LINEAR REGRESSION

PREPARED BY
MURALIDHARAN N

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 best and J the worst.
Clarity	Cubic zirconia 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 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). Perform Univariate and Bivariate Analysis.

Loading all the necessary library for the model building.

Now, reading the head and tail of the dataset to check whether data has been properly fed.

HEAD OF THE DATA

	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

TAIL OF THE DATA

	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
26962	26963	1.11	Premium	G	SI1	62.3	58.0	6.61	6.52	4.09	5408
26963	26964	0.33	Ideal	H	IF	61.9	55.0	4.44	4.42	2.74	1114
26964	26965	0.51	Premium	E	VS2	61.7	58.0	5.12	5.15	3.17	1656
26965	26966	0.27	Very Good	F	VVS2	61.8	56.0	4.19	4.20	2.60	682
26966	26967	1.25	Premium	J	SI1	62.0	58.0	6.90	6.88	4.27	5166

Checking the shape of the data (26967, 11)

Checking the info the data

```
<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
```

We have float, int and object data types in the data.

DATA DESCRIPTION

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Unnamed: 0	26967	NaN	NaN	NaN	13484	7784.85	1	6742.5	13484	20225.5	26967
carat	26967	NaN	NaN	NaN	0.798375	0.477745	0.2	0.4	0.7	1.05	4.5
cut	26967	5	Ideal	10816	NaN	NaN	NaN	NaN	NaN	NaN	NaN
color	26967	7	G	5661	NaN	NaN	NaN	NaN	NaN	NaN	NaN
clarity	26967	8	SI1	6571	NaN	NaN	NaN	NaN	NaN	NaN	NaN
depth	26270	NaN	NaN	NaN	61.7451	1.41286	50.8	61	61.8	62.5	73.6
table	26967	NaN	NaN	NaN	57.4561	2.23207	49	56	57	59	79
x	26967	NaN	NaN	NaN	5.72985	1.12852	0	4.71	5.69	6.55	10.23
y	26967	NaN	NaN	NaN	5.73357	1.16606	0	4.71	5.71	6.54	58.9
z	26967	NaN	NaN	NaN	3.53806	0.720624	0	2.9	3.52	4.04	31.8
price	26967	NaN	NaN	NaN	3939.52	4024.86	326	945	2375	5360	18818

We have both categorical and continuous data,

For categorical data we have cut, colour and clarity

For continuous data we have carat, depth, table, x. y, z and price

Price will be target variable.

Checking the duplicates in the data,

```
dups = df.duplicated()
print('Number of duplicate rows = %d' % (dups.sum()))
```

```
Number of duplicate rows = 0
```

Unique values in the categorical data

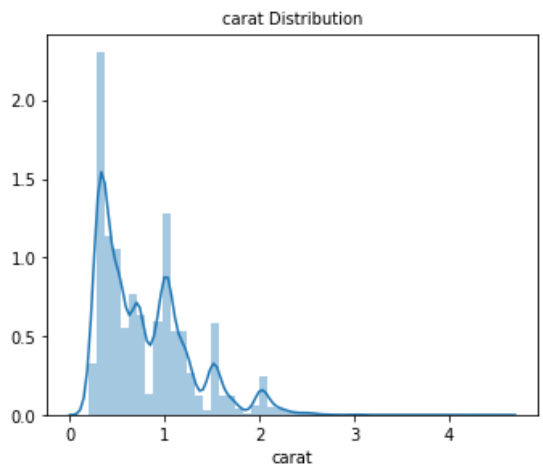
CUT: 5
Fair 781
Good 2441
Very Good 6030
Premium 6899
Ideal 10816

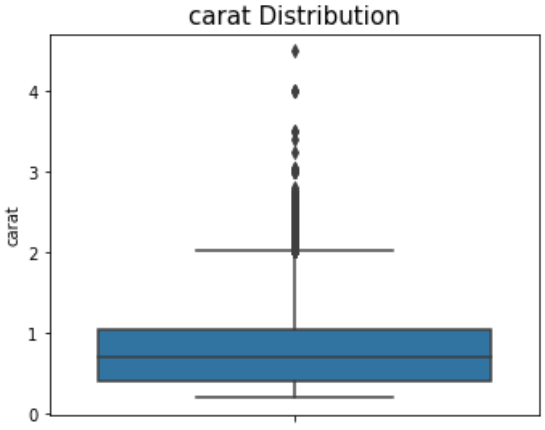
We have 5 cuts and the ideal seems to be most preferred cut

COLOR: 7
J 1443
I 2771
D 3344
H 4102
F 4729
E 4917
G 5661

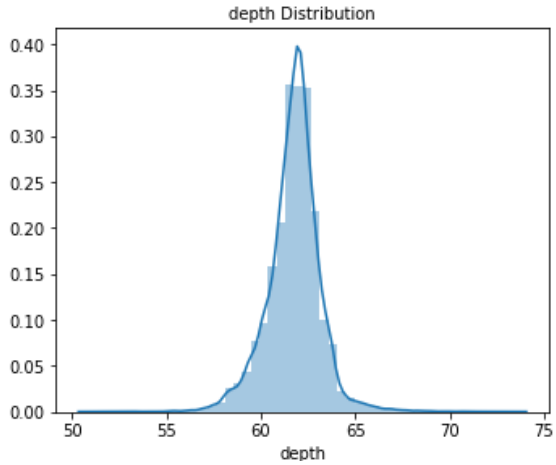
CLARITY: 8
I1 365
IF 894
VVS1 1839
VVS2 2531
VS1 4093
SI2 4575
VS2 6099
SI1 6571

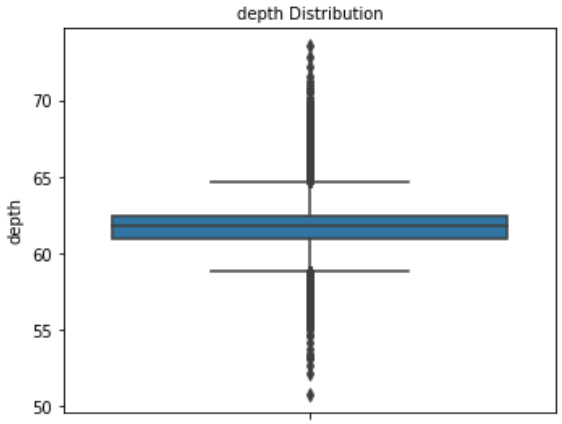
Univariate / Bivariate analysis





The distribution of data in carat seems to positively skewed, as there are multiple peaks points in the distribution there could multimode and the box plot of carat seems to have large number of outliers. In the range of 0 to 1 where majority of data lies.

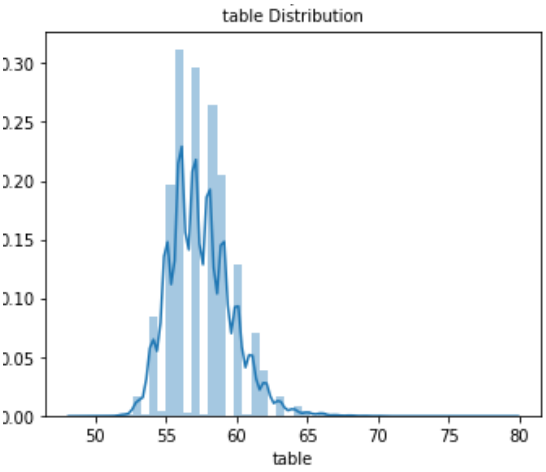


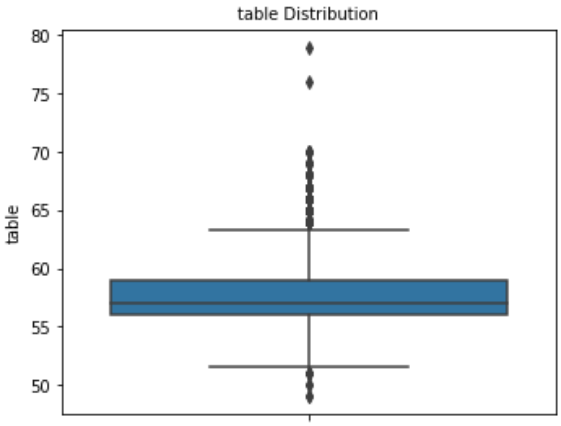


The distribution of depth seems to be normal distribution,

The depth ranges from 55 to 65

The box plot of the depth distribution holds many outliers.

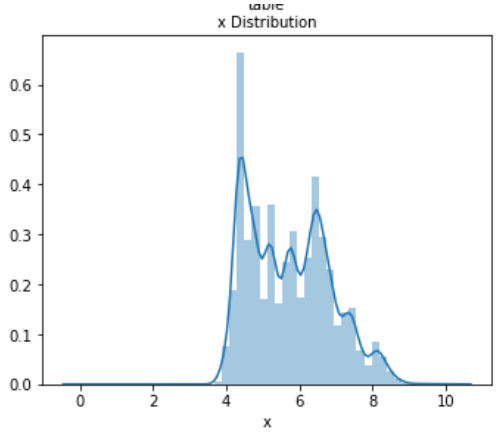


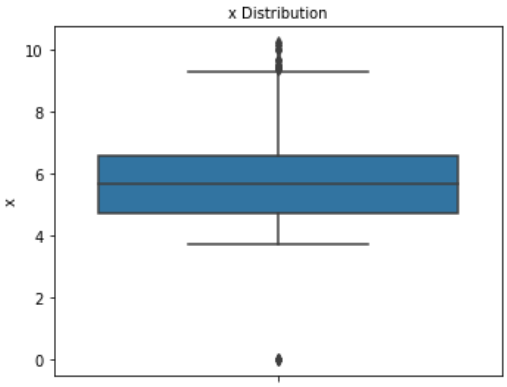


The distribution of table also seems to be positively skewed

The box plot of table has outliers

The data distribution where there is maximum distribution is between 55 to 65.

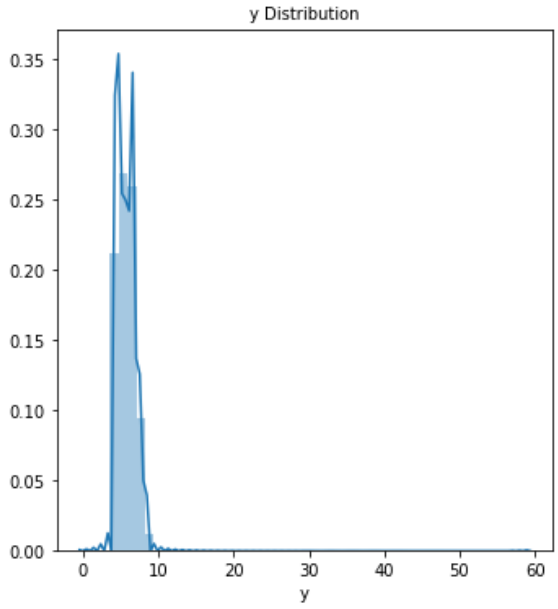


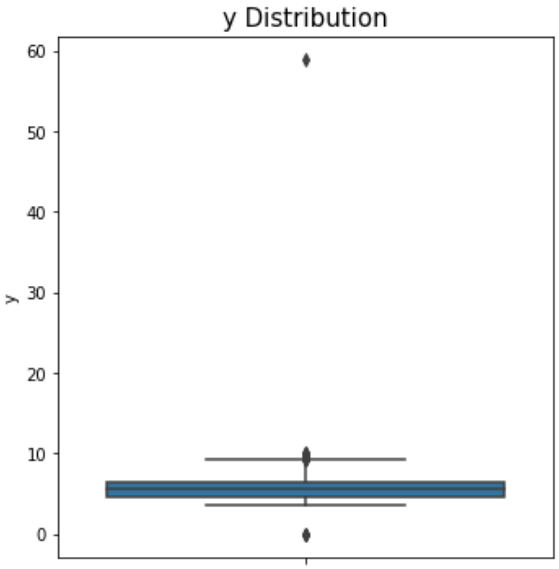


The distribution of x (Length of the cubic zirconia in mm.) is positively skewed

The box plot of the data consists of many outliers

The distribution rages from 4 to 8

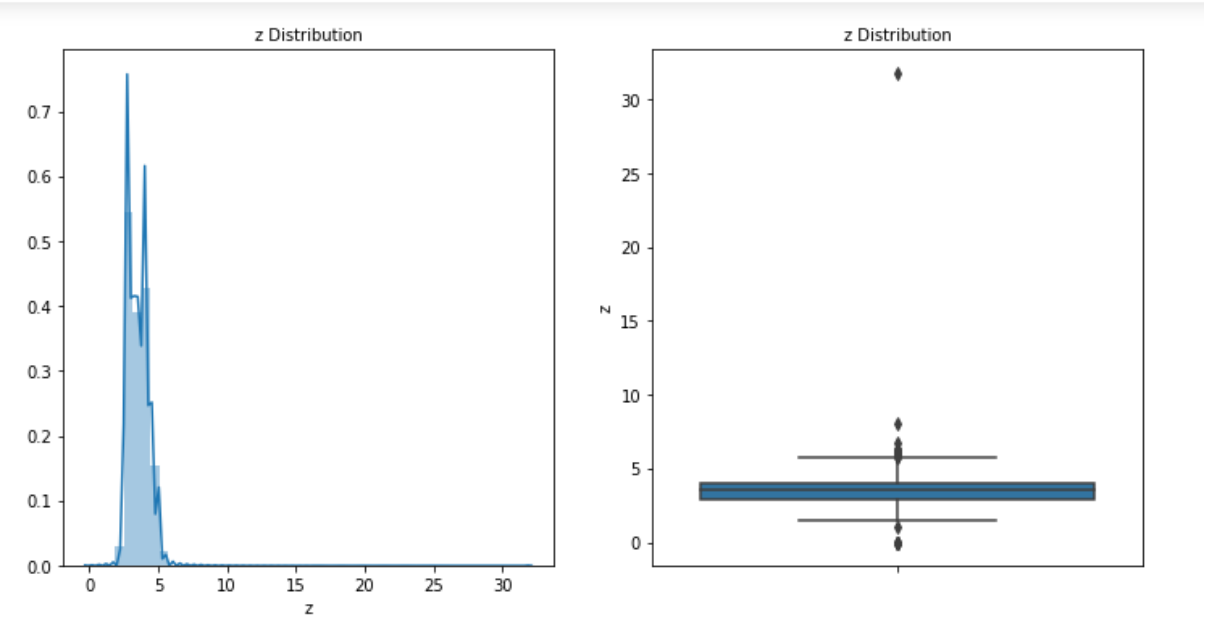




The distribution of Y (Width of the cubic zirconia in mm.) is positively skewed

The box plot also consists of outliers

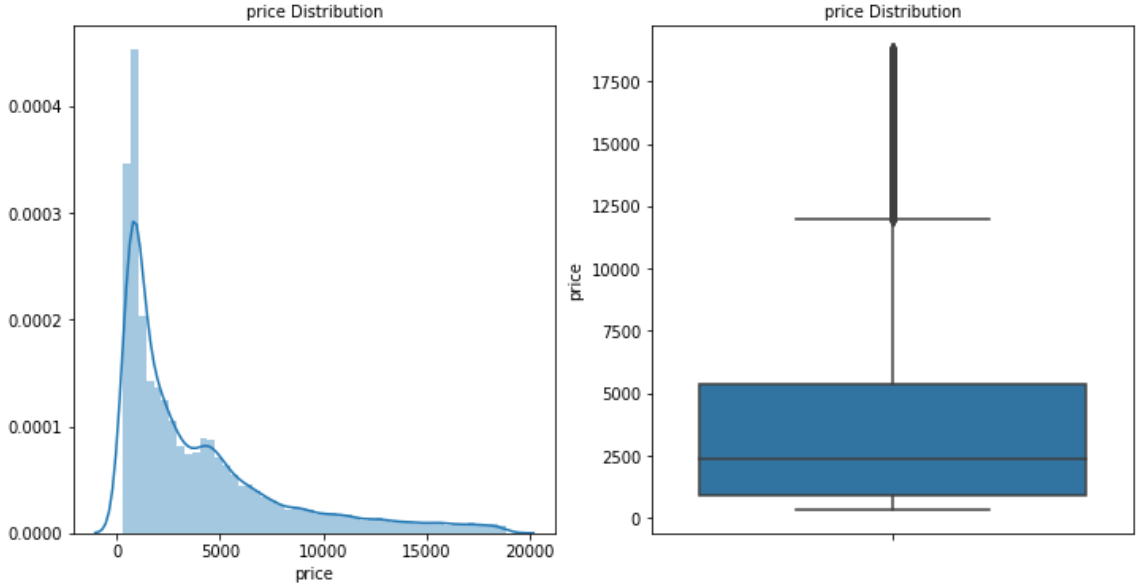
The distribution too much positively skewed. The skewness may be due to the diamonds are always made in specific shape. There might not be too much sizes in the market



The distribution of z (Height of the cubic zirconia in mm.) is positively skewed

The box plot also consists of outliers

The distribution too much positively skewed. The skewness may be due to the diamonds are always made in specific shape. There might not be too much sizes in the market

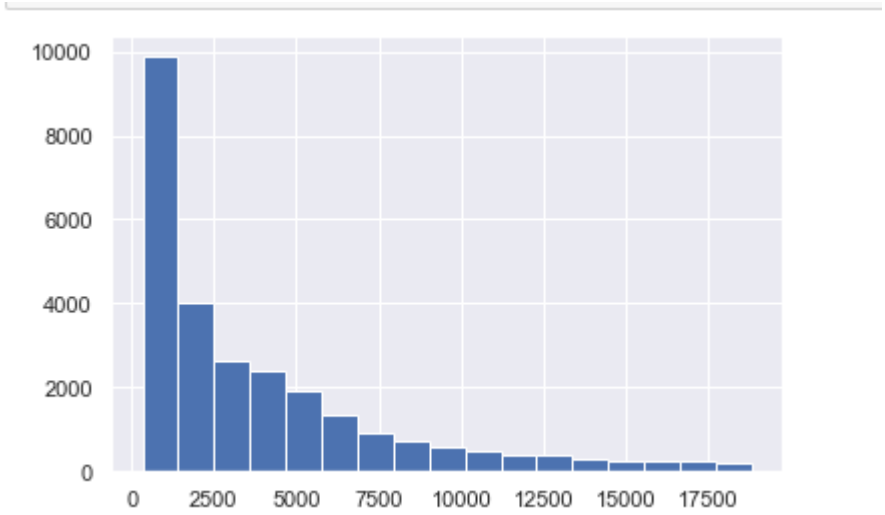


The price has seems to be positively skewed. The skew is positive

The price has outliers in the data

The price distribution is from rs 100 to 8000.

PRICE – HIST



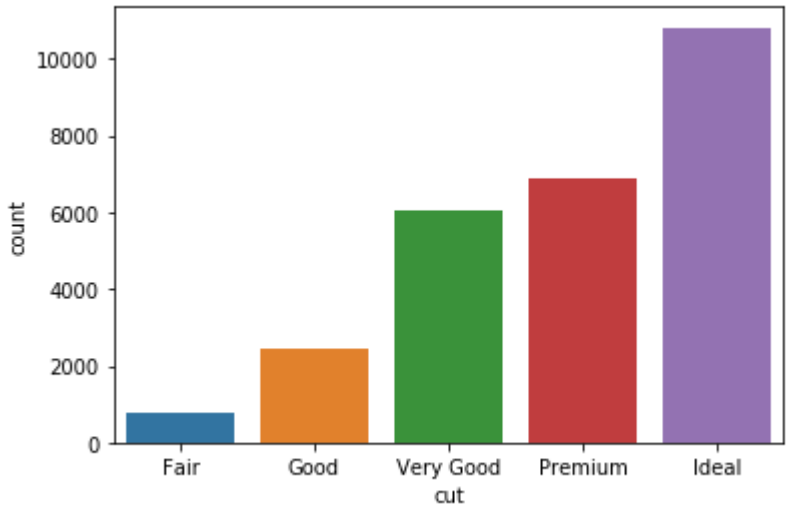
skew

```
carat         1.116481
depth        -0.028618
table         0.765758
x             0.387986
y             3.850189
z             2.568257
price         1.618550
dtype: float64
```

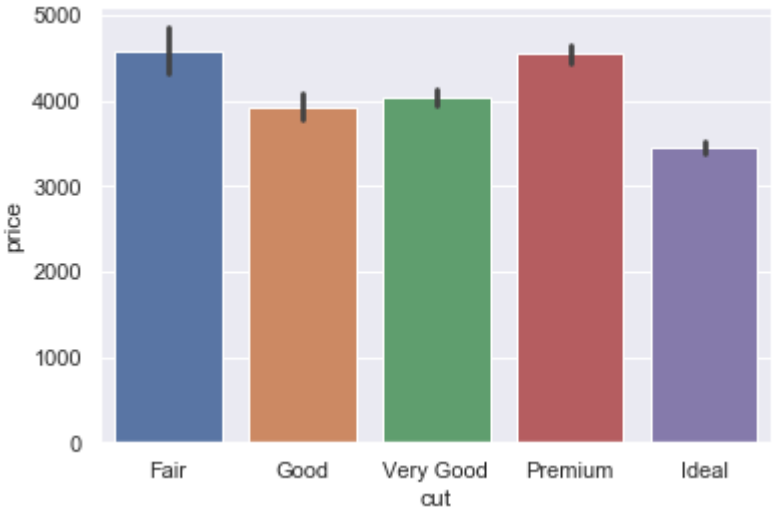
BIVARIATE ANALYSIS

CUT :

Quality is increasing order Fair, Good, Very Good, Premium, Ideal.



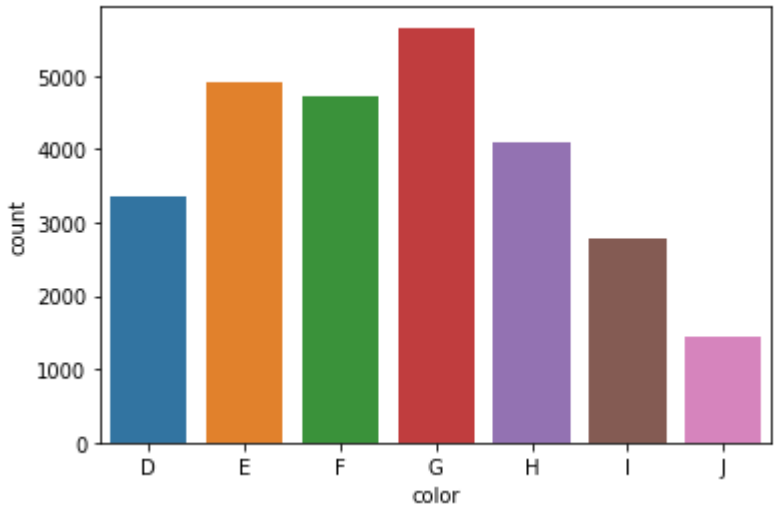
The most preferred cut seems to be ideal cut for diamonds.



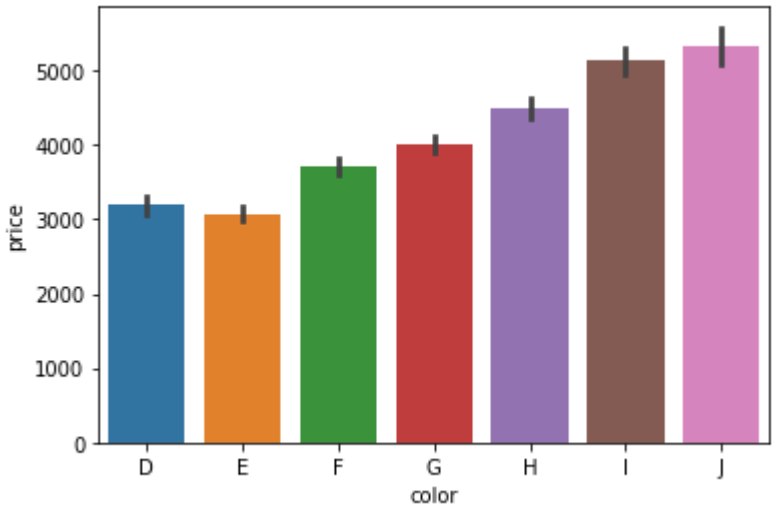
The reason for the most preferred cut ideal is because those diamonds are priced lower than other cuts.

COLOR:

D being the best and J the worst.



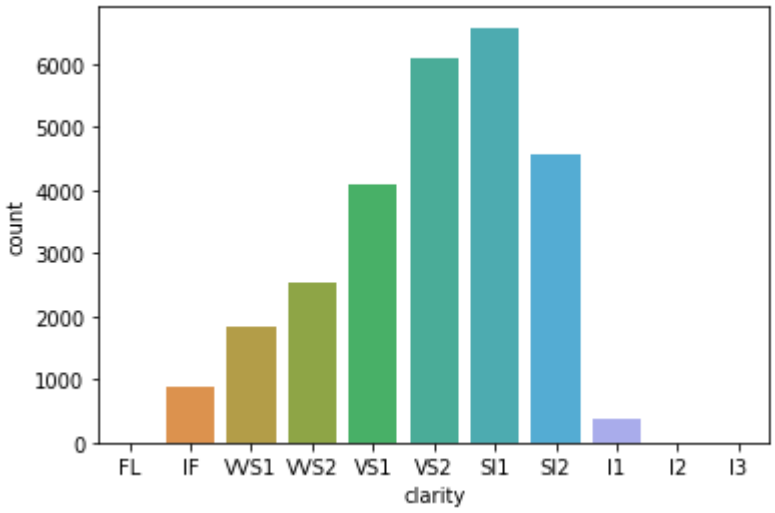
We have 7 colours in the data, The G seems to be the preferred colour,



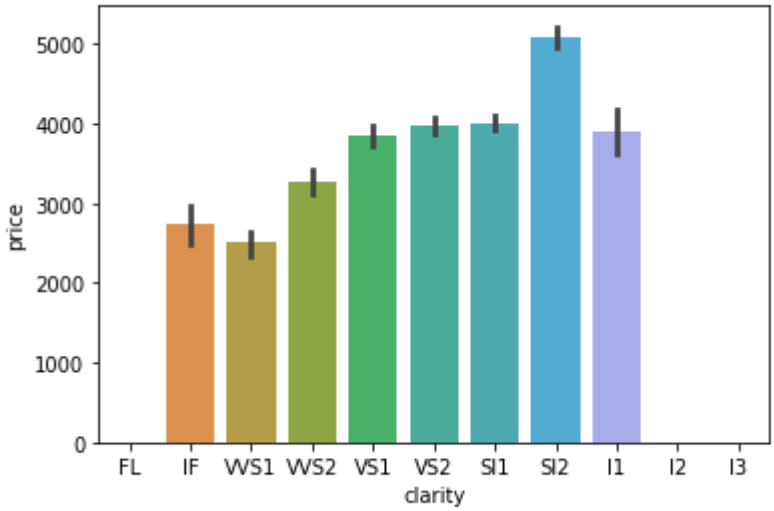
We see the G is priced in the middle of the seven colours, whereas J being the worst colour price seems too high.

CLARITY:

Best to Worst, FL = flawless, I3= level 3 inclusions) FL, IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1, I2, I3



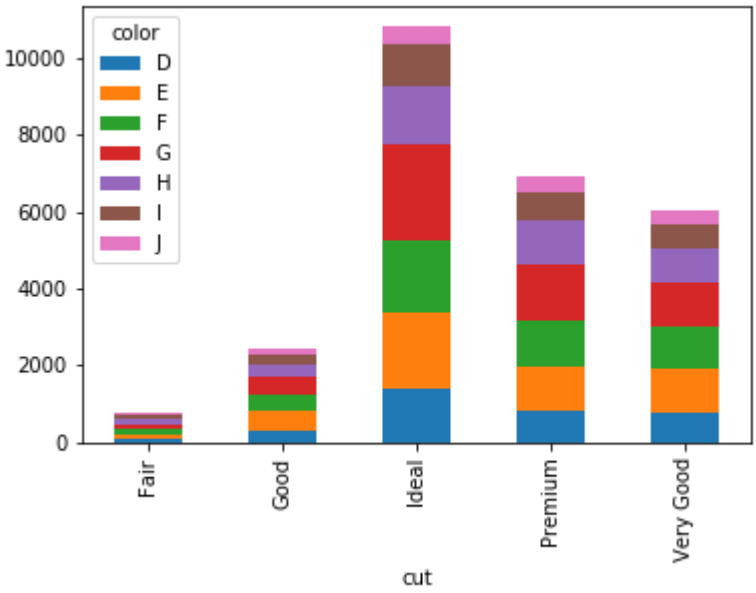
The clarity VS2 seems to be preferred by people



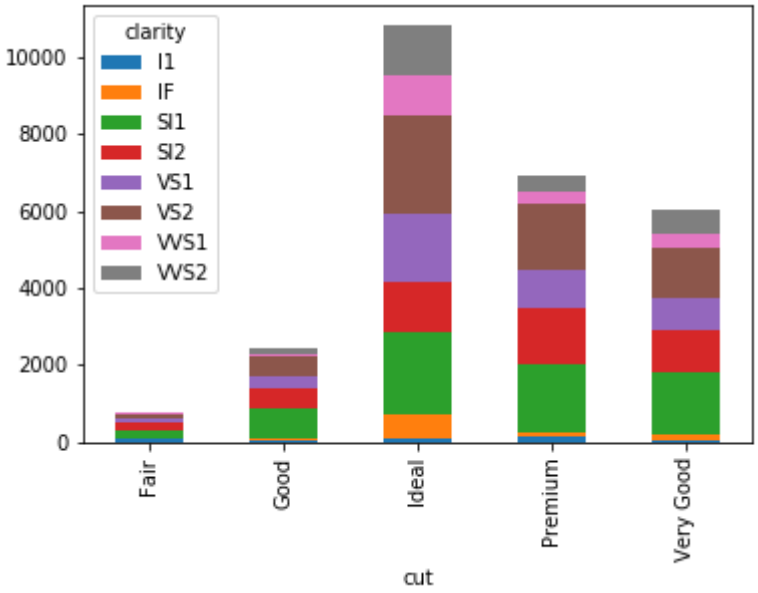
The data has No FL diamonds, from this we can clearly understand the flawless diamonds are not bringing any profits to the store.

More relationship between categorical variables

Cut and colour

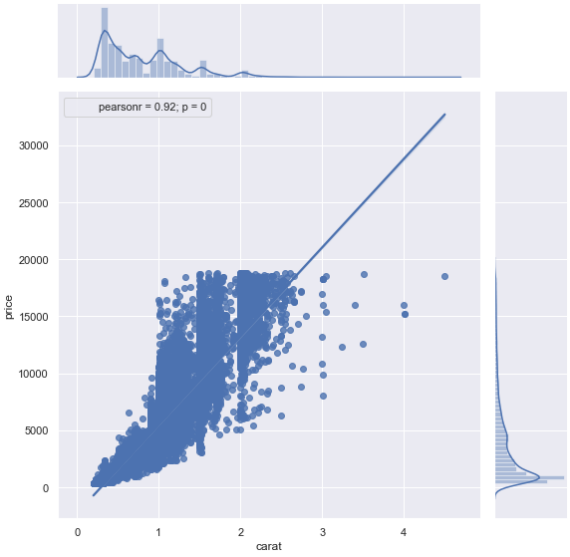


Cut and clarity

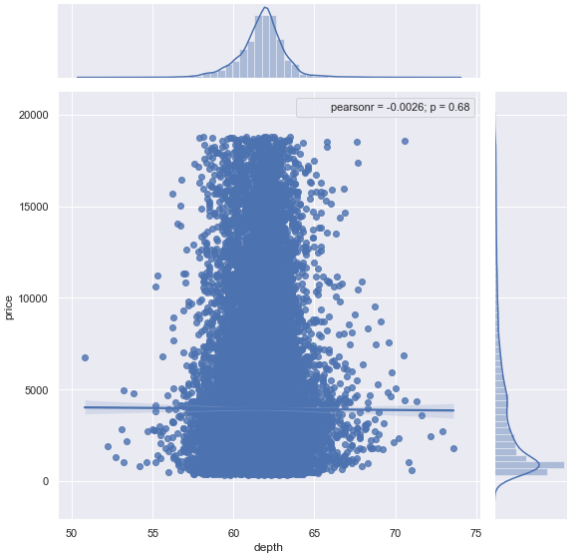


CORRLEATION

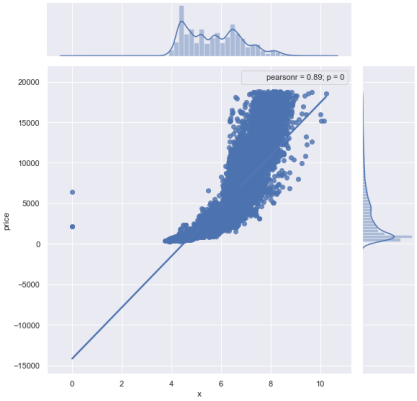
CARAT VS PRICE



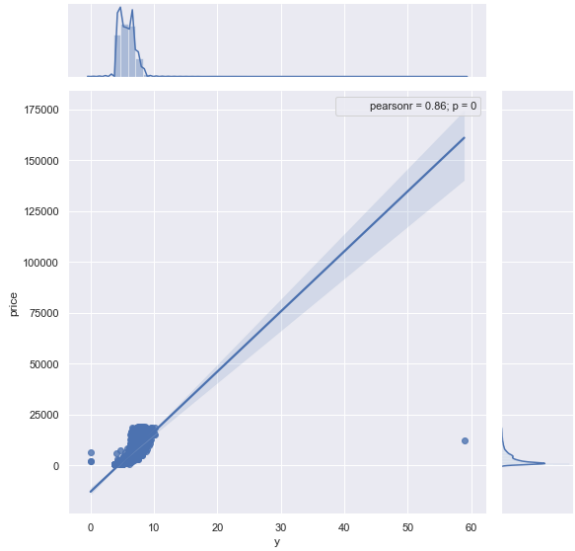
DEPTH VS PRICE



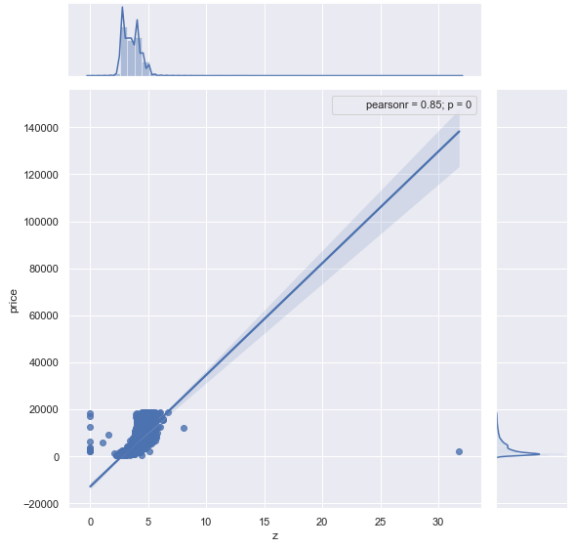
X VS PRICE



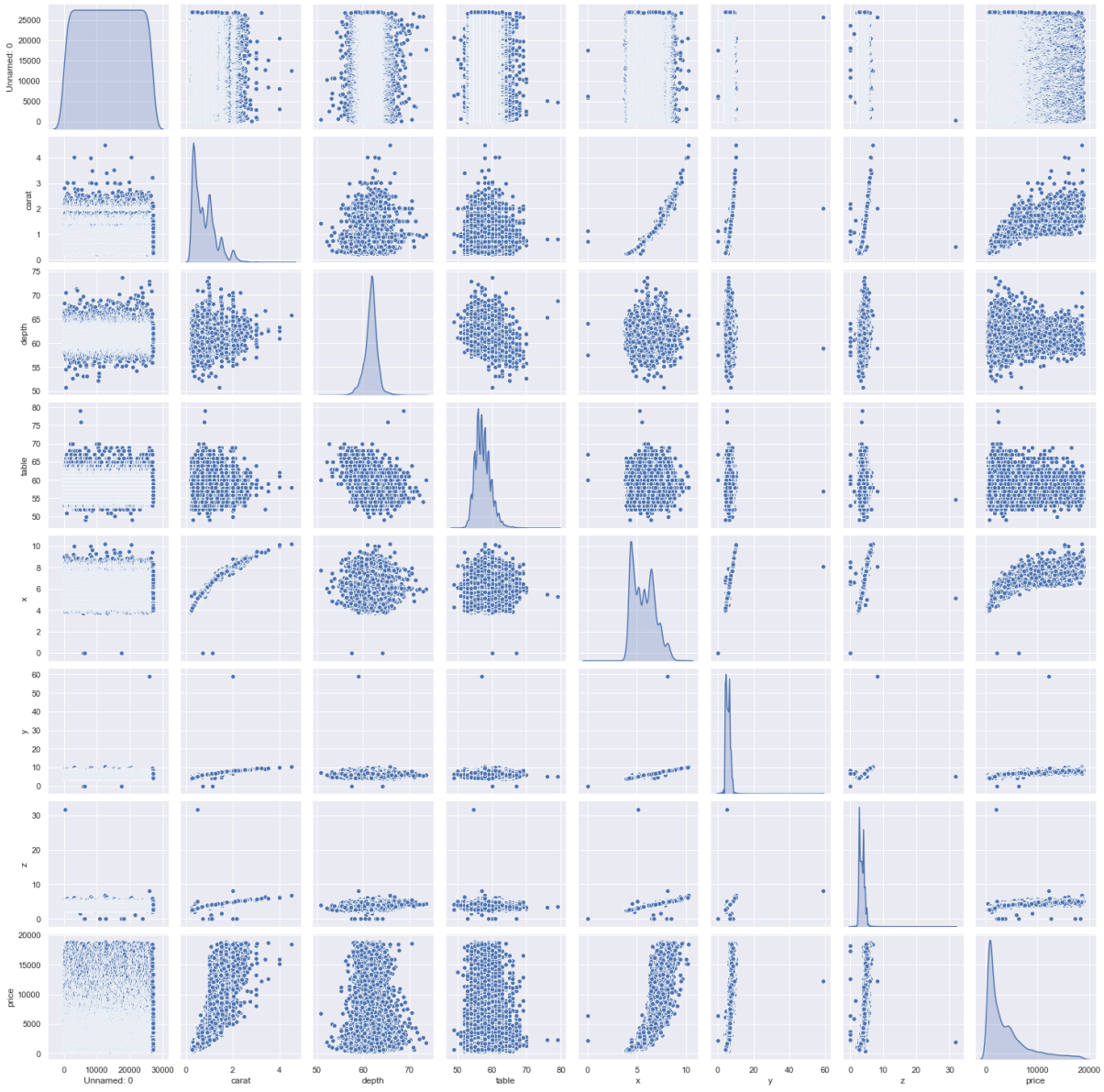
Y VS PRICE



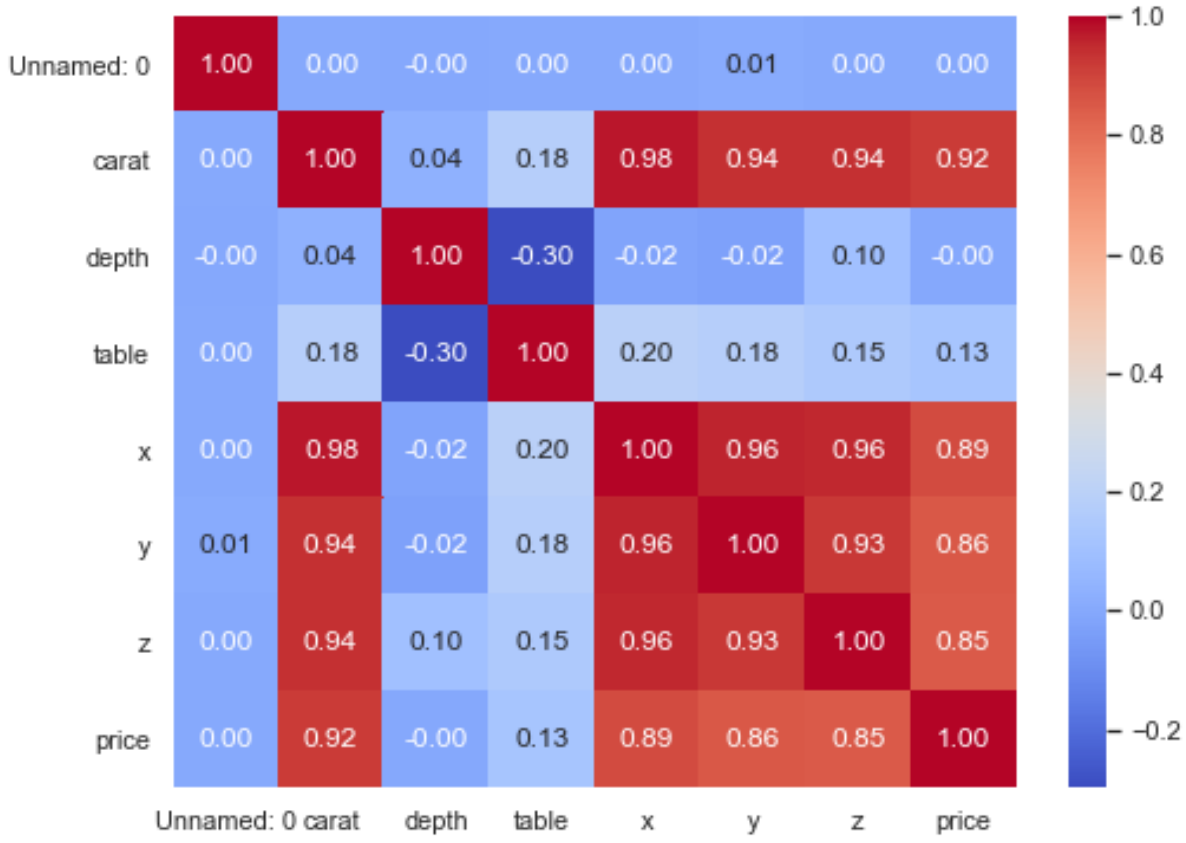
Z VS PRICE



DATA DISTRIBUTION



CORRELATIOM MATRIX



This matrix clearly shows the presence of multi collinearity in 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? Do you think scaling is necessary in this case?

```
: df.isnull().sum()

: 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
```

Yes we have Null values in depth, since depth being continuous variable mean or median imputation can be done.

The percentage of Null values is less than 5%, we can also drop these if we want.

After median imputation, we don't have any null values in the dataset.

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

Checking if there is value that is "0"

	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
5821	5822	0.71	Good	F	SI2	64.1	60.0	0.00	0.00	0.0	2130
6034	6035	2.02	Premium	H	VS2	62.7	53.0	8.02	7.95	0.0	18207
6215	6216	0.71	Good	F	SI2	64.1	60.0	0.00	0.00	0.0	2130
10827	10828	2.20	Premium	H	SI1	61.2	59.0	8.42	8.37	0.0	17265
12498	12499	2.18	Premium	H	SI2	59.4	61.0	8.49	8.45	0.0	12631
12689	12690	1.10	Premium	G	SI2	63.0	59.0	6.50	6.47	0.0	3696
17506	17507	1.14	Fair	G	VS1	57.5	67.0	0.00	0.00	0.0	6381
18194	18195	1.01	Premium	H	I1	58.1	59.0	6.66	6.60	0.0	3167
23758	23759	1.12	Premium	G	I1	60.4	59.0	6.71	6.67	0.0	2383

We have certain rows having values zero, the x, y, z are the dimensions of a diamond so this can't take into model. As there are very less rows.

We can drop these rows as don't have any meaning in model building.

SCALING

Scaling can be useful to reduce or check the multi collinearity in the data, so if scaling is not applied I find the VIF – variance inflation factor values very high. Which indicates presence of multi collinearity

These values are calculated after building the model of linear regression. To understand the multi collinearity in the model

The scaling had no impact in model score or coefficients of attributes nor the intercept.

BEFORE SCALING – VIF VALUES

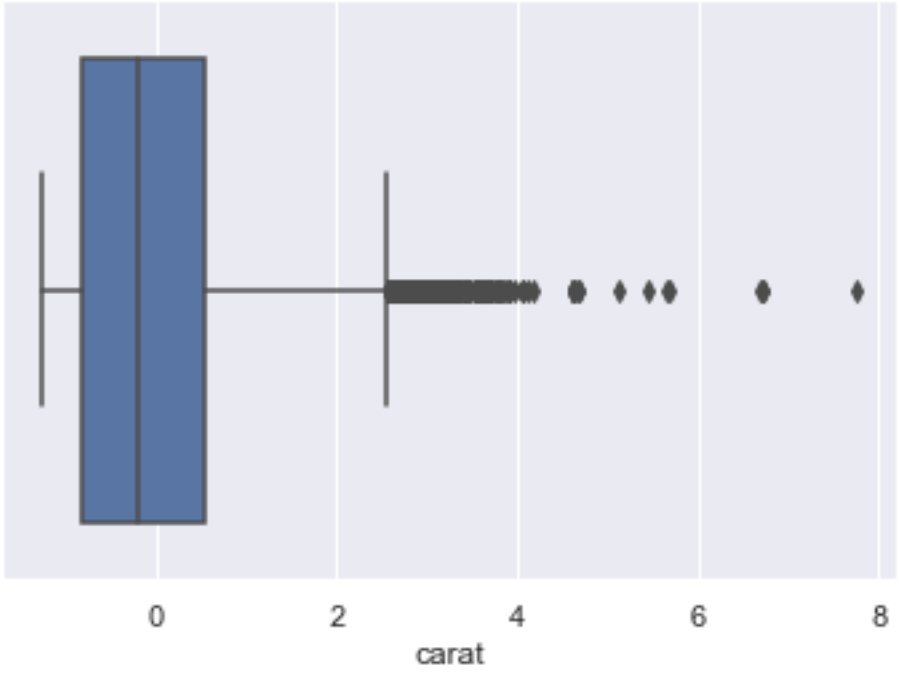
```
carat ---> 124.32595405062301
depth ---> 1407.6352441517224
table ---> 1002.8676766903022
x ---> 12004.212489729716
y ---> 11533.491914672948
z ---> 3442.374035538099
cut_Good ---> 4.5067464355335405
cut_Ideal ---> 18.17410430875144
cut_Premium ---> 10.884031423492264
cut_Very Good ---> 10.062010659328736
color_E ---> 2.479875675651354
```

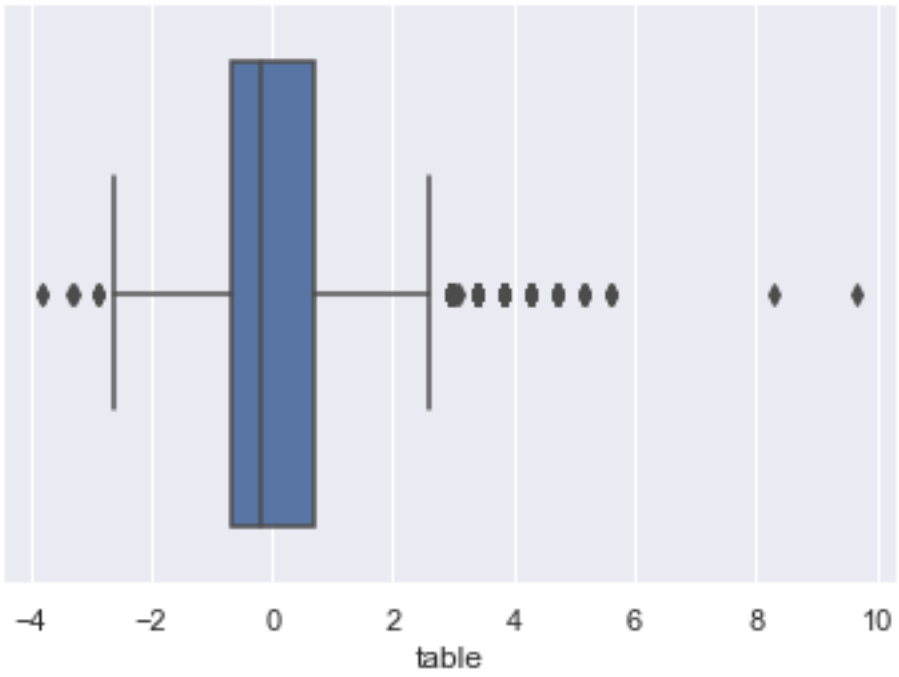
AFTER SCALING – VIF VALUES

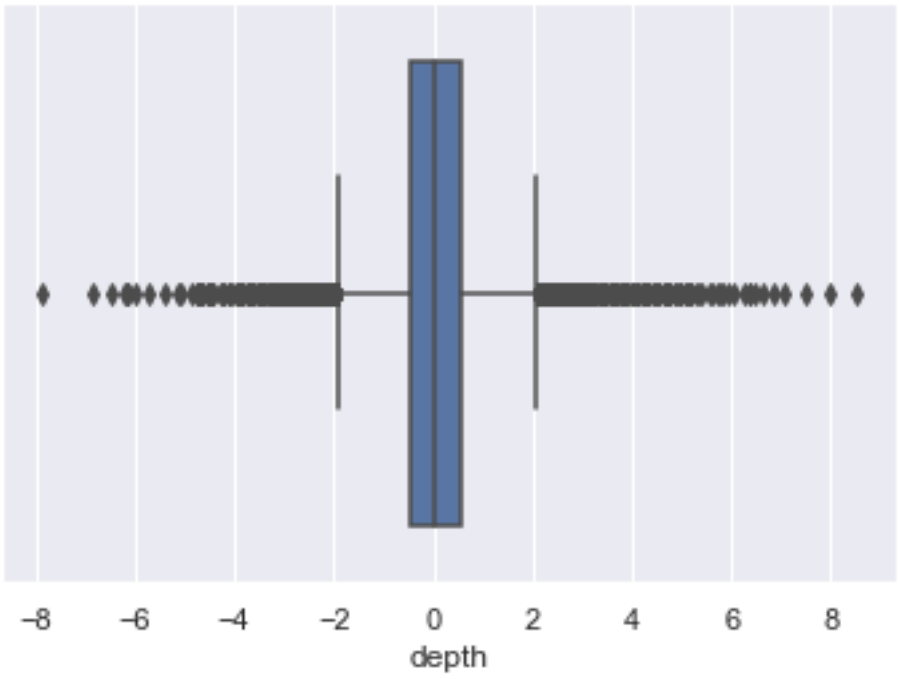
```
carat ---> 33.35287649550623
depth ---> 4.574003842337535
table ---> 1.7722022611198975
x ---> 463.94494858728734
y ---> 463.08309600508517
z ---> 238.6002431605187
cut_Good ---> 3.6104961328079184
cut_Ideal ---> 14.347409690217962
cut_Premium ---> 8.623207030351887
cut_Very Good ---> 7.852218650260111
color_E ---> 2.371053795458172
```

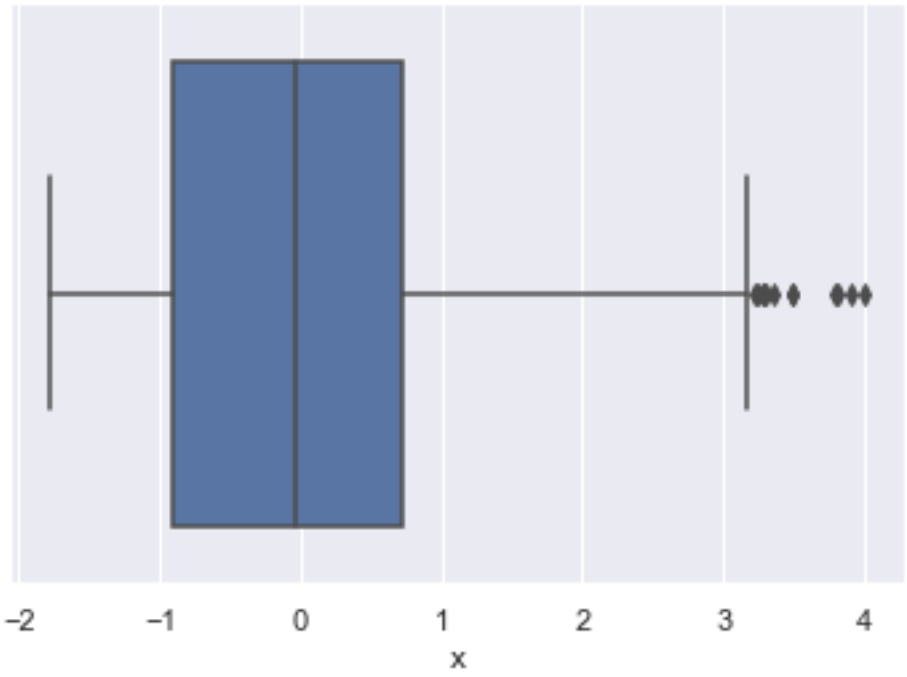
CHECKING THE OUTLIERS IN THE DATA

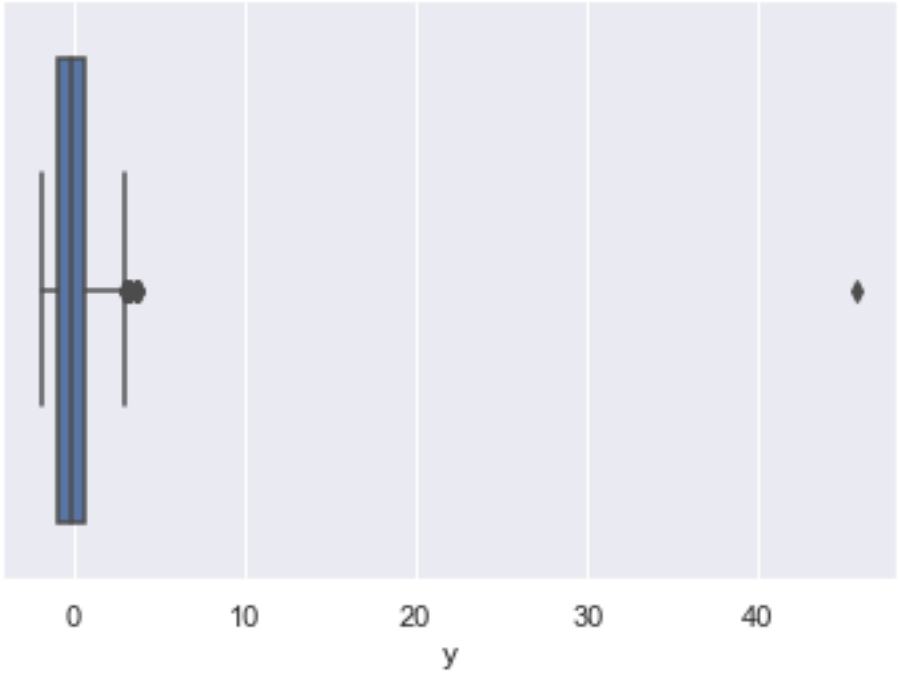
BEFORE TREATING OUTLIERS

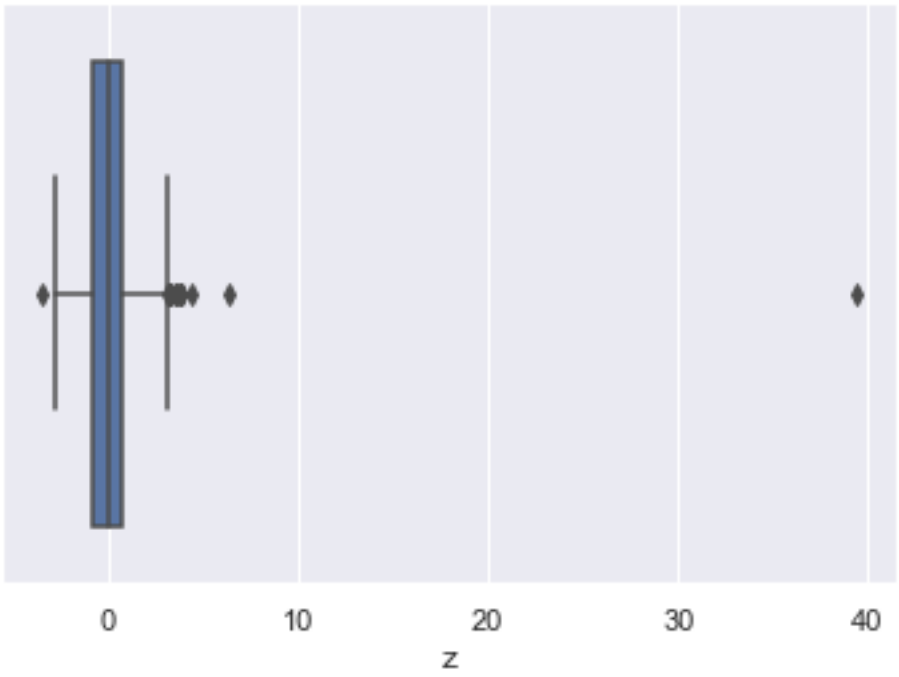


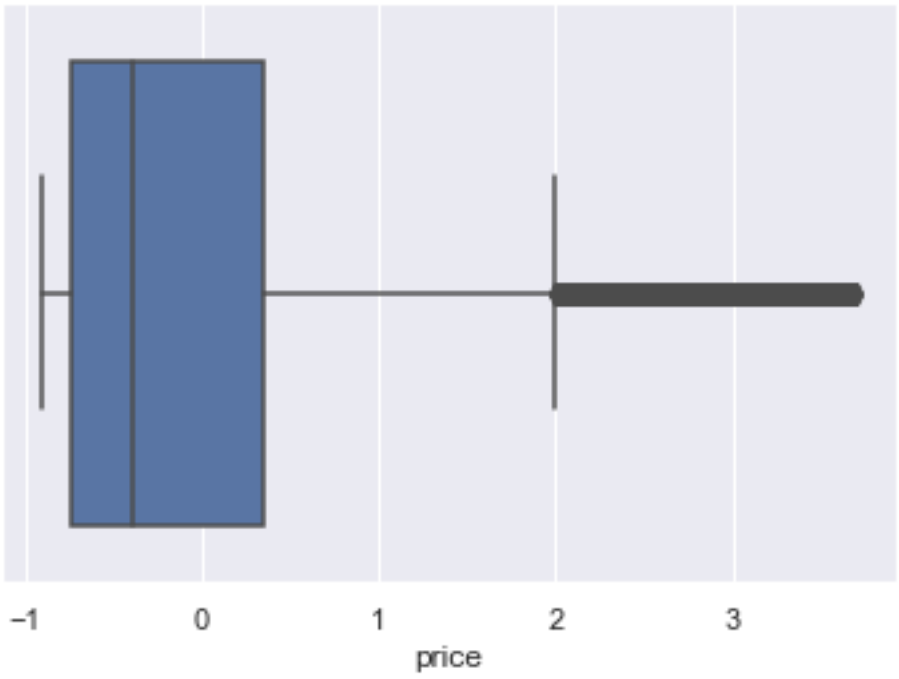

−4
−2
0
2
4
6
8
10
table


−8
−6
−4
−2
0
2
4
6
8
depth

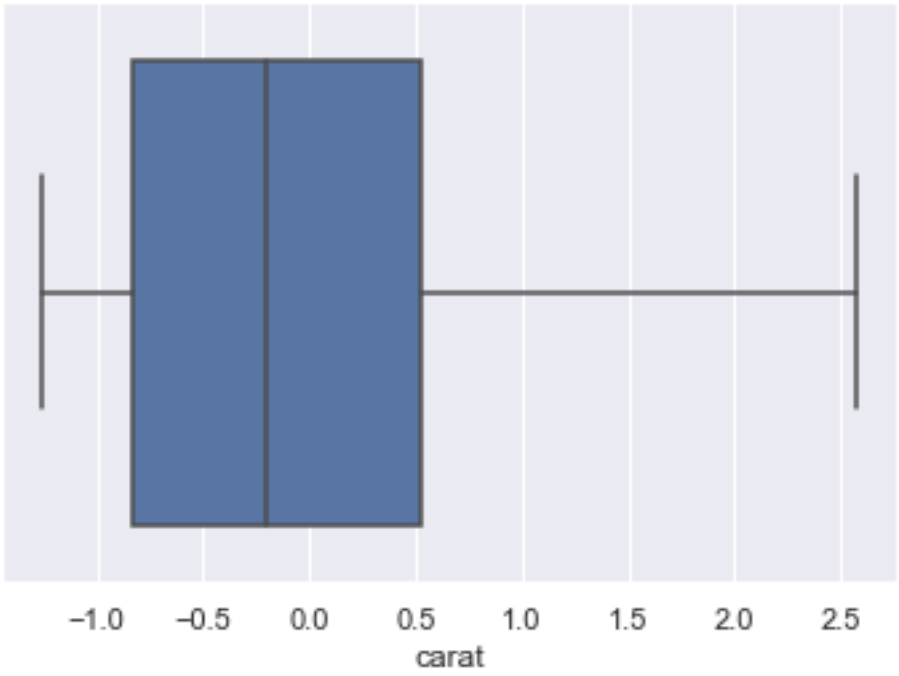

−2
−1
0
1
2
3
4
x

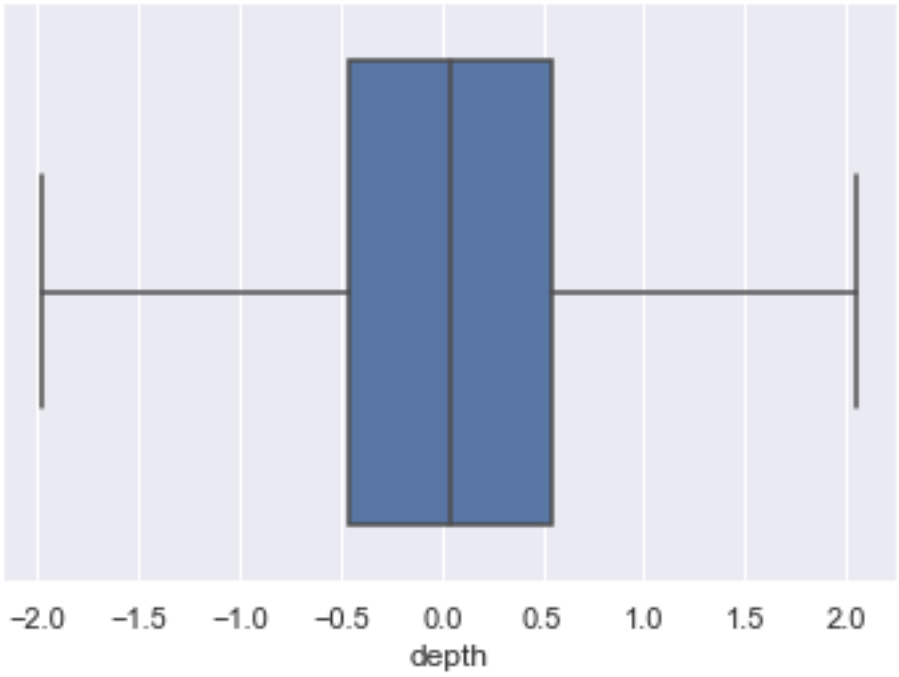

0
10
20
30
40
y

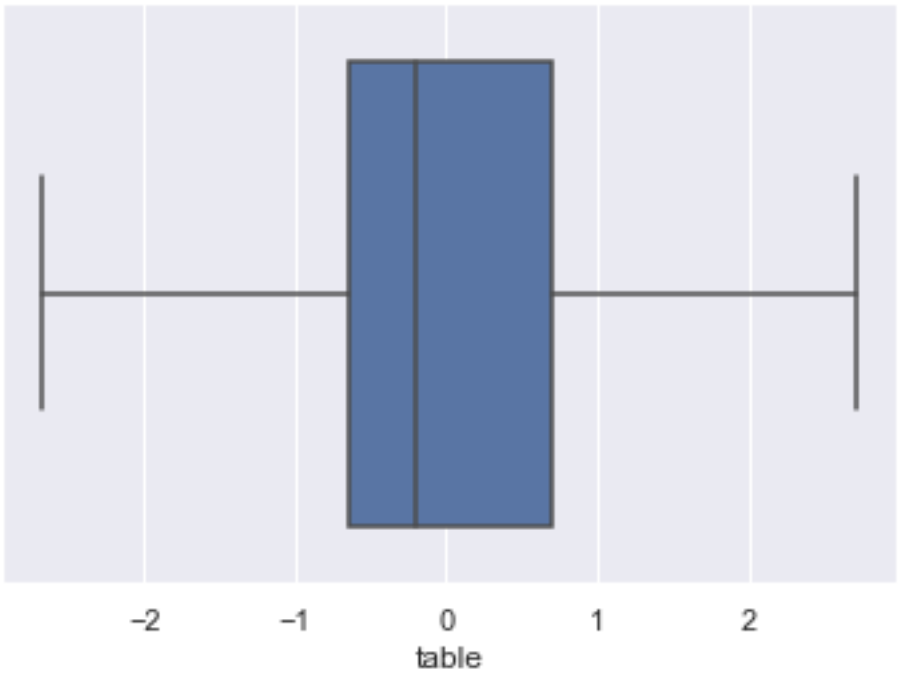

0
10
20
30
40
z

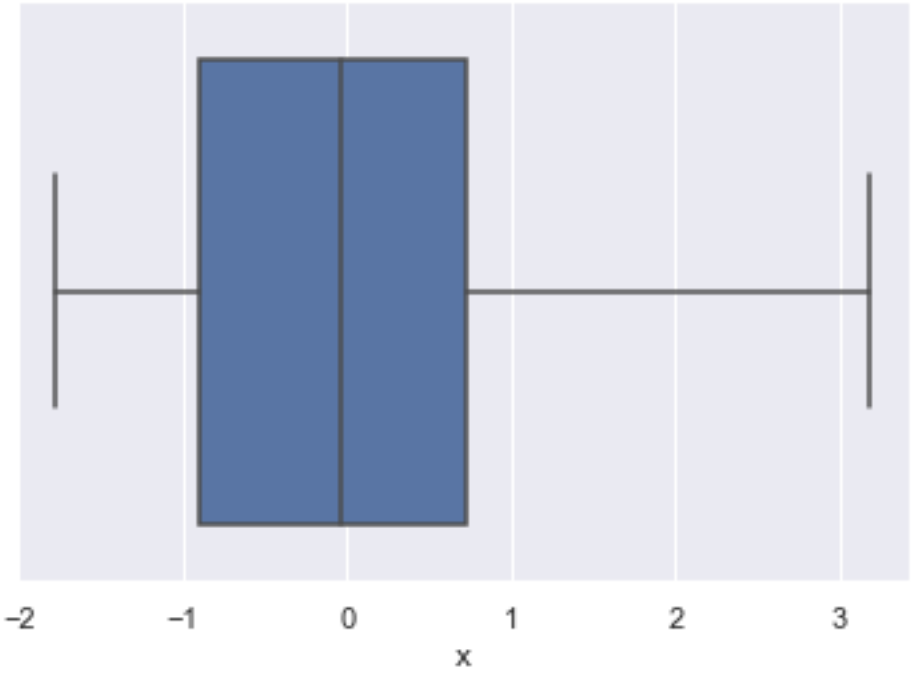

−1
0
1
2
3
price

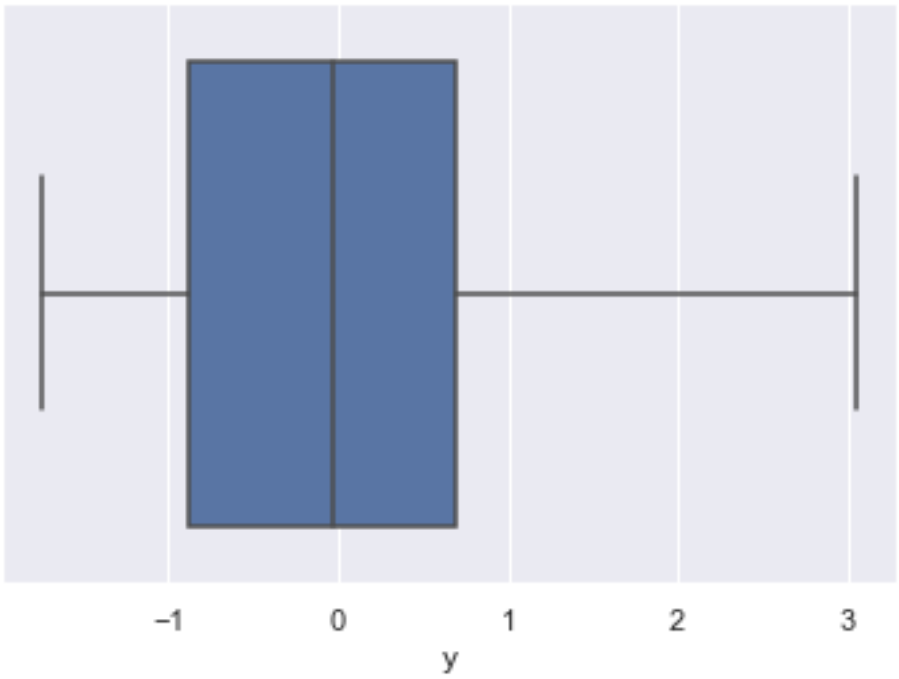
AFTER TREATING OUTLIERS

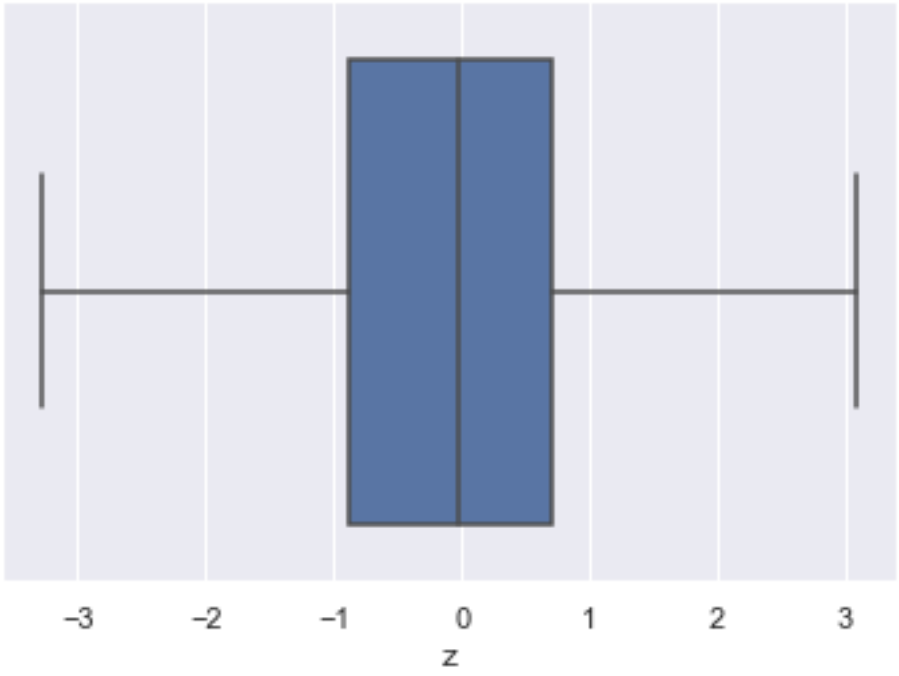


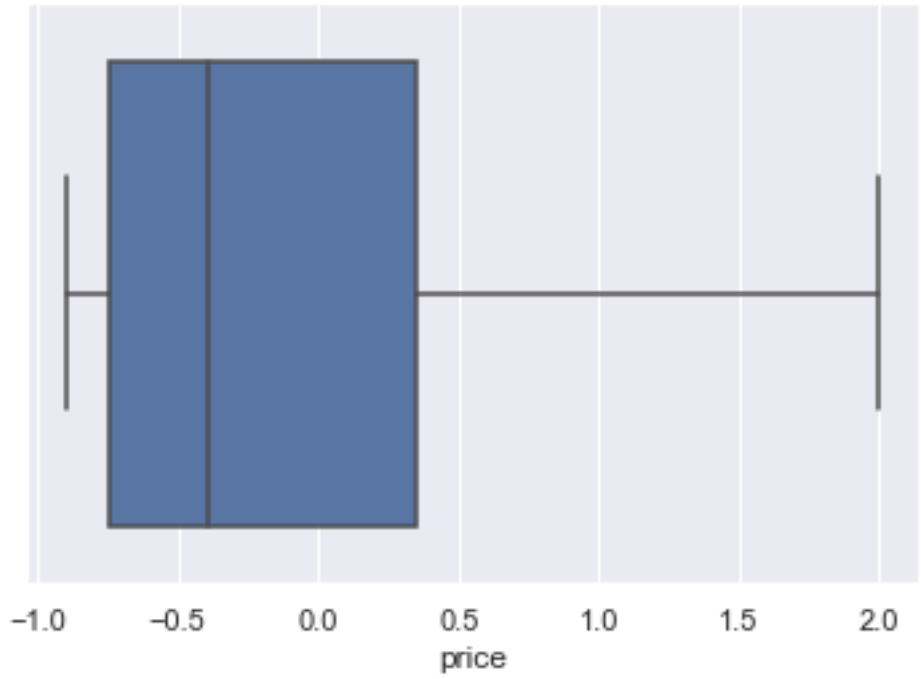



−2
−1
0
1
2
table


−2
−1
0
1
2
3
x


−1
0
1
2
3
y


−3
−2
−1
0
1
2
3
z



1.3 Encode the data (having string values) for Modelling. Data Split: Split the data into test and train (70:30). Apply Linear regression. Performance Metrics: Check the performance of Predictions on Train and Test sets using Rsquare, RMSE.

ENCODING THE STRING VALUES

GET DUMMIES

```
data = pd.get_dummies(df, columns=['cut','color','clarity'],drop_first=True)
```

	Unnamed: 0	carat	depth	table	x	y	z	price	cut_Good	cut_Ideal	...	color_H	color_I	color_J	clarity_IF	cl
0	-1.731904	-1.043125	0.253399	0.244112	-1.295920	-1.240065	-1.224865	-0.854851	0	1	...	0	0	0	0	
1	-1.731776	-0.980310	-0.679158	0.244112	-1.162787	-1.094057	-1.169142	-0.734303	0	0	...	0	0	0	1	
2	-1.731647	0.213173	0.325134	1.140496	0.275049	0.331668	0.335404	0.584271	0	0	...	0	0	0	0	
3	-1.731519	-0.791865	-0.105277	-0.652273	-0.807766	-0.802041	-0.806936	-0.709945	0	1	...	0	0	0	0	
4	-1.731390	-1.022187	-0.966099	0.692304	-1.224916	-1.119823	-1.238796	-0.785257	0	1	...	0	0	0	0	

5 rows × 25 columns

```
Index(['Unnamed: 0', 'carat', 'depth', 'table', 'x', 'y', 'z', 'price',
       'cut_Good', 'cut_Ideal', 'cut_Premium', 'cut_Very Good', 'color_E',
       'color_F', 'color_G', 'color_H', 'color_I', 'color_J', 'clarity_IF',
       'clarity_SI1', 'clarity_SI2', 'clarity_VS1', 'clarity_VS2',
       'clarity_VVS1', 'clarity_VVS2'],
      dtype='object')
```

Dummies have been encoded.

Linear regression model does not take categorical values so that we have encoded categorical values to integer for better results.

DROPING UNWANTED COLUMNS

Train/ Test split

```
# drop the id column as it is useless for the model
data_model = data.drop(columns=['Unnamed: 0'], axis=1)
```

```
data_model.columns
```

```
Index(['carat', 'depth', 'table', 'x', 'y', 'z', 'price', 'cut_Good',
       'cut_Ideal', 'cut_Premium', 'cut_Very Good', 'color_E', 'color_F',
       'color_G', 'color_H', 'color_I', 'color_J', 'clarity_IF', 'clarity_SI1',
       'clarity_SI2', 'clarity_VS1', 'clarity_VS2', 'clarity_VVS1',
       'clarity_VVS2'],
      dtype='object')
```

```
: # Split X and y into training and test set in 70:30 ratio
```

```
: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3 , random_state=1)
```

Linear Regression Model

```
[ ]: # invoke the LinearRegression function and find the bestfit model on training data

regression_model = LinearRegression()
regression_model.fit(X_train, y_train)
```

The coefficient for carat is 1.1009417847804501
The coefficient for depth is 0.005605143445570377
The coefficient for table is -0.013319500386804035
The coefficient for x is -0.30504349819633475
The coefficient for y is 0.30391448957926553
The coefficient for z is -0.13916571567987943
The coefficient for cut_Good is 0.09403402912977911
The coefficient for cut_Ideal is 0.1523107462056746
The coefficient for cut_Premium is 0.14852774839849378
The coefficient for cut_Very Good is 0.12583881878452705
The coefficient for color_E is -0.04705442233369822
The coefficient for color_F is -0.06268437439142825
The coefficient for color_G is -0.10072161838356786

The coefficient for color_H is -0.20767313311661612
The coefficient for color_I is -0.3239541927462737
The coefficient for color_J is -0.46858930275015803
The coefficient for clarity_IF is 0.9997691394634902
The coefficient for clarity_SI1 is 0.6389785818271332
The coefficient for clarity_SI2 is 0.42959662348315514
The coefficient for clarity_VS1 is 0.8380875826737564
The coefficient for clarity_VS2 is 0.7660244466083613
The coefficient for clarity_VVS1 is 0.9420769630114072
The coefficient for clarity_VVS2 is 0.9313670288415696

```
# R square on training data
regression_model.score(X_train, y_train)
```

```
0.9419557931252712
```

```
# R square on testing data
regression_model.score(X_test, y_test)
```

```
0.9381643998102491
```

```
#RMSE on Training data
predicted_train=regression_model.fit(X_train, y_train).predict(X_train)
np.sqrt(metrics.mean_squared_error(y_train,predicted_train))
```

```
0.20690072466418796
```

```
#RMSE on Testing data
predicted_test=regression_model.fit(X_train, y_train).predict(X_test)
np.sqrt(metrics.mean_squared_error(y_test,predicted_test))
```

```
0.21647817772382869
```

VIF –VALUES

```
carat ---> 33.35086119845924
depth ---> 4.573918951598579
table ---> 1.7728852812618958
x ---> 463.5542785436457
y ---> 462.769821646584
z ---> 238.65819968687333
cut_Good ---> 3.609618194943713
cut_Ideal ---> 14.34812508118844
cut_Premium ---> 8.623414379121153
cut_Very Good ---> 7.848451571723695
color_E ---> 2.371070464762613
```

We still find we have multi collinearity in the dataset, to drop these values to Lower level we can drop columns after doing stats model.

From stats model we can understand the features that do not contribute to the Model

We can remove those features after that the Vif Values will be reduced

Ideal value of VIF is less tha 5%.

STATSMODEL

BEST PARAMS SUMMARY

OLS Regression Results

Dep. Variable:	price	R-squared:	0.942
Model:	OLS	Adj. R-squared:	0.942
Method:	Least Squares	F-statistic:	1.330e+04
Date:	Fri, 15 Jan 2021	Prob (F-statistic):	0.00
Time:	22:15:37	Log-Likelihood:	2954.6
No. Observations:	18870	AIC:	-5861.
Df Residuals:	18846	BIC:	-5673.
Df Model:	23		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.7568	0.016	-46.999	0.000	-0.788	-0.725
carat	1.1009	0.009	121.892	0.000	1.083	1.119
depth	0.0056	0.004	1.525	**0.127**	-0.002	0.013
table	-0.0133	0.002	-6.356	0.000	-0.017	-0.009
x	-0.3050	0.032	-9.531	0.000	-0.368	-0.242
y	0.3039	0.034	8.934	0.000	0.237	0.371
z	-0.1392	0.024	-5.742	0.000	-0.187	-0.092
cut_Good	0.0940	0.011	8.755	0.000	0.073	0.115
cut_Ideal	0.1523	0.010	14.581	0.000	0.132	0.173

```
cut_Premium        0.1485      0.010     14.785      0.000        0.129
0.168
cut_Very_Good      0.1258      0.010     12.269      0.000        0.106
0.146
color_E           -0.0471      0.006     -8.429      0.000       -0.058
-0.036
color_F           -0.0627      0.006    -11.075      0.000       -0.074
-0.052
color_G           -0.1007      0.006    -18.258      0.000       -0.112
-0.090
color_H           -0.2077      0.006    -35.323      0.000       -0.219
-0.196
color_I           -0.3240      0.007    -49.521      0.000       -0.337
-0.311
color_J           -0.4686      0.008    -58.186      0.000       -0.484
-0.453
clarity_IF         0.9998      0.016     62.524      0.000        0.968
1.031
clarity_SI1        0.6390      0.014     46.643      0.000        0.612
0.666
clarity_SI2        0.4296      0.014     31.177      0.000        0.403
0.457
clarity_VS1        0.8381      0.014     59.986      0.000        0.811
0.865
clarity_VS2        0.7660      0.014     55.618      0.000        0.739
0.793
clarity_VVS1       0.9421      0.015     63.630      0.000        0.913
0.971
clarity_VVS2       0.9314      0.014     64.730      0.000        0.903
0.960
==============================================================================
Omnibus:                     4696.785   Durbin-Watson:
1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            17
654.853
Skew:                           1.208   Prob(JB):
0.00
Kurtosis:                       7.076   Cond. No.
57.0
```

After dropping the depth variable

OLS Regression Results

```
==============================================================================
======================
```

Dep. Variable: price R-squared: 0.942

Model: OLS Adj. R-squared: 0.942

Method: Least Squares F-statistic: 1.390e+04

Date: Fri, 15 Jan 2021 Prob (F-statistic): 0.00

Time: 22:16:56 Log-Likelihood: 2953.5

No. Observations: 18870 AIC: -5861.

Df Residuals: 18847 BIC: -5680.

Df Model: 22

Covariance Type: nonrobust

==========

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.7567	0.016	-46.991	0.000	-0.788	-0.725
carat	1.1020	0.009	122.331	0.000	1.084	1.120
table	-0.0139	0.002	-6.770	0.000	-0.018	-0.010
x	-0.3156	0.031	-10.101	0.000	-0.377	-0.254
y	0.2834	0.031	9.069	0.000	0.222	0.345
z	-0.1088	0.014	-7.883	0.000	-0.136	-0.082
cut_Good	0.0951	0.011	8.876	0.000	0.074	0.116
cut_Ideal	0.1512	0.010	14.508	0.000	0.131	0.172
cut_Premium	0.1474	0.010	14.711	0.000	0.128	0.167
cut_Very_Good	0.1255	0.010	12.239	0.000	0.105	0.146
color_E	-0.0471	0.006	-8.439	0.000	-0.058	-0.036
color_F	-0.0627	0.006	-11.082	0.000	-0.074	-0.052
color_G	-0.1007	0.006	-18.246	0.000	-0.111	-0.090
color_H	-0.2076	0.006	-35.306	0.000	-0.219	-0.196
color_I	-0.3237	0.007	-49.497	0.000	-0.337	-0.311
color_J	-0.4684	0.008	-58.169	0.000	-0.484	-0.453
clarity_IF	1.0000	0.016	62.544	0.000	0.969	1.031
clarity_SI1	0.6398	0.014	46.738	0.000	0.613	0.667
clarity_SI2	0.4302	0.014	31.232	0.000	0.403	0.457

clarity_VS1	0.8386	0.014	60.042	0.000	0.811	0.866
clarity_VS2	0.7667	0.014	55.691	0.000	0.740	0.794
clarity_VVS1	0.9424	0.015	63.655	0.000	0.913	0.971
clarity_VVS2	0.9319	0.014	64.784	0.000	0.904	0.960

==========

Omnibus:	4699.504	Durbin-Watson:	1.994
Prob(Omnibus):	0.000	Jarque-Bera (JB):	17704.272
Skew:	1.208	Prob(JB):	0.00
Kurtosis:	7.084	Cond. No.	56.5

To ideally bring down the values to lower levels we can drop one of the variable that is highly correlated.

Dropping variables would bring down the multi collinearity level down.

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

We had a business problem to predict the price of the stone and provide insights for the company on the profits on different prize slots. From the EDA analysis we could understand the cut, ideal cut had number profits to the company. The colours H, I, J have bought profits for the company. In clarity if we could see there were no flawless stones and they were no profits coming from I1, I2, I3 stones. The ideal, premium and very good types of cut were bringing profits where as fair and good are not bringing profits.

The predictions were able to capture 95% variations in the price and it is explained by the predictors in the training set.

Using stats model if we could run the model again we can have P values and coefficients which will give us better understanding of the relationship, so that values more 0.05 we can drop those variables and re run the model again for better results.

For better accuracy dropping depth column in iteration for better results.

The equation, **(-0.76) * Intercept + (1.1) * carat + (-0.01) * table + (-0.32) * x + (0.28) * y + (-0.11) * z + (0.1) * cut_Good + (0.15) * cut_Ideal + (0.15) * cut_Premium + (0.13) * cut_Very_Good + (-0.05) * color_E + (-0.06) * color_F + (-0.1) * color_G + (-0.21) * color_H + (-0.32) * color_I + (-0.47) * color_J + (1.0) * clarity_IF + (0.64) * clarity_SI1 + (0.43) * clarity_SI2 + (0.84) * clarity_VS1 + (0.77) * clarity_VS2 + (0.94) * clarity_VVS1 + (0.93) * clarity_VVS2 +**

Recommendations

1. The ideal, premium, very good cut types are the one which are bringing profits so that we could use marketing for these to bring in more profits.
2. The clarity of the diamond is the next important attributes the more the clear is the stone the profits are more

The five best attributes are

Carat,
Y the diameter of the stone
clarity_IF
clarity_SI1
clarity_SI2
clarity_VS1
clarity_VS2
clarity_VVS1
clarity_VVS2

THE END