Using R with Two Variables

I continue to explore the dimoands dataset which I worked with in 'problemsetone'. So, let's load the dataset into the environment.

A Scatterplot of Price vs X(Length in mm)

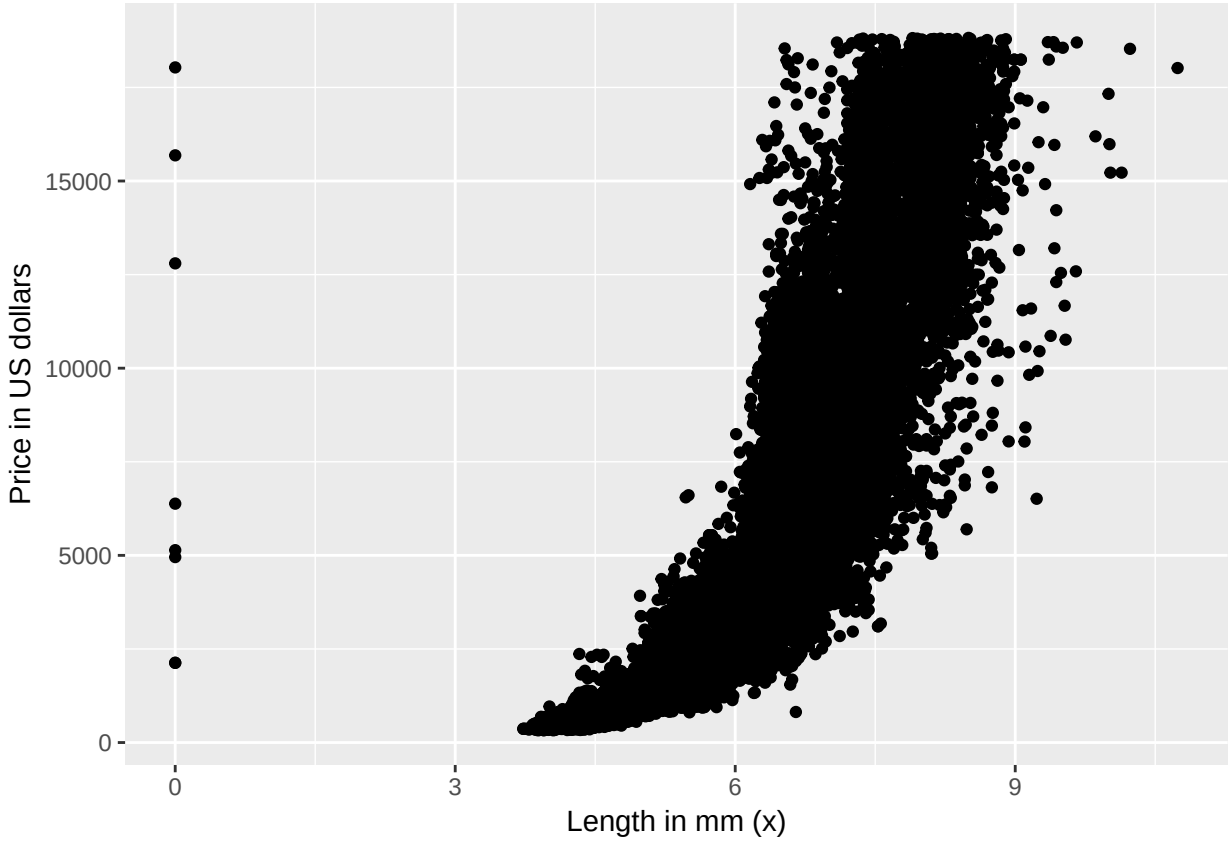
First, I load the data and get a short summary on what I have in it.

```
#Load the ggplot package
library(ggplot2)

#Load the diamonds dataset
data("diamonds")

#Getting a summary from the dataset
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
##
```

Now I create the scatterplot of price vs x. I use `geom_point()` function to get the scattered format from my plot.

```
ggplot(aes(x = x, y = price),
       data = diamonds) +
  xlab('Length in mm (x)') +
  ylab('Price in US dollars') +
  geom_point()
```



Observations:

- In general the majority length of the diamond is concentrated somewhere above 3mm up to 9mm
- There are a few outliers with length of zero, and a few more with length above 9mm
- The price goes higher as the length of the diamond increases. The distribution looks exponential. The change in price is more steep between 0-5000 dollars, which is for lengths between 3mm-7mm, and from there the change seems to be much slower

Some Correlations:

What is the correlation between price and x (length in mm)?

```
with(diamonds, cor.test(diamonds$price, diamonds$x, method = 'pearson'))
```

```
##
##  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
```

What is the correlation between price and y (width in mm)?

```
with(diamonds, 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
```

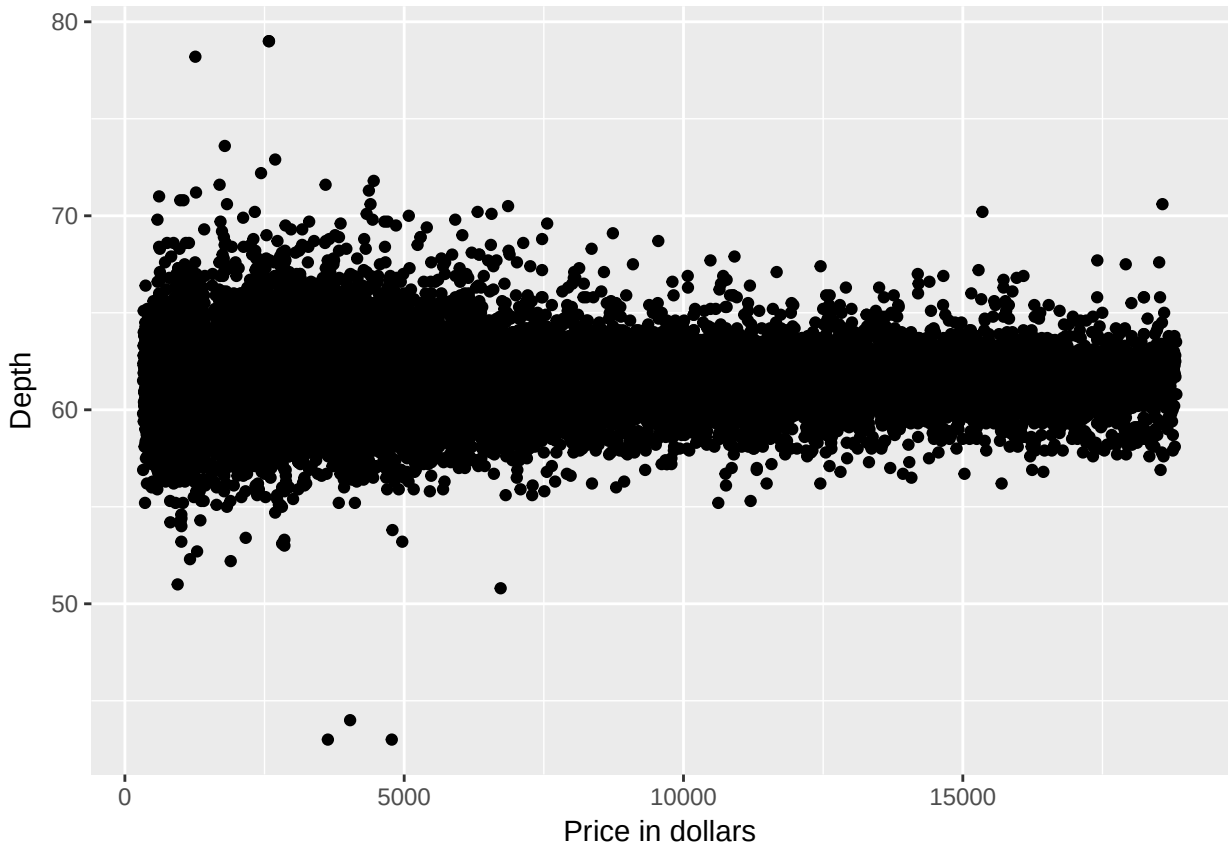
What is the correlation between price and z (depth in mm)?

```
with(diamonds, 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
```

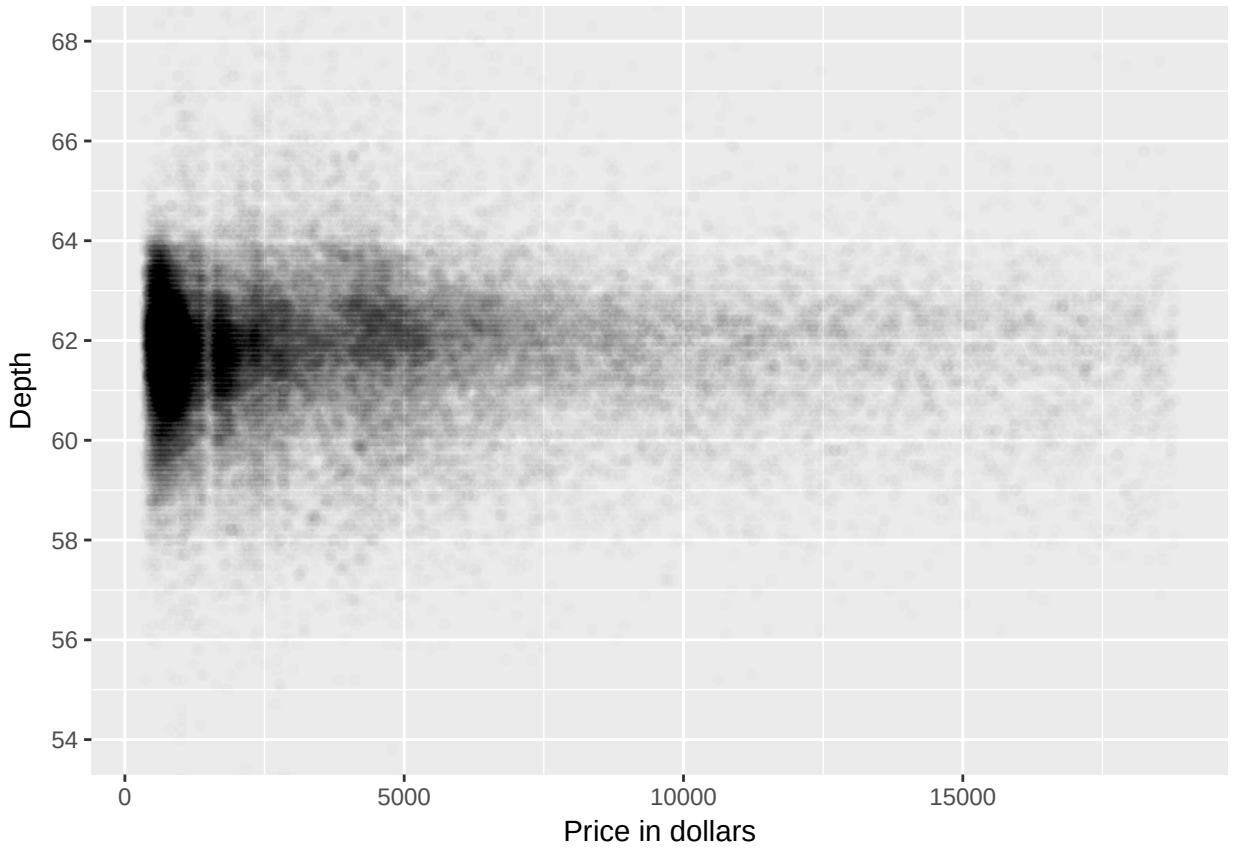
A Simple Scatterplot of Price vs Depth

```
ggplot(aes(x = price, y = depth),
       data = diamonds) +
  xlab('Price in dollars') +
  ylab('Depth') +
  geom_point()
```



To get a better look of this plot and play around with it a bit, I want to change the code to make the transparency of the points to be 1/100 of what they are now and mark the x-axis every 2 units. - The transparency of 1/100 means that for every 100 dots on the scatterplot, 1 will be shown - Marking the x-axis every 2 unit is with regards to how I want to scale the x-axis

```
ggplot(aes(x = price, y = depth),
       data = diamonds) +
  xlab('Price in dollars') +
  ylab('Depth') +
  geom_point(alpha = 1/100) +
  scale_y_continuous(breaks = seq(0, 80, 2)) +
  coord_cartesian(ylim = c(54, 68)) #To zoom in on the plot on the y-axis
```



Based on the scatterplot most diamonds are between 58-64 value of depth.

Some Correlations:

What is the correlation of depth vs price?

```
with(diamonds, cor.test(diamonds$price, diamonds$depth), method = 'pearson')
```

```
##
##  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
```

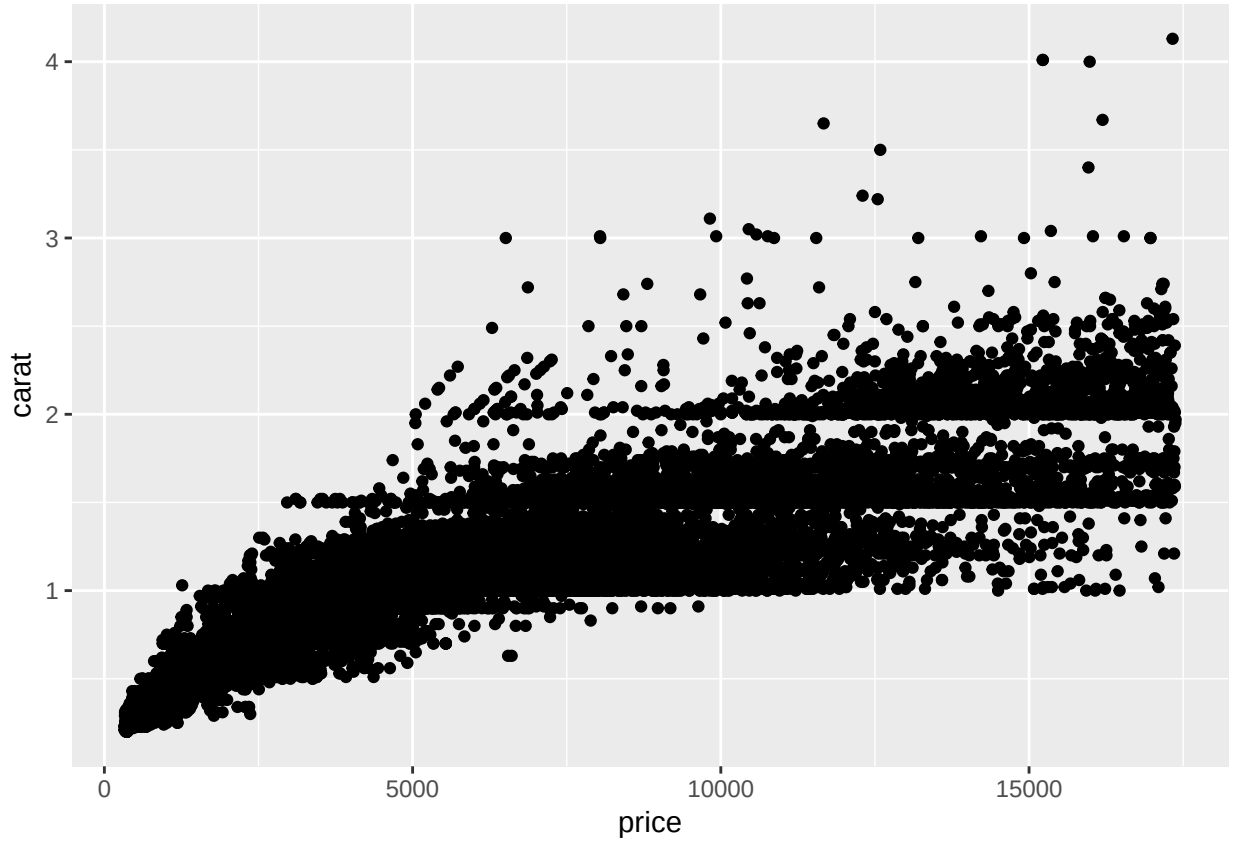
Based on the correlation coefficient, would you use depth to predict the price of a diamond?

My response to this question would be no. The correlation coefficient -0.01 is too small to be able to give insights with regards to predicting one variable based on the other.

A Simple Scatterplot of Price vs Carat

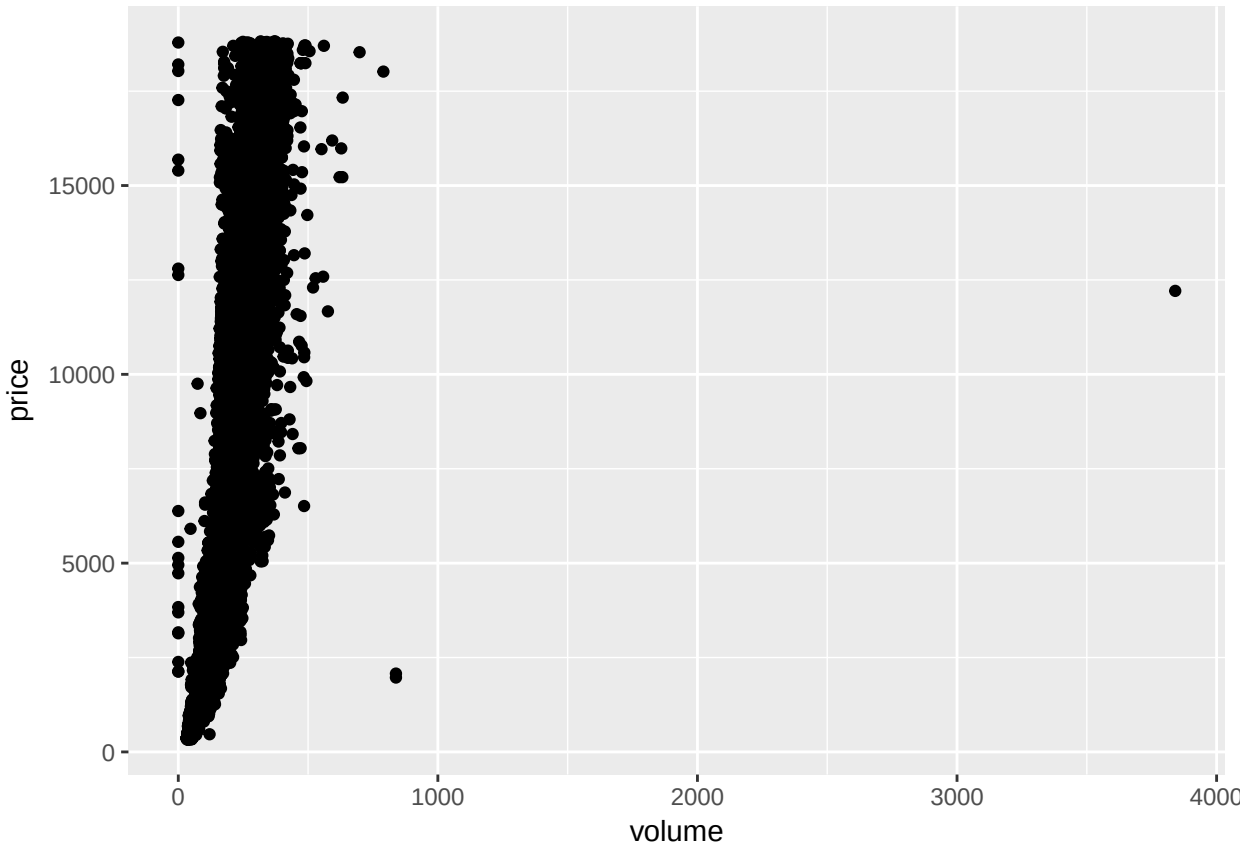
With this plot, I also omit the top 1% of price and carat values.

```
ggplot(aes(x = price, y = carat),
       data = subset(diamonds, price < quantile(diamonds$price, 0.99),
                     carat < quantile(diamonds$carat, 0.99))) +
  geom_point()
```



Scatterplot of Price vs. Volume (x x y x z)

```
#Calculating a rough estimation of volume based on length, width and depth of the diamong
diamonds$volume <- diamonds$x * diamonds$y * diamonds$z

ggplot(aes(x = volume, y = price),
       data = diamonds) +
  geom_point()
```



** Observations **

- There are two outliers which are significantly placed for a volum close to 1000. There is one more significant outlier with a volume close to 4000.
- There are also a few outliers with a volume of zero (this goes back to any of the x, y, or z variables being zero which makes the multiplication zero)
- The relationship between volume and price seems to be exponential– as volume increases, the price increases as well but with a much faster speed

Some Correlations:

What is the correlation of price and volume, excluding the outliers that are either zero or greater than 800?

To calculate the correlation, I create a new dataframe which contains the volume without the outliers mentioned in the question. This makes my job easier, later when I want to see the correlation between price and volume.

I name the dataframe 'diamonds_volume_without_outliers'. This dataframe will have the same number of variables as the diamonds dataset (i.e. 12), but will have fewer observations, since I am excluding any volume that is zero or above 800.

```
#
diamonds_volume_without_outliers <- subset(diamonds, volume > 0 & volume < 800)

with(diamonds, cor.test(diamonds_volume_without_outliers$price, diamonds_volume_without_outliers$volume
```

```
##
##  Pearson's product-moment correlation
```

```
##
## data:  diamonds_volume_without_outliers$price and diamonds_volume_without_outliers$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
```

More Adjusments

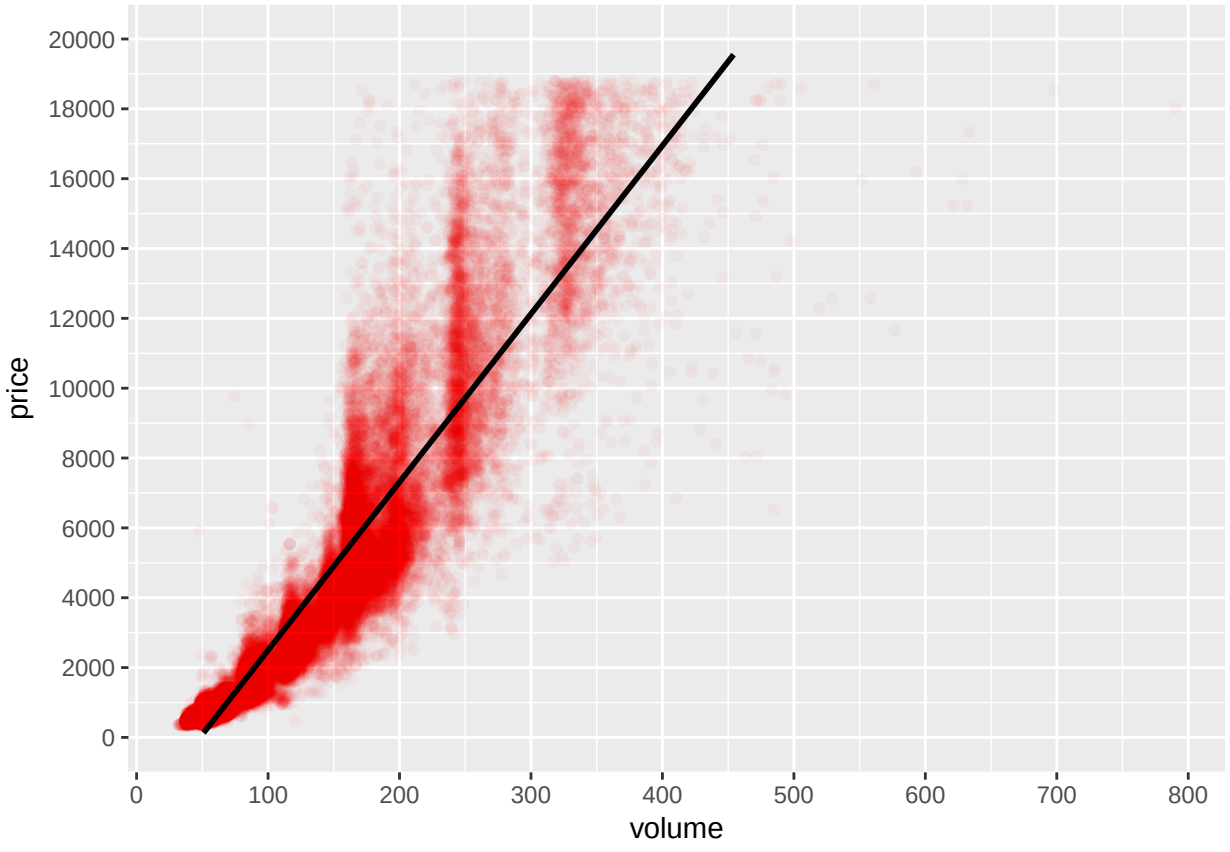
After subseting the data to exclude volumes that are zero or greater than 800, I want to adjust the transparency of the points and add a linear model to the plot.

I will also scale the x and y axis to show part of the plot that contains scatters (aka. points).

I use geom_smooth(method = 'lm') to add a linear model to the plot and I chose the color red for it to be more visible on black.

```
ggplot(aes(x = volume, y = price),
       data = diamonds_volume_without_outliers) +
  scale_y_continuous(limits = c(0, 20000), breaks = seq(0, 20000, 2000)) +
  scale_x_continuous(breaks = seq(0, 800, 100)) +
  geom_point(alpha = 1/30, color = 'red') +
  geom_smooth(method = 'lm', color = 'black')
```

```
## Warning: Removed 37 rows containing missing values (geom_smooth).
```



Is This Plot a Suitable Model to Estimate the Price od Diamonds?

Although the correlation value that I calculated for price vs. volume is 0.92 which is a high number, using a plot with a linear model is not helpful to estimate the price of the diamonds.

Mean Price by Clarity

I create a new dataframe and group the diamonds using their clarity, and calculate their mean, median, minimum and maximum values:

```
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
```

```
diamondsByClarity <- diamonds %>%
  group_by(clarity) %>%
  summarise(mean_price = mean(price),
            median_price = median(price),
            min_price = min(price),
            max_price = max(price),
            n = n()) #number of diamonds with a specific clarity
head(diamondsByClarity)
```

```
## # A tibble: 6 × 6
##   clarity mean_price median_price min_price max_price     n
##     <ord>      <dbl>        <dbl>     <int>     <int> <int>
## 1      I1   3924.169         3344       345     18531   741
## 2     SI2   5063.029         4072       326     18804  9194
## 3     SI1   3996.001         2822       326     18818 13065
## 4     VS2   3924.989         2054       334     18823 12258
## 5     VS1   3839.455         2005       327     18795  8171
## 6    VVS2   3283.737         1311       336     18768  5066
```

Bar Charts of Mean Price

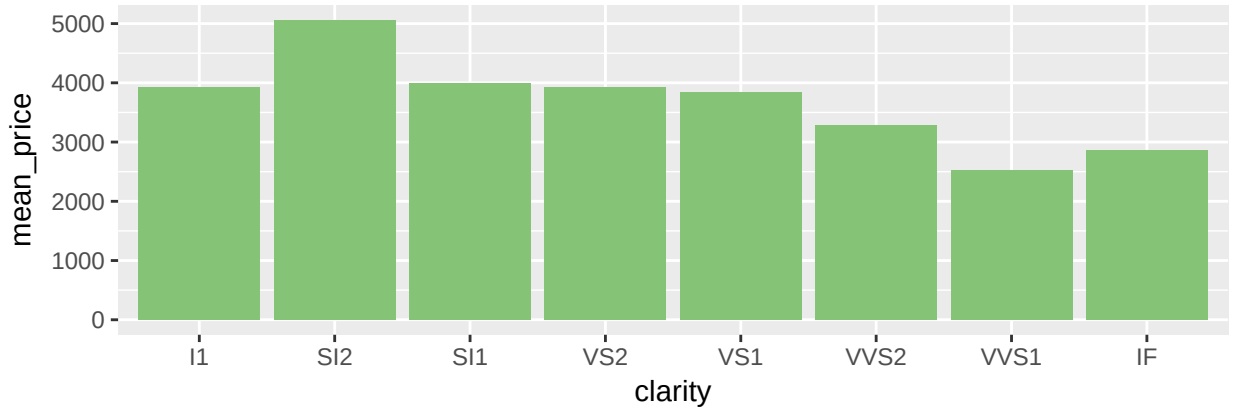
I created summary data frames with the mean price by clarity and color

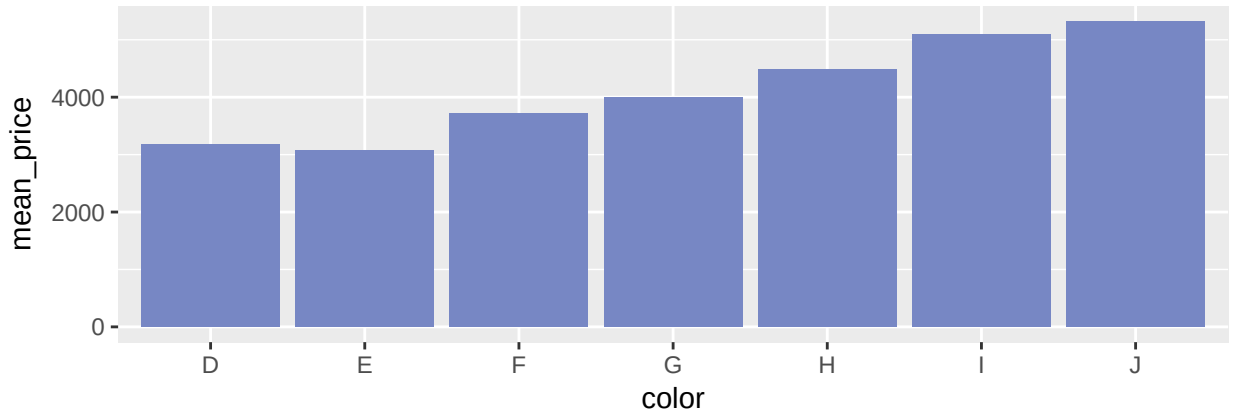
```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
#By clarity
diamonds_by_clarity <- group_by(diamonds, clarity)

#For each category of clarity, what is the mean price?
diamonds_mp_by_clarity <- summarise(diamonds_by_clarity, mean_price = mean(price))

p1 <- ggplot(aes(x = clarity, y = mean_price),
             data = diamonds_mp_by_clarity) +
  geom_bar(stat = 'identity', fill = I('#85C477'))

#By color
diamonds_by_color <- group_by(diamonds, color)

#For each category of color, what is the mean price?
diamonds_mp_by_color <- summarise(diamonds_by_color, mean_price = mean(price))

p2 <- ggplot(aes(x = color, y = mean_price),
             data = diamonds_mp_by_color) +
  geom_bar(stat = 'identity', fill = I('#7787C4'))

grid.arrange(p1, p2, ncol = 1)
```





** Observations **

- Regarding the clarity, seems the best clarity has a lower mean price than the worst calrity. SI2 with a low clarity has the largest mean price while clarity wise it is not on a good level.
- Regarding the color, seems the best color (D) also has the lowest mean price after E. The worst color

(J) has the highest mean price.

With these observations, it looks odd how diamonds with bad clarity and bad colors have higher mean prices.

Gapminder Data

The Gapminder website contains over 500 data sets with information about the world's population. My task will be to download a data set of my choice, and create 2-5 plots based on its data.

I decide to go with the data about female students who are out of primary school. I downloaded the main file in xlsx, and converted it to CSV to read it in R.

```
getwd()
```

```
## [1] "/Users/nazaninmirarab/Desktop/Data Science/P4"
```

```
female_out_of_school <- read.csv('data.csv', header = T, check.names = F)
```

fter executing the command above, I have 'female_out_of_school' dataset created in my environment.

There are many values in the dataset marked as '..' and I want to change them to NA, so that I can later filter them out of the actual values when I am creating plots.

I also name the first column in the dataset 'Country' since it is representing the countries.

```
#Replacing the '..' value with NA
female_out_of_school[female_out_of_school == '..'] = NA

#Changing the name of the first column to 'country'
colnames(female_out_of_school)[1] <- 'Country'
```

I do a little bit more cleanup on the original dataset. In my dataset, the values are between years 1999-2006. So I want to have a table with only these years present.

Also, I want to change the original table to a table where there are three columns: country, year, and count. Country and year are self-explanatory. Count will be the number of female students out of school for the specific year and country.

To do this tidying up, I use the 'tidyr' package.

```
#install.packages("tidyr")
library(tidyr)

#Creating a new dataset 'female_out_gather' with 3 columns: country, year and count
female_out_gather <- gather(female_out_of_school, year, count, 2:ncol(female_out_of_school))
```

```
## Warning: attributes are not identical across measure variables; they will
## be dropped
```

```
#Extracting years 1999-2006 from the dataset as only these years have meaningful values in them
female_out_gather_1999_2006 <- subset(female_out_gather, year >= 1999, year <= 2006)
```

For getting plots, I use my 'female_out_gather_1999_2006' dataset that I have cleaned up already. I want to get a plot from the data, excluding the NA values.

One way to show to plots would be to group the data by years (from 1999 to 2006) and see how the distribution looks like.

Here are the steps I perform to create my new dataframe dropout_by_user:

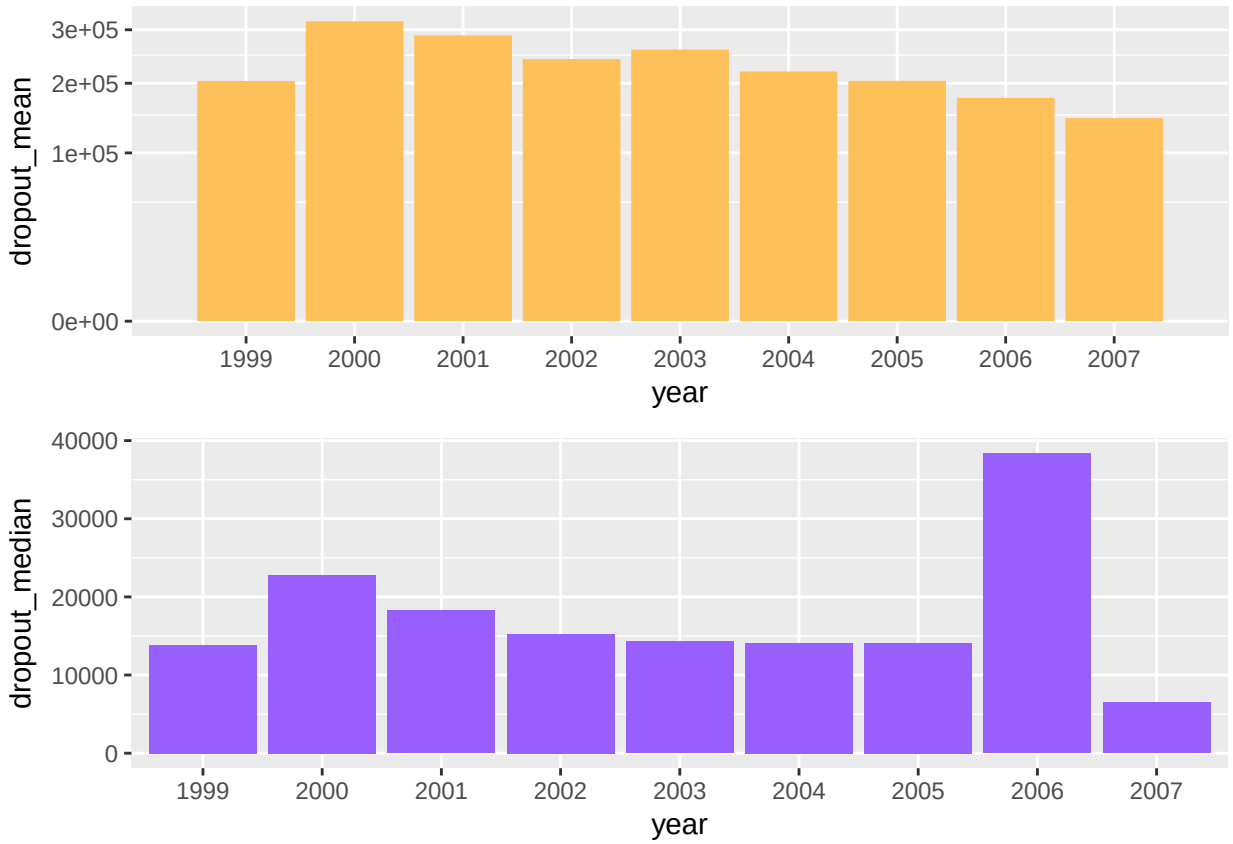
- Create a new dataframe dropout_by_user and start grouping the data from female_out_gather_1999_2006 to that dataframe
- Change the 'count' variable from type = 'character' to type = 'numeric'. This will allow me to perform mathematical analysis on this variable
- Filter those counts that are NA
- Group the data by year
- Take the mean and media of each year, along with the total count of people in that year
- Print the first few rows to see how the data looks

```
dropout_by_year <- female_out_gather_1999_2006 %>%
  mutate(count = as.numeric(count)) %>%
  filter(!is.na(count)) %>%
  group_by(year) %>%
  summarise(dropout_mean = mean(count),
            dropout_median = median(count),
            n = n())

head(dropout_by_year)
```

```
## # A tibble: 6 × 4
##    year dropout_mean dropout_median     n
##   <chr>        <dbl>          <dbl> <int>
## 1  1999     203830.9          13834   132
## 2  2000     317771.8          22791   128
## 3  2001     288614.7          18252   130
## 4  2002     242759.2          15231   131
## 5  2003     260660.6          14314   134
## 6  2004     220189.0          14110   135
```

I managed to successfully create the dropout_by_year dataframe. Now I want to get some plots out of it.

```
library(gridExtra)

pl1 <- ggplot(aes(x = year, y = dropout_mean),
              data = dropout_by_year) +
  geom_bar(stat = 'identity', fill = I('#FFC05A')) +
  coord_trans(y = 'sqrt')

pl2 <- ggplot(aes(x = year, y = dropout_median),
              data = dropout_by_year) +
  geom_bar(stat = 'identity', fill = I('#995FFE'))

grid.arrange(pl1, pl2, ncol = 1)
```



** Observations **

- The mean of school dropouts has had its lowest in year 2007.
- There is a peak of dropout seen from year 1999 to 2000, and from 2002 to 2003, but besides these, it seems after 2003, the average number of primary school dropouts among female students has dropped.
- In the median plot (purple), year 2007 has the lowest median.
- From year 1999 to 2000 there is noticable peak in the median, but median decreases after 2000 onwards, with one exception in 2006. It looks the median number of dropouts in 2006 has had a significant steep which doesn't really go with the previous plot I had regarding the average number of students.