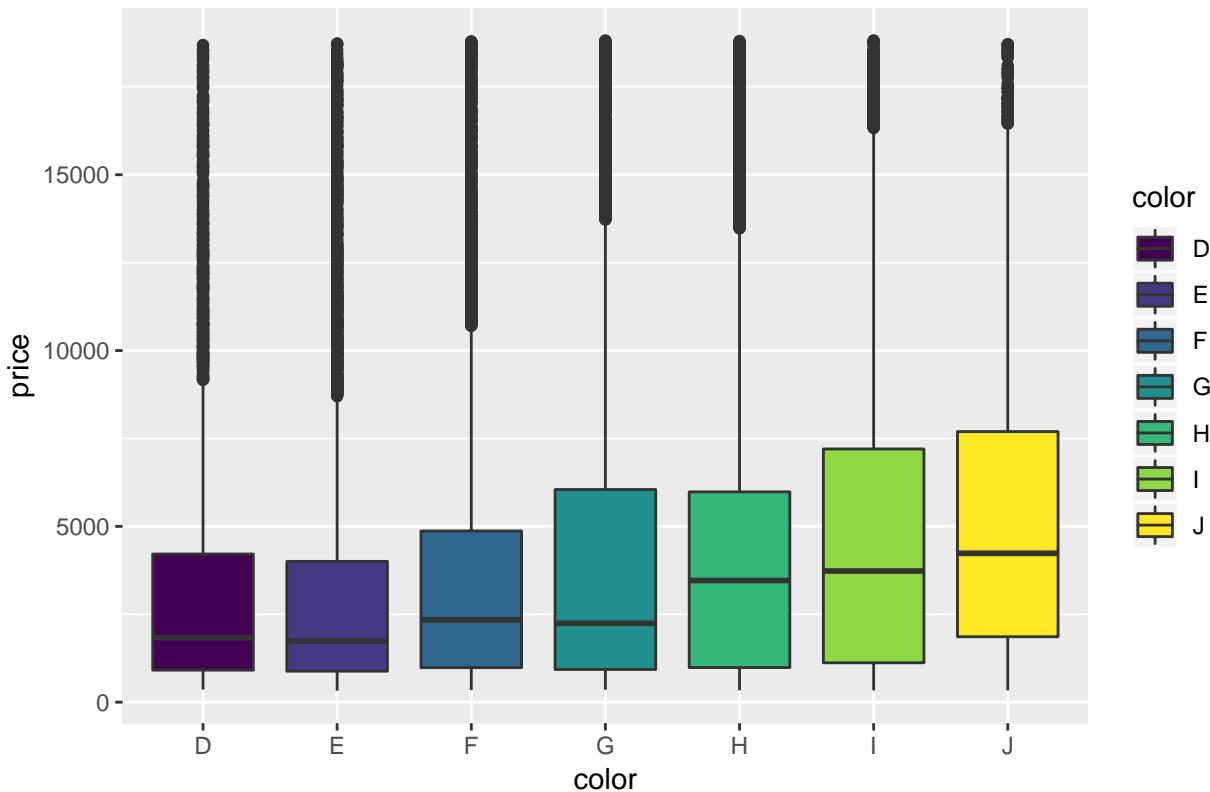
Diamond Price



Investigate the price of diamonds using box plots

```
ggplot(diamonds) +  
  geom_boxplot(aes(x = color, y = price, fill = color)) +  
  ggtitle("Diamonds Price by Color.") +  
  theme(plot.title = element_text(hjust = 0.5))
```

Diamonds Price by Color.



what is the price range for the middle 50% of diamonds with color D?

what is the price range for the middle 50% of diamonds with color J?

what is the IQR for the diamonds with best color?

what is the IQR for the diamonds with worst color?

```
summary(diamonds$price[diamonds$color == 'D'])

##    Min. 1st Qu. Median     Mean 3rd Qu.    Max.
##      357      911     1838     3170     4214   18693

summary(diamonds$price[diamonds$color == 'J'])

##    Min. 1st Qu. Median     Mean 3rd Qu.    Max.
##      335     1860     4234     5324     7695   18710

IQR(diamonds$price[diamonds$color == "D"])

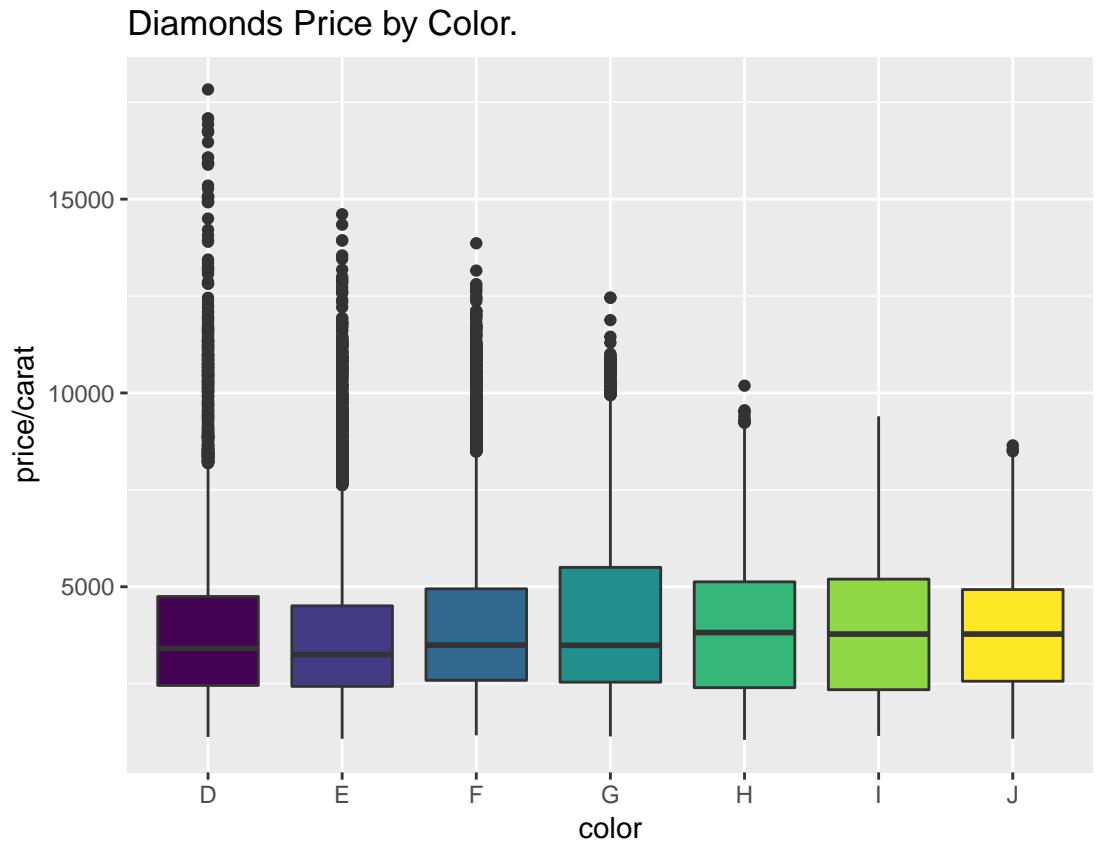
## [1] 3302.5

IQR(diamonds$price[diamonds$color == "J"])

## [1] 5834.5
```

Investigating the price per carat of diamonds across the different colors of diamonds using boxplots.

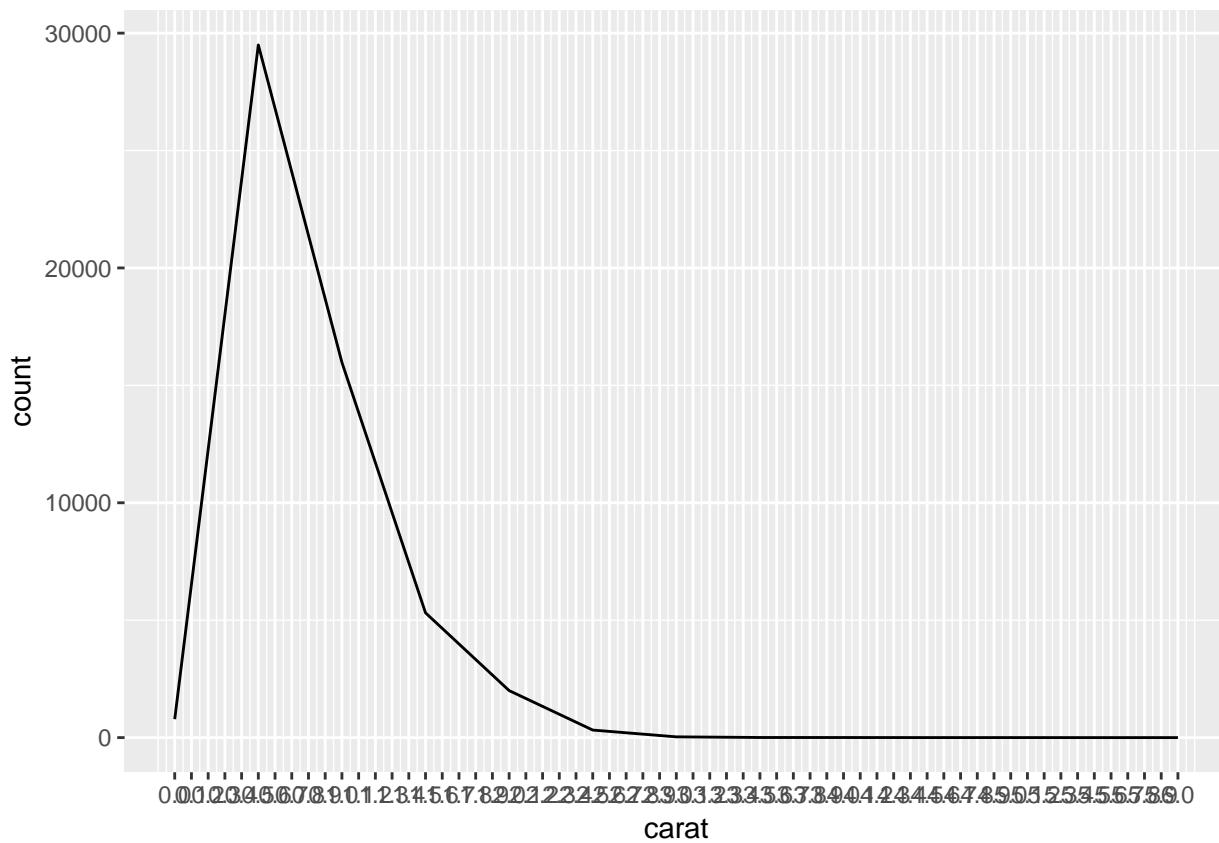
```
ggplot(diamonds) +
  geom_boxplot(aes(x = color, y = price/carat, fill = color)) +
  ggtitle("Diamonds Price by Color.")
```



investigate the weight of the diamonds using frequency polygon. ### what carat size has a count greater than 2000?

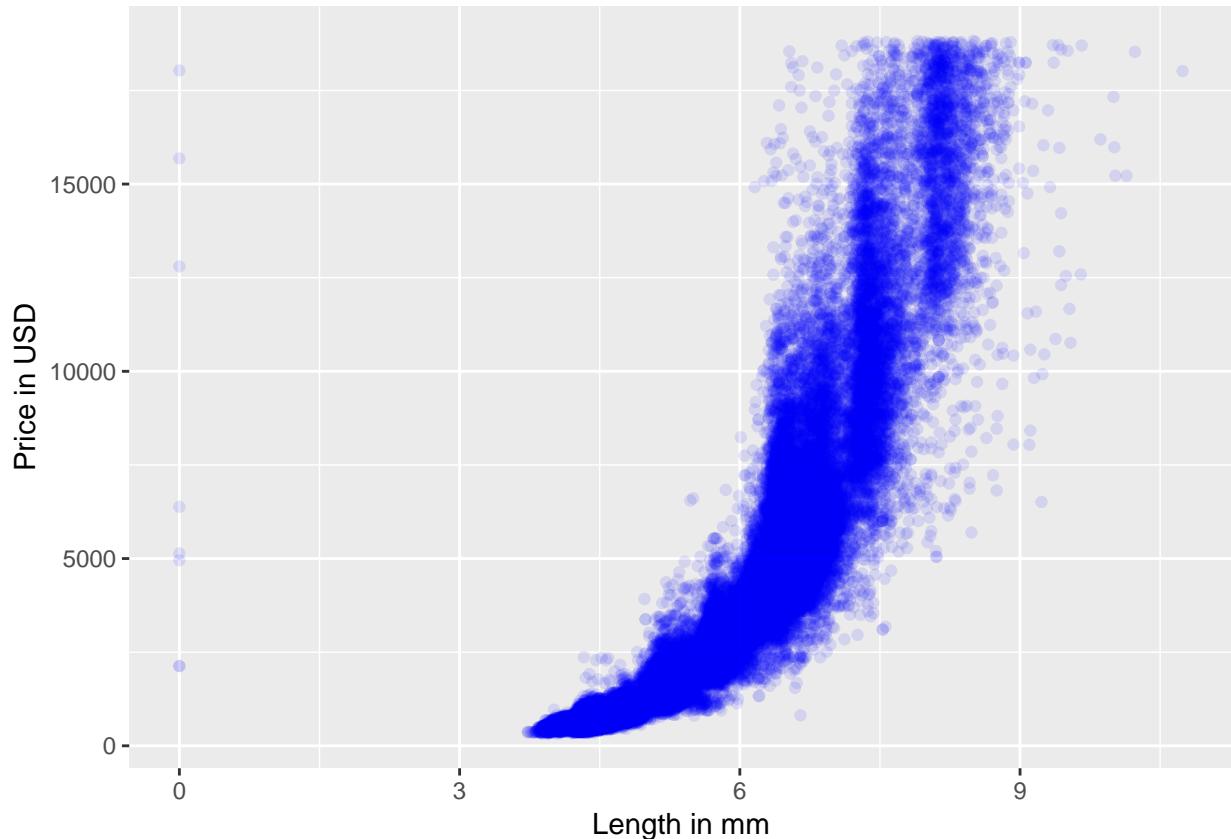
```
ggplot(diamonds) + geom_freqpoly(aes(x = carat), binwidth = 0.5) +
  scale_x_continuous(limits = c(0,6), breaks = seq(0,6,0.1))
```

Warning: Removed 2 rows containing missing values (geom_path).



Create a scatterplot of price vs x

```
ggplot(data = diamonds) +  
  xlab('Length in mm') +  
  ylab('Price in USD') +  
  geom_point(aes(x = x, y = price), color = 'blue', alpha = 1/10)
```



Correlation between price and x, between price and y, between price and z

```
cor.test(diamonds$price,diamonds$x)

##
##  Pearson's product-moment correlation
##
## data: diamonds$price and diamonds$x
## t = 440.16, df = 53938, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8825835 0.8862594
## sample estimates:
##      cor
## 0.8844352

cor.test(diamonds$price,diamonds$y)

##
##  Pearson's product-moment correlation
##
## data: diamonds$price and diamonds$y
## t = 401.14, df = 53938, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8632867 0.8675241
```

```

## sample estimates:
##      cor
## 0.8654209
cor.test(diamonds$price,diamonds$z)

##
## Pearson's product-moment correlation
##
## data: diamonds$price and diamonds$z
## t = 393.6, df = 53938, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.8590541 0.8634131
## sample estimates:
##      cor
## 0.8612494

```

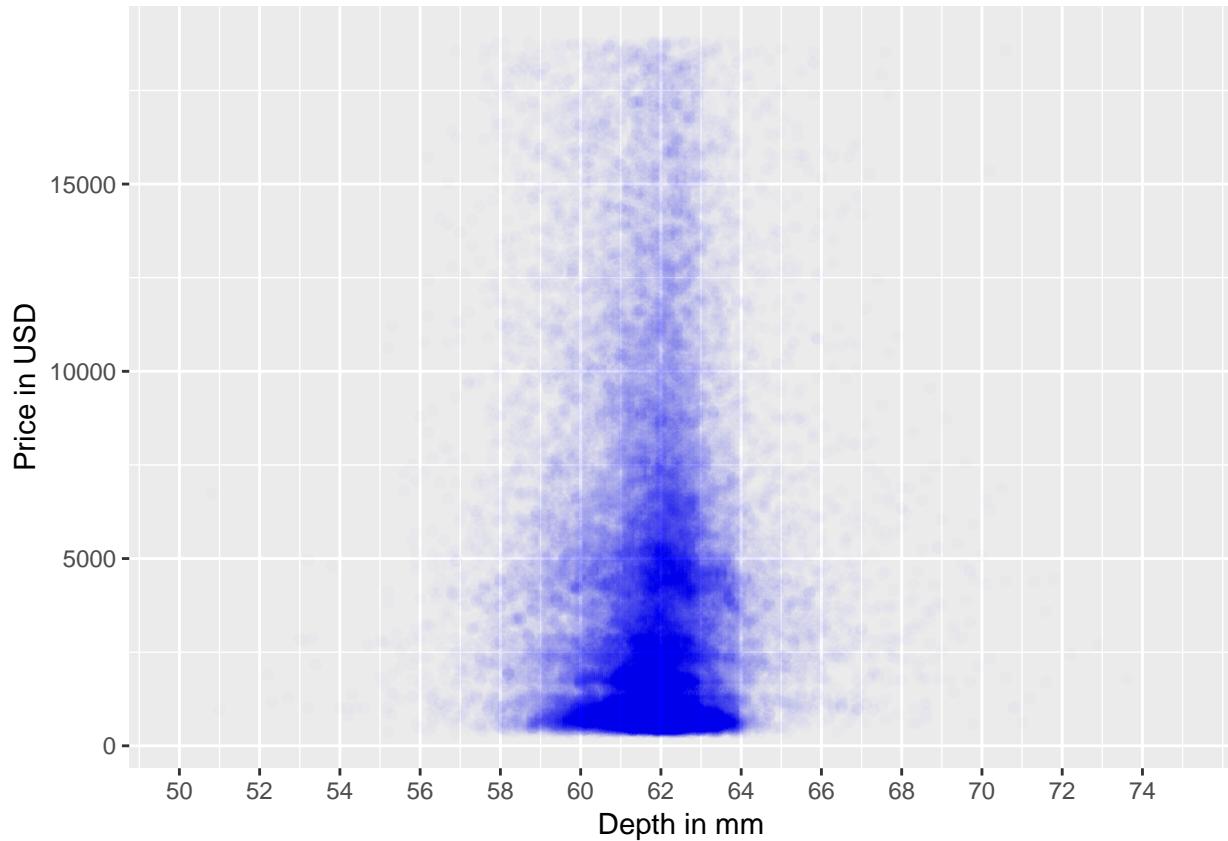
Create a scatterplot of price vs depth

```

ggplot(data = diamonds) +
  xlab('Depth in mm') +
  ylab('Price in USD') +
  geom_point(aes(x = depth, y = price),alpha=1/100, color = 'blue') +
  scale_x_continuous(limits = c(50, 75),
                     breaks = seq(50,75, 2))

## Warning: Removed 6 rows containing missing values (geom_point).

```



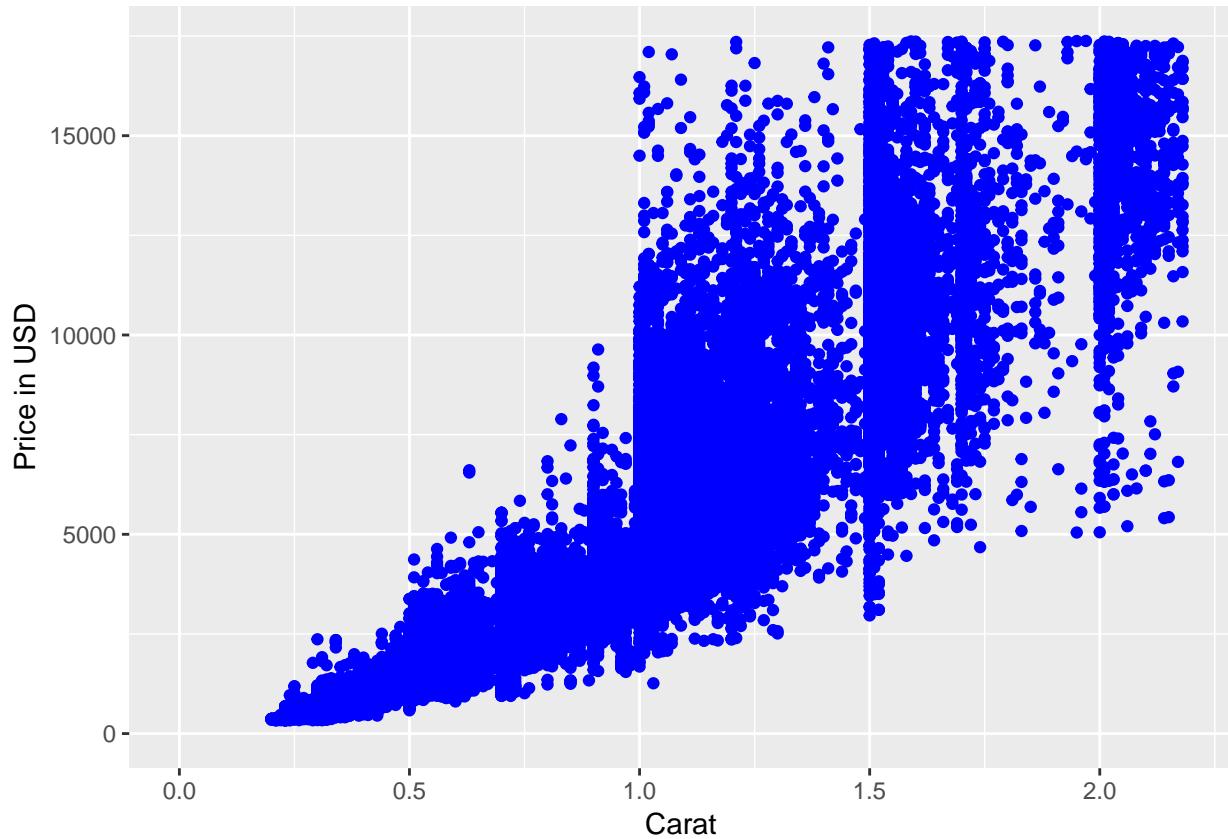
```
### Correlation between price and depth
cor.test(diamonds$price,diamonds$depth)

##
## Pearson's product-moment correlation
##
## data: diamonds$price and diamonds$depth
## t = -2.473, df = 53938, p-value = 0.0134
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.019084756 -0.002208537
## sample estimates:
## cor
## -0.0106474
```

Creating a scatterplot of price vs carat and omit the top 1% of price and carat values.

```
ggplot( data = diamonds) +
  xlab('Carat') +
  ylab('Price in USD') +
  geom_point(aes(x = carat, y = price),color = 'blue') +
  xlim(0, quantile(diamonds$carat, probs = 0.99)) + # 99% percentile on x variable
  ylim(0, quantile(diamonds$price, probs = 0.99))

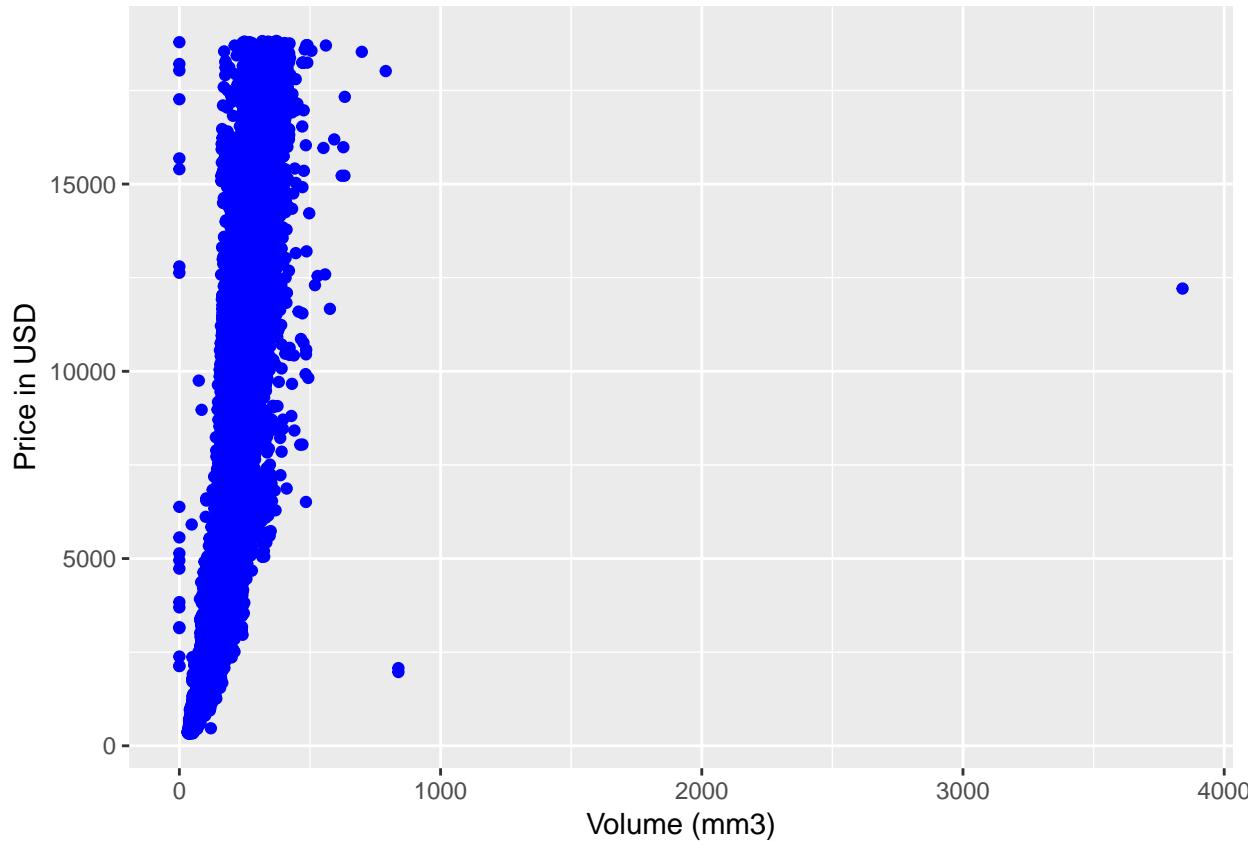
## Warning: Removed 926 rows containing missing values (geom_point).
```



Create a scatterplot of price vs volume (xyz)

```
diamonds$volume <- diamonds$x * diamonds$y * diamonds$z

ggplot( data = diamonds) +
  xlab('Volume (mm3)') +
  ylab('Price in USD') +
  geom_point(aes(x = volume, y = price),color = 'blue')
```



number of diamonds with volume=0

```
dim(diamonds[diamonds$volume == 0,])
## [1] 20 11
```

Correlation between price vs volume (xyz)

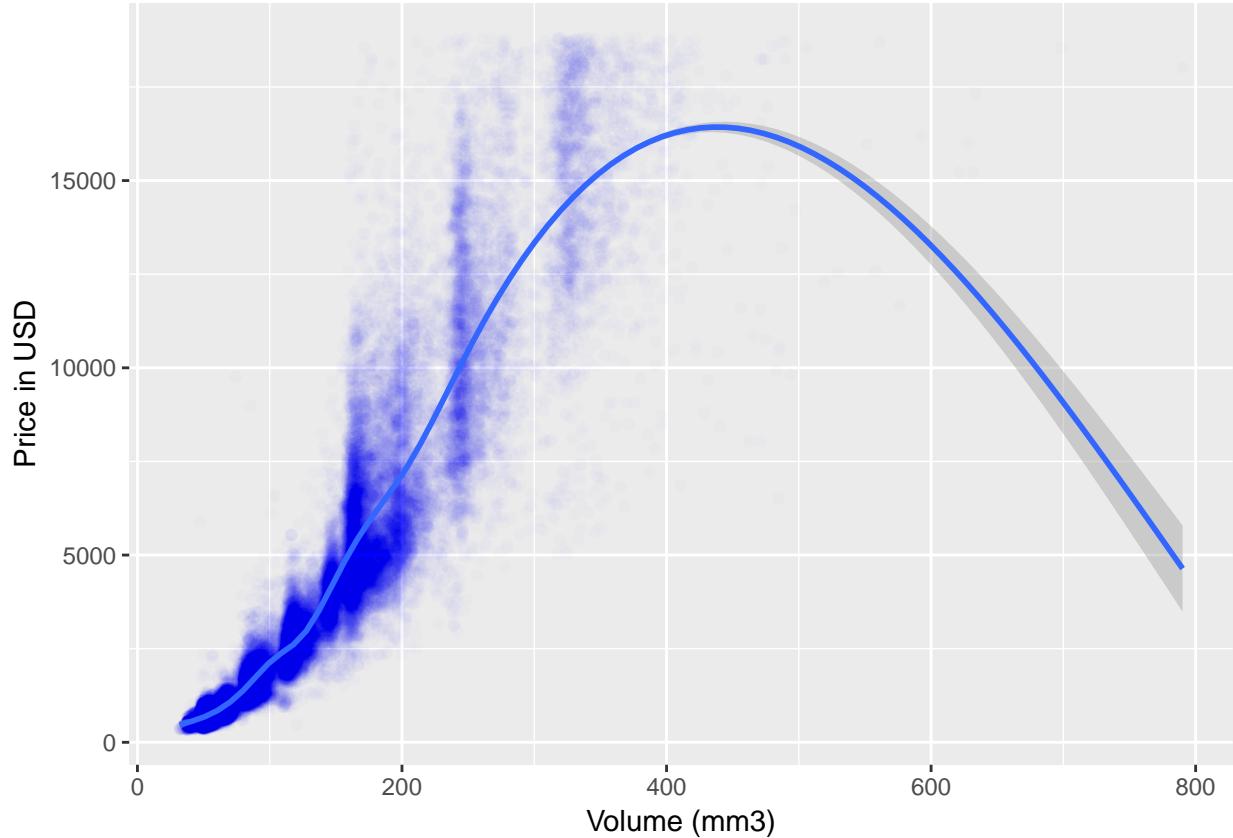
Exclude diamonds with volume=0 and volume >= 800

```
temp_df <- diamonds[ diamonds$volume > 0 & diamonds$volume <= 800 , ]
cor.test(temp_df$price,temp_df$volume)

##
## Pearson's product-moment correlation
##
## data: temp_df$price and temp_df$volume
## t = 559.19, df = 53915, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.9222944 0.9247772
## sample estimates:
##      cor
## 0.9235455
```

Create a scatterplot of price vs volume (xyz)

```
temp_df <- diamonds[ diamonds$volume > 0 & diamonds$volume <= 800 , ]  
  
ggplot( data = temp_df ) +  
  xlab('Volume (mm3)') +  
  ylab('Price in USD') +  
  geom_point(aes(x = volume, y = price),color = 'blue',alpha=1/100) +  
  geom_smooth(aes(x = volume, y = price))  
  
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Create a new data frame containing info on diamonds by clarity

```
library(dplyr)  
  
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##     filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##     intersect, setdiff, setequal, union  
clarity_groups <- group_by(diamonds,clarity)      # first groups data by clarity  
  
diamondsByClarity <- summarise(clarity_groups,           # then summarizes
```

```

mean_price = mean(price),                                # and creates new variables
median_price = median(price),
min_price = min(price),
max_price = max(price),
n = n()                                                 # number of items in each group
)

diamondsByClarity

## # A tibble: 8 x 6
##   clarity mean_price median_price min_price max_price     n
##   <ord>      <dbl>        <dbl>      <dbl>      <dbl> <int>
## 1 I1          3924.       3344      345    18531     741
## 2 SI2         5063.       4072      326    18804    9194
## 3 SI1         3996.       2822      326    18818   13065
## 4 VS2         3925.       2054      334    18823   12258
## 5 VS1         3839.       2005      327    18795     8171
## 6 VVS2        3284.       1311      336    18768     5066
## 7 VVS1        2523.       1093      336    18777     3655
## 8 IF           2865.       1080      369    18806     1790

```

Create two bar plots : mean_price vs. clarity, mean_price vs. color

```

library(dplyr)
library(ggplot2)

diamonds_by_clarity <- group_by(diamonds, clarity)
diamonds_mp_by_clarity <- summarise(diamonds_by_clarity, mean_price = mean(price))
p1 <- ggplot( data = diamonds_mp_by_clarity ) +
  geom_bar(aes(x = clarity, y = mean_price),stat="identity",color = 'black', fill = 'orange')

diamonds_by_color <- group_by(diamonds, color)
diamonds_mp_by_color <- summarise(diamonds_by_color, mean_price = mean(price))
p2 <- ggplot( data = diamonds_mp_by_color ) +
  geom_bar(aes(x = color, y = mean_price),stat="identity",color = 'black',fill = 'blue')

library(gridExtra)

##
## Attaching package: 'gridExtra'
## The following object is masked from 'package:dplyr':
##   combine
grid.arrange(p1,p2,ncol=1)

```

