VNR Vignana Jyothi Institute of Engineering and Technology

(Affiliated to J.N.T.U, Hyderabad)

Bachupally(v), Hyderabad, Telangana, India.

ANALYSIS OF DIAMOND DATASET

A course based project submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

CSE-CYBER SECURITY

Submitted by

B.Reshma (21071A6209)

Ch.Sadhwick (21071A6210)

Under the guidance of

Mrs.E.Lalitha Assistant Professor

Dept. of CSE-CS,DS,AI&DS

VNR Vignana Jyothi Institute of Engineering and Technology
(Affiliated to J.N.T.U, Hyderabad)
Bachupally(v), Hyderabad, Telangana, India.

CERTIFICATE

This is to certify that **B.Reshma (21071A6209), Ch.Sadhwick (21071A6210)** have completed their course based project work at CYBER SECURITY Department of VNR VJIET, Hyderabad entitled **"Analysis of diamond dataset"** in partial fulfilment of the requirements for the award of B.Tech degree during the academic year 2022-2023. This work is carried out under my supervision and has not been submitted to any other University/Institute for award of any degree/diploma.

Mrs.E.Lalitha

Assistant Professor

Department of
CSE-CYS,DS,AI&DS

VNRVJIET

Dr.M.Raja Sekhar

Professor and HOD

Department of
CSE-CYS,DS,AI&DS

VNRVJIET

DECLARATION

This is to certify that our project report titled "**Analysis of diamond dataset**" submitted to Vallurupalli Nageswara Rao Institute of Engineering and Technology in complete fulfilment of requirement for the award of Bachelor of Technology in CSE-Cyber Security is a bonafide report to the work carried out by us under the guidance and supervision of **Mr.E.Lalitha** , Assistant Professor, Department of CSE-Cyber Security, Vallurupalli Nageswara Rao Institute of Engineering and Technology. To the best of our knowledge, this has not been submitted in any form to other university or institution for the award of any degree or diploma.

B.Reshma (21071A6209), **Ch.Sadhwick** (21071A6210)

ACKNOWLEDGEMENT

Behind every achievement lies an unfathomable sea of gratitude to those who activated it, without which it would ever never have come into existence. To them I lay the words of gratitude imprinting within us.

VNRVJIET has helped us transform ourselves from mere amateurs in the field of Computer Science into skilled engineers capable of handling any given situation in real time. I am are highly indebted to the institute for everything that it has given us.

We would like to express our gratitude towards the principal of our institute, **Dr.ChallaDhanunjaya Naidu** and the Head of the CSE-CYS,DS,AI&DS Department, **Dr.M.Raja Sekhar** for their kind co-operation and encouragement which helped us complete the project in the stipulated time.

Although we had spent a lot of time and put in a lot effort into this project, it would not have been possible without the motivating support and help of our project guide **Mr.E.Lalitha .** We thank her for her guidance, constant supervision and for providing necessary information to complete this project. Our thanks and appreciations also go to all the faculty members, staff members of VNRVJIET, and all our friends who have helped me put this project together.

Scheme of Course Based Project

Name of the course : Course Based project
Year / Semester : II-B.Tech I-semester
Project Title : Analysis of diamond dataset

Done by : B.Reshma (21071A6209)
Ch.Sadhwick (21071A6210)

Project Objectives : The project objectives for analysis of a diamond dataset include understanding the data, exploring relationships between variables, developing predictive models, evaluating model performance, and providing insights and recommendations based on the results. The goal is to gain a better understanding of the factors that influence the value of a diamond and provide useful information for business decisions and strategy.

Description : The analysis of a diamond dataset involves exploring the characteristics and relationships between variables, developing predictive models to determine the value of a diamond based on its features, evaluating the performance of the models, and providing insights and recommendations based on the results of the analysis.

ABSTRACT

This project involves the analysis of a diamond dataset to gain insights into the factors that influence the value of a diamond. The objectives of the project include understanding the characteristics of the dataset, exploring relationships between variables, developing predictive models, evaluating model performance, and providing insights and recommendations based on the results of the analysis. The dataset includes information such as carat weight, cut, color, clarity, and price. Machine learning techniques will be used to develop models that can predict the value of a diamond based on its features. The results of the analysis will be used to provide insights and recommendations to stakeholders, such as identifying which features are most important in determining the value of a diamond and which types of diamonds are more likely to be undervalued in the market. The project aims to provide useful information for business decisions and strategy related to the diamond industry.

1. Explore Dataset & Examine what Features affect the Price of Diamonds
   - 1.1 Importing Libraries
   - 1.2 Extract Dataset
   - 1.3 Features
   - 1.4 Drop the 'Unnamed: 0' column as we already have Index
   - 1.5 Examine Nan Values
   - 1.6 Dropping Rows with Dimensions 'Zero'
   - 1.7 Scaling of all Features
2. Correlation Between Features
3. Visualization Of Features Through Graphs
4. Feature Engineering
5. Linear Regression Model
6. Finding R2 Value
7. Predicting price of diamond
8. KMeans clustering

INDEX

Contents **PageNo**

CHAPTER 1
INTRODUCTION

1.1 Introduction

What are Diamonds?

- **Diamonds are the Precious stone consisting of a clear and colorless Crystalline form of pure carbon.**
- **They are the hardest Gemstones known to man and can be scratched only by other Diamonds.**

A diamond is one of the most expensive stones. The price of diamonds varies irrespective of the size because of the factors affecting the price of a diamond.

Why are Diamonds so Valuable?

- **Whether it is a Rare book, a fine bottle of Scotch, or a Diamond, something that is Rare and Unique is often expensive.**
- **But what makes it truly Valuable is that this Rarity coincides with the desire of many to possess it. ;)**
- **Diamonds are Rare because of the Incredibly powerful forces needed to create them.**
- **And** therefore, **Diamonds** are considered to be **Very Costly.**

CHAPTER 2
FEATURES OF DIAMOND DATASET

- **Carat** : Carat weight of the Diamond.
- **Cut** : Describe cut quality of the diamond.
  - **Quality in increasing order Fair, Good, Very Good, Premium, Ideal .**
- **Color** : Color of the Diamond.
  - **With D being the best and J the worst.**
- **Clarity** : Diamond Clarity refers to the absence of the Inclusions and Blemishes.
  - **(In order from Best to Worst, FL = flawless, I3= level 3 inclusions) FL, IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1, I2, I3**
- **Depth** : The Height of a Diamond, measured from the Culet to the table, divided by its average Girdle Diameter.
- **Table** : The Width of the Diamond's Table expressed as a Percentage of its Average Diameter.
- **Price** : the Price of the Diamond.
- X : Length of the Diamond in mm.
- Y : Width of the Diamond in mm.
- Z : Height of the Diamond in mm.

Qualitative Features (Categorical) : Cut, Color, Clarity.

Quantitative Features (Numerical) : Carat, Depth , Table , Price , X , Y, Z.

CHAPTER 3

TECHNOLOGIES USED

LANGUAGE: Python

USED: Data analysis, data visualization, in built functions, regression models.

IDE: Jupyter notebook

LIBERARIES:

pandas,numpy,mathplotlib, seaborn, train_test_split from sklearn.model_selection

CHAPTER 4

CODING SNIPPETS

```
import warnings
warnings.filterwarnings('ignore')

# Handle table-like data and matrices :
import numpy as np
import pandas as pd
import math

# Modelling Algorithms :
# Classification
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.cluster import KMeans

# Regression
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor

# Modelling Helpers :
from sklearn.model_selection import train_test_split

# Regression
from sklearn.metrics import mean_squared_log_error,mean_squared_error, r2_score,mean_absolute_error
```

```
# Classification
from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score

#visualization
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.pylab as pylab
import seaborn as sns

# Configure visualisations
%matplotlib inline
mpl.style.use( 'ggplot' )
plt.style.use('fivethirtyeight')
sns.set(context="notebook", palette="dark", style = 'whitegrid' , color_codes=True)
params = {
    'axes.labelsize': "large",
    'xtick.labelsize': 'x-large',
    'legend.fontsize': 20,
    'figure.dpi': 150,
    'figure.figsize': [25, 7]
}
plt.rcParams.update(params)
```

```
dataset=pd.read_csv("diamonds.csv")
dataset.head()
```

	Unnamed: 0	carat	cut	color	clarity	depth	table	price	x	y	z
0	1	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	2	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	3	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	4	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	5	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 53940 entries, 0 to 53939
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  53940 non-null  int64
 1   carat       53940 non-null  float64
 2   cut         53940 non-null  object
 3   color       53940 non-null  object
 4   clarity     53940 non-null  object
 5   depth       53940 non-null  float64
 6   table       53940 non-null  float64
 7   price       53940 non-null  int64
 8   x           53940 non-null  float64
 9   y           53940 non-null  float64
 10  z           53940 non-null  float64
dtypes: float64(6), int64(2), object(3)
memory usage: 4.5+ MB
```

```
dataset.describe()
```

	Unnamed: 0	carat	depth	table	price	x	y	z
count	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000
mean	26970.500000	0.797940	61.749405	57.457184	3932.799722	5.731157	5.734526	3.538734
std	15571.281097	0.474011	1.432621	2.234491	3989.439738	1.121761	1.142135	0.705699
min	1.000000	0.200000	43.000000	43.000000	326.000000	0.000000	0.000000	0.000000
25%	13485.750000	0.400000	61.000000	56.000000	950.000000	4.710000	4.720000	2.910000
50%	26970.500000	0.700000	61.800000	57.000000	2401.000000	5.700000	5.710000	3.530000
75%	40455.250000	1.040000	62.500000	59.000000	5324.250000	6.540000	6.540000	4.040000
max	53940.000000	5.010000	79.000000	95.000000	18823.000000	10.740000	58.900000	31.800000

In [8]:
```
dataset.drop(['Unnamed: 0'] , axis=1 , inplace=True)
dataset.head()
```

Out[8]:

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

In [9]:
```
dataset.shape
```

Out[9]: (53940, 10)

```
In [10]: dataset.isnull().sum()
```

```
Out[10]: carat      0
         cut        0
         color      0
         clarity    0
         depth      0
         table      0
         price      0
         x          0
         y          0
         z          0
         dtype: int64
```

```
#Lets see the diamonds with either height,length or width that are zero
dataset.loc[(dataset['x']==0) | (dataset['y']==0) | (dataset['z']==0)]
```

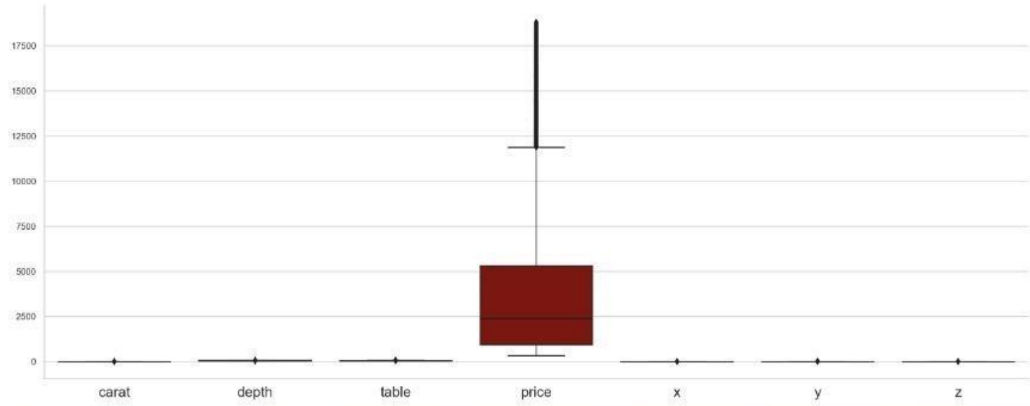
	carat	cut	color	clarity	depth	table	price	x	y	z
2207	1.00	Premium	G	SI2	59.1	59.0	3142	6.55	6.48	0.0
2314	1.01	Premium	H	I1	58.1	59.0	3167	6.66	6.60	0.0
4791	1.10	Premium	G	SI2	63.0	59.0	3696	6.50	6.47	0.0
5471	1.01	Premium	F	SI2	59.2	58.0	3837	6.50	6.47	0.0
10167	1.50	Good	G	I1	64.0	61.0	4731	7.15	7.04	0.0
11182	1.07	Ideal	F	SI2	61.6	56.0	4954	0.00	6.62	0.0
11963	1.00	Very Good	H	VS2	63.3	53.0	5139	0.00	0.00	0.0
13601	1.15	Ideal	G	VS2	59.2	56.0	5564	6.88	6.83	0.0
15951	1.14	Fair	G	VS1	57.5	67.0	6381	0.00	0.00	0.0
24394	2.18	Premium	H	SI2	59.4	61.0	12631	8.49	8.45	0.0
24520	1.56	Ideal	G	VS2	62.2	54.0	12800	0.00	0.00	0.0
26123	2.25	Premium	I	SI1	61.3	58.0	15397	8.52	8.42	0.0
26243	1.20	Premium	D	VVS1	62.1	59.0	15686	0.00	0.00	0.0

```
In [12]: dataset = dataset[(dataset[['x','y','z']] != 0).all(axis=1)]
```

Scaling of all features

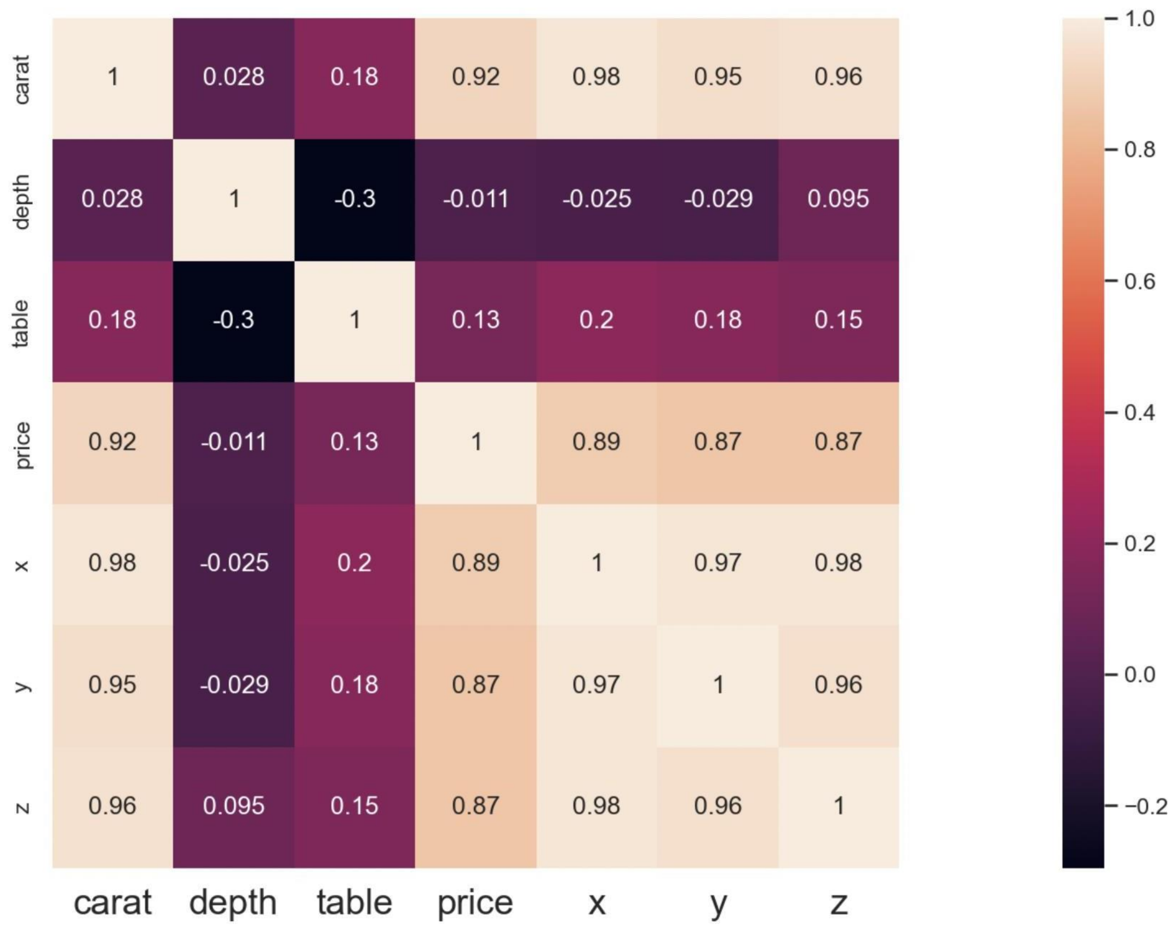
In [14]:
```
sns.factorplot(data=dataset , kind='box' , size=7, aspect=2.5)
```

Out[14]: <seaborn.axisgrid.FacetGrid at 0x1fdf2b627f0>




```
corr = dataset.corr()
sns.heatmap(data=corr, square=True , annot=True, cbar=True)
```

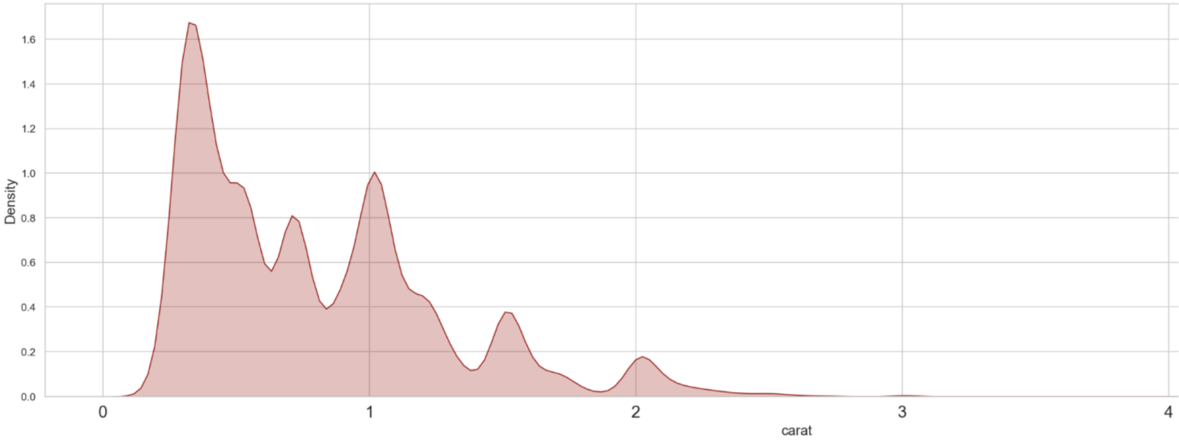
<AxesSubplot:>



1.Carat

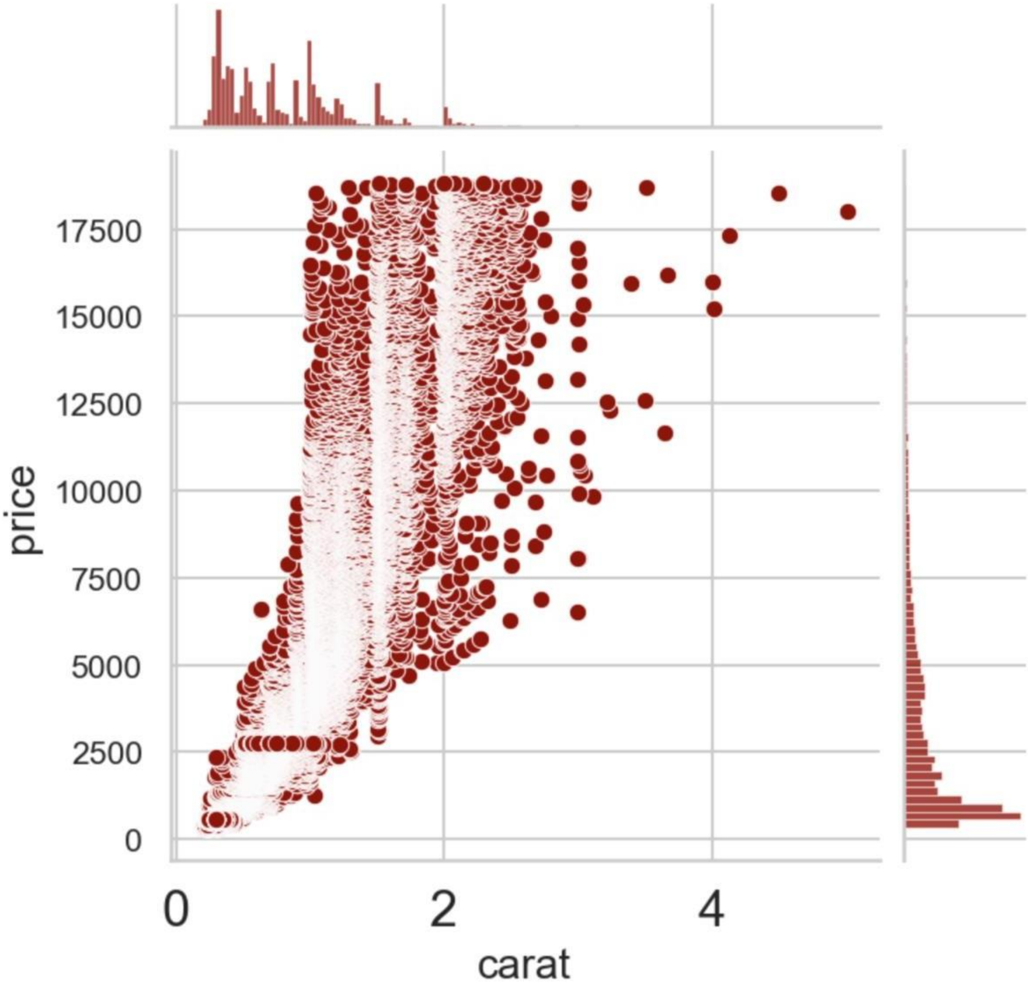
```
sns.kdeplot(dataset['carat'], shade=True , color='r')
```

<AxesSubplot:xlabel='carat', ylabel='Density'>




```
sns.jointplot(x='carat' , y='price' , data=dataset , size=5 ,color='r')
```

<seaborn.axisgrid.JointGrid at 0x1fdf3a79cd0>

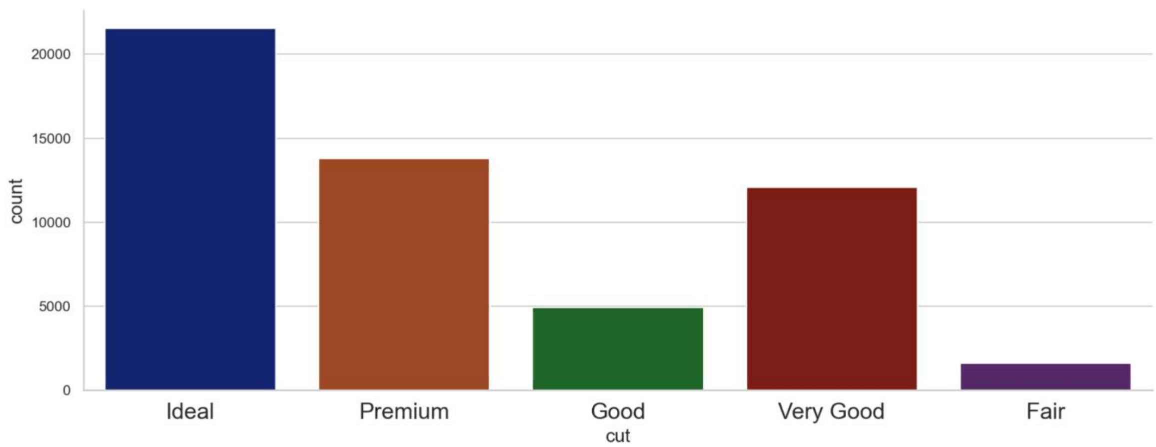


2.Cut

```
sns.factorplot(x='cut', data=dataset , kind='count',aspect=2.5 )
```

Python

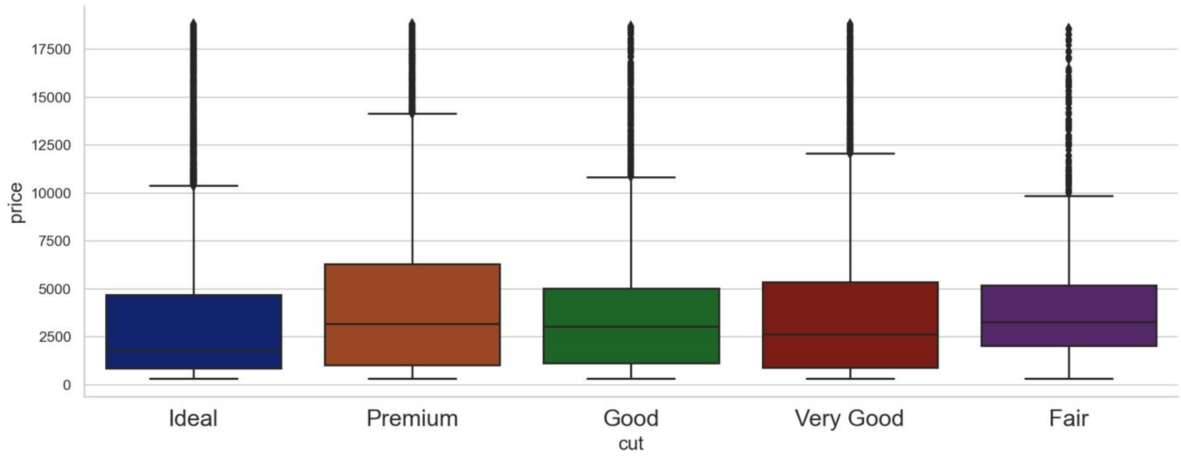
<seaborn.axisgrid.FacetGrid at 0x1fdf39fb7c0>



cut Vs price

```python
sns.factorplot(x='cut', y='price', data=dataset, kind='box' ,aspect=2.5 )
```

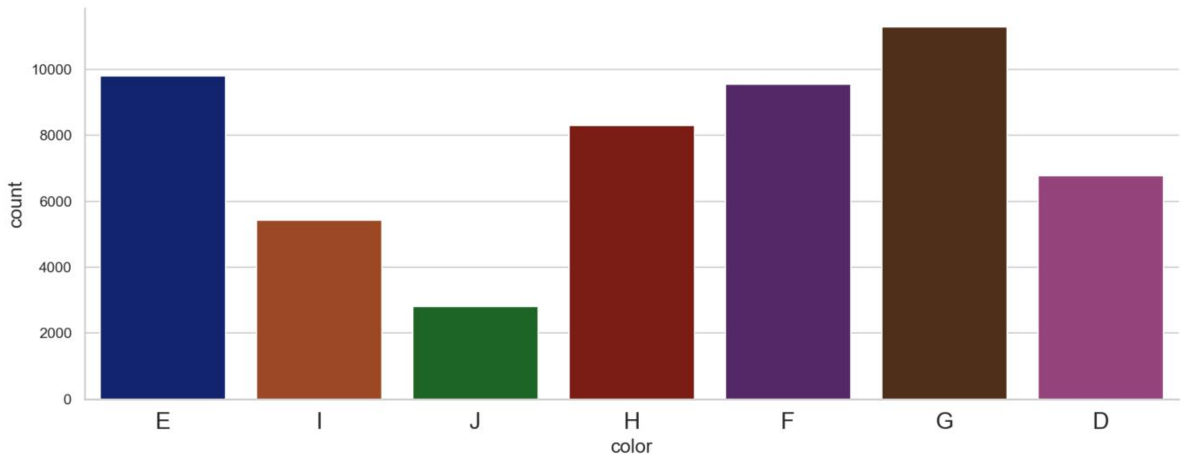
```
<seaborn.axisgrid.FacetGrid at 0x1fdf7823b20>
```



3.Color

```python
sns.factorplot(x='color', data=dataset , kind='count',aspect=2.5 )
```

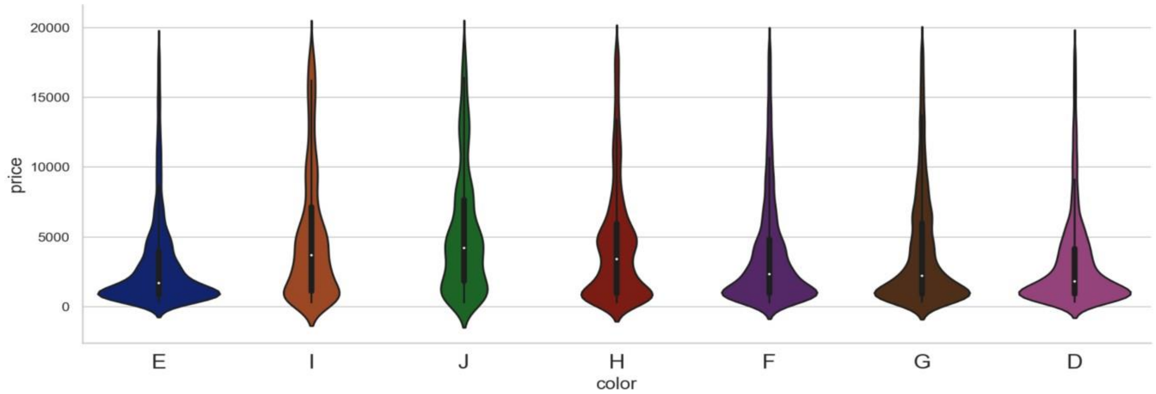
```
<seaborn.axisgrid.FacetGrid at 0x1fdf3c81c70>
```



color vs price

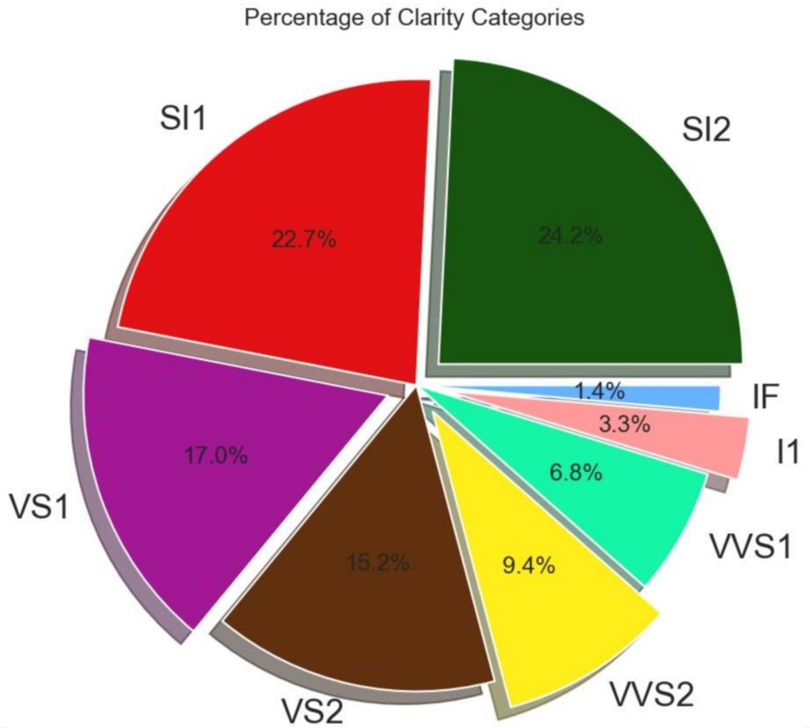
```python
sns.factorplot(x='color', y='price' , data=dataset , kind='violin', aspect=2.5)
```

```
<seaborn.axisgrid.FacetGrid at 0x1fdf7c23e20>
```



4.Clarity

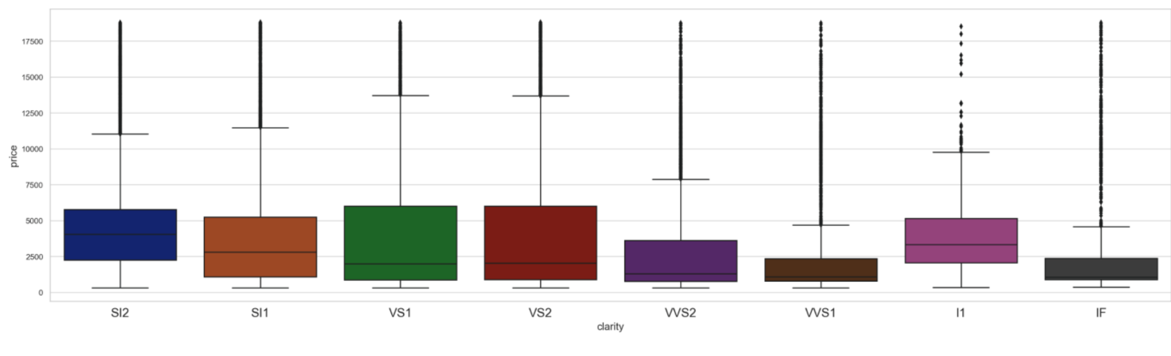
```
labels = dataset.clarity.unique().tolist()
sizes = dataset.clarity.value_counts().tolist()
colors = ['#005400', '#E20E00', '#A00994', '#613205', '#FFED0D', '#16F5A7','#ff9999','#66b3ff']
explode = (0.1, 0.0, 0.1, 0, 0.1, 0, 0.1,0)
plt.pie(sizes, explode=explode, labels=labels, colors=colors,autopct='%1.1f%%', shadow=True, startangle=0)
plt.axis('equal')
plt.title("Percentage of Clarity Categories")
plt.plot()
fig=plt.gcf()
fig.set_size_inches(6,6)
plt.show()
```




```
sns.boxplot(x='clarity', y='price', data=dataset )
```

Python

<AxesSubplot:xlabel='clarity', ylabel='price'>

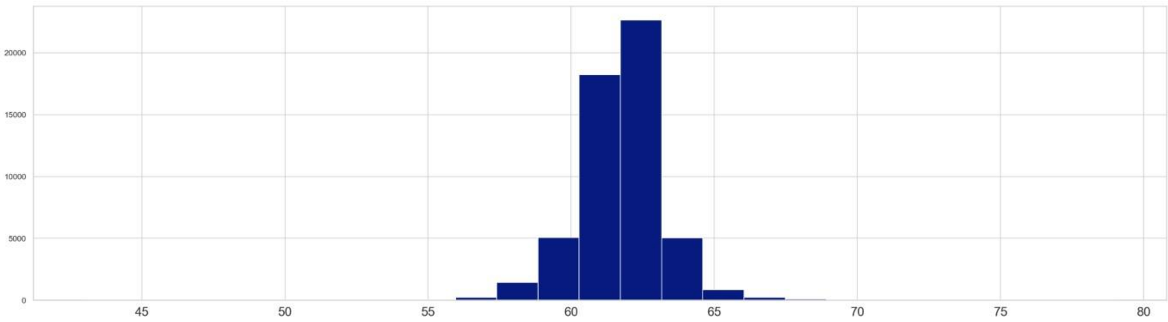


5.Depth

```python
plt.hist('depth' , data=dataset , bins=25)
```

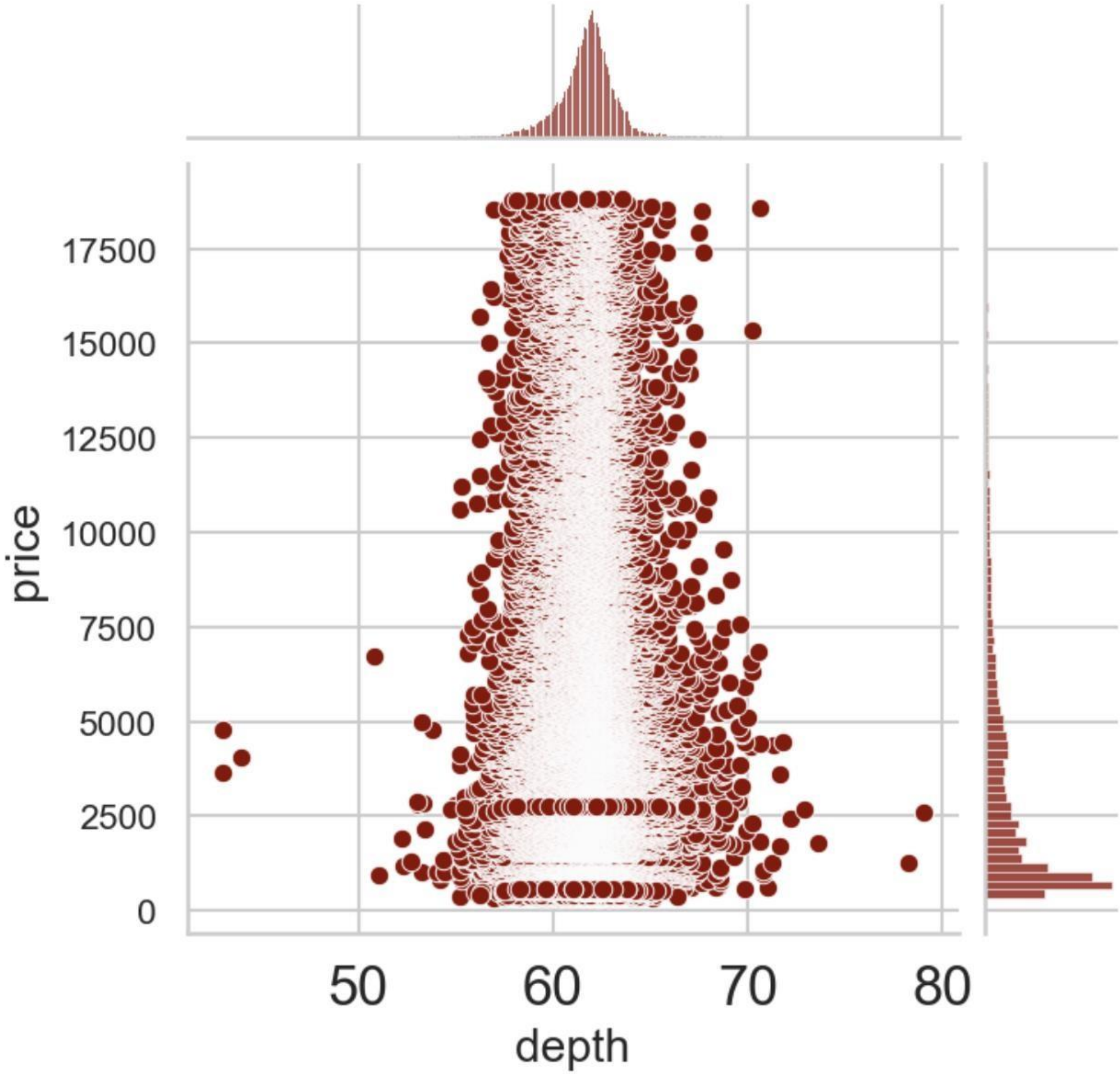
Python

```
(array([3.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
        2.0000e+00, 4.0000e+00, 1.1000e+01, 4.3000e+01, 2.1900e+02,
        1.4240e+03, 5.0730e+03, 1.8242e+04, 2.2649e+04, 5.0330e+03,
        8.5100e+02, 2.3400e+02, 8.7000e+01, 2.7000e+01, 1.1000e+01,
        3.0000e+00, 1.0000e+00, 0.0000e+00, 0.0000e+00, 3.0000e+00]),
 array([43.  , 44.44, 45.88, 47.32, 48.76, 50.2 , 51.64, 53.08, 54.52,
        55.96, 57.4 , 58.84, 60.28, 61.72, 63.16, 64.6 , 66.04, 67.48,
        68.92, 70.36, 71.8 , 73.24, 74.68, 76.12, 77.56, 79.  ]),
 <BarContainer object of 25 artists>)
```



```
sns.jointplot(x='depth' , y='price' , data=dataset , size=5 ,color='r')
```

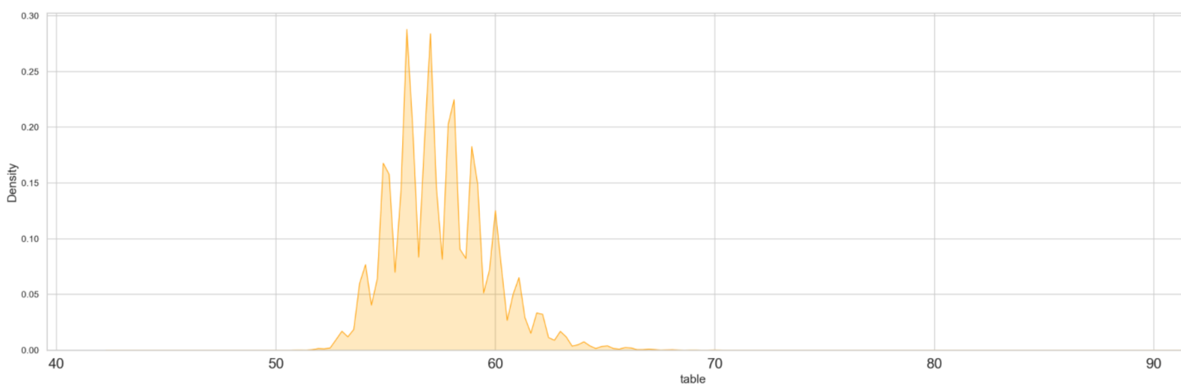
```
<seaborn.axisgrid.JointGrid at 0x1fdf832fa00>
```



6.Table

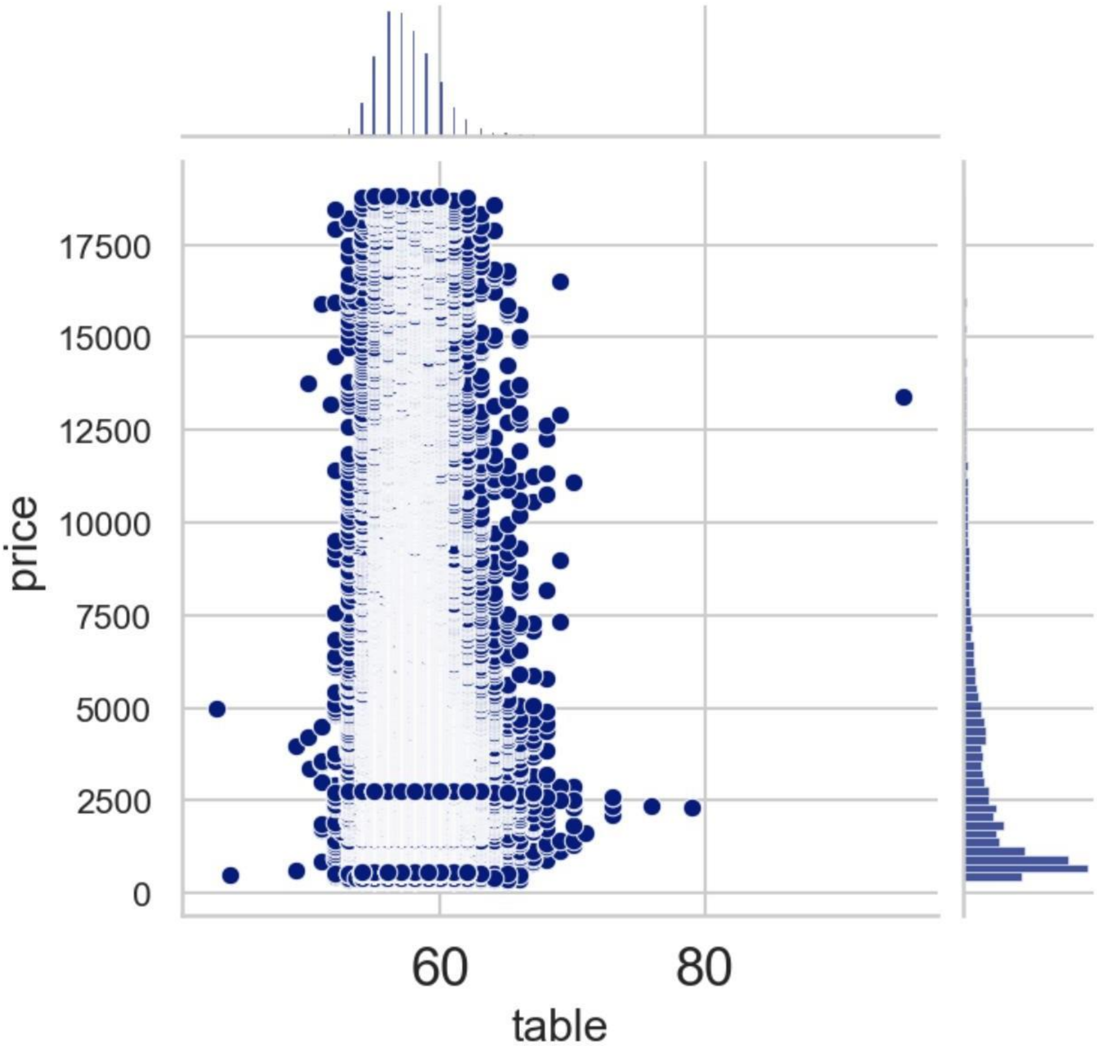
```
sns.kdeplot(dataset['table'] ,shade=True , color='orange')
```

```
<AxesSubplot:xlabel='table', ylabel='Density'>
```



```
sns.jointplot(x='table', y='price', data=dataset , size=5)
```

```
<seaborn.axisgrid.JointGrid at 0x1fdfbc77af0>
```

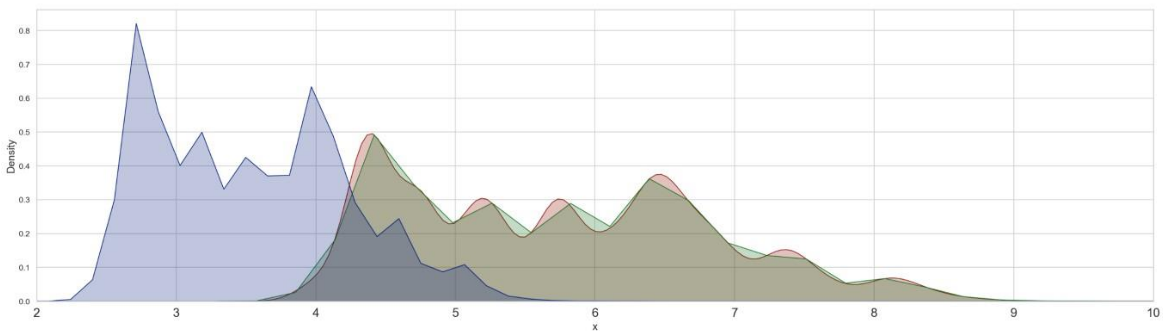


7.Dimensions

```python
sns.kdeplot(dataset['x'] ,shade=True , color='r' )
sns.kdeplot(dataset['y'] , shade=True , color='g' )
sns.kdeplot(dataset['z'] , shade= True , color='b')
plt.xlim(2,10)
```

Python

```
(2.0, 10.0)
```



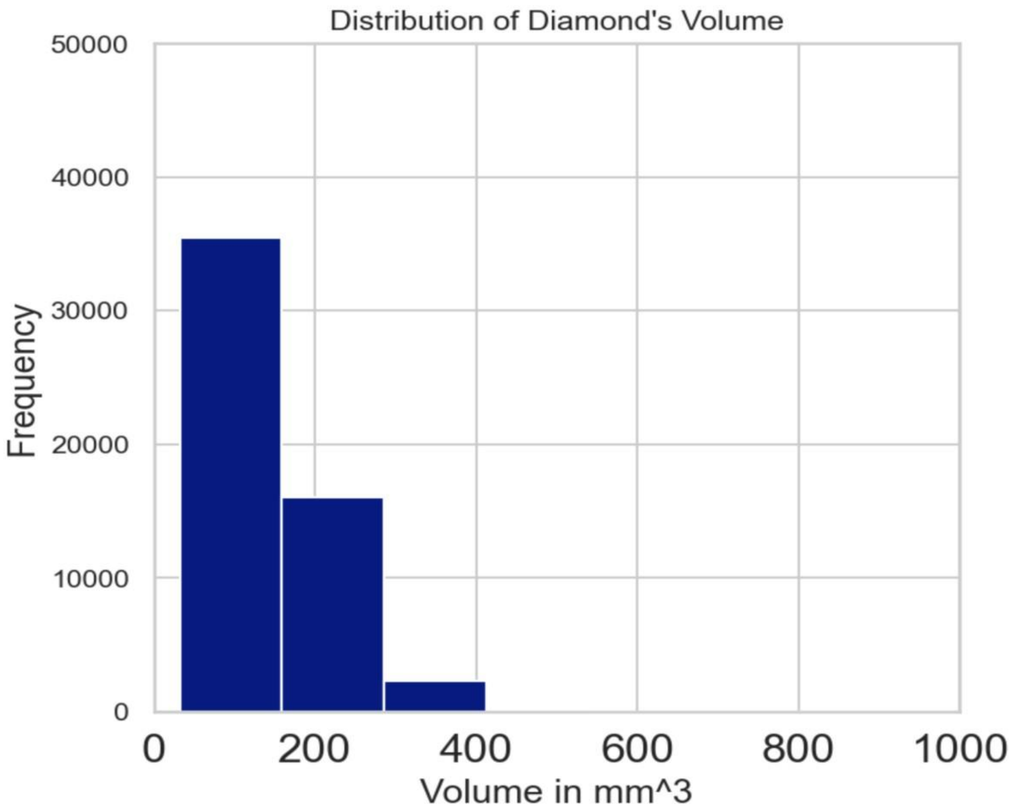
1.Creating new feature volume

```
dataset['volume'] = dataset['x']*dataset['y']*dataset['z']
dataset.head()
```

	carat	cut	color	clarity	depth	table	price	x	y	z	volume
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43	38.202030
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31	34.505856
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31	38.076885
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63	46.724580
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75	51.917250

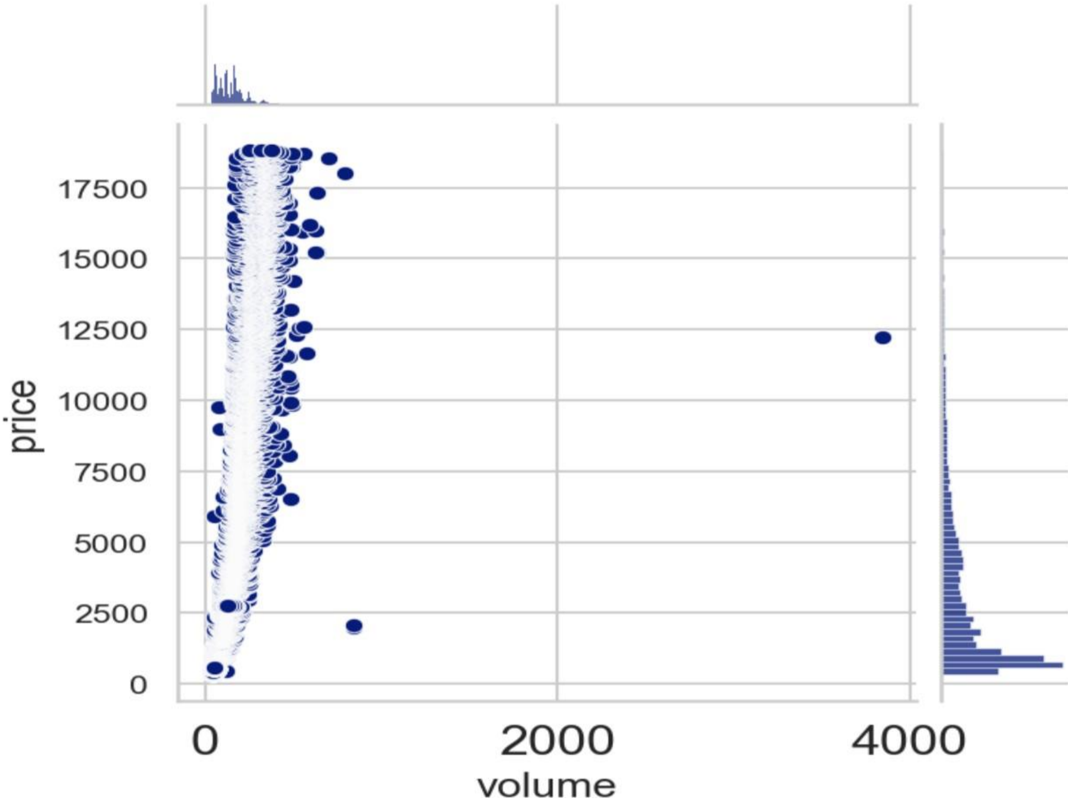
```
plt.figure(figsize=(5,5))
plt.hist( x=dataset['volume'] , bins=30 ,color='b')
plt.xlabel('Volume in mm^3')
plt.ylabel('Frequency')
plt.title('Distribution of Diamond\'s Volume')
plt.xlim(0,1000)
plt.ylim(0,50000)
```

```
(0.0, 50000.0)
```



```python
sns.jointplot(x='volume', y='price' , data=dataset, size=5)
```

```
<seaborn.axisgrid.JointGrid at 0x1fd80972310>
```



```python
X = dataset.drop(['price'], axis=1)
y = dataset['price']

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2, random_state=66)
```

```python
dataset.cut.replace({'Ideal':5, 'Premium':4, 'Good':2, 'Very Good':3, 'Fair':1}, inplace=True)
```

```python
dataset.color.replace({'E':2, 'I':6, 'J':7, 'H':5, 'F':3, 'G':4, 'D':1}, inplace=True)
```

```python
dataset.clarity.replace({'SI2':1, 'SI1':2, 'VS1':3, 'VS2':4, 'VVS2':5, 'VVS1':6, 'I1':7, 'IF':8}, inplace=True)
```

```python
X = dataset.drop(['price'], axis=1)
X.head()
y = dataset['price']
y.head()
```

```
0    326
1    326
2    327
3    334
4    335
Name: price, dtype: int64
```

```python
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
```

```python
import sklearn.linear_model as sl
linreg = sl.LinearRegression()
linreg.fit(X_train, y_train)
```

```
LinearRegression()
```

```python
print('R squared of the Linear Regression on training set: {:.2%}'.format(linreg.score(X_train, y_train)))
print('R squared of the Linear Regression on test set: {:.2%}'.format(linreg.score(X_test, y_test)))
```

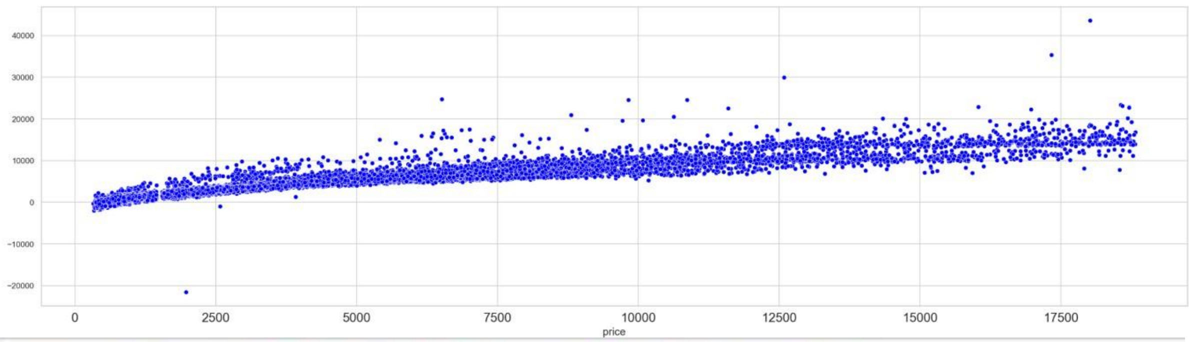
Python

```
R squared of the Linear Regression on training set: 88.42%
R squared of the Linear Regression on test set: 88.82%
```

```python
y_pred = linreg.predict(X_test)
sns.scatterplot(x=y_test , y=y_pred, color="blue")
```

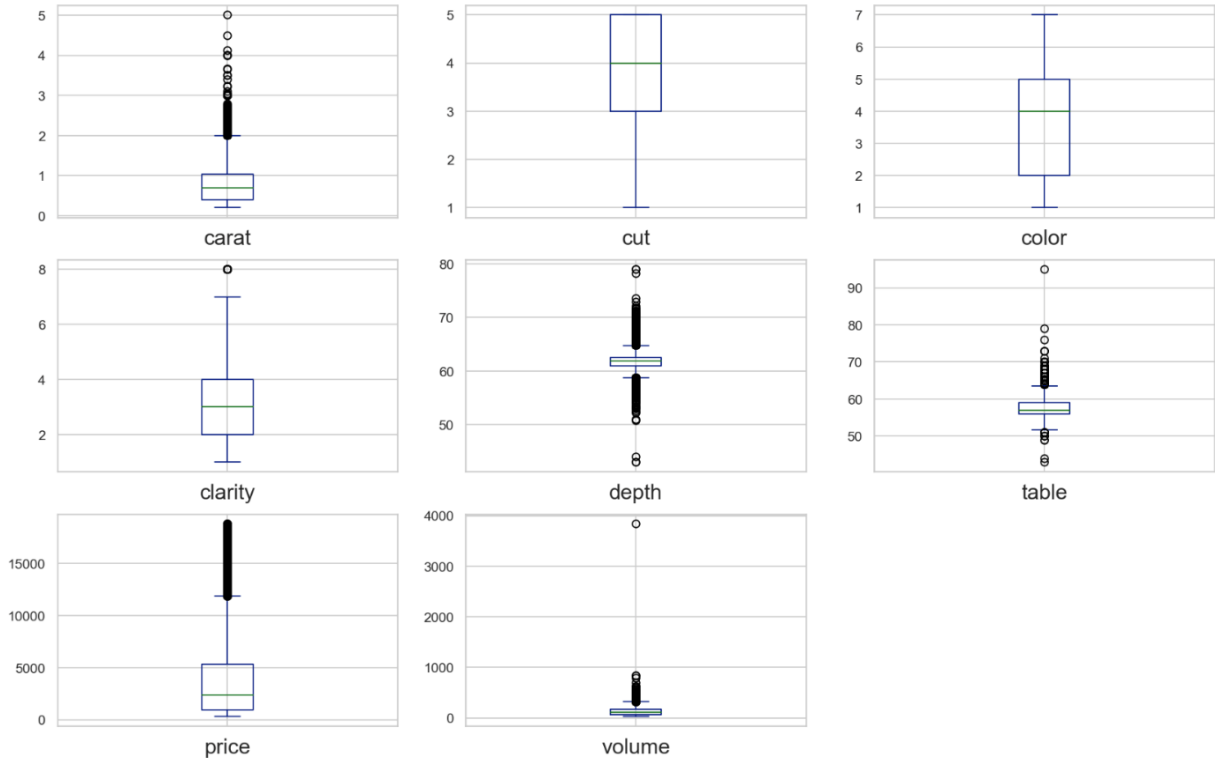
Python

```
<AxesSubplot:xlabel='price'>
```




```python
dataset.drop(['x', 'y', 'z'], axis=1, inplace=True)
```

```python
dataset.plot(kind='box',figsize=(15,10),subplots=True,layout=(3,3))
plt.show()
```




```
import sklearn.ensemble as se
rf = se.RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
model = rf
new_diamond = [0.23, 5, 2, 1, 61.5, 55, 3.95,3.98 , 2.43 , 38.20]
prediction = model.predict([new_diamond])[0]
print("\033[1m The market price of this new diamond is ${:.2f}".format(prediction))
```

The market price of this new diamond is $393.73

```
kmeans=KMeans(n_clusters=5,random_state=0).fit(dataset)
print("KMeans Clusters : ",kmeans.cluster_centers_)
print(kmeans.labels_)
```

```
KMeans Clusters :  [[8.98714684e-01 3.66693195e+00 3.60803820e+00 2.62451253e+00
  6.18460247e+01 5.78502746e+01 3.71543971e+03 1.45757671e+02]
 [1.49038642e+00 3.93348946e+00 4.06018735e+00 3.29274005e+00
  6.16578220e+01 5.77651522e+01 1.05481000e+04 2.42877059e+02]
 [4.30183006e-01 4.03792938e+00 3.34236062e+00 3.67025791e+00
  6.17186953e+01 5.70972328e+01 1.13132159e+03 7.05828230e+01]
 [1.90609006e+00 3.86416510e+00 4.28480300e+00 2.81313321e+00
  6.16569231e+01 5.79865666e+01 1.57297441e+04 3.09186965e+02]
 [1.17761514e+00 3.81399481e+00 3.97854312e+00 2.94765614e+00
  6.17851715e+01 5.77403444e+01 6.51254490e+03 1.91138053e+02]]
[2 2 2 ... 0 0 0]
```

CHAPTER 5
CONCLUSION

- We would like to summarize that our dataset is 88.42% accurate hence we can predict the price of a diamond.
- We predicted the price of a diamond with the following features => carat: 0.23, cut: 5, color: 2, clarity: 1, depth: 61.5, table: 55, x: 3.95, y: 3.98 , z: 2.43 , volume: 38.20.
- The price of the above diamond is $393.73.
- We can also conclude that our dataset has no '0' or 'NAN' values and a very few outliers which makes our analysis more accurate.
- We constructed scatterplot for visualizing the relation between our x values (which contains all parameters except price) and predicted y values (predicted prices).
- From the heatmap we can conclude that the darkest shade shows the most negative correlation between the features which is –0.3 between table and depth and the lightest shade shows the most positive correlation between the features which is 0.98 between carat and x; z and x.
- Using KMeans clustering analysis, we also divided our dataset into 5 clusters.

CHAPTER 6
REFERENCES

[1] https://www.kaggle.com/datasets/shivam2503/diamonds