**Load and Explore Data**

1-I started by importing libraries and loading the diamonds csv. I saw that the columns x,y, and z stand for length, width, and depth per Kaggle. To make things more intuitive, I renamed those columns length, width, and deep. I had to use deep instead of depth because there was another column already named depth.

*#import libraries and rename columns*

*import pandas as pd*

*from matplotlib import pyplot as plt*

*import seaborn as sns*

*import numpy as np*

*df=pd.read\_csv(r'C:\Users\rryan\Desktop\Personal\Python\diamonds.csv')*

*df.rename(columns={'x': 'length'}, inplace=True)*

*df.rename(columns={'y': 'width'}, inplace=True)*

*df.rename(columns={'z': 'deep'}, inplace=True)*

2-Columns length, width, and deep on its own doesn’t mean much. But the product of all 3 results in the diamond volume. I created a new column named volume which is the product of length, width, and deep.

*#add column volume*

*df['volume']=df['length']\*df['width']\*df['deep']*

3-I checked the data and found there are 53,940 rows and 12 columns. Price is an integer and carat, cut, and color are objects. The rest are float. There was no missing data.

*#check data*

*print(df.shape)*

*print(df.dtypes)*

*print(df.isnull().sum())*

**Clean Data (remove outliers and rows with 0s)**

1-After executing .describe() I can see price has a standard deviation that is larger than the mean (3,989>3,932). Also, there are values for length, width, and deep with 0s. That is incorrect since no diamond can have a size of 0. I created a new dataframe named dfclean and in dfclean, removed rows where price is an outlier and/or volume is 0. Note that it’s better to remove values of 0 in volume since volume is the product of length, width, and deep. In other words, a value of 0 in volume implies any one and/or all length, width, or deep could be 0.

*#check stats*

*print(df.describe())*

Unnamed: 0 carat depth table price length width \

count 53,940.00 53,940.00 53,940.00 53,940.00 53,940.00 53,940.00 53,940.00

mean 26,970.50 0.80 61.75 57.46 **3,932.80** 5.73 5.73

std 15,571.28 0.47 1.43 2.23 **3,989.44** 1.12 1.14

min 1.00 0.20 43.00 43.00 326.00 0.00 0.00

25% 13,485.75 0.40 61.00 56.00 950.00 4.71 4.72

50% 26,970.50 0.70 61.80 57.00 2,401.00 5.70 5.71

75% 40,455.25 1.04 62.50 59.00 5,324.25 6.54 6.54

max 53,940.00 5.01 79.00 95.00 18,823.00 10.74 58.90

deep volume

count 53,940.00 53,940.00

mean 3.54 129.85

std 0.71 78.25

min 0.00 **0.00**

25% 2.91 65.14

50% 3.53 114.81

75% 4.04 170.84

max 31.80 3,840.60

*#in a new dataframe dfclean, remove price outliers*

*dfclean=df.copy()*

*meanprice=dfclean['price'].mean()*

*stdprice=dfclean['price'].std()*

*topprice=meanprice + stdprice\*1.96*

*botprice=meanprice - stdprice\*1.96*

*dfclean = dfclean.drop(dfclean[dfclean['price'] > topprice].index)*

*dfclean = dfclean.drop(dfclean[dfclean['price'] < botprice].index)*

*#in dfclean, remove rows where volume has a vlue of 0*

*dfclean.drop(dfclean.loc[dfclean['volume']==0].index, inplace=True)*

2-I checked stats in dfclean after removing price outliers and rows with volume as 0. The standard deviation of price was reduced to 2,740 which is less than the mean of 3,142. There were no rows where volume is 0. This confirms the appropriate rows were removed.

Unnamed: 0 carat depth table price length width \

count 50,288.00 50,288.00 50,288.00 50,288.00 50,288.00 50,288.00 50,288.00

mean 27,063.33 0.72 61.76 57.42 **3,142.05** 5.58 5.58

std 16,116.94 0.38 1.43 2.23 **2,740.52** 0.99 0.99

min 1.00 0.20 43.00 43.00 326.00 3.73 3.68

25% 12,579.75 0.38 61.10 56.00 909.00 4.67 4.68

50% 28,793.50 0.68 61.80 57.00 2,145.00 5.58 5.57

75% 41,365.25 1.01 62.50 59.00 4,656.00 6.42 6.42

max 53,940.00 3.65 79.00 79.00 11,748.00 9.54 31.80

deep volume

count 50,288.00 50,288.00

mean 3.45 117.55

std 0.63 61.70

min 1.07 **31.71**

25% 2.88 63.14

50% 3.42 108.82

75% 3.98 163.73

max 31.80 838.50

**Binning Data**

1-There are too many values for columns volume and carat. I put those data values into binned categories, for volume 0-900 mm cubed in increments of 50 mm cubed and carat from 0-4 carats in increments of 0.5 carats. Volume has 6 bins and carats has 7 bins.

*#bin volume and carat*

*bins = [0,50,100,150,200,250,900]*

*names=['extra small','small','medium','medium large','large','extra large']*

*dfclean['Diamond Vol.']=pd.cut(dfclean['volume'],bins,labels=names)*

*bins1 = [0,1,1.5,2,2.5,3,3.5,4]*

*names1=['0-1c','1-1.5c','1.5-2c','2-2.5c','2.5-3c','3-3.5c','3.5-4c']*

*dfclean['Diamond Carat']=pd.cut(dfclean['carat'],bins1,labels=names1)*

**Groupby Data**

1-I used the .groupby() function to see what factors impacted price. The volume of the diamond had a substantial impact on price. The average price of a diamond nearly doubled for every 50 mm cubed increase in volume until over 150 mm cubed (medium large) and even then; average price still increased by ~50% from above 150 mm cubed to 900 mm cubed.

Diamond Vol.

extra small 617.35

small 1,119.88

medium 2,974.24

medium large 5,632.92

large 7,925.80

extra large 9,456.72

2-There was also a relationship between carats (weight) and price although not quite as sensitive as volume. Diamonds below 1 carat had an average price of 1,785. Diamonds within the range of 1.5-3.5 carats were generally ~$9,000. Diamonds above 3.5 carats had an average price of 11,668. Medium weight diamonds between 1.5-3.5 carat were not very sensitive to price.

Diamond Carat

0-1c 1,785.27

1-1.5c 6,205.62

**1.5-2c 9,211.66**

**2-2.5c 9,509.78**

**2.5-3c 9,529.77**

**3-3.5c 9,929.83**

3.5-4c 11,668.00

3-Grouping by color and cut, there does not appear to be a substantial impact on price. In fact, Very Good cut had a lower average price than Good. There must be other stronger factors influencing price.

cut

Fair 3,664.21

Good 3,274.99

Ideal 2,797.52

Premium 3,517.15

Very Good 3,224.26

color

D 2,672.68

E 2,603.25

F 3,065.45

G 3,223.41

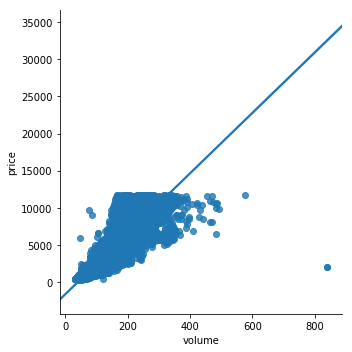
H 3,547.87

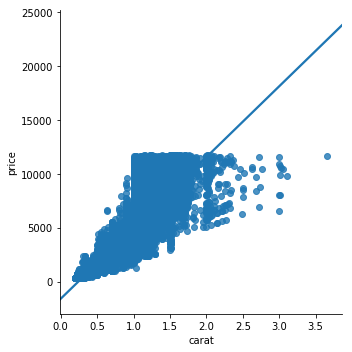
I 3,722.19

J 3,995.23

**Graph Data**

1-I made a scatter plot graph of diamond volume and diamond carat vs. price.





2- I made a violinplot of diamond volume and diamond carat vs. price.

