Problem 1: Linear Regression

You are hired by a company Gem Stones co ltd, which is a cubic zirconia manufacturer. You are provided with the dataset containing the prices and other attributes of almost 27,000 cubic zirconia (which is an inexpensive diamond alternative with many of the same qualities as a diamond). The company is earning different profits on different prize slots. You have to help the company in predicting the price for the stone on the bases of the details given in the dataset so it can distinguish between higher profitable stones and lower profitable stones so as to have better profit share. Also, provide them with the best 5 attributes that are most important.

Data Dictionary:

Variable Name	**Description**
Carat	Carat weight of the cubic zirconia.
Cut	Describe the cut quality of the cubic zirconia. Quality is increasing order Fair, Good, Very Good, Premium, Ideal.
Color	Colour of the cubic zirconia.With D being the worst and J the best.
Clarity	Clarity refers to the absence of the Inclusions and Blemishes. (In order from Worst to Best in terms of avg price) IF, VVS1, VVS2, VS1, VS2, Sl1, Sl2, l1
Depth	The Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter.
Table	The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter.
Price	the Price of the cubic zirconia.
X	Length of the cubic zirconia in mm.
Y	Width of the cubic zirconia in mm.
Z	Height of the cubic zirconia in mm.

1.1 Read the data and do exploratory data analysis. Describe the data briefly. (Check the null values, Data types, shape, EDA, duplicate values). Perform Univariate and Bivariate Analysis.

Table 1: Head of the dataset showing the first 5 records

	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
0	1	0.30	Ideal	E	SI1	62.1	58.0	4.27	4.29	2.66	499
1	2	0.33	Premium	G	IF	60.8	58.0	4.42	4.46	2.70	984
2	3	0.90	Very Good	E	VVS2	62.2	60.0	6.04	6.12	3.78	6289
3	4	0.42	Ideal	F	VS1	61.6	56.0	4.82	4.80	2.96	1082
4	5	0.31	Ideal	F	VVS1	60.4	59.0	4.35	4.43	2.65	779

The dataset was loaded and the head of the dataset was checked. Table 1 shows the first 5 records of the dataset. From this table, we can see the different variables or columns of the dataset.

Table 2: Information of the dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26967 entries, 0 to 26966
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  26967 non-null  int64
 1   carat       26967 non-null  float64
 2   cut         26967 non-null  object
 3   color       26967 non-null  object
 4   clarity     26967 non-null  object
 5   depth       26270 non-null  float64
 6   table       26967 non-null  float64
 7   x           26967 non-null  float64
 8   y           26967 non-null  float64
 9   z           26967 non-null  float64
 10  price       26967 non-null  int64
dtypes: float64(6), int64(2), object(3)
memory usage: 2.3+ MB
```

The dataset has 11 columns and 26967 records except ‘depth’ which has 26270 records seen in Table 2. There are 26967 non-null records in all the columns except ‘depth’ meaning there are missing records in ‘depth’ based on this initial analysis that was done.

Table 3: Data type of the columns in the dataset

```
Unnamed: 0      int64
carat         float64
cut            object
color          object
clarity        object
depth         float64
table         float64
x             float64
y             float64
z             float64
price           int64
dtype: object
```

The column ‘cut’, ‘color’, ‘clarity’ is of object data type while all the other columns are integer or float data type. This is seen in Table 1, Table 2 and Table 3. ‘Unnamed: 0’ is the serial number. Therefore, it is not considered as an independent variable. Hence there are 9 independent variables and one target variable – ‘price’.

The shape of the data is (26967, 11) meaning the dataset has 26967 rows and 11 columns.

Table 4: Missing value of the columns in the dataset

```
Unnamed: 0      0
carat           0
cut             0
color           0
clarity         0
depth         697
table           0
x               0
y               0
z               0
price           0
dtype: int64
```

The dataset was further checked for missing values and it is seen from Table 4 that there are missing values in ‘depth’ column of the dataset.

The ‘Unnamed: 0’ is basically the serial number and hence it is dropped as it may interfere with the exploratory data analysis. The dataset is now checked for duplicates.

```
df1_drop.duplicated().sum()
```

```
34
```

Figure 1: Number of duplicates in the dataset

The dataset is checked for duplicate values and it was found that there 34 duplicates seen in Figure 1. The duplicates are dropped and upon dropping the duplicates, the shape of the dataset is (26933, 10) meaning the dataset now has 26933 rows and 10 columns

Table 5: Description of the dataset

	carat	depth	table	x	y	z	price
count	26933.000000	26236.000000	26933.000000	26933.000000	26933.000000	26933.000000	26933.000000
mean	0.798010	61.745285	57.455950	5.729346	5.733102	3.537769	3937.526120
std	0.477237	1.412243	2.232156	1.127367	1.165037	0.719964	4022.551862
min	0.200000	50.800000	49.000000	0.000000	0.000000	0.000000	326.000000
25%	0.400000	61.000000	56.000000	4.710000	4.710000	2.900000	945.000000
50%	0.700000	61.800000	57.000000	5.690000	5.700000	3.520000	2375.000000
75%	1.050000	62.500000	59.000000	6.550000	6.540000	4.040000	5356.000000
max	4.500000	73.600000	79.000000	10.230000	58.900000	31.800000	18818.000000

Table 5 shows the description or the summary of the numerical columns in the dataset after dropping the 'Unnamed: 0' column. The 'Unnamed: 0' is basically the serial number and hence it is dropped as it may interfere with the exploratory data analysis. It can be seen that all the columns in this dataframe have 26933 values except 'depth' column. There are missing values in the 'depth' column. By looking at Table 5, we are able to deduce that 'price' has the highest mean value while 'carat' has the lowest mean value. 'price' has the highest standard deviation value while 'carat' has the lowest standard deviation value. This is probably because price and carat are two different measurements. 'price' is the piece of the cubic zirconia while 'carat' is the carat weight of the cubic zirconia. The mean and median values are approximately equal in all variables.

Univariate Analysis:

Numerical variables:

Table 6: Head of the numerical dataset showing the first 5 records

	carat	depth	table	x	y	z	price
0	0.30	62.1	58.0	4.27	4.29	2.66	499
1	0.33	60.8	58.0	4.42	4.46	2.70	984
2	0.90	62.2	60.0	6.04	6.12	3.78	6289
3	0.42	61.6	56.0	4.82	4.80	2.96	1082
4	0.31	60.4	59.0	4.35	4.43	2.65	779

Table 6 shows the first 5 records of the numerical dataset. The numerical dataset was used to calculate the skewness, plot the univariate distribution and the boxplot.

Table 7: Skewness of the variables of the dataset

```
y         3.867764
z         2.580665
price     1.619116
carat     1.114789
table     0.765805
x         0.392290
depth    -0.026086
dtype: float64
```

1. Carat:

Table 8: Description of 'carat'

```
Description of carat
----------------------------------------------------
count    26933.000000
mean         0.798010
std          0.477237
min          0.200000
25%          0.400000
50%          0.700000
75%          1.050000
max          4.500000
Name: carat, dtype: float64 Distribution of carat
```

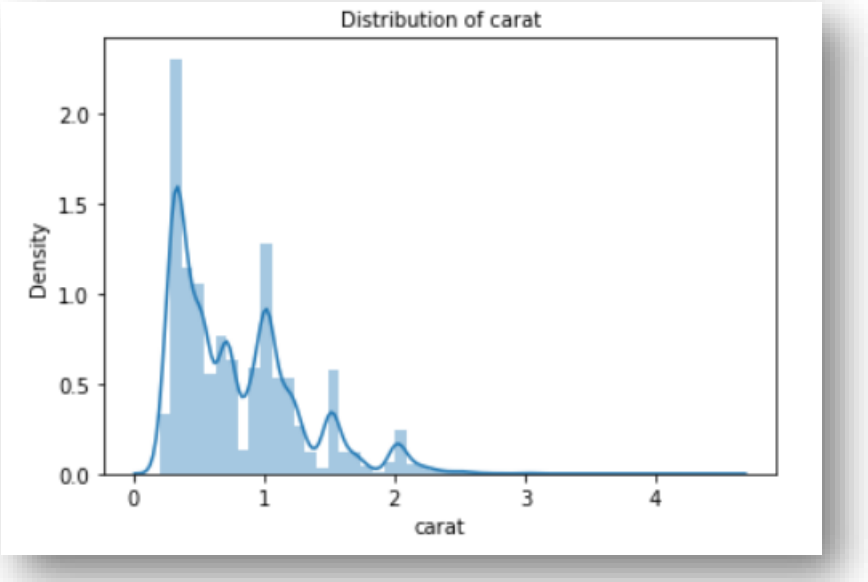


Figure 2: Univariate distribution of 'carat'

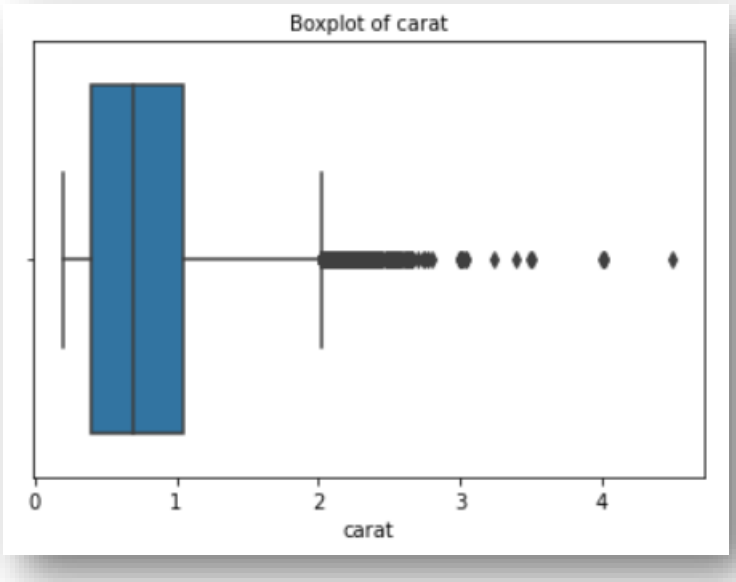


Figure 3: Boxplot showing the distribution of 'carat'

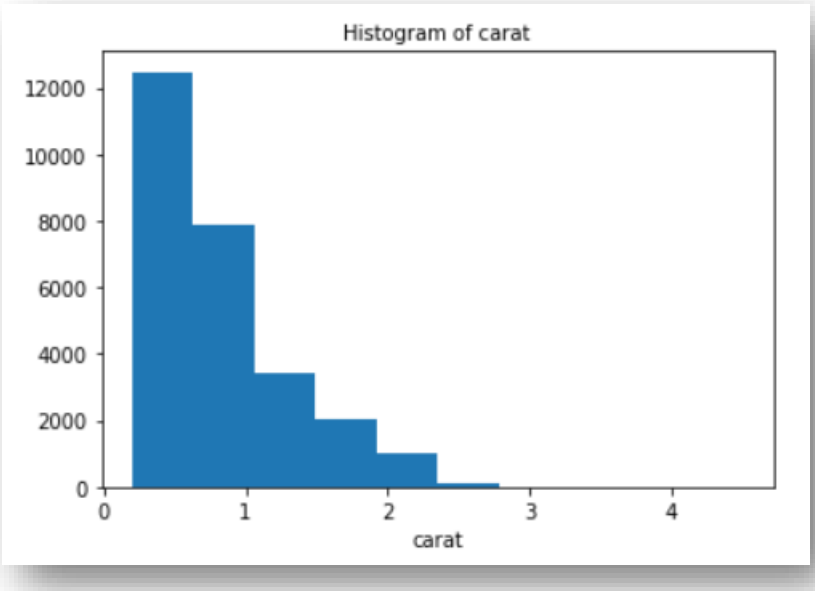


Figure 4: Histogram of 'carat'

Univariate analysis of ‘carat’ is done to understand the patterns and distribution of the data. From Figure 3, we can see that the Box plot of ‘carat’ variable has outliers. The distribution of the data is moderately right skewed which is seen in Figure 2. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘carat’ variable is 1.114789. We can also observe from Figure 2 that the distribution of ‘carat’ variable has multiple modes. From Table 8, it is seen that the mean of the data is 0.798010 meaning the carat weight of the cubic zirconia is 0.798 on average. The dist plot shows the distribution of data from 0 to 4.

2. Depth:

Table 9: Description of 'depth'

```
Description of depth
-----------------------------------------------------
count    26236.000000
mean        61.745285
std          1.412243
min         50.800000
25%         61.000000
50%         61.800000
75%         62.500000
max         73.600000
Name: depth, dtype: float64 Distribution of depth
```

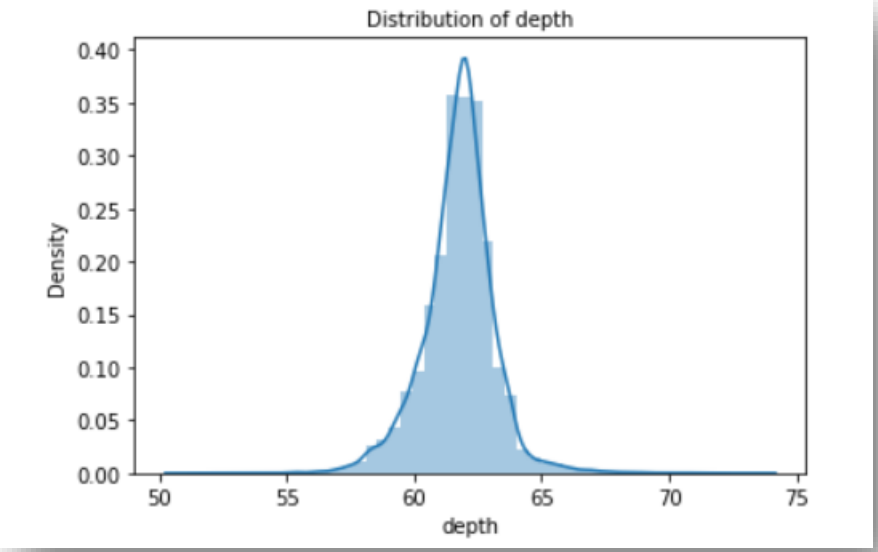


Figure 5: Univariate distribution of 'depth'

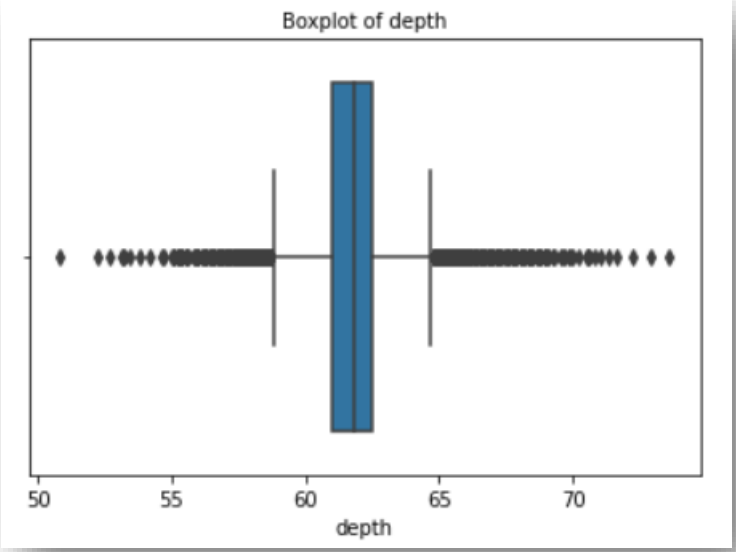


Figure 6: Boxplot showing the distribution of 'depth'

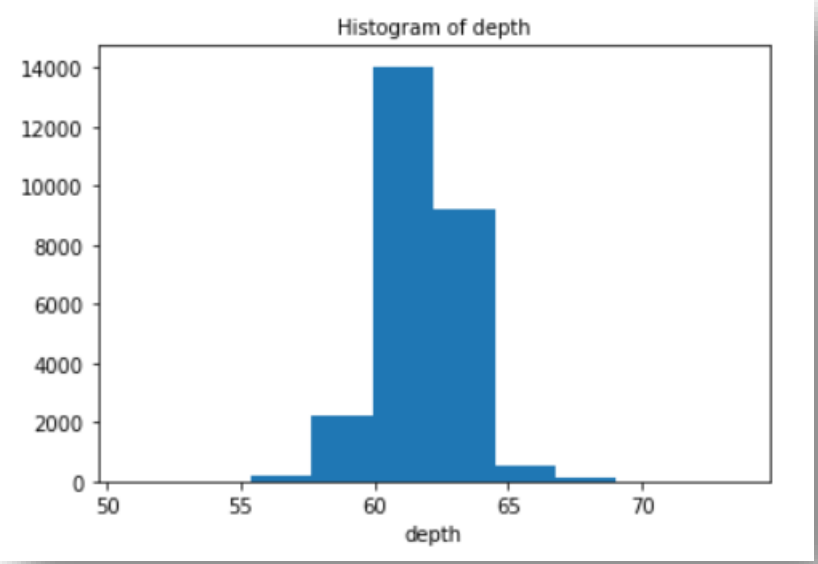


Figure 7: Histogram of 'depth'

Univariate analysis of ‘depth’ is done to understand the patterns and distribution of the data. From Figure 6, we can see that the Box plot of ‘depth’ variable has outliers. The distribution of the data is moderately left skewed which is seen in Figure 5. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘depth’ variable is -0.026086. From Table 9, it is seen that the mean of the data is 61.745285 meaning the height of cubic zirconia, measured from the culet to the table, divided by its average Girdle Diameter is 61.75 on average. The dist plot shows the distribution of data from 50 to 70.

3. Table:

Table 10: Description of 'table'

```
Description of table
----------------------------------------------------
count    26933.000000
mean        57.455950
std          2.232156
min         49.000000
25%         56.000000
50%         57.000000
75%         59.000000
max         79.000000
Name: table, dtype: float64 Distribution of table
```

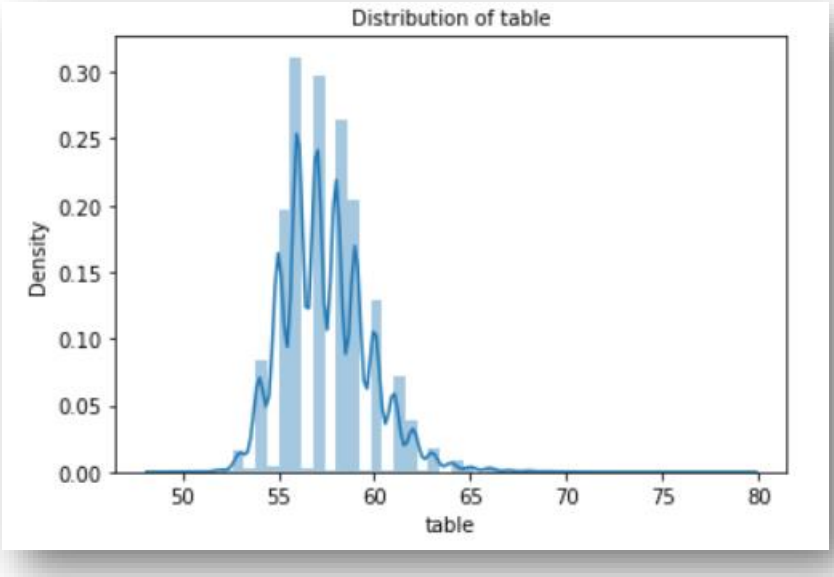


Figure 8: Univariate distribution of 'table'

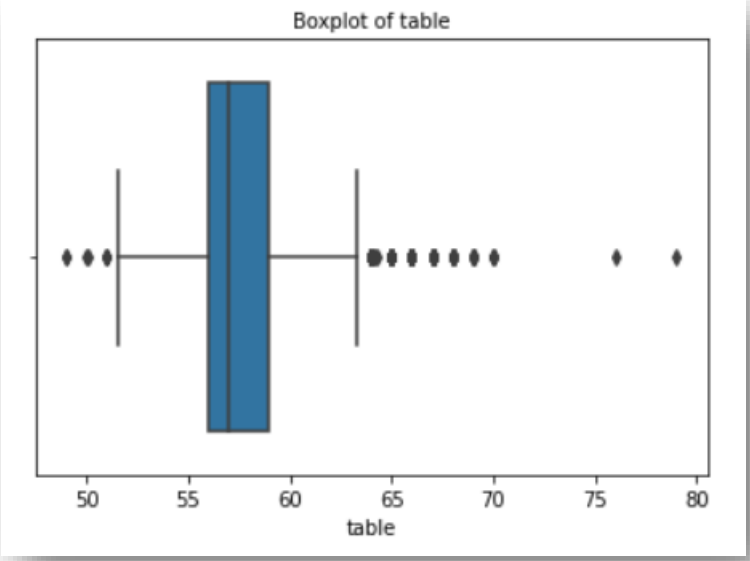


Figure 9: Boxplot showing the distribution of 'table'

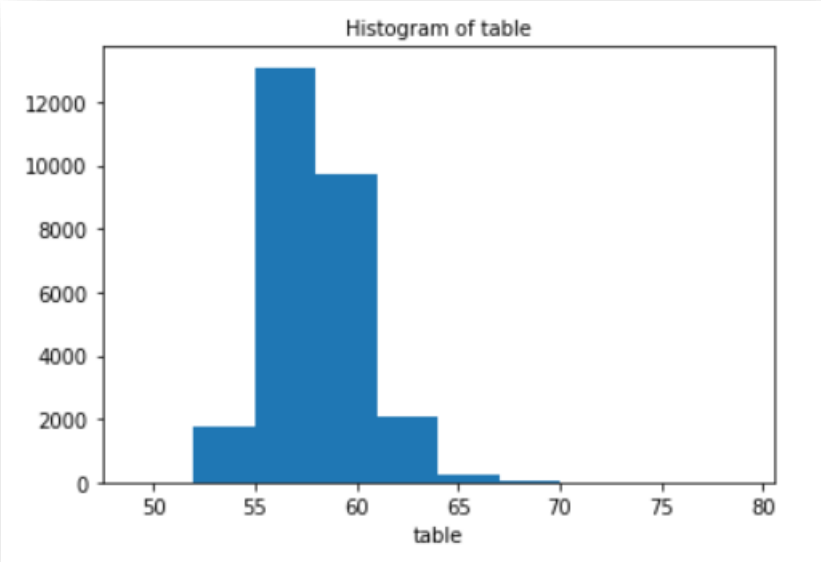


Figure 10: Histogram of 'table'

Univariate analysis of ‘table’ is done to understand the patterns and distribution of the data. From Figure 9, we can see that the Box plot of ‘table’ variable has outliers. The distribution of the data is moderately right skewed which is seen in Figure 8. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘table’ variable is 0.765805. We can also observe from Figure 8 that the distribution of ‘table’ variable has multiple modes. From Table 10, it is seen that the mean of the data is 57.455950 meaning the width of the cubic zirconia's table expressed as a percentage of its average diameter is 57.46 on average. The dist plot shows the distribution of data from 50 to 80.

4. X:

Table 11: Description of 'x'

```
Description of x
-----------------------------------------
count    26933.000000
mean         5.729346
std          1.127367
min          0.000000
25%          4.710000
50%          5.690000
75%          6.550000
max         10.230000
Name: x, dtype: float64 Distribution of x
```

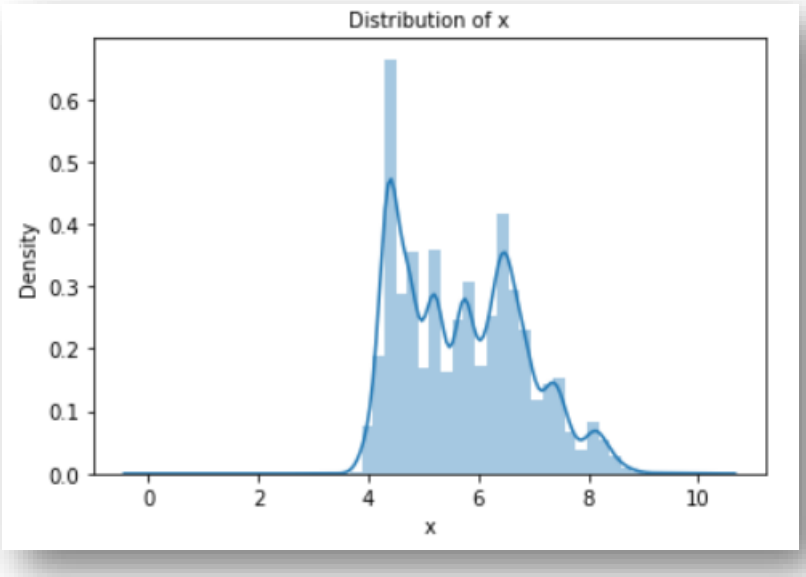


Figure 11: Univariate distribution of 'x'

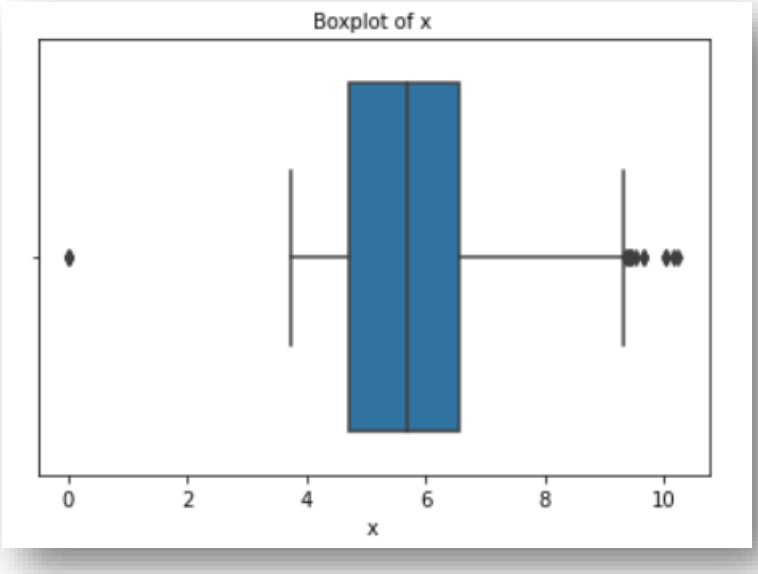


Figure 12: Boxplot showing the distribution of 'x'

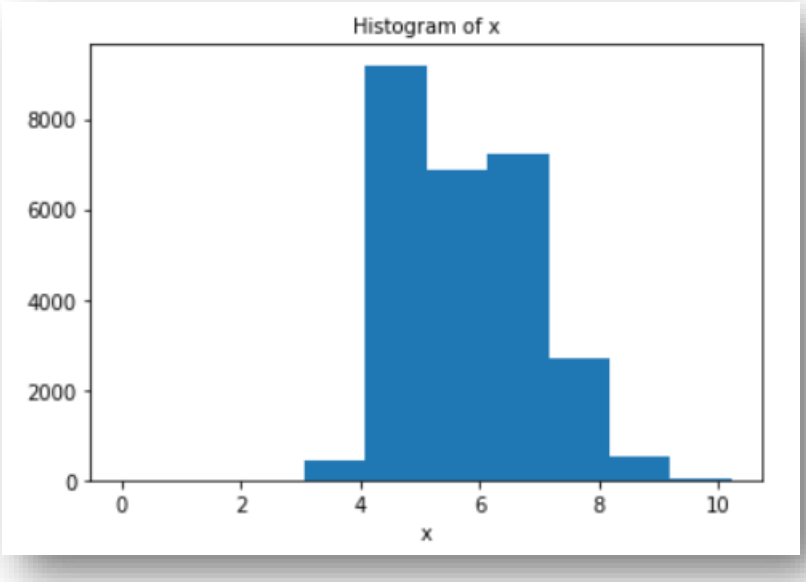


Figure 13: Histogram of 'x'

Univariate analysis of ‘x’ is done to understand the patterns and distribution of the data. From Figure 12, we can see that the Box plot of ‘x’ variable has outliers. The distribution of the data is moderately right skewed which is seen in Figure 11. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘x’ variable is 0.392290. We can also observe from Figure 11 that the distribution of ‘x’ variable has multiple modes. From Table 11, it is seen that the mean of the data is 5.729346 meaning the length of the cubic zirconia is 5.73mm on average. The dist plot shows the distribution of data from 0 to 10.

5. Y:

Table 12: Description of 'y'

```
Description of y
-----------------------------------------
count    26933.000000
mean         5.733102
std          1.165037
min          0.000000
25%          4.710000
50%          5.700000
75%          6.540000
max         58.900000
Name: y, dtype: float64 Distribution of y
```

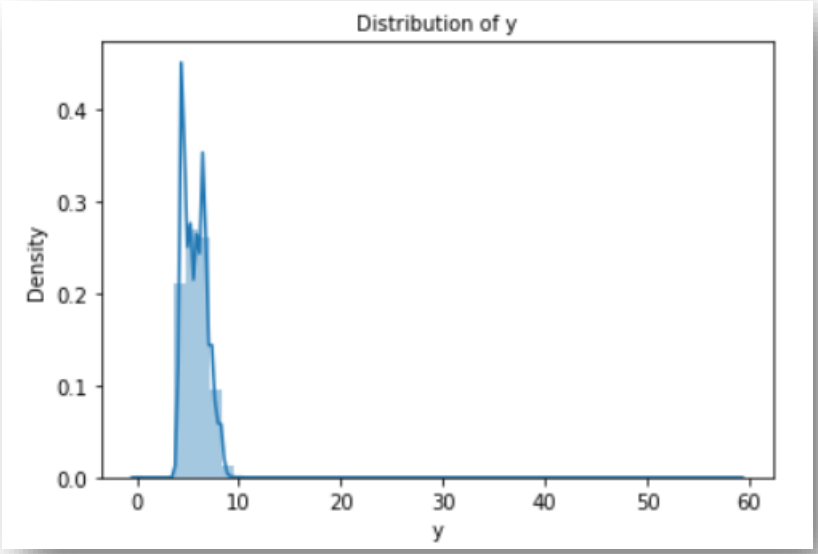


Figure 14: Univariate distribution of 'y'

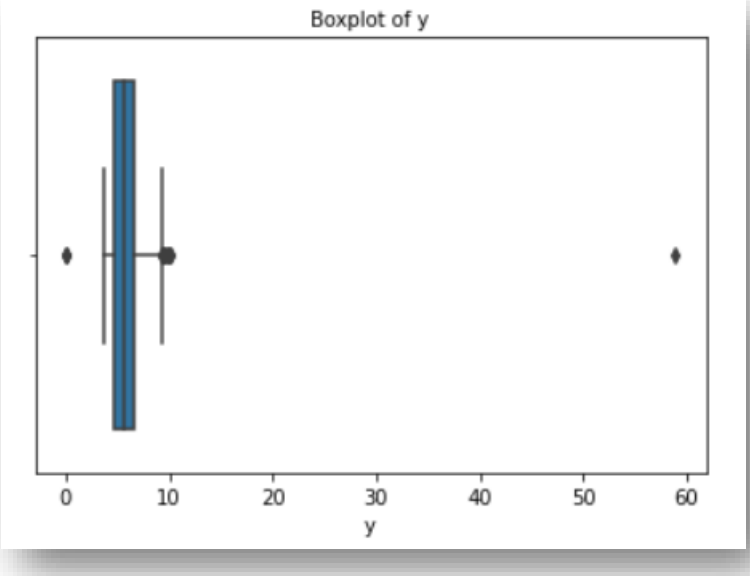


Figure 15: Boxplot showing the distribution of 'y'

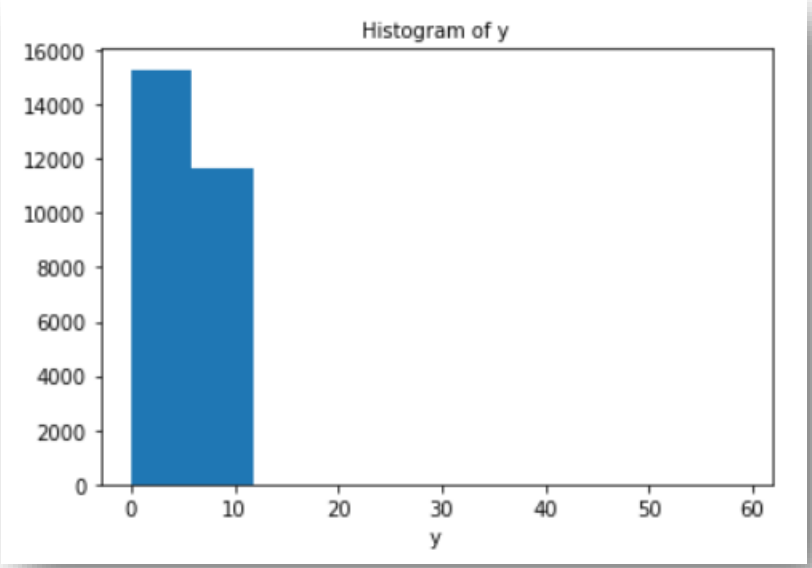


Figure 16: Histogram of 'y'

Univariate analysis of ‘y’ is done to understand the patterns and distribution of the data. From Figure 15, we can see that the Box plot of ‘y’ variable has outliers. The distribution of the data is moderately right skewed which is seen in Figure 14. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘y’ variable is 3.867764. We can also observe from Figure 14 that the distribution of ‘y’ variable has multiple modes. From Table 12, it is seen that the mean of the data is 5.733102 meaning the width of the cubic zirconia is 5.73mm on average. The dist plot shows the distribution of data from 0 to 60.

6. Z:

Table 13: Description of 'z'

```
Description of z
-----------------------------------------
count    26933.000000
mean         3.537769
std          0.719964
min          0.000000
25%          2.900000
50%          3.520000
75%          4.040000
max         31.800000
Name: z, dtype: float64 Distribution of z
```

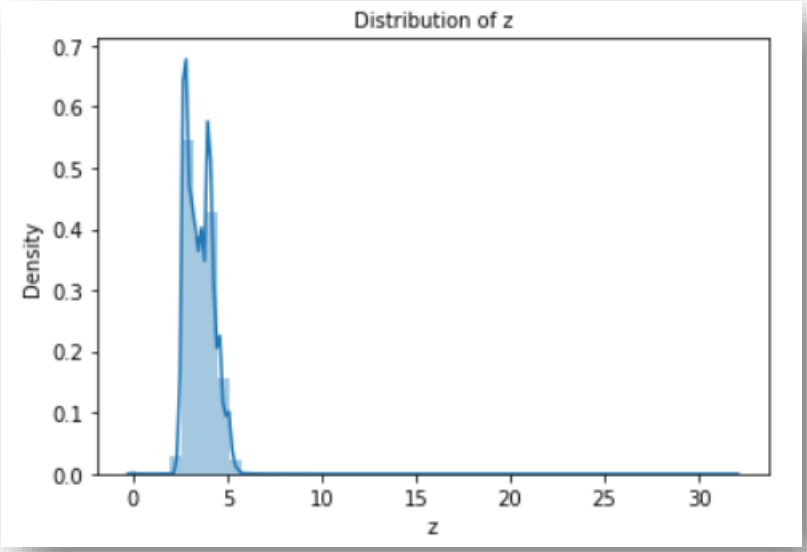


Figure 17: Univariate distribution of 'z'

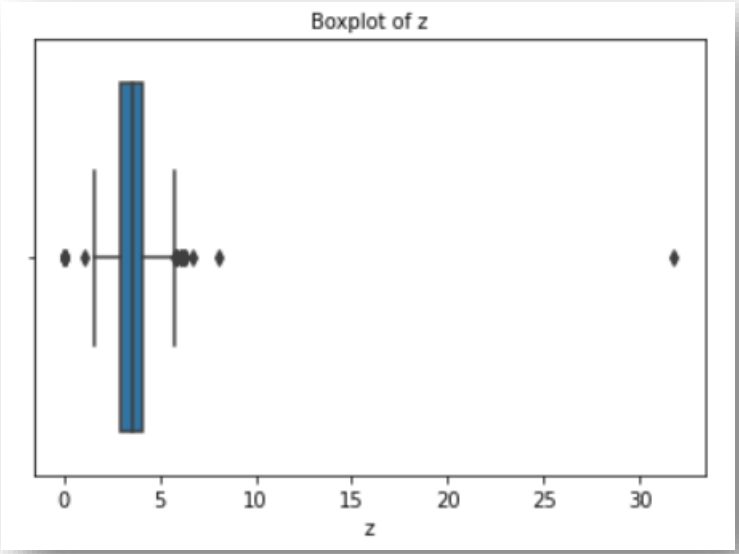


Figure 18: Boxplot showing the distribution of 'z'

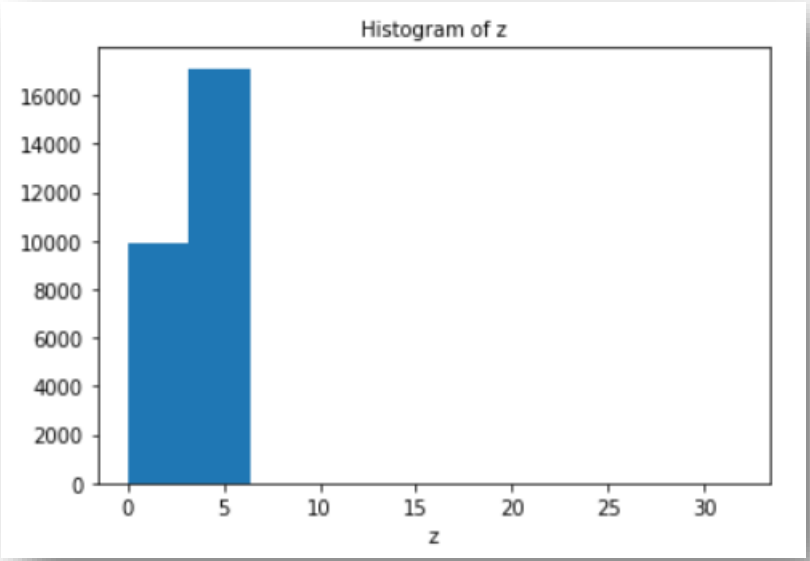


Figure 19: Histogram of 'z'

Univariate analysis of ‘z’ is done to understand the patterns and distribution of the data. From Figure 18, we can see that the Box plot of ‘z’ variable has outliers. The distribution of the data is moderately right skewed which is seen in Figure 17. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘z’ variable is 2.580665. We can also observe from Figure 17 that the distribution of ‘z’ variable has multiple modes. From Table 13, it is seen that the mean of the data is 3.537769 meaning the height of the cubic zirconia is 3.54mm on average. The dist plot shows the distribution of data from 0 to 30.

7. Price:

Table 14: Description of 'price'

```
Description of price
-----------------------------------------------------
count    26933.000000
mean      3937.526120
std       4022.551862
min        326.000000
25%        945.000000
50%       2375.000000
75%       5356.000000
max      18818.000000
Name: price, dtype: float64 Distribution of price
```

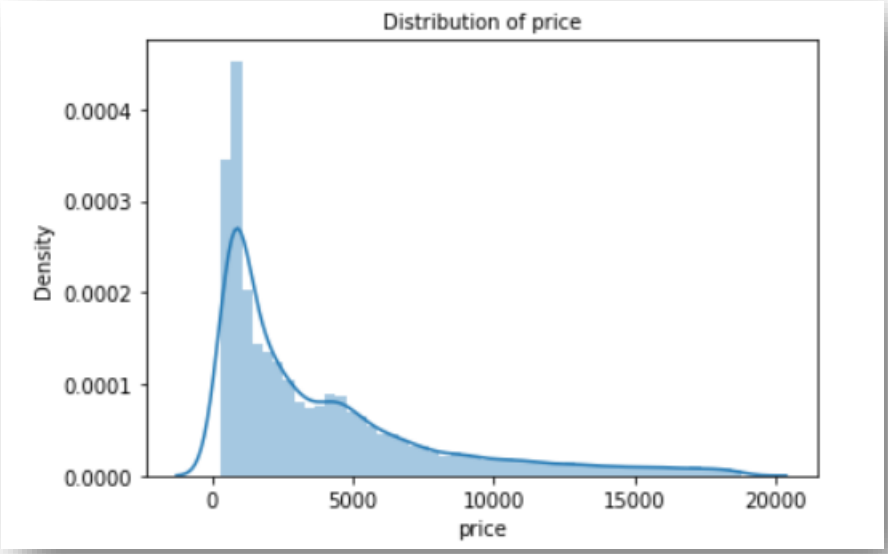


Figure 20: Univariate distribution of 'price'

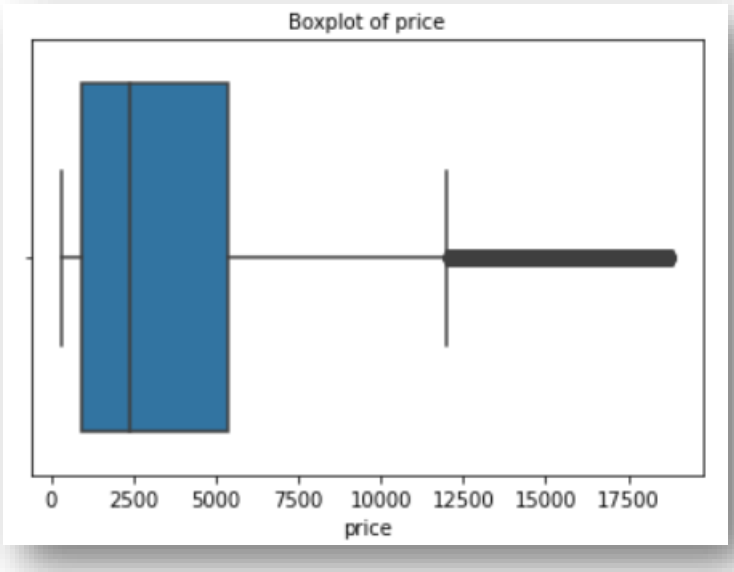


Figure 21: Boxplot showing the distribution of 'price'

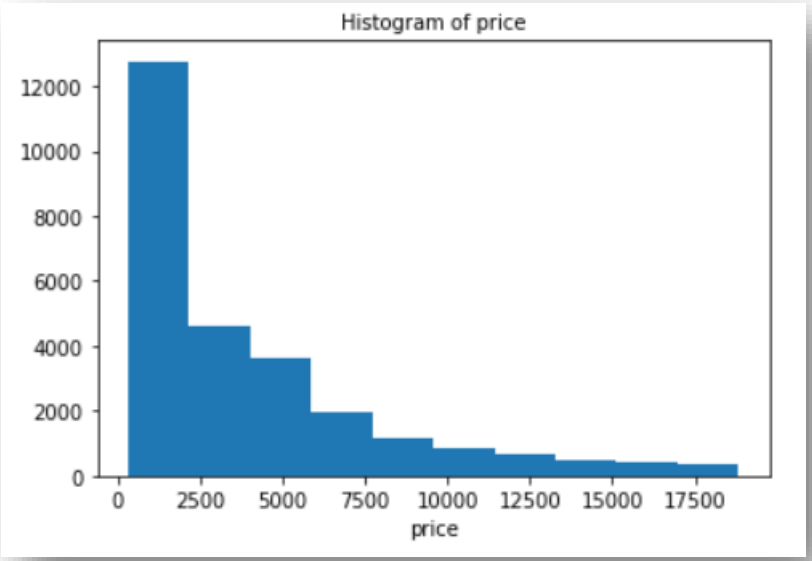


Figure 22: Histogram of 'price'

Univariate analysis of ‘price’ is done to understand the patterns and distribution of the data. From Figure 21, we can see that the Box plot of ‘price’ variable has outliers. The distribution of the data is moderately right skewed which is seen in Figure 20. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘price’ variable is 1.619116. From Table 14, it is seen that the mean of the data is 3937.526120 meaning the price of cubic zirconia is 3937.53 on average. The dist plot shows the distribution of data from 0 to 17500.

Categorical variables:

8. **Cut:**

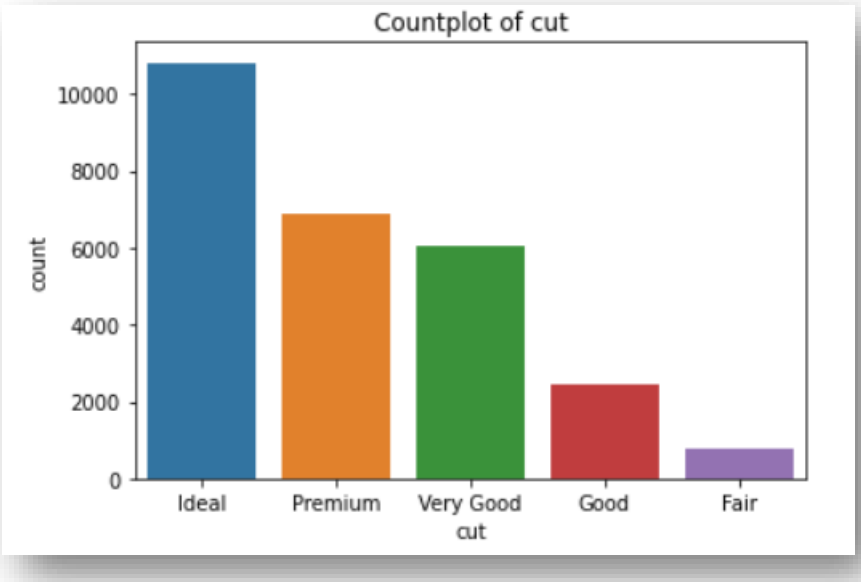


Figure 23: Countplot of 'cut'

Univariate analysis of 'cut' is done to understand the patterns and distribution of the data. From Figure 23, we can understand that 'Ideal' cut is the most sold cut quality of the cubic zirconia and 'Fair' cut is the least sold cut quality. Quality is in the increasing order - Fair, Good, Very Good, Premium, Ideal.

9. **Color:**

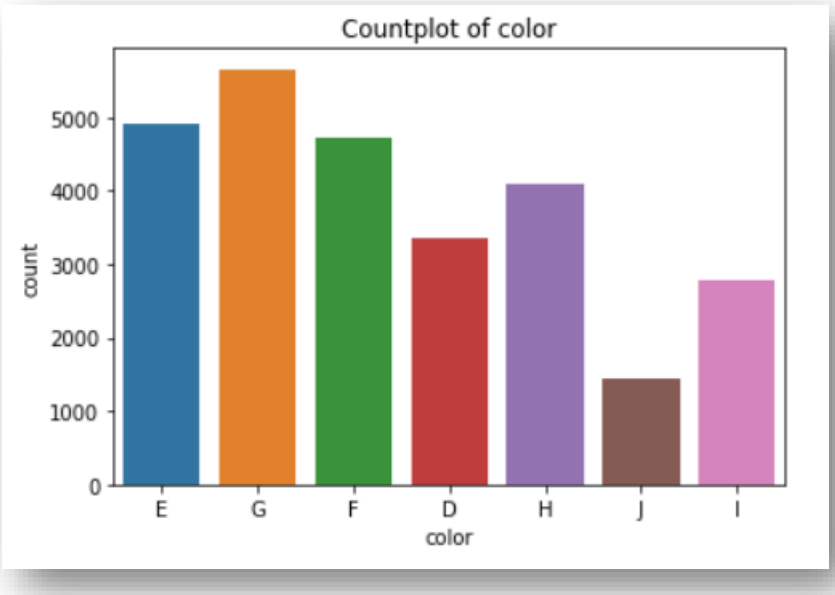


Figure 24: Countplot of 'color'

Univariate analysis of ‘color’ is done to understand the patterns and distribution of the data. From Figure 24, we can understand that ‘G’ color is the most sold color of the cubic zirconia and ‘J’ is the least sold color. However, from the description of the dataset, it is seen that ‘D’ is the worst and ‘J’ is the best color.

10. Clarity:

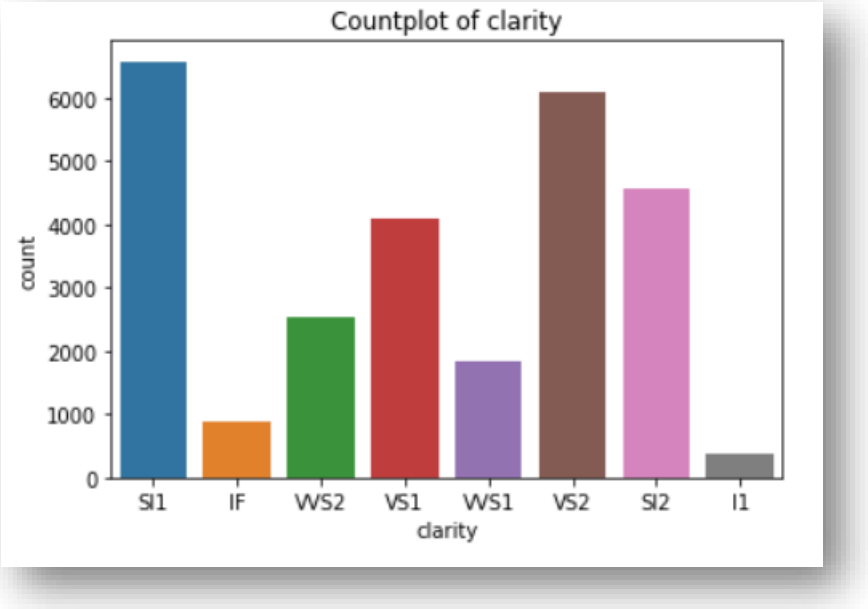


Figure 25: Countplot of 'clarity'

Univariate analysis of ‘clarity’ is done to understand the patterns and distribution of the data. From Figure 25, we can understand that ‘SI1’ clarity is the most sold clarity of the cubic zirconia and ‘I1’ is the least sold clarity. Clarity refers to the absence of the inclusions and blemishes. From the description of the dataset, it is seen that the order from worst to best in terms of average price is - IF, VVS1, VVS2, VS1, VS2, Sl1, Sl2, l1.

Bivariate Analysis:

Numerical variables:

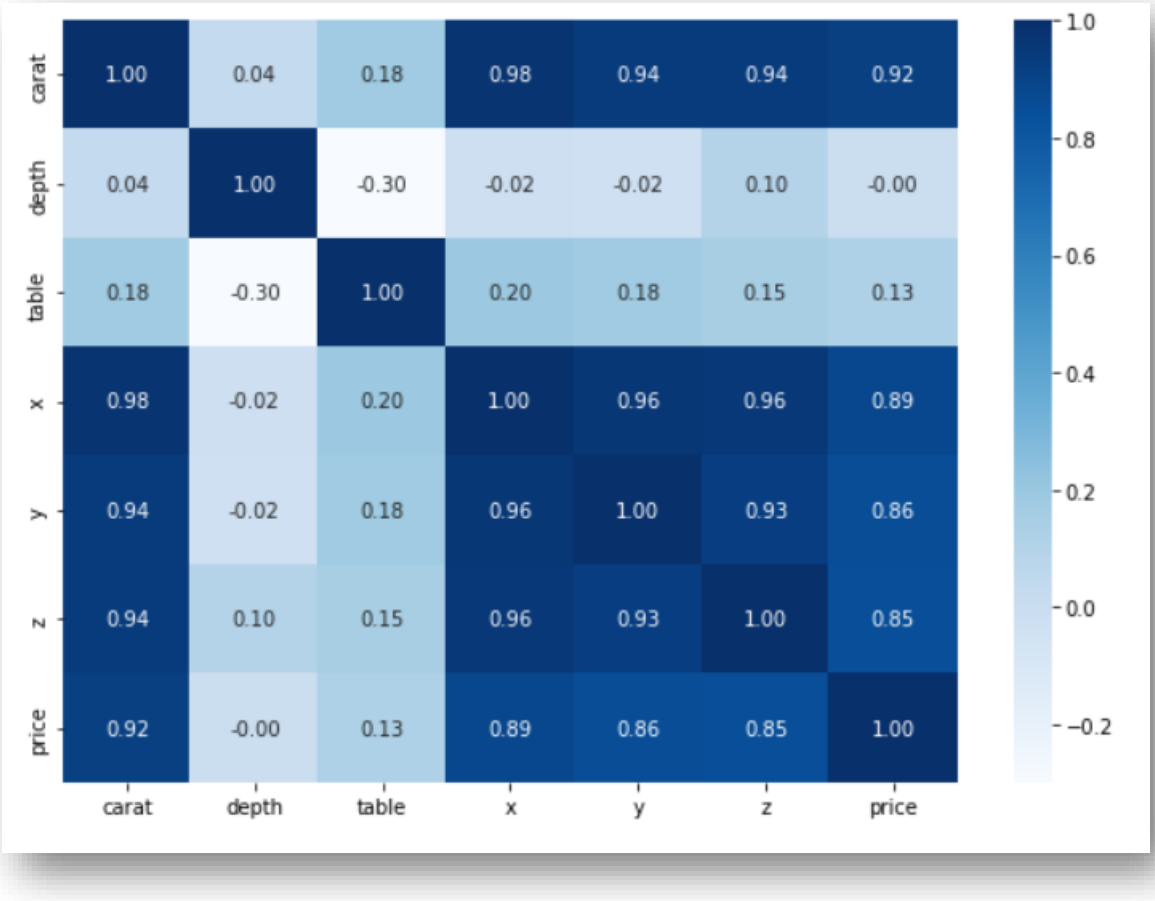


Figure 26: Heat map showing the bivariate analysis of the dataset

Bivariate analysis is done using the help of a heat map. A heat map is used to understand the correlation between two numerical values in a dataset. Figure 26 shows the heatmap of the dataset.

Observations:

- There is high correlation between the different features like carat, x, y, z and price
- There is less correlation between table with the other features
- Depth is negatively correlated with most the other features except for carat

Categorical variables:

1. **Cut:**

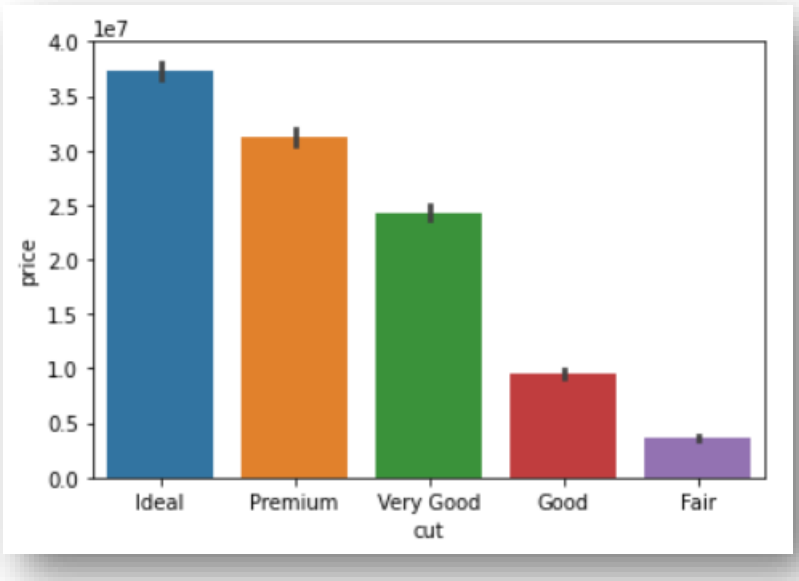


Figure 27: Countplot of 'cut' and 'price'

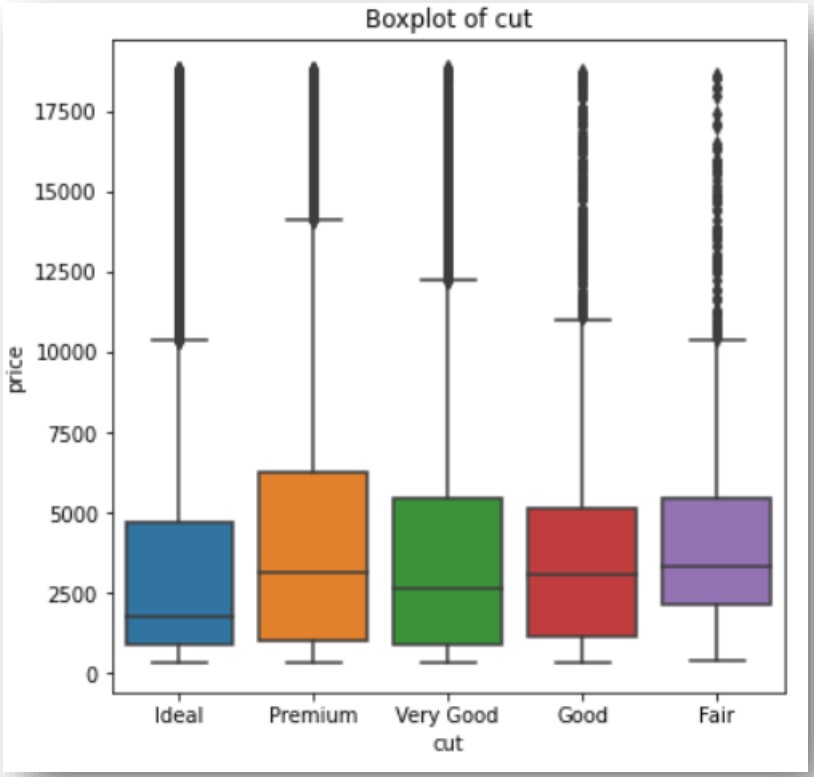


Figure 28: Box plot of 'cut' and 'price'

From the Box plot in Figure 28, we can see that all cut type gems have outliers with respect to price. From Figure 27Figure 29, it is seen that the least priced cubic zirconia gem seems to be ‘Fair’ type while ‘Premium’ and ‘Ideal’ cut type cubic zirconia gems seem to be expensive.

2. **Color:**

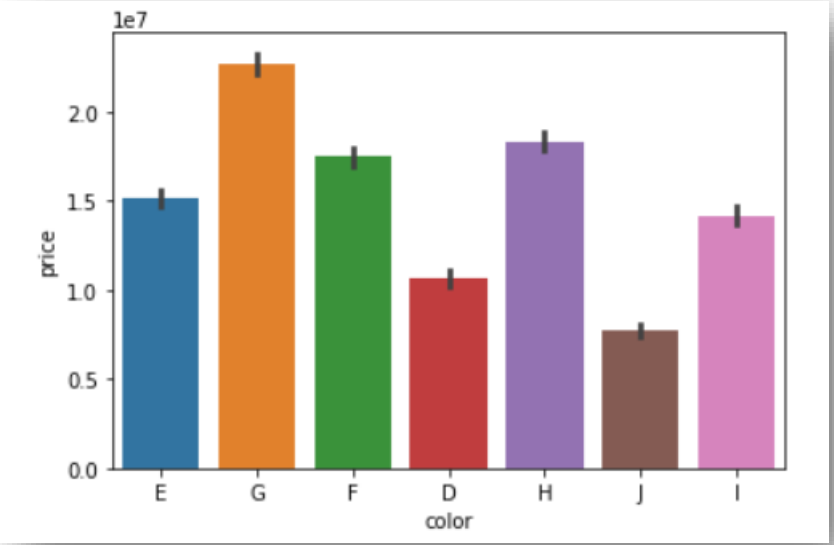


Figure 29: Countplot of 'color' and 'price'

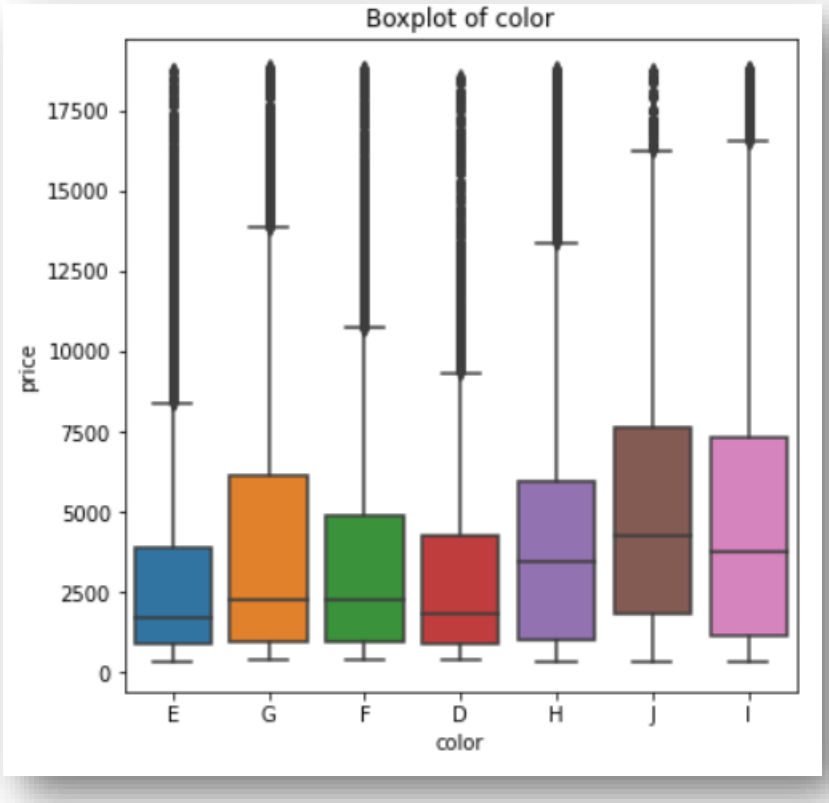


Figure 30: Box plot of 'color' and 'price'

From the Box plot in Figure 30, we can see that all color type gems have outliers with respect to price. From Figure 29, it is seen that the least priced cubic zirconia gem seems to be ‘E’ type while ‘J’ and ‘I’ coloured cubic zirconia gems seem to be expensive.

3. Clarity:

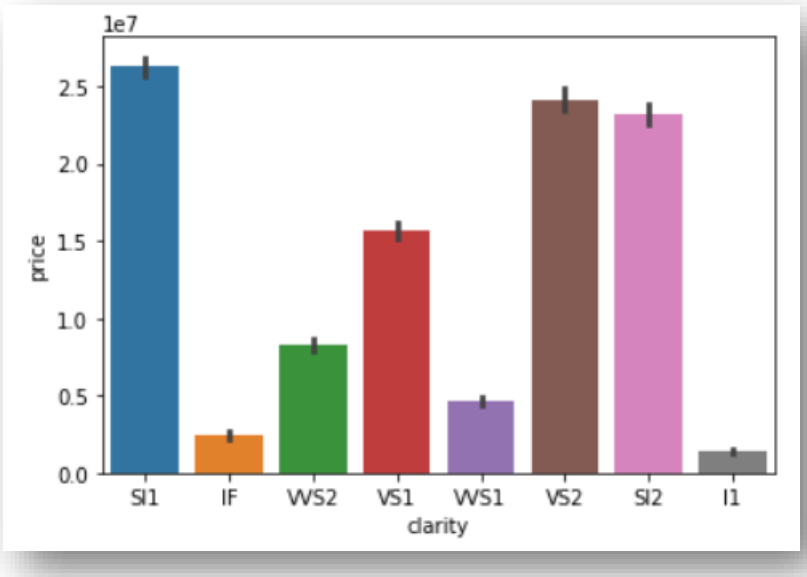


Figure 31: Countplot of 'clarity' and 'price'

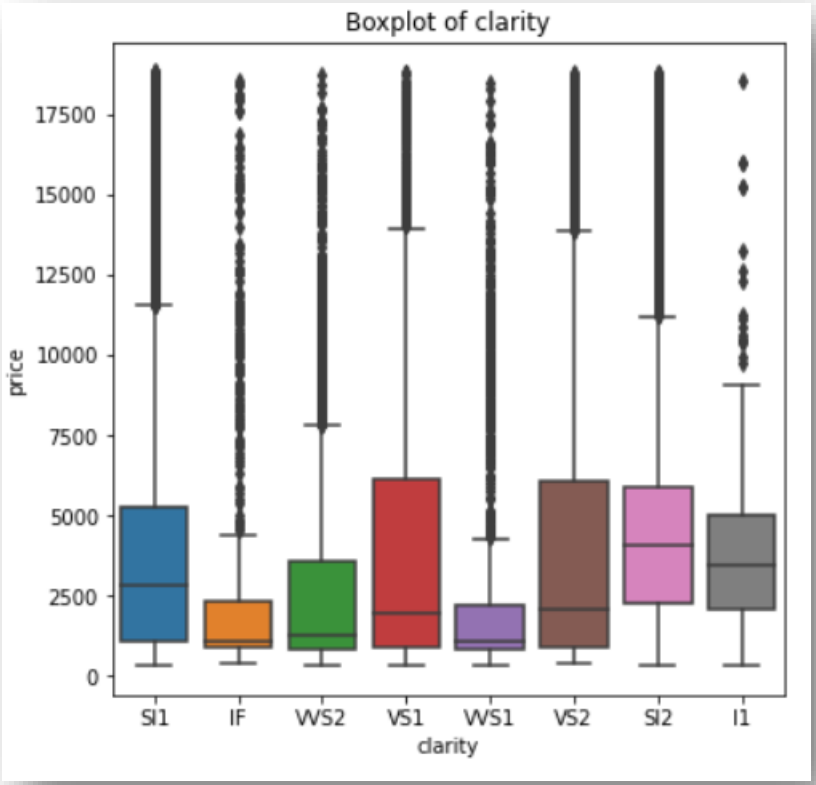


Figure 32: Box plot of 'clarity' and 'price'

From the Box plot in Figure 32, we can see that all clarity type gems have outliers with respect to price. From Figure 31, it is seen that the least priced cubic zirconia gem seems to be 'I1' type while 'VS2', 'SI1' and 'SI2' clarity cubic zirconia gems seem to be expensive.

Multivariate Analysis:

Numerical variables:

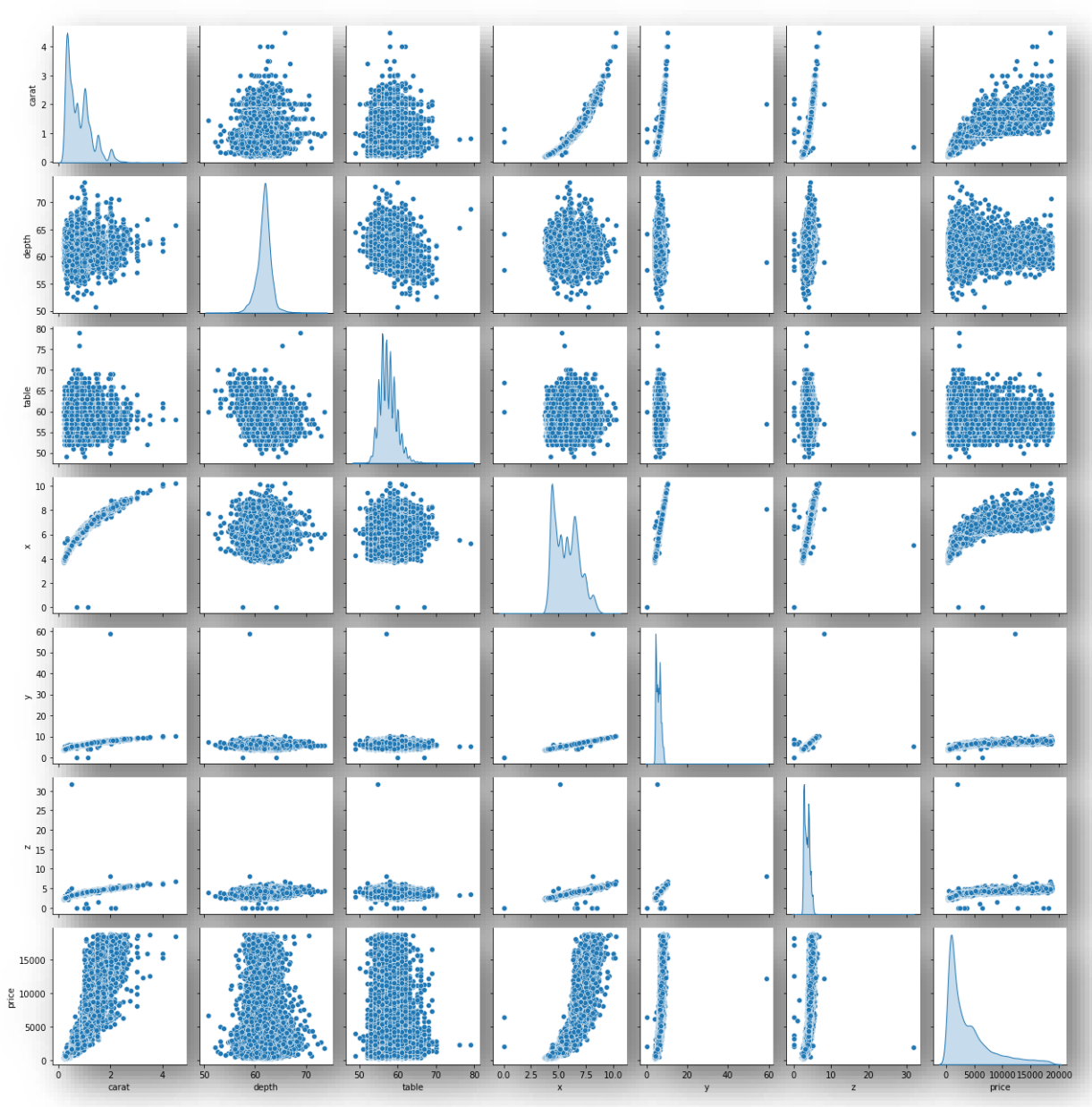


Figure 33: Pair plot showing the multivariate analysis of the numerical variables in the dataset

Multivariate analysis is done using the help of a pair plot to understand the relationship between all the numerical values in the dataset. Pair plot can be used to compare all the variables with each

other to understand the patterns or trends in the dataset. Figure 33 shows the pair plot of the dataset.

1.2 Impute null values if present, also check for the values which are equal to zero. Do they have any meaning or do we need to change them or drop them? Check for the possibility of combining the sub levels of a ordinal variables and take actions accordingly. Explain why you are combining these sub levels with appropriate reasoning.

```
df1_drop.isnull().sum()

carat      0
cut        0
color      0
clarity    0
depth      0
table      0
x          0
y          0
z          0
price      0
dtype: int64
```

Figure 34: Missing value of the columns in the dataset after imputing with median

'depth' column has 674 missing or null values which is seen in Table 4. These null values of 'depth' column have been imputed using the median value of that column. Figure 34 shows the missing values of the columns after imputing 'depth' missing values with the median value.

The dataset is now checked for the presence of '0' values.

```
Count of zeros in column  carat  is:  0
Count of zeros in column  cut  is:  0
Count of zeros in column  color  is:  0
Count of zeros in column  clarity  is:  0
Count of zeros in column  depth  is:  0
Count of zeros in column  table  is:  0
Count of zeros in column  x  is:  2
Count of zeros in column  y  is:  2
Count of zeros in column  z  is:  8
Count of zeros in column  price  is:  0
```

Figure 35: Count of '0' values in each column

Figure 35 shows the number of '0' values present in each column. It is seen that there are zero values in 'x', 'y' and 'z' column. 'x'. 'y' and 'z' are the length, width and height of the cubic zirconia

respectively. These measurements cannot be zero as it is a gem and it does not make sense when the length, width or height is zero. Therefore, these zero values are also imputed with the median values.

```
Count of zeros in column  carat  is:  0
Count of zeros in column  cut  is:  0
Count of zeros in column  color  is:  0
Count of zeros in column  clarity  is:  0
Count of zeros in column  depth  is:  0
Count of zeros in column  table  is:  0
Count of zeros in column  x  is:  0
Count of zeros in column  y  is:  0
Count of zeros in column  z  is:  0
Count of zeros in column  price  is:  0
```

Figure 36: Count of '0' values in each column after imputing with median

Figure 36 shows the number of '0' values present in each column after imputing with the median values.

From Table 5, we can see that the maximum value of 'y' and 'z' is 58.9 and 31.8 respectively. The 'depth' value must have been incorrectly entered in 'y' and 'z' columns. Therefore, this extreme maximum value is replaced with the median value of their respective columns.

Table 15: Description of the dataset after imputing values for 'y' and 'z'

	carat	depth	table	x	y	z	price
count	26933.000000	26933.000000	26933.000000	26933.000000	26933.000000	26933.000000	26933.000000
mean	0.798010	61.746701	57.455950	5.729769	5.731550	3.537765	3937.526120
std	0.477237	1.393875	2.232156	1.126285	1.117994	0.696400	4022.551862
min	0.200000	50.800000	49.000000	3.730000	3.710000	1.070000	326.000000
25%	0.400000	61.100000	56.000000	4.710000	4.720000	2.900000	945.000000
50%	0.700000	61.800000	57.000000	5.690000	5.700000	3.520000	2375.000000
75%	1.050000	62.500000	59.000000	6.550000	6.540000	4.040000	5356.000000
max	4.500000	73.600000	79.000000	10.230000	10.160000	8.060000	18818.000000

Table 15 shows the description of the dataset after imputing the maximum value of 'y' and 'z' column with their median values.

```
CUT :  5
Fair          780
Good         2435
Very Good    6027
Premium      6886
Ideal       10805
Name: cut, dtype: int64


COLOR :  7
J    1440
I    2765
D    3341
H    4095
F    4723
E    4916
G    5653
Name: color, dtype: int64


CLARITY :  8
I1        364
IF        891
VVS1     1839
VVS2     2530
VS1      4087
SI2      4564
VS2      6093
SI1      6565
Name: clarity, dtype: int64
```

Figure 37: Value counts of the categorical variables

There are 3 categorical variables in this dataset – 'cut', 'color' and 'clarity'. Figure 37 shows the different categorical variables and the value counts for each of the types in the different categories. Linear Regression requires all the columns to be numerical. Hence these categorical columns have to be changed into numerical columns. **Label encoding method** can be used to convert the categorical columns to numerical columns.

Without combining the sub levels of ordinal variables:

Each type in the categorical column is assigned a unique number as seen in Figure 38.

```
df1_drop = df1_drop.replace({'Ideal':5,'Premium':4,'Very Good':3,'Good':2,'Fair':1})
df1_drop = df1_drop.replace({'J':7, 'I':6,'H':5,'G':4,'F':3,'E':2,'D':1})
df1_drop = df1_drop.replace({'I1':8, 'SI2':7,'SI1':6,'VS2':5,'VS1':4,'VVS2':3,'VVS1':2,'IF':1})
```

Figure 38: Label encoding for categorical column without combining the sub levels of ordinal variables

Table 16: Head of the dataset after doing label encoding without combining the sub levels of ordinal variables

	carat	cut	color	clarity	depth	table	x	y	z	price
0	0.30	5	2	6	62.1	58.0	4.27	4.29	2.66	499
1	0.33	4	4	1	60.8	58.0	4.42	4.46	2.70	984
2	0.90	3	2	3	62.2	60.0	6.04	6.12	3.78	6289
3	0.42	5	3	4	61.6	56.0	4.82	4.80	2.96	1082
4	0.31	5	3	2	60.4	59.0	4.35	4.43	2.65	779

Table 16 shows the head of the dataset after performing label encoding.

Combining the sub levels of ordinal variables:

Each type in the categorical column is assigned a number as seen in Figure 39. However, in this case, 1 or 2 types each category has been assigned with the same number in order to combine the sub levels of the ordinal variables. **This is done to reduce the number of types and hence make the prediction better.**

```
df1_combine = df1_combine.replace({'Ideal':4,'Premium':3,'Very Good':2,'Good':1,'Fair':1})
df1_combine = df1_combine.replace({'J':4, 'I':3,'H':3,'G':2,'F':2,'E':1,'D':1})
df1_combine = df1_combine.replace({'I1':5, 'SI2':4,'SI1':4,'VS2':3,'VS1':3,'VVS2':2,'VVS1':2,'IF':1})
```

Figure 39: Label encoding for categorical column after combining the sub levels of ordinal variables

Table 17: Head of the dataset after doing label encoding after combining the sub levels of ordinal variables

	carat	cut	color	clarity	depth	table	x	y	z	price
0	0.30	4	1	4	62.1	58.0	4.27	4.29	2.66	499
1	0.33	3	2	1	60.8	58.0	4.42	4.46	2.70	984
2	0.90	2	1	2	62.2	60.0	6.04	6.12	3.78	6289
3	0.42	4	2	3	61.6	56.0	4.82	4.80	2.96	1082
4	0.31	4	2	2	60.4	59.0	4.35	4.43	2.65	779

Table 17 shows the head of the dataset after performing label encoding after combining the sub levels of ordinal variables. It can be seen that all the columns are now numerical columns.

Table 18: Data type after label encoding

```
carat      float64
cut          int64
color        int64
clarity      int64
depth      float64
table      float64
x          float64
y          float64
z          float64
price        int64
dtype: object
```

It can be seen from Table 18 that all the columns are now numerical columns.

1.3 Encode the data (having string values) for Modelling. Split the data into train and test (70:30). Apply Linear regression using scikit learn. Perform checks for significant variables using appropriate method from statsmodel. Create multiple models and check the performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare. Compare these models and select the best one with appropriate reasoning.

Linear regression using scikit learn:

Without combining the sub levels of ordinal variables:

```
X = df1_drop.drop('price', axis=1)
y = df1_drop.pop('price')

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30 , random_state=1)
```

Figure 40: Splitting the data into train and test without combining the sub levels of ordinal variables

The data is first split into train and test data as shown in Figure 40**Error! Reference source not found.** before building the linear regression model without combining the sub levels of ordinal levels. There is no fixed rule for separation training and testing data sets. Most of the researchers use **70:30 ratio** for separation data sets. The same ratio was used in this dataset to split the data into train and test. The random state was set to be 1.

```
The coefficient for carat is 11113.015235152121
The coefficient for cut is 116.82821221730455
The coefficient for color is -333.96821278390996
The coefficient for clarity is -499.8134377440133
The coefficient for depth is -17.496305203938242
The coefficient for table is -23.049959237648125
The coefficient for x is -2012.041277516459
The coefficient for y is 1636.639565290995
The coefficient for z is -991.4279415804899
```

Figure 41: Coefficient values for each column

There are 9 attributes in this dataset and hence there are 9 coefficients. For every one-unit change in x (the different independent variables), y (the dependent variable) changes m (the coefficient value) times. The coefficient values for each column are shown in Figure 41.

```
The intercept for our model is 6351.735739185806
```

Figure 42: Intercept of the model

The intercept of the model without combining the sub levels of ordinal levels is found to be 6351.74 which is seen in Figure 42. The intercept is the constant or the bias. The intercept is the value of the dependent variable when the product of coefficient and independent variable is 0.

```
linear_model.score(X_train, y_train)
0.9093303376203457

linear_model.score(X_test, y_test)
0.908779109283248
```

Figure 43: Coefficient of determinant value for train and test data without combining the sub levels of ordinal levels

The coefficient of determinant or r^2 determines the fitness of a model. The coefficient of determinant value ranges from 0 to 1. The closer the value is to 1, the better the model is. Figure 43 shows the coefficient of determinant value for the train and test data without combining the sub levels of ordinal levels. It is seen that the value is 0.909 for the train data and 0.908 for the test data.

```
mse_without_combine_train_lr = np.mean((linear_model.predict(X_train)-y_train)**2)
math.sqrt(mse_without_combine_train_lr)
print('RMSE train:', math.sqrt(mse_without_combine_train_lr))
```

```
RMSE train: 1207.3039449259259
```

Figure 44: Square root of mean square error for training data

```
mse_without_combine_test_lr = np.mean((linear_model.predict(X_test)-y_test)**2)
math.sqrt(mse_without_combine_test_lr)
print('RMSE test:', math.sqrt(mse_without_combine_test_lr))
```

```
RMSE test: 1224.0230643756115
```

Figure 45: Square root of mean square error for test data

The sum of squared errors is checked by predicting value of y for test cases and subtracting from the actual y for the test cases. Square root of mean square error is the standard deviation i.e., average variance between predicted and actual. This is done for both training and test data and the RMSE values are shown in Figure 44 and Figure 45. So, there is an average of 1207.3 and 1224.0 price difference for the training and test data respectively from real price.

Combining the sub levels of ordinal variables:

```
X1 = df1_combine.drop('price', axis=1)
y1 = df1_combine.pop('price')

X1_train, X1_test, y1_train, y1_test = train_test_split(X1, y1, test_size=0.30 , random_state=1)
```

Figure 46: Splitting the data into train and test combining the sub levels of ordinal variables

The data is first split into train and test data as shown in Figure 46 before building the linear regression model combining the sub levels of ordinal levels. There is no fixed rule for separation training and testing data sets. Most of the researchers use **70:30 ratio** for separation data sets. The same ratio was used in this dataset to split the data into train and test. The random state was set to be 1.

```
The coefficient for carat is 11062.020191930626
The coefficient for cut is 98.84116571058883
The coefficient for color is -611.3928235427279
The coefficient for clarity is -897.6524587507201
The coefficient for depth is -28.31698801772922
The coefficient for table is -31.62443962400107
The coefficient for x is -2124.7818292157544
The coefficient for y is 1723.3930641483184
The coefficient for z is -987.8784627376036
```

Figure 47: Coefficient values for each column

There are 9 attributes in this dataset and hence there are 9 coefficients. For every one-unit change in x (the different independent variables), y (the dependent variable) changes m (the coefficient value) times. The coefficient values for each column are shown in Figure 47.

```
The intercept for our model is 8318.014886555964
```

Figure 48: Intercept of the model

The intercept of the model without combining the sub levels of ordinal levels is found to be 8318.01 which is seen in Figure 48. The intercept is the constant or the bias. The intercept is the value of the dependent variable when the product of coefficient and independent variable is 0.

```
linear_model_combine.score(X1_train, y1_train)
```

```
0.9050077417696945
```

```
linear_model_combine.score(X1_test, y1_test)
```

```
0.9049388530007738
```

Figure 49: Coefficient of determinant value for train and test data combining the sub levels of ordinal levels

The coefficient of determinant or r^2 determines the fitness of a model. The coefficient of determinant value ranges from 0 to 1. The closer the value is to 1, the better the model is. Figure 49 shows the coefficient of determinant value for the train and test data combining the sub levels of ordinal levels. It is seen that the value is 0.905 for the train data and 0.905 for the test data.

```
mse_combine_train_lr = np.mean((linear_model_combine.predict(X1_train)-y1_train)**2)
math.sqrt(mse_combine_train_lr)
print('RMSE train:', math.sqrt(mse_combine_train_lr))
```

```
RMSE train: 1235.747460132434
```

Figure 50: Square root of mean square error for training data

```
mse_combine_test_lr = np.mean((linear_model_combine.predict(X1_test)-y1_test)**2)
math.sqrt(mse_combine_test_lr)
print('RMSE test:', math.sqrt(mse_combine_test_lr))
```

```
RMSE test: 1249.5221873002079
```

Figure 51: Square root of mean square error for test data

The sum of squared errors is checked by predicting value of y for test cases and subtracting from the actual y for the test cases. Square root of mean square error is the standard deviation i.e., average variance between predicted and actual. This is done for both training and test data and the RMSE values are shown in Figure 50 and Figure 51. So, there is an average of 1235.7 and 1249.5 price difference for the training and test data respectively from real price.

Linear Regression using Stats model:

The coefficient of determinant or r^2 determines the fitness of a model. The coefficient of determinant value ranges from 0 to 1. The closer the value is to 1, the better the model is. However, the coefficient of determinant or r^2 is not a reliable metric as it always increases with addition of more attributes even if the attributes have no influence on the predicted variable. Therefore, adjusted r^2 should be used which removes the statistical chance that improves r^2. When useless attributes are added, adjusted r^2 will decrease and vice versa.

Adjusted r^2 = r^2 – statistical fluke

Adjusted r^2 will always be less than or equal to r^2. it will never be more than r^2.

Scikit does not provide a facility for adjusted r^2. Therefore, statsmodel is used.

Statsmodel is a library that gives results similar to what is obtained in R language. This library expects the independent and dependent variables to be given in one single dataframe without being split into train and test.

Without combining the sub levels of ordinal variables:

The independent and dependent variables are concatenated into one single dataframe shown in Table 19.

Table 19: Concatenating the independent and dependent variables for statsmodel

	carat	cut	color	clarity	depth	table	x	y	z	price
22114	0.34	3	5	5	62.4	60.0	4.41	4.44	2.76	537
2275	0.30	5	2	5	61.2	55.0	4.35	4.31	2.65	844
19183	0.50	5	4	6	62.5	57.0	5.09	5.05	3.17	1240
5030	1.10	2	2	7	63.3	56.0	6.53	6.58	4.15	4065
25414	1.02	4	3	7	61.1	62.0	6.54	6.49	3.98	4057

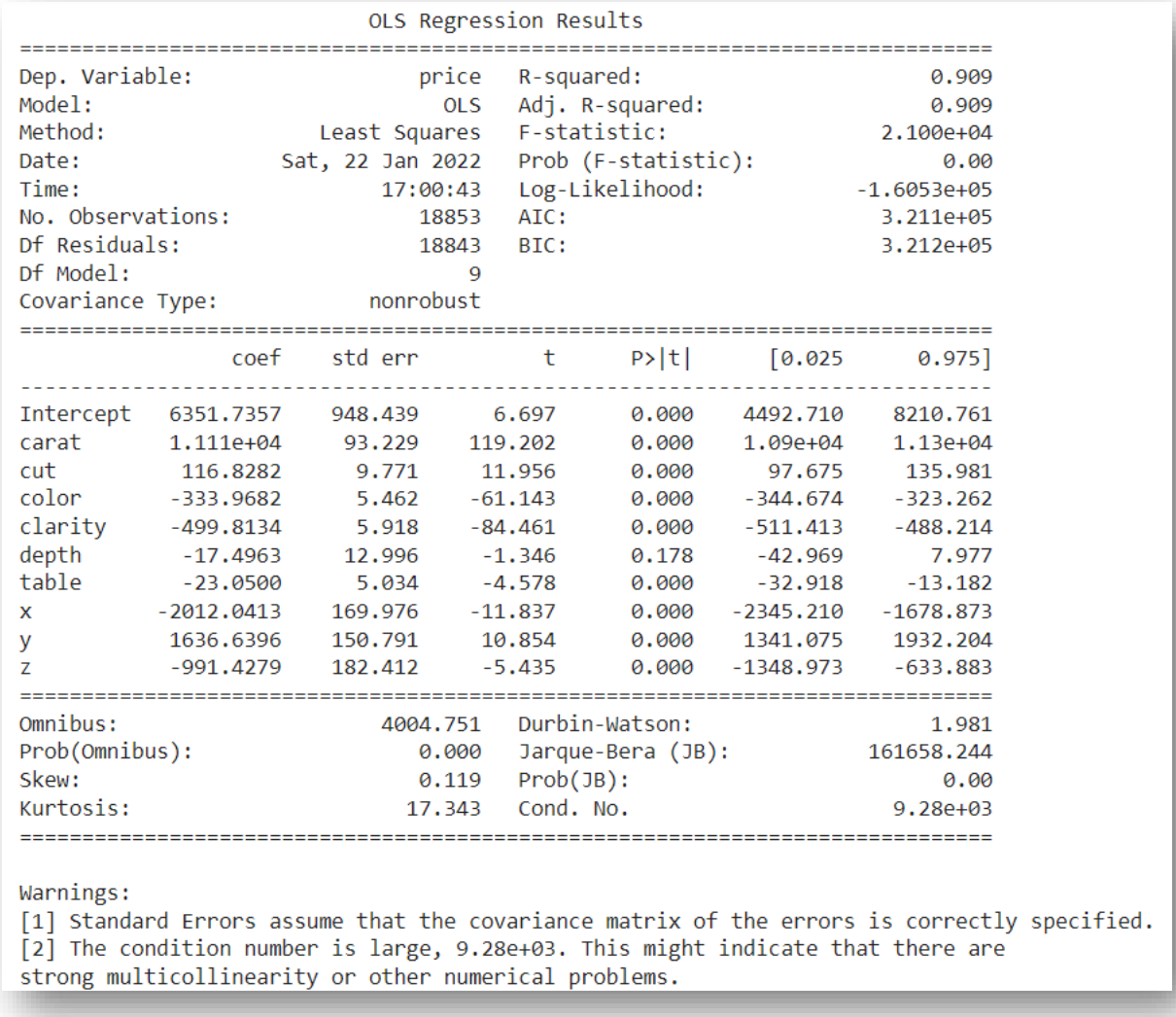
```
Intercept     6351.735739
carat        11113.015235
cut            116.828212
color         -333.968213
clarity       -499.813438
depth          -17.496305
table          -23.049959
x            -2012.041278
y             1636.639565
z             -991.427942
dtype: float64
```

Figure 52: Intercept and Coefficient values for each column using statsmodel

There are 9 attributes in this dataset and hence there are 9 coefficients. For every one-unit change in x (the different independent variables), y (the dependent variable) changes m (the coefficient value) times. The coefficient values for each column are shown in Figure 52.

The intercept of the statsmodel without combining the sub levels of ordinal levels is found to be 6351.74 which is seen in Figure 52. The intercept is the constant or the bias. The intercept is the value of the dependent variable when the product of coefficient and independent variable is 0.

Inferential statistics for the model without combining the sub levels of ordinal levels is calculated and shown in Figure 53.



OLS Regression Results

Dep. Variable:	price	R-squared:	0.909
Model:	OLS	Adj. R-squared:	0.909
Method:	Least Squares	F-statistic:	2.100e+04
Date:	Sat, 22 Jan 2022	Prob (F-statistic):	0.00
Time:	17:00:43	Log-Likelihood:	-1.6053e+05
No. Observations:	18853	AIC:	3.211e+05
Df Residuals:	18843	BIC:	3.212e+05
Df Model:	9		
Covariance Type:	nonrobust		

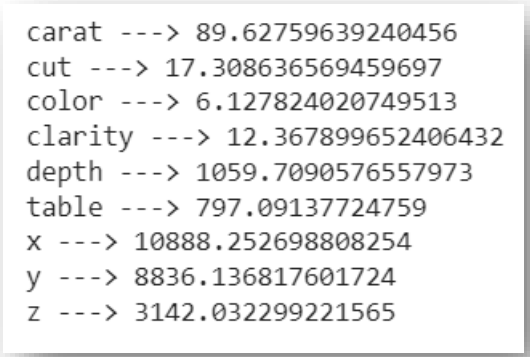
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	6351.7357	948.439	6.697	0.000	4492.710	8210.761
carat	1.111e+04	93.229	119.202	0.000	1.09e+04	1.13e+04
cut	116.8282	9.771	11.956	0.000	97.675	135.981
color	-333.9682	5.462	-61.143	0.000	-344.674	-323.262
clarity	-499.8134	5.918	-84.461	0.000	-511.413	-488.214
depth	-17.4963	12.996	-1.346	0.178	-42.969	7.977
table	-23.0500	5.034	-4.578	0.000	-32.918	-13.182
x	-2012.0413	169.976	-11.837	0.000	-2345.210	-1678.873
y	1636.6396	150.791	10.854	0.000	1341.075	1932.204
z	-991.4279	182.412	-5.435	0.000	-1348.973	-633.883

Omnibus:	4004.751	Durbin-Watson:	1.981
Prob(Omnibus):	0.000	Jarque-Bera (JB):	161658.244
Skew:	0.119	Prob(JB):	0.00
Kurtosis:	17.343	Cond. No.	9.28e+03

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.28e+03. This might indicate that there are strong multicollinearity or other numerical problems.

Figure 53: Inferential statistics of statsmodel without combining the sub levels of ordinal variables

It is seen from Figure 53 that the adjusted r^2 value is equal to r^2 value. The condition number is large, 9.28e+03. This might indicate that there are strong multicollinearity or other numerical problems.



```
carat ---> 89.62759639240456
cut ---> 17.308636569459697
color ---> 6.127824020749513
clarity ---> 12.367899652406432
depth ---> 1059.7090576557973
table ---> 797.09137724759
x ---> 10888.252698808254
y ---> 8836.136817601724
z ---> 3142.032299221565
```

Figure 54: Variance Inflation Factor without combining the sub levels of ordinal variables

The variance inflation factor was calculated for combining the sub levels of ordinal variables and the values are shown in Figure 54. All the values are found to be above 5. This indicates the presence of a strong multicollinearity.

Combining the sub levels of ordinal variables:

The independent and dependent variables are concatenated into one single dataframe shown in Table 20.

Table 20: Concatenating the independent and dependent variables for statsmodel

	carat	cut	color	clarity	depth	table	x	y	z	price
22114	0.34	2	3	3	62.4	60.0	4.41	4.44	2.76	537
2275	0.30	4	1	3	61.2	55.0	4.35	4.31	2.65	844
19183	0.50	4	2	4	62.5	57.0	5.09	5.05	3.17	1240
5030	1.10	1	1	4	63.3	56.0	6.53	6.58	4.15	4065
25414	1.02	3	2	4	61.1	62.0	6.54	6.49	3.98	4057

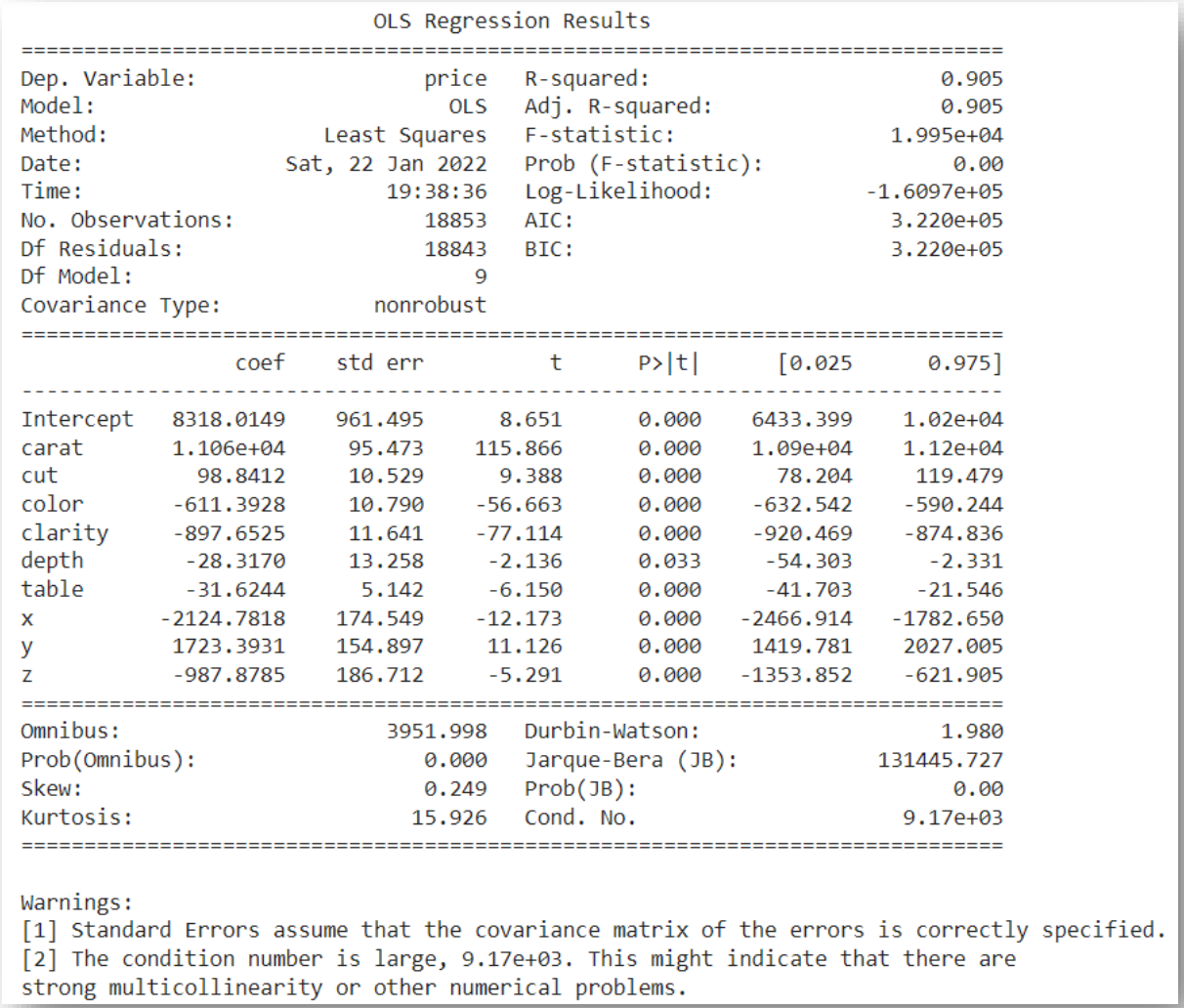
```
Intercept     8318.014887
carat        11062.020192
cut             98.841166
color         -611.392824
clarity       -897.652459
depth          -28.316988
table          -31.624440
x            -2124.781829
y             1723.393064
z             -987.878463
dtype: float64
```

Figure 55: Intercept and Coefficient values for each column using statsmodel

There are 9 attributes in this dataset and hence there are 9 coefficients. For every one-unit change in x (the different independent variables), y (the dependent variable) changes m (the coefficient value) times. The coefficient values for each column are shown in Figure 55.

The intercept of the statsmodel combining the sub levels of ordinal levels is found to be 8318.01 which is seen in Figure 55. The intercept is the constant or the bias. The intercept is the value of the dependent variable when the product of coefficient and independent variable is 0.

Inferential statistics for the model combining the sub levels of ordinal levels is calculated and shown in Figure 56.



OLS Regression Results

Dep. Variable:	price	R-squared:	0.905
Model:	OLS	Adj. R-squared:	0.905
Method:	Least Squares	F-statistic:	1.995e+04
Date:	Sat, 22 Jan 2022	Prob (F-statistic):	0.00
Time:	19:38:36	Log-Likelihood:	-1.6097e+05
No. Observations:	18853	AIC:	3.220e+05
Df Residuals:	18843	BIC:	3.220e+05
Df Model:	9		
Covariance Type:	nonrobust		

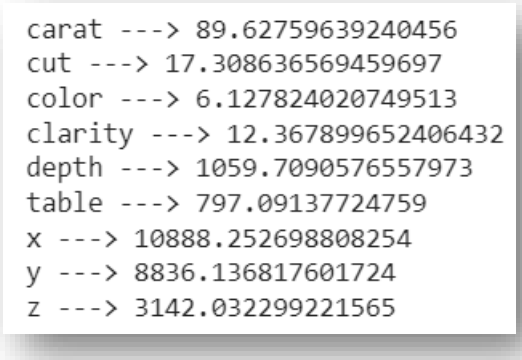
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	8318.0149	961.495	8.651	0.000	6433.399	1.02e+04
carat	1.106e+04	95.473	115.866	0.000	1.09e+04	1.12e+04
cut	98.8412	10.529	9.388	0.000	78.204	119.479
color	-611.3928	10.790	-56.663	0.000	-632.542	-590.244
clarity	-897.6525	11.641	-77.114	0.000	-920.469	-874.836
depth	-28.3170	13.258	-2.136	0.033	-54.303	-2.331
table	-31.6244	5.142	-6.150	0.000	-41.703	-21.546
x	-2124.7818	174.549	-12.173	0.000	-2466.914	-1782.650
y	1723.3931	154.897	11.126	0.000	1419.781	2027.005
z	-987.8785	186.712	-5.291	0.000	-1353.852	-621.905

Omnibus:	3951.998	Durbin-Watson:	1.980
Prob(Omnibus):	0.000	Jarque-Bera (JB):	131445.727
Skew:	0.249	Prob(JB):	0.00
Kurtosis:	15.926	Cond. No.	9.17e+03

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.17e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

Figure 56: Inferential statistics of statsmodel combining the sub levels of ordinal variables

It is seen from Figure 56 that the adjusted r^2 value is equal to r^2 value. The condition number is large, 9.17e+03. This might indicate that there are strong multicollinearity or other numerical problems.



```
carat ---> 89.62759639240456
cut ---> 17.308636569459697
color ---> 6.127824020749513
clarity ---> 12.367899652406432
depth ---> 1059.7090576557973
table ---> 797.09137724759
x ---> 10888.252698808254
y ---> 8836.136817601724
z ---> 3142.032299221565
```

Figure 57: Variance Inflation Factor combining the sub levels of ordinal variables

The variance inflation factor was calculated for combining the sub levels of ordinal variables and the values are shown in Figure 57. All the values are found to be above 5. This indicates the presence of a strong multicollinearity.

Inference:

```
Linear Regression model without combining the sub levels of ordinal variables:
R-Squared train: 0.9093303376203457
R-Squared test: 0.908779109283248
RMSE train: 1207.3039449259259
RMSE test: 1224.0230643756115
Adjusted R-Squared: 0.909

Linear Regression model combining the sub levels of ordinal variables:
R-Squared train: 0.9050077417696945
R-Squared test: 0.9049388530007738
RMSE train: 1235.747460132434
RMSE test: 1249.5221873002079
Adjusted R-Squared: 0.905
```

Figure 58: Linear Regression Model R-Squared, RMSE and Adjusted R-squared scores

Figure 58 shows the coefficient of determinant value for the train and test data without combining the sub levels of ordinal levels. It is seen that the value is 0.909 for the train data and 0.908 for the test data. The RMSE values are 1207.3 and 1224.0 for the training and test data respectively.

Figure 58 shows the coefficient of determinant value for the train and test data combining the sub levels of ordinal levels. It is seen that the value is 0.905 for the train data and 0.905 for the test data. The RMSE values are 1235.7 and 1249.5 for the training and test data respectively.

From these data we can observe that the coefficient of determinant value without combining the sub levels of ordinal levels and combining the sub levels of ordinal levels is good for both.

However, in both cases the **RMSE values are really high** and this can be **improved by scaling the data.**

Scaled model:

Scaling is a way of representing a dataset. Scaling needs to be done as a dataset has features or variables with different ‘weights’ for each feature. The independent attributes have different units and scales of measurement. In such cases, it is suggested to transform the features so that all the features are in the same ‘scale’. This is called scaling. Scaling also makes it easier to compare the features now that the weightage on each feature is the same.

Scaling can be done by Z-score method and Min-max method. Z-score method is used when you want to centralize the data (weight-based). Example: principal component analysis (PCA), neural network etc., Min-max method is used when the data is distance based. Example: Clustering, KNN etc., Models can be implemented after scaling is done.

Scaling is done on all the numerical variables of the dataset. Among the 10 numerical variables, depth, x, y and z are the variables in which the values are associated with the measurement of the cubic zirconia and carat is the variable in which the values are given as weight of the cubic zirconia.

Therefore, the dataset has to be scaled. Only then all the variables can be compared and weighed equally.

Z-score method is used for scaling this dataset.

Scaled Linear regression using scikit learn:

Combining the sub levels of ordinal variables:

Scaling is done for the dataset combining the sub levels of ordinal levels as the model with combining the sub levels of ordinal levels is proposed as the best model.

Table 21: Description of the dataset before scaling

	carat	cut	color	clarity	depth	table	x	y	z	price
count	26933.000000	26933.000000	26933.000000	26933.000000	26933.000000	26933.000000	26933.000000	26933.000000	26933.000000	26933.000000
mean	0.798010	2.938663	2.055063	3.211859	61.746701	57.455950	5.729769	5.731550	3.537765	3937.526120
std	0.477237	1.048197	0.878716	0.846734	1.393875	2.232156	1.126285	1.117994	0.696400	4022.551862
min	0.200000	1.000000	1.000000	1.000000	50.800000	49.000000	3.730000	3.710000	1.070000	326.000000
25%	0.400000	2.000000	1.000000	3.000000	61.100000	56.000000	4.710000	4.720000	2.900000	945.000000
50%	0.700000	3.000000	2.000000	3.000000	61.800000	57.000000	5.690000	5.700000	3.520000	2375.000000
75%	1.050000	4.000000	3.000000	4.000000	62.500000	59.000000	6.550000	6.540000	4.040000	5356.000000
max	4.500000	4.000000	4.000000	5.000000	73.600000	79.000000	10.230000	10.160000	8.060000	18818.000000

Table 22: Description of the dataset after scaling

	carat	cut	color	clarity	depth	table	x	y	z	price
count	2.693300e+04	2.693300e+04	2.693300e+04	2.693300e+04	2.693300e+04	2.693300e+04	2.693300e+04	2.693300e+04	2.693300e+04	2.693300e+04
mean	6.794979e-17	-2.013281e-16	4.237587e-17	-5.163961e-16	1.612401e-15	-2.574881e-15	-9.027297e-16	-4.386480e-16	1.531137e-16	-1.112985e-19
std	1.000019e+00	1.000019e+00	1.000019e+00	1.000019e+00	1.000019e+00	1.000019e+00	1.000019e+00	1.000019e+00	1.000019e+00	1.000019e+00
min	-1.253091e+00	-1.849555e+00	-1.200708e+00	-2.612272e+00	-7.853574e+00	-3.788313e+00	-1.775577e+00	-1.808227e+00	-3.543671e+00	-8.978363e-01
25%	-8.340039e-01	-8.955183e-01	-1.200708e+00	-2.502119e-01	-4.639676e-01	-6.522737e-01	-9.054438e-01	-9.048067e-01	-9.158200e-01	-7.439510e-01
50%	-2.053739e-01	5.851811e-02	-6.266366e-02	-2.502119e-01	3.823865e-02	-2.042681e-01	-3.531027e-02	-2.822050e-02	-2.551000e-02	-3.884487e-01
75%	5.280277e-01	1.012555e+00	1.075381e+00	9.308182e-01	5.404449e-01	6.917430e-01	7.282763e-01	7.231391e-01	7.212016e-01	3.526369e-01
max	7.757273e+00	1.012555e+00	2.213426e+00	2.111848e+00	8.504002e+00	9.651855e+00	3.995716e+00	3.961141e+00	6.493857e+00	3.699331e+00

From Table 21 and Table 22, we are able to see how the values have changed in scale. The values may seem to be different but however they are only scaled. The dataset is brought into one unit of comparison.

```
X1_scaled = df1_combine_scaled.drop('price', axis=1)
y1_scaled = df1_combine_scaled['price']

X1_train_scaled, X1_test_scaled, y1_train_scaled, y1_test_scaled = train_test_split(X1_scaled, y1_scaled, test_size=0.30 , random_state=1)
```

Figure 59: Splitting the data into train and test combining the sub levels of ordinal variables

The data is first split into train and test data as shown in Figure 59**Error! Reference source not found.** before building the linear regression scaled model combining the sub levels of ordinal levels.

There is no fixed rule for separation training and testing data sets. Most of the researchers use **70:30 ratio** for separation data sets. The same ratio was used in this dataset to split the data into train and test. The random state was set to be 1.

```
The coefficient for carat is 1.312402403431688
The coefficient for cut is 0.025756052855796546
The coefficient for color is -0.13355723444478676
The coefficient for clarity is -0.18895293750800624
The coefficient for depth is -0.009812267287761415
The coefficient for table is -0.01754873540226526
The coefficient for x is -0.5949232395882987
The coefficient for y is 0.4789853369172661
The coefficient for z is -0.17102529688534732
```

Figure 60: Coefficient values for each column

There are 9 attributes in this dataset and hence there are 9 coefficients. For every one-unit change in x (the different independent variables), y (the dependent variable) changes m (the coefficient value) times. The coefficient values for each column are shown in Figure 60.

```
The intercept for our scaled model is 0.0004478928342919454
```

Figure 61: Intercept of the scaled model

The intercept of the scaled model without combining the sub levels of ordinal levels is found to be 0.000448 which is seen in Figure 61. The intercept is the constant or the bias. The intercept is the value of the dependent variable when the product of coefficient and independent variable is 0.

```
linear_model_combine_scaled.score(X1_train_scaled, y1_train_scaled)
```

```
0.9050077417696946
```

```
linear_model_combine_scaled.score(X1_test_scaled, y1_test_scaled)
```

```
0.9049388530007738
```

Figure 62: Coefficient of determinant value for scaled train and test data combining the sub levels of ordinal levels

The coefficient of determinant or r^2 determines the fitness of a model. The coefficient of determinant value ranges from 0 to 1. The closer the value is to 1, the better the model is. Figure 62 shows the coefficient of determinant value for the scaled train and test data combining the sub levels of ordinal levels. It is seen that the value is 0.905 for the train data and 0.904 for the test data.

```
mse_combine_train_lr_scaled = np.mean((linear_model_combine_scaled.predict(X1_train_scaled)-y1_train_scaled)**2)
math.sqrt(mse_combine_train_lr_scaled)
print('RMSE train:', math.sqrt(mse_combine_train_lr_scaled))
```

```
RMSE train: 0.30721055793092117
```

Figure 63: Square root of mean square error for scaled training data

```
mse_combine_test_lr_scaled = np.mean((linear_model_combine_scaled.predict(X1_test_scaled)-y1_test_scaled)**2)
math.sqrt(mse_combine_test_lr_scaled)
print('RMSE test:', math.sqrt(mse_combine_test_lr_scaled))
```

```
RMSE test: 0.31063499678682394
```

Figure 64: Square root of mean square error for scaled test data

The sum of squared errors is checked by predicting value of y for test cases and subtracting from the actual y for the test cases. Square root of mean square error is the standard deviation i.e., average variance between predicted and actual. This is done for both training and test data and the RMSE values are shown in Figure 63 and Figure 64. So, there is an average of 0.307 and 0.310 price difference for the training and test data respectively from real price.

Scaled Linear Regression using Stats model:

Combining the sub levels of ordinal variables:

The independent and dependent variables are concatenated into one single dataframe shown in Table 23.

Table 23: Concatenating the independent and dependent variables for statsmodel

	carat	cut	color	clarity	depth	table	x	y	z	price
22114	-0.959730	-0.895518	1.075381	-0.250212	0.468701	1.139749	-1.171811	-1.155260	-1.116858	-0.845381
2275	-1.043547	1.012555	-1.200708	-0.250212	-0.392224	-1.100279	-1.225085	-1.271542	-1.274816	-0.769060
19183	-0.624461	1.012555	-0.062664	0.930818	0.540445	-0.204268	-0.568045	-0.609630	-0.528104	-0.670613
5030	0.632799	-1.849555	-1.200708	0.930818	1.114395	-0.652274	0.710518	0.758918	0.879160	0.031690
25414	0.465165	0.058518	-0.062664	0.930818	-0.463968	2.035760	0.719397	0.678415	0.635043	0.029702

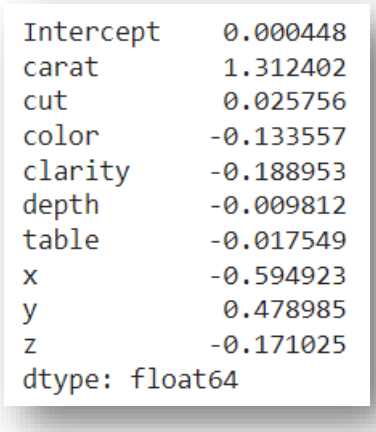


Figure 65: Intercept and Coefficient values for each column using statsmodel

There are 9 attributes in this dataset and hence there are 9 coefficients. For every one-unit change in x (the different independent variables), y (the dependent variable) changes m (the coefficient value) times. The coefficient values for each column are shown in Figure 65.

The intercept of the statsmodel combining the sub levels of ordinal levels is found to be 0.000448 which is seen in Figure 65. The intercept is the constant or the bias. The intercept is the value of the dependent variable when the product of coefficient and independent variable is 0.

Inferential statistics for the model combining the sub levels of ordinal levels is calculated and shown in Figure 66.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.905
Model:                            OLS   Adj. R-squared:                  0.905
Method:                 Least Squares   F-statistic:                 1.995e+04
Date:                Wed, 26 Jan 2022   Prob (F-statistic):               0.00
Time:                        09:27:11   Log-Likelihood:                -4500.5
No. Observations:               18853   AIC:                             9021.
Df Residuals:                   18843   BIC:                             9099.
Df Model:                           9
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0004      0.002      0.200      0.841      -0.004       0.005
carat          1.3124      0.011    115.866      0.000       1.290       1.335
cut            0.0258      0.003      9.388      0.000       0.020       0.031
color         -0.1336      0.002    -56.663      0.000      -0.138      -0.129
clarity       -0.1890      0.002    -77.114      0.000      -0.194      -0.184
depth         -0.0098      0.005     -2.136      0.033      -0.019      -0.001
table         -0.0175      0.003     -6.150      0.000      -0.023      -0.012
x             -0.5949      0.049    -12.173      0.000      -0.691      -0.499
y              0.4790      0.043     11.126      0.000       0.395       0.563
z             -0.1710      0.032     -5.291      0.000      -0.234      -0.108
==============================================================================
Omnibus:                     3951.998   Durbin-Watson:                   1.980
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           131445.727
Skew:                           0.249   Prob(JB):                         0.00
Kurtosis:                      15.926   Cond. No.                         56.5
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Figure 66: Inferential statistics of scaled statsmodel combining the sub levels of ordinal variables

It is seen from Figure 66that the adjusted r^2 value is equal to r^2 value.

```
carat ---> 89.62759639240456
cut ---> 17.308636569459697
color ---> 6.127824020749513
clarity ---> 12.367899652406432
depth ---> 1059.7090576557973
table ---> 797.09137724759
x ---> 10888.252698808254
y ---> 8836.136817601724
z ---> 3142.032299221565
```

Figure 67: Variance Inflation Factor combining the sub levels of ordinal variables

The variance inflation factor was calculated for combining the sub levels of ordinal variables and the values are shown in Figure 67. All the values are found to be above 5. This indicates the presence of a strong multicollinearity.

Inference:

```
Linear Regression model combining the sub levels of ordinal variables:
R-Squared train: 0.9050077417696945
R-Squared test: 0.9049388530007738
RMSE train: 1235.747460132434
RMSE test: 1249.5221873002079
Adjusted R-Squared: 0.905

Scaled Linear Regression model combining the sub levels of ordinal variables:
R-Squared train: 0.9050077417696946
R-Squared test: 0.9049388530007738
RMSE train: 0.30721055793092117
RMSE test: 0.31063499678682394
Adjusted R-Squared: 0.905
```

Figure 68: Linear Regression Model R-Squared, RMSE and Adjusted R-squared scores

Figure 68 shows the coefficient of determinant value for the train and test data combining the sub levels of ordinal levels. It is seen that the value is 0.905 for the train data and 0.905 for the test data. The RMSE values are 1235.7 and 1249.5 for the training and test data respectively. However, the RMSE values are really high and this was improved by scaling the data.

After scaling the data using Z-score method, the coefficient of determinant value for the train and test data and the adjusted R-squared value remained the same. At the same time, the RMSE values have been significantly reduced.

Taking these values into account, the **scaled model with combining the sub levels of ordinal levels is proposed as the best model** because in this model, the ordinal levels are combined. **This is done to reduce the number of types in the categorical variables and hence make the prediction better.** I believe lesser the categories of ordinal variables, better the prediction of the model will be. From the data, we can see that the coefficient of determinant value is also good. Therefore, the scaled model with combining the sub levels of ordinal levels is proposed as the best model.

1.4 Inference: Basis on these predictions, what are the business insights and recommendations.

Please explain and summarise the various steps performed in this project. There should be proper business interpretation and actionable insights present.

The recommendations for this problem are that more real time unstructured data and past data should be collected.

Carat is the most dominant factor in deciding the price of the cubic zirconia. It has a correlation value of 0.92 with price. Higher the carat, higher the price of the cubic zirconia.

Carat is measure of weight which has direct correlation with physical dimensions (x,y,z). Therefore, indirectly the length, width and height of the gem also have an influence on the price of the cubic zirconia.

Carat, x, y and z are the most important attributes in deciding the price of the cubic zirconia.

Ideal cut, colour G with clarity SI1 have a higher price. The price is also high for VS1, VS2 and SI2 clarity gems. Colour E, F, H and I also have high prices. In terms of cut, apart from Ideal, Premium Very Good cut have high prices. Therefore, the company can invest in purchasing more of these types of gems to improve their sales and thereby their profits. Having good quality gems will attract more customers. It will also add to the brand value and face value of the company if the quality of the gems is almost always good. The customers will feel that the company is reliable and will always have good quality gems. Therefore, the sales will increase.

It advisable to avoid diamonds of cut fair and good. Regarding Colour J and D will have less price. Clarity IF, VVS1, VVs2 and I1 have lower price and should be avoided.

Using these parameters cubic zirconia of higher price can be selected and avoid lower price for better marketability and profit.