PREDICTIVE MODELLING PROJECT

BUSINESS REPORT

Report by:

Tushar Babbar
PGPDSBA Online
DSBA Dec20 Group 5

Table of Contents

1 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.

1.1.1 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.

1.1.2 Importing the necessary libraries for the model building.

1.1.3 Reading the head and tail of the dataset to get an overview of our data¶

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

The total number of rows present in the dataset above is : 26967
The total number of columns/variables present in the dataset above is :10

- **Here we will be omitting "Unnamed: 0" column in further calculations.**

1.1.4 Checking the info of 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
```

```
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
```

- **We have float, int and object data types in the data.**
- **It can also be observed that there are some null values present in "depth".**

1.1.5 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

Description and 5 point summary of the variables in the dataset.

- 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 our target variable

.

1.1.6 Checking for Duplicates and removing the Unnamed: 0 column

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

1.1.7 Checking for unique values of the categorical variables

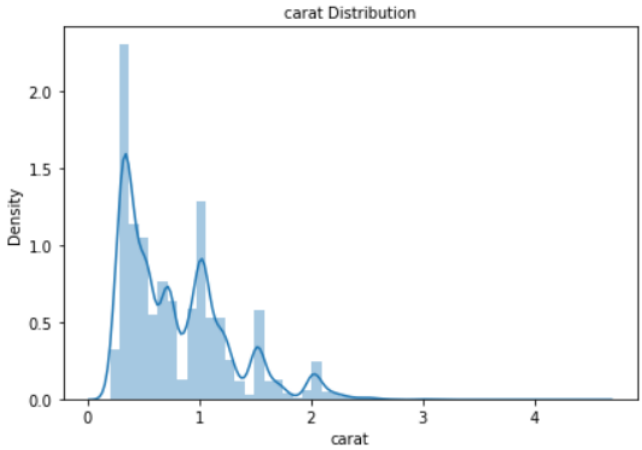
```
CUT :  5
Fair           781
Good          2441
Very Good     6030
Premium       6899
Ideal        10816
Name: cut, dtype: int64


COLOR :  7
J    1443
I    2771
D    3344
H    4102
F    4729
E    4917
G    5661
Name: color, dtype: int64


CLARITY :  8
I1       365
IF       894
VVS1    1839
VVS2    2531
VS1     4093
SI2     4575
VS2     6099
SI1     6571
Name: clarity, dtype: int64
```

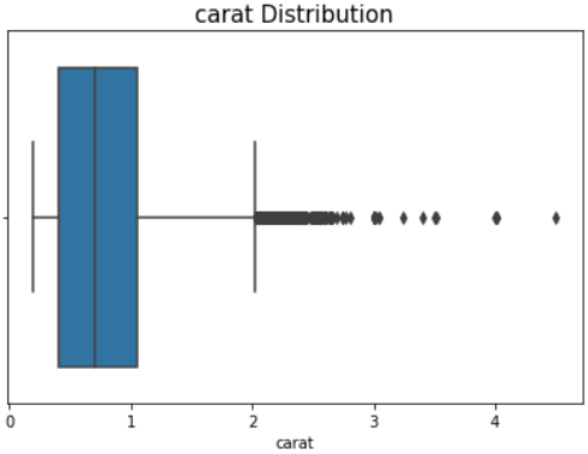
1.1.8 Checking for unique values of the numerical variables

```
carat   -  257
depth   - 169
table   -  112
x       -  531
y       -  526
z       -  356
price   - 8742
```

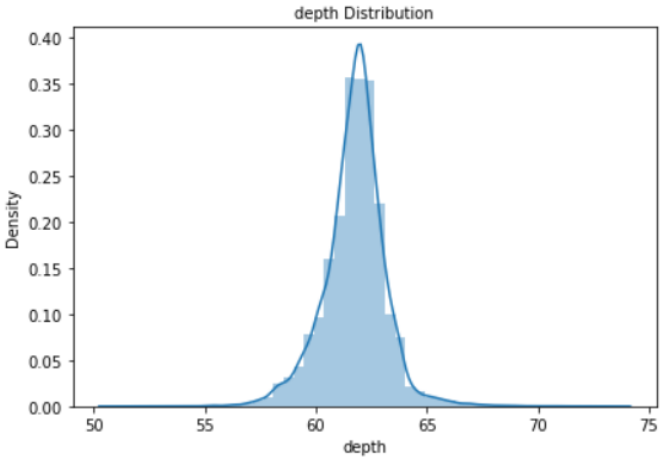
1.1.9 Performing Exploratory Data Analysis

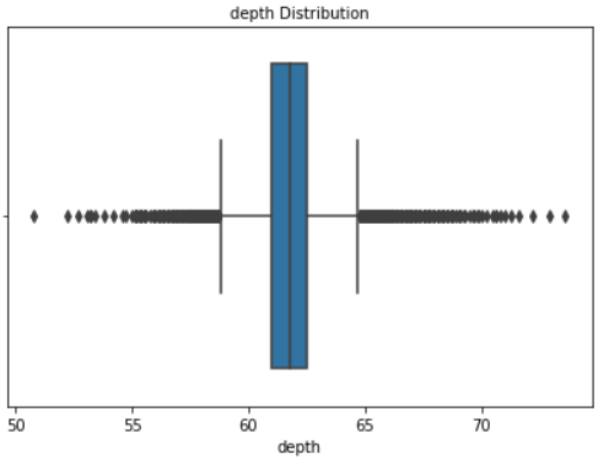
1.1.10 Univariate/Bivariate Analysis



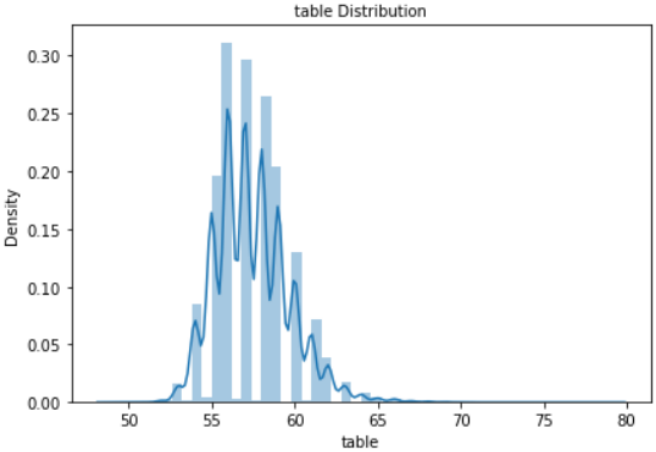


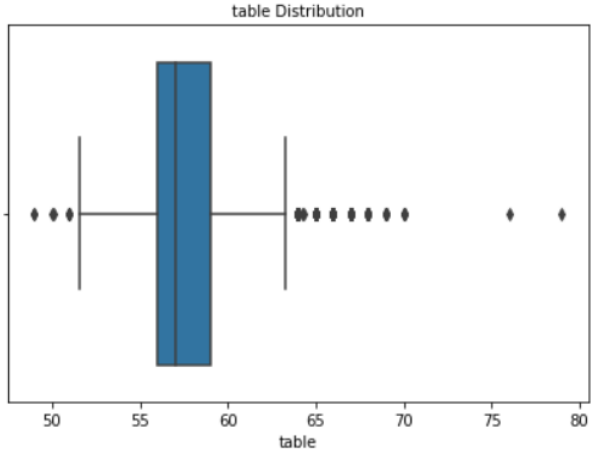
- **The distribution of data in "carat" seems to positively skewed, as there are multiple peaks points in the distribution and**
- **The box plot of carat seems to have large number of outliers.**
- **Majority of data lies In the range of 0 to 1 .**



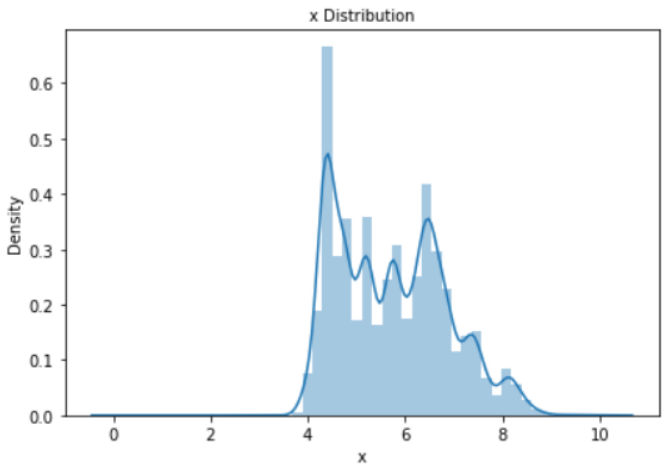


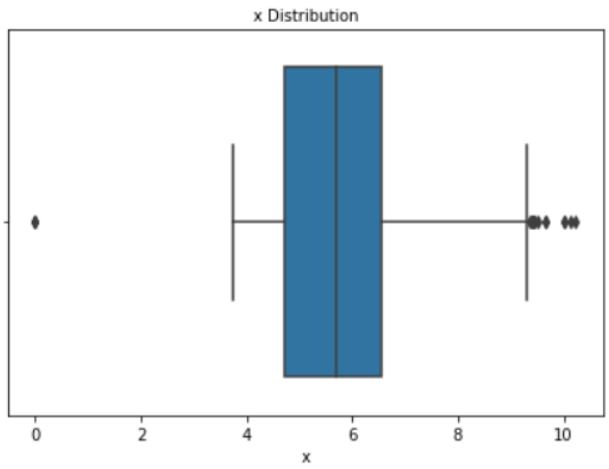
- **The distribution of "depth" seems to be a normal distribution.**
- **The data ranges from 55 to 65.**
- **The box plot of the depth distribution holds many outliers.**



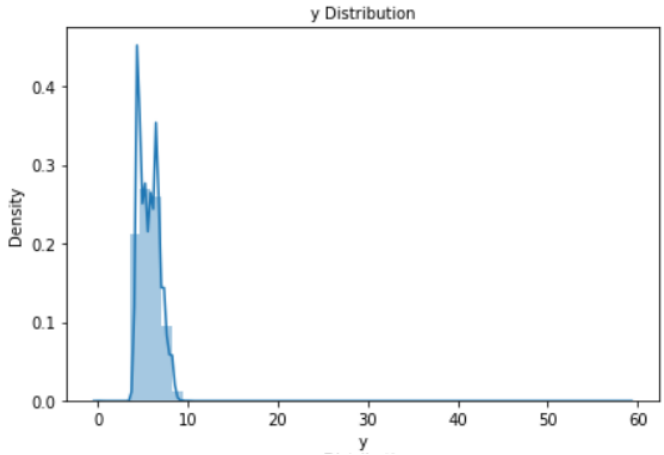


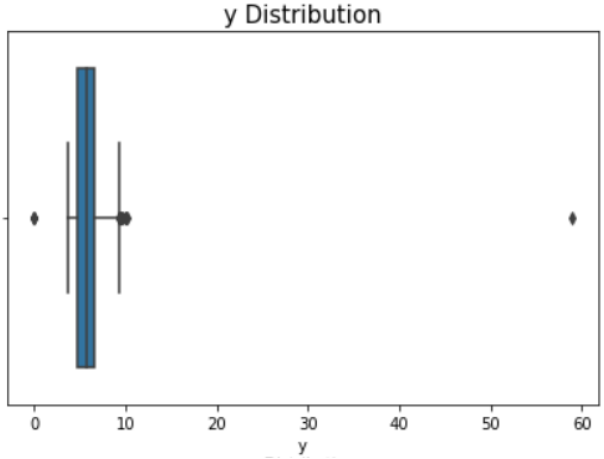
- **The distribution of "table" also seems to be positively skewed.**
- **The box plot of table has outliers.**
- **The data ranges from 55 to 65.**



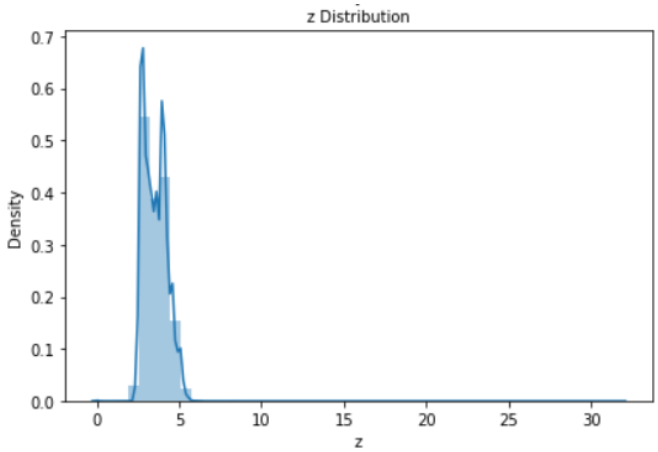


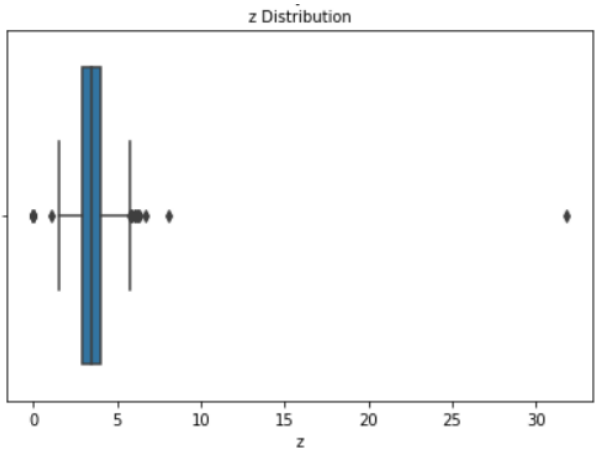
- **The distribution of x (Length of the cubic zirconia in mm.) is positively skewed.**
- **The box plot of the data consists of many outliers**
- **The distribution rages from 4 to 8**



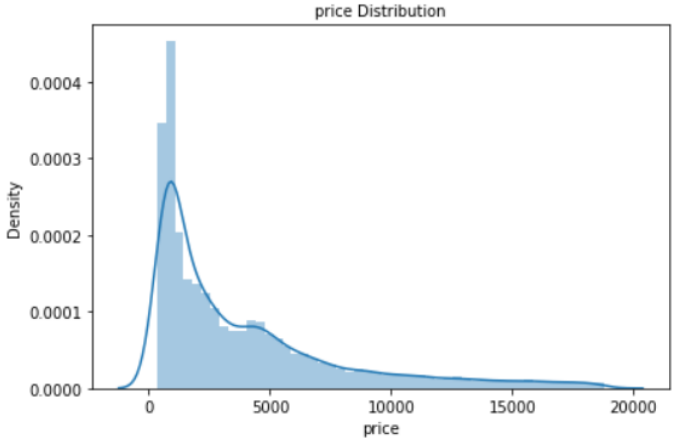


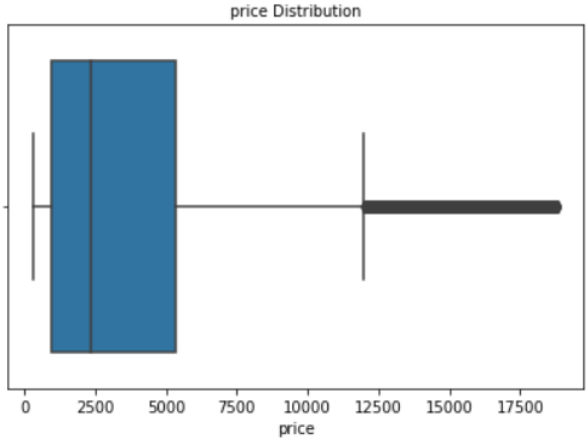
- **The distribution of "Y" (Width of the cubic zirconia in mm.) is positively skewed**
- **The skewness may be due to the diamonds are always made in specific shape.**
- **The box plot also consists of outliers.**
- **There might not be too much sizes in the market.**





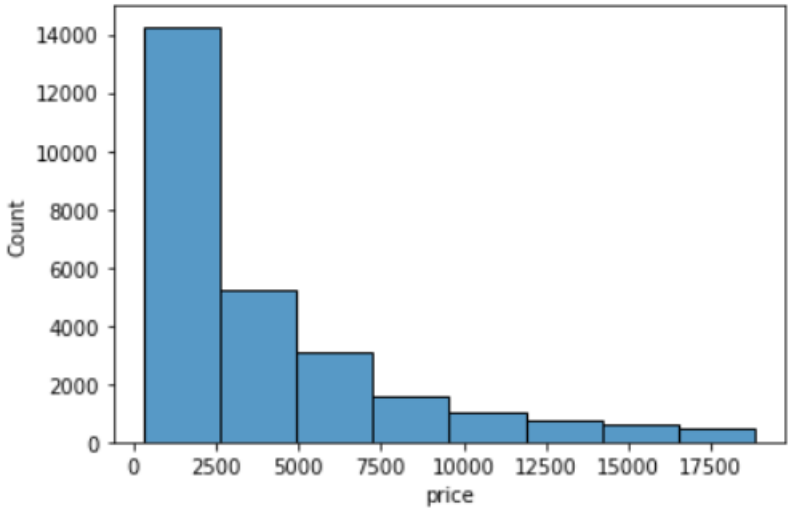
- **The distribution of "z" (Height of the cubic zirconia in mm.) is positively skewed**
- **The box plot also consists of outliers**
- **There might not be too much sizes in the market.**





- **The distribution of "price" seems to be positively skewed.**
- **The price boxplot has outliers in the data.**
- **The price distribution ranges from Rs 100 to 8000**

PRICE-HISTOGRAM



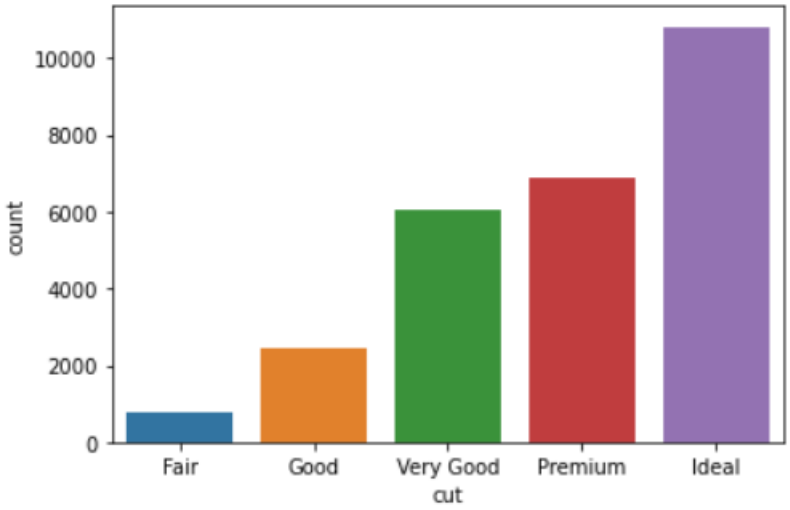
1.1.11 Observing the Skewness Present in the variables

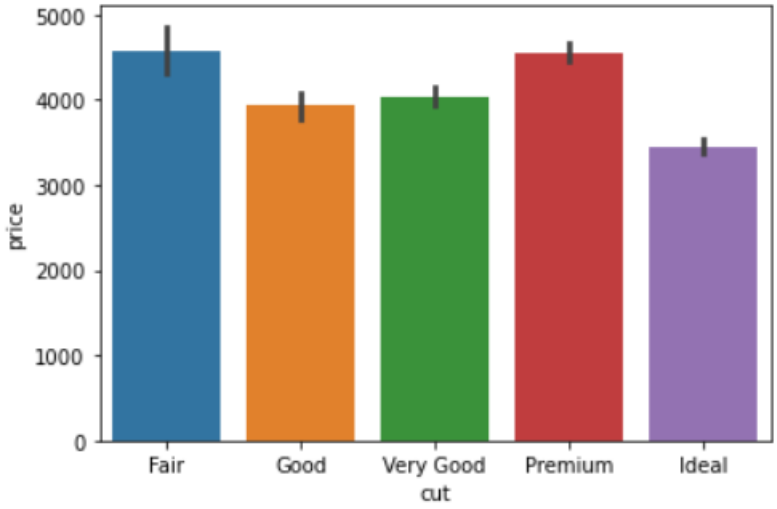
```
carat   -  1.116481
depth   -  -0.028618
table   -  0.765758
x       -  0.387986
y       -  3.850189
z       -  2.568257
price   -  1.618550
```

1.1.12 CATEGORICAL VARIABLES

CUT

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

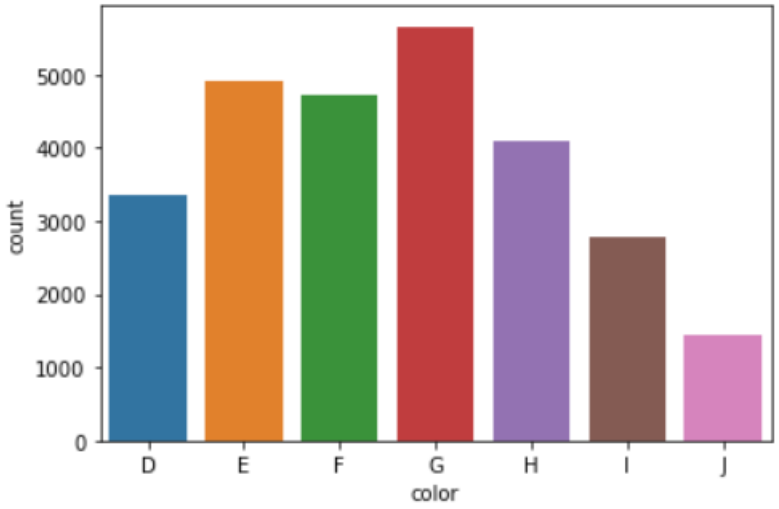


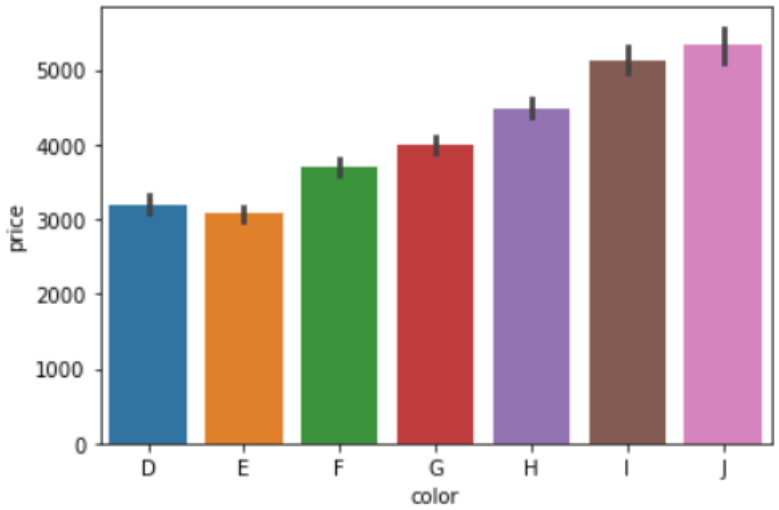


- **The most preferred cut seems to be ideal cut for diamonds.**
- **The reason for the most preferred cut ideal is because those diamonds are priced lower than other cuts.**

COLOR

D being the best and J the worst.

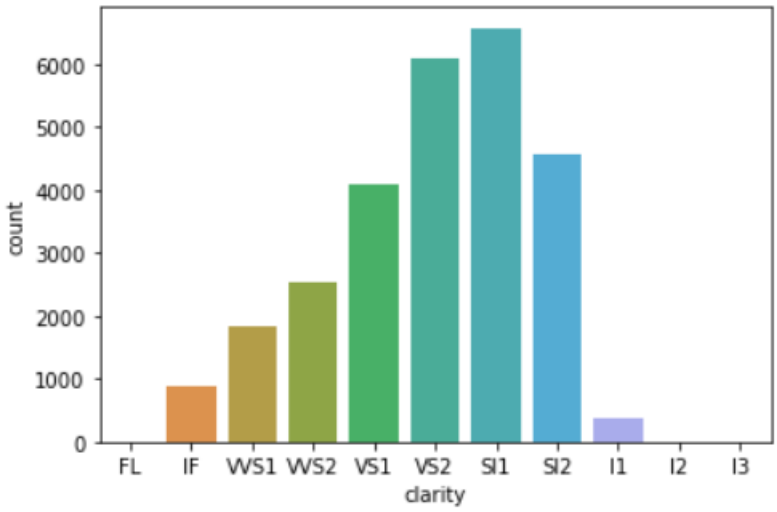


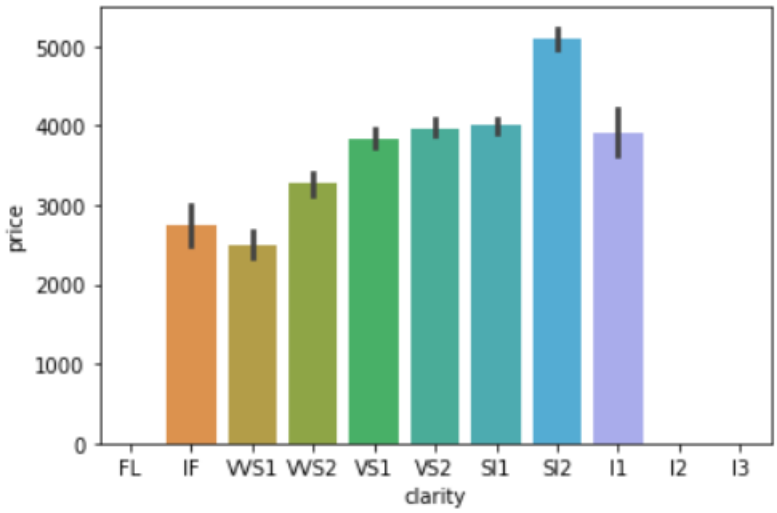


- **We have 7 colours in the data,**
- **The G seems to be the preferred colour as we see the G is priced in the middle of the seven colours, whereas J being the worst colour price seems too high.**

CLARITY:

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



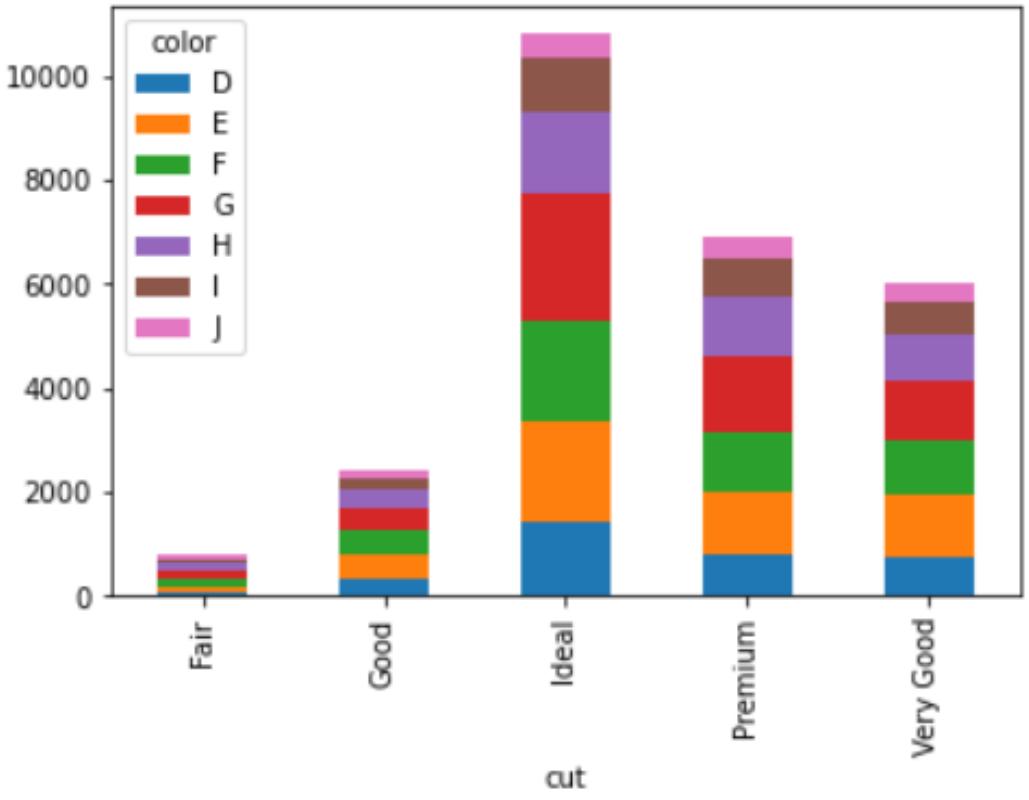


- **The clarity VS2 seems to be preferred by people as its performing the best with resect to its price.**
- **The data has No FL,I2,I3 diamonds, from this we can clearly understand the flawless diamonds are not bringing any profits to the store.**

MORE RELATIONSHIP BETWEEN CATEGORICAL VARIABLES

Cut and Color

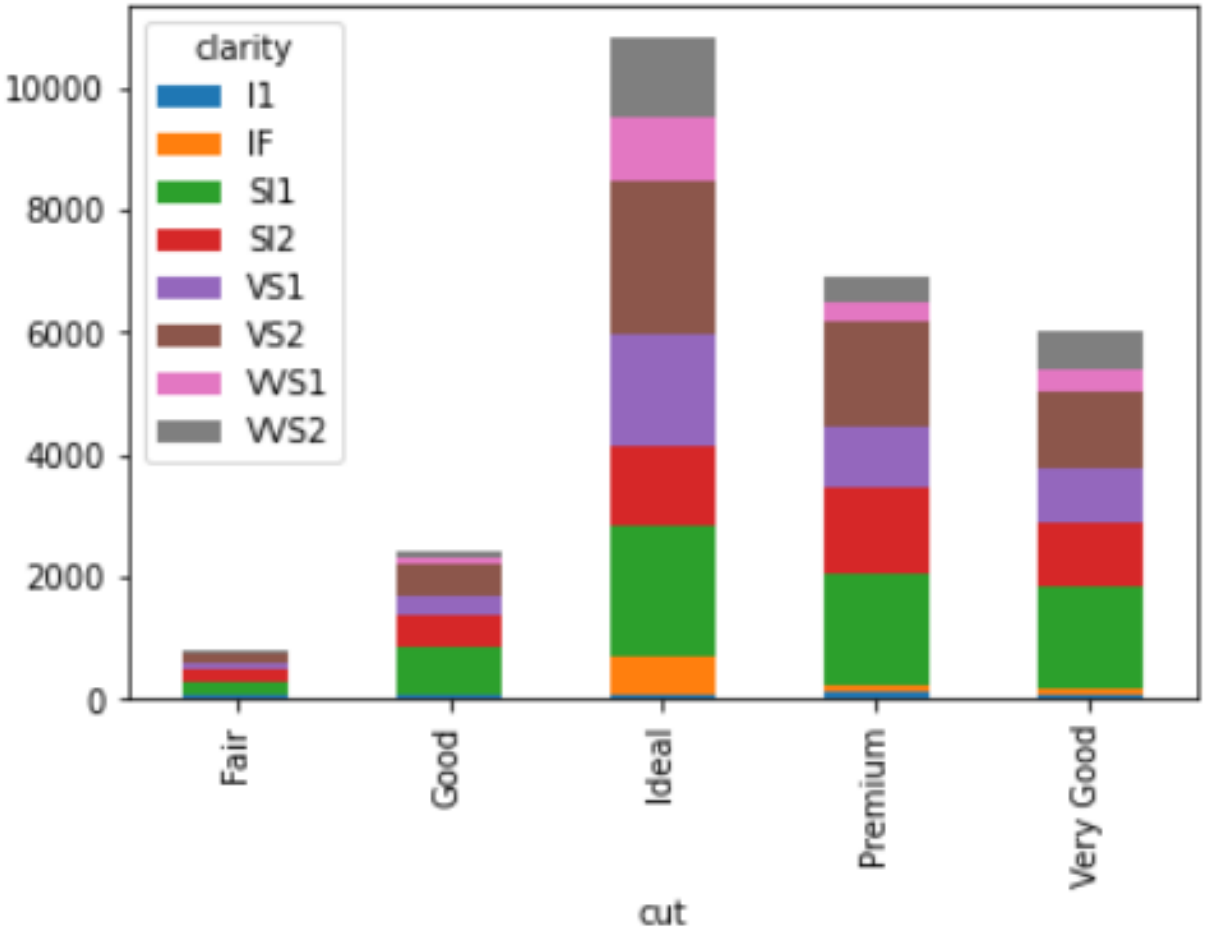
color cut	D	E	F	G	H	I	J
Fair	74	100	148	147	150	94	68
Good	311	491	454	419	352	253	161
Ideal	1409	1966	1893	2470	1552	1073	453
Premium	808	1174	1167	1471	1161	711	407
Very Good	742	1186	1067	1154	887	640	354



- **As it can be observed and stated before Ideal cut is the best performing**
- **With G as the most preferred type.**

Cut and Clarity

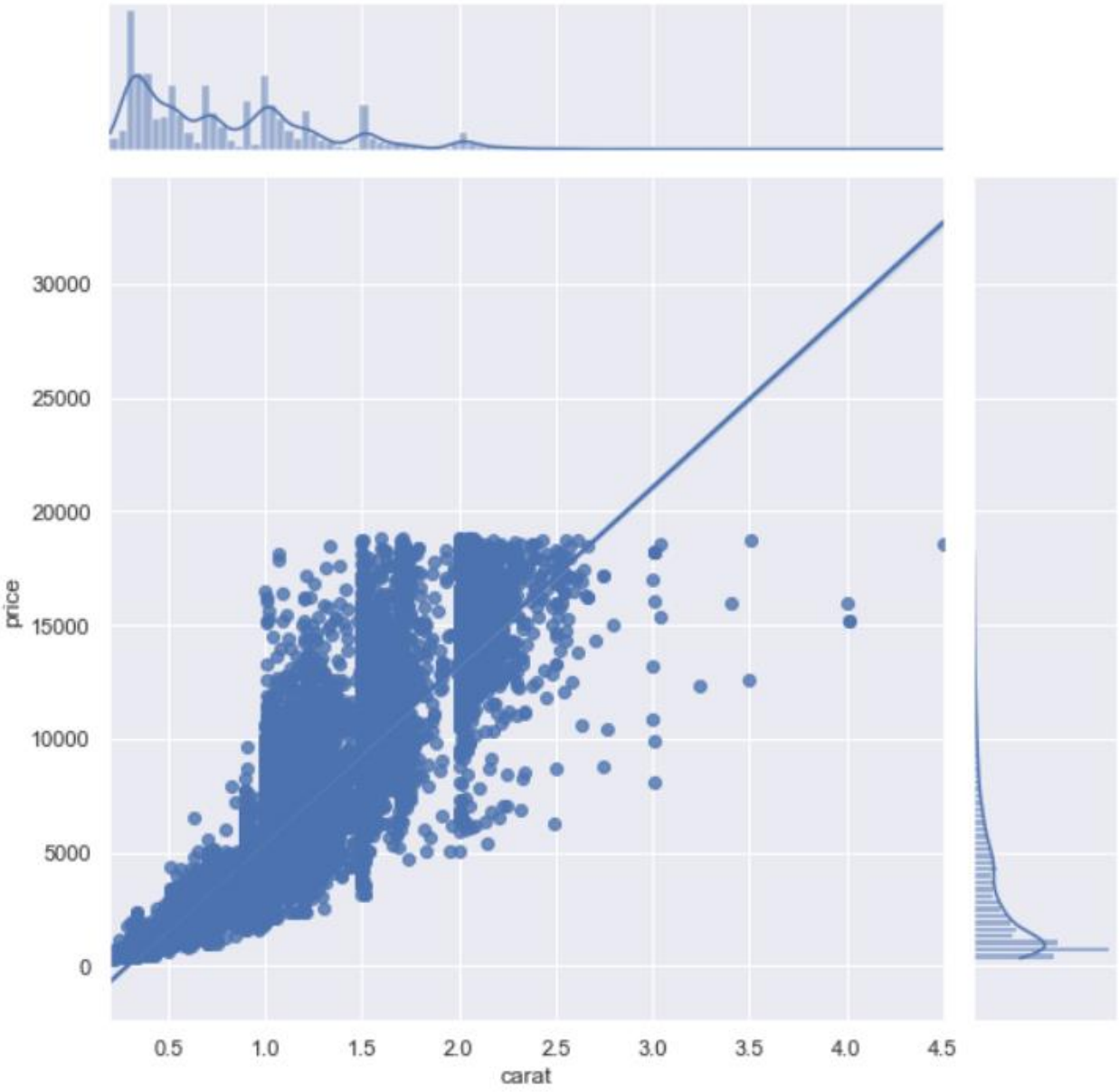
clarity cut	I1	IF	SI1	SI2	VS1	VS2	VVS1	VVS2
Fair	89	4	193	225	93	129	10	38
Good	51	30	765	530	331	491	100	143
Ideal	74	613	2150	1324	1784	2528	1036	1307
Premium	108	115	1809	1449	998	1697	307	416
Very Good	43	132	1654	1047	887	1254	386	627



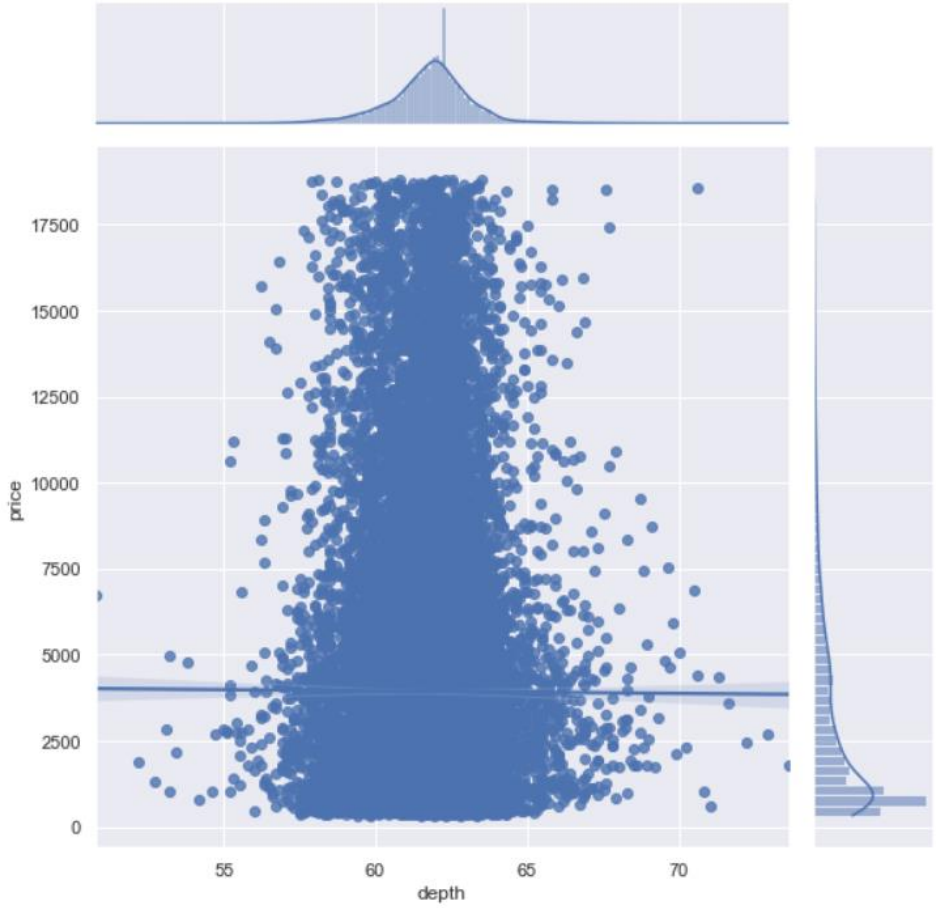
1.3 BIVARIATE ANALYSIS

1.3.2 Correlation w.r.t. Price

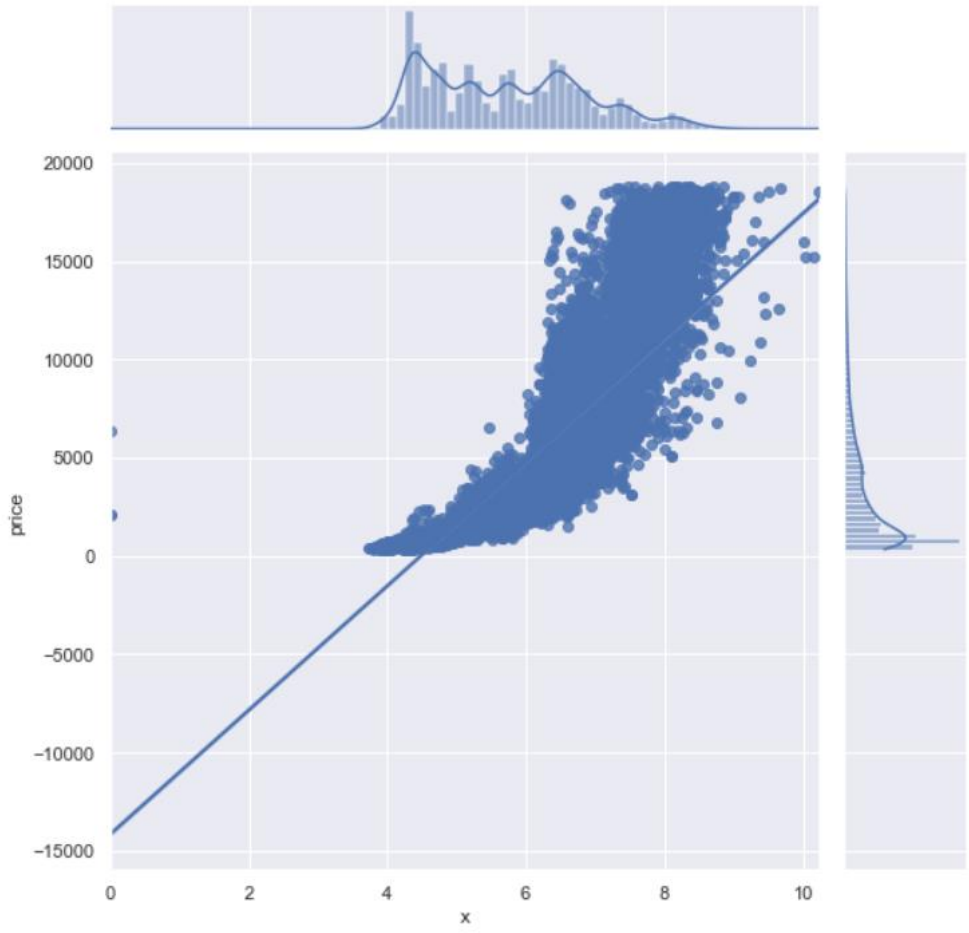
Carat v/s Price



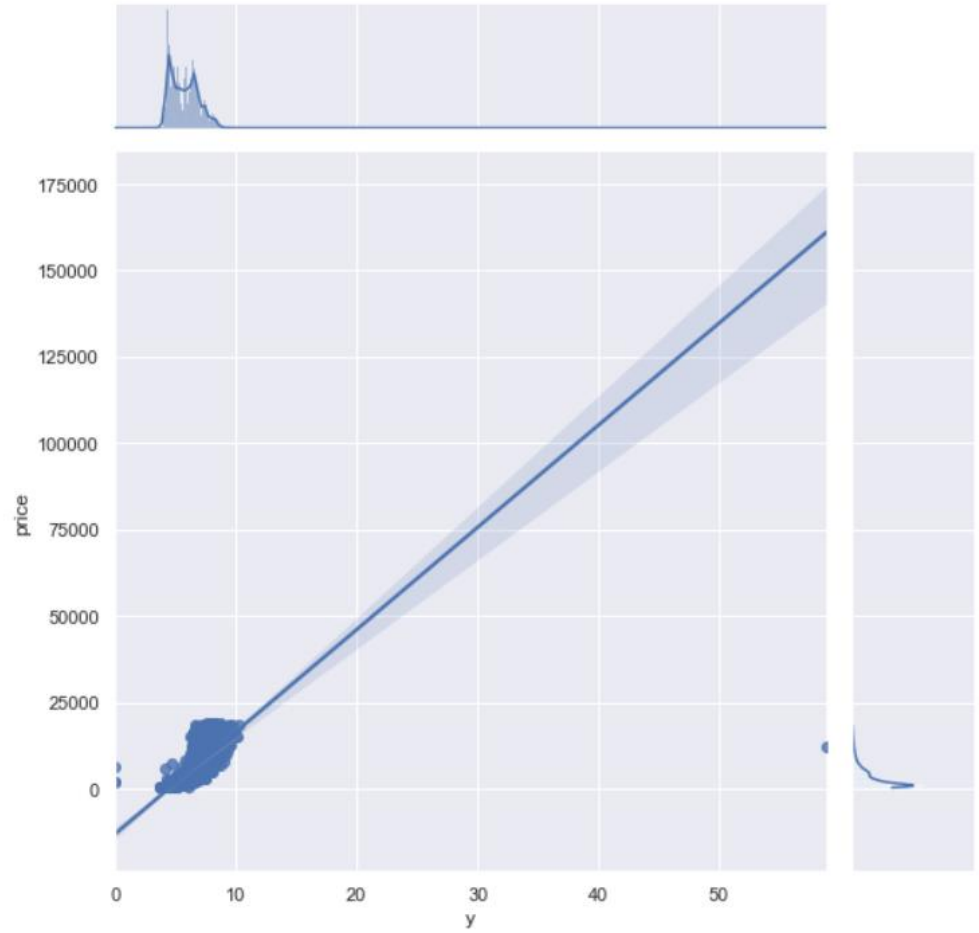
Depth v/s Price



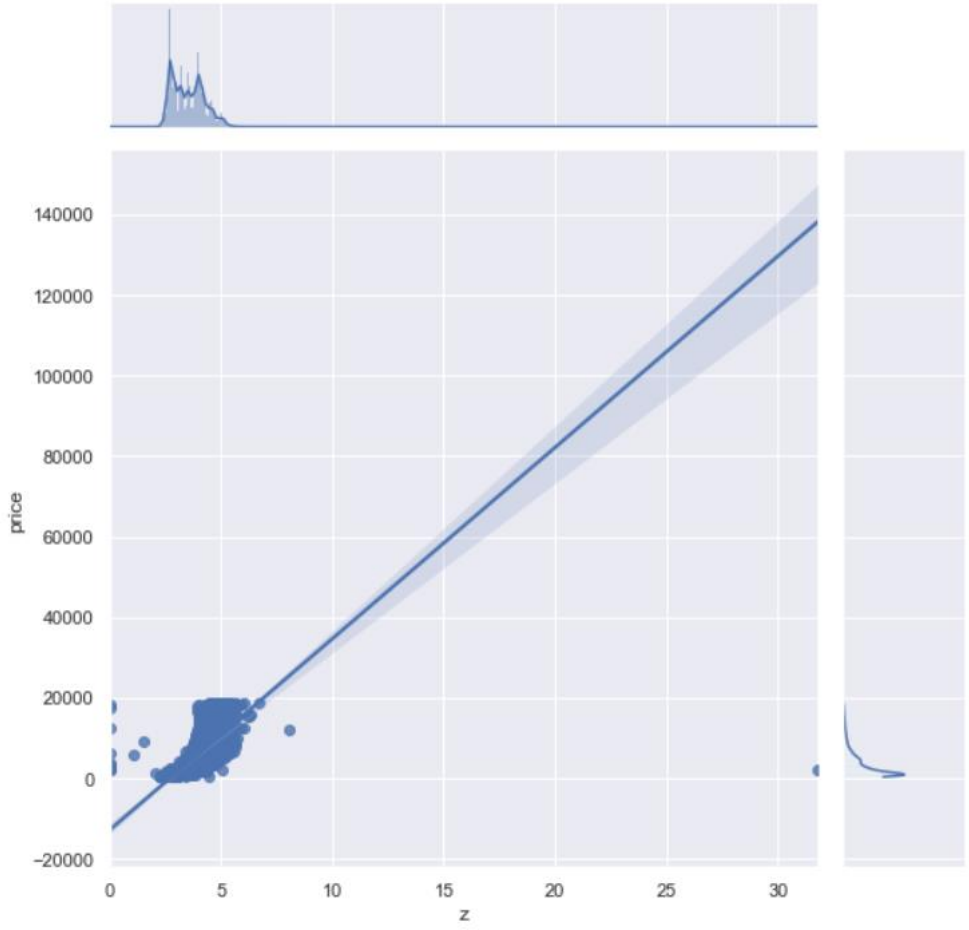
X v/s Price



Y v/s Price

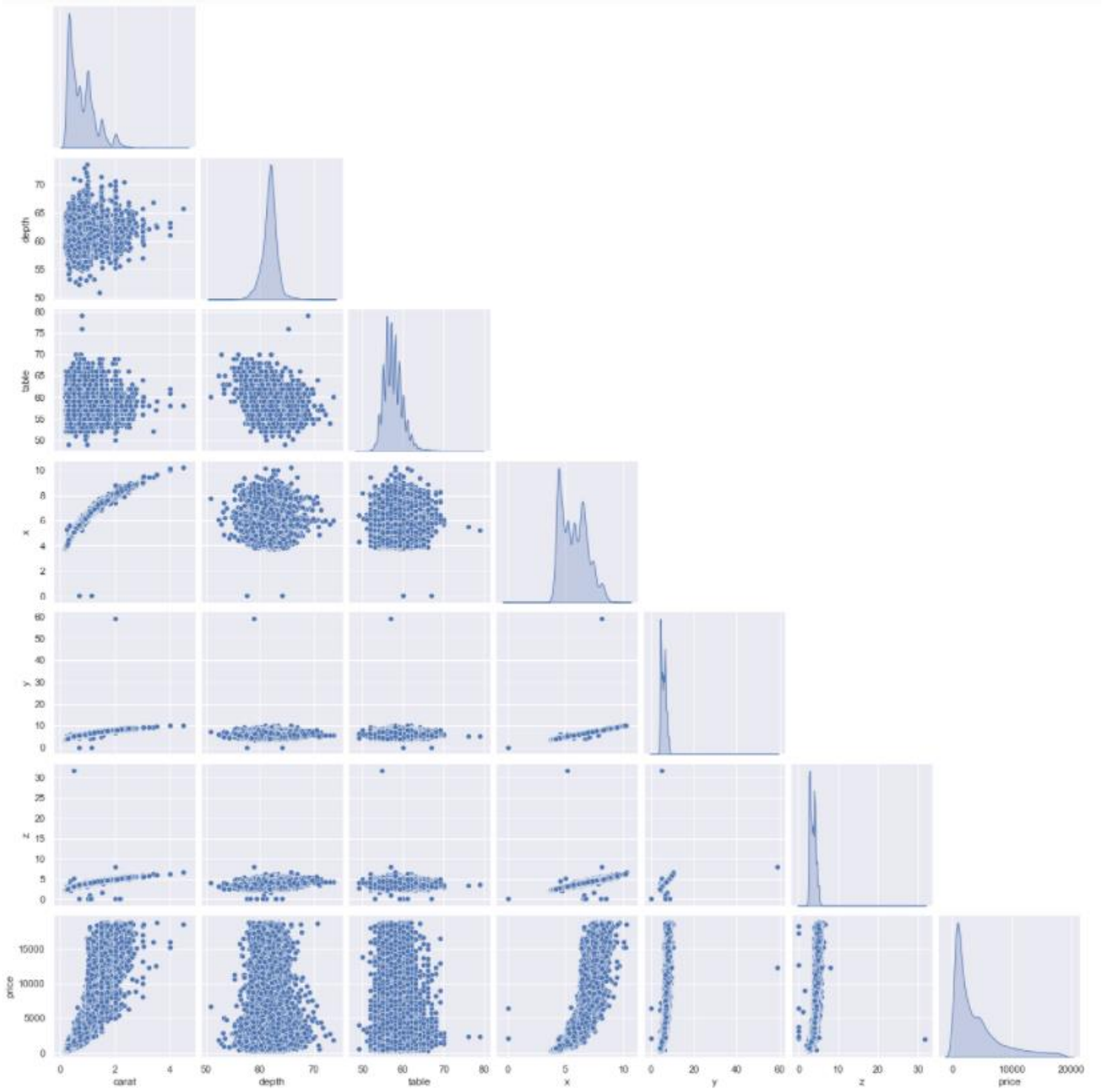


Z v/s Price



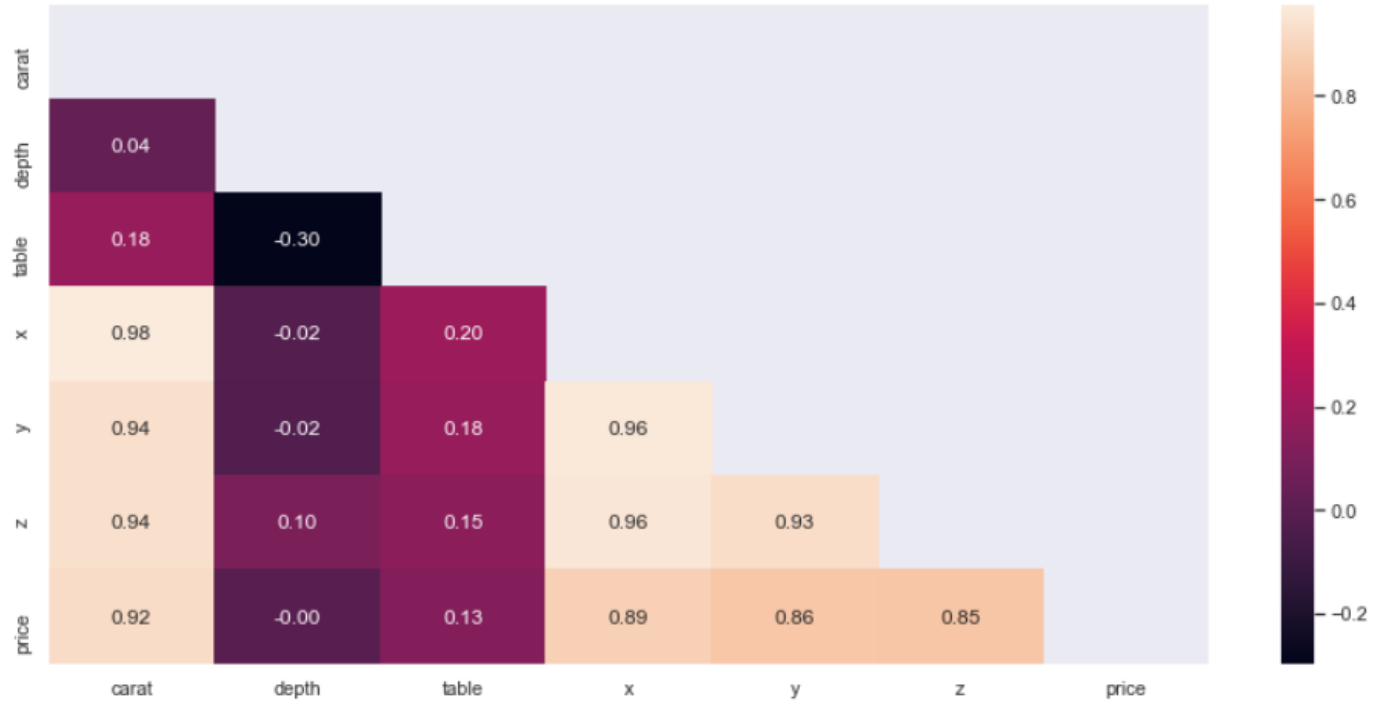
1.3.3 Pair Plot

A pair plot plots the relationships between all numeric variables in a dataset. The diagonal below is the histogram for each variable and shows the distribution. From the below plot, we can observe if there are relationships between every two pair of variables.



1.4.4 Correlation Heat Map

The correlation coefficient shown in the table below shows the degree of correlation between the two variables represented in X axis and Y axis. It varies between -1 (maximum negative correlation) to +1 (maximum positive correlation).



- **This matrix clearly shows the presence of multi collinearity in the dataset where lighter colours depict a high correlation and darker colours as the weakest correlation.**
- **Here carat has very high correlation to x,y,z and price.**

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?

Null Values

```
carat        0
cut          0
color        0
clarity      0
depth      697
table        0
x            0
y            0
z            0
price        0
```

Checking if there is value that is “0”

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

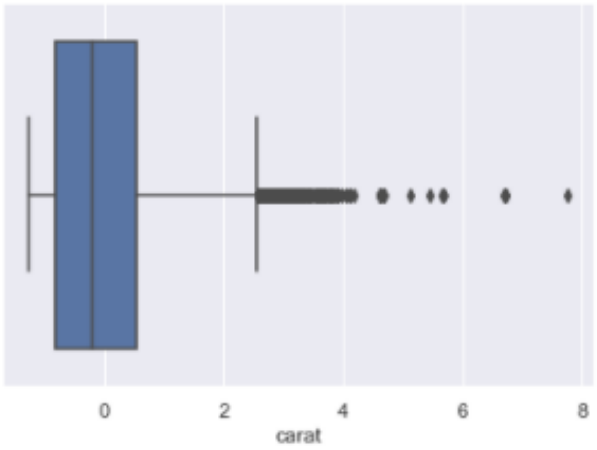
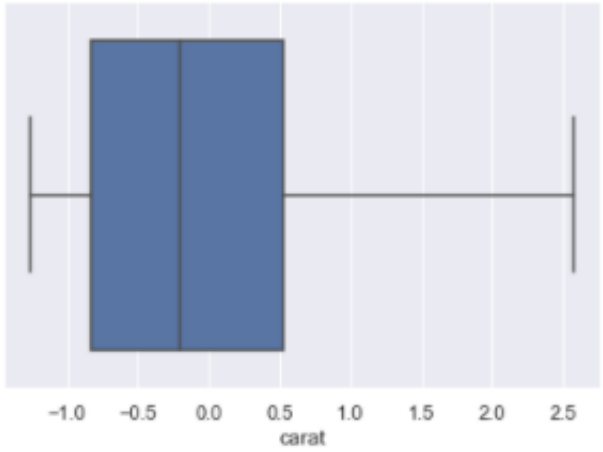
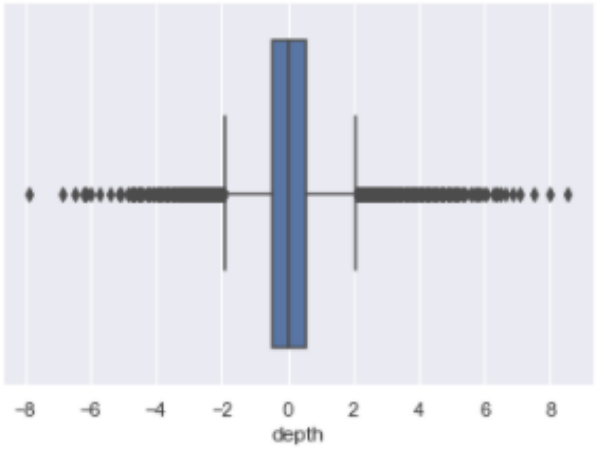
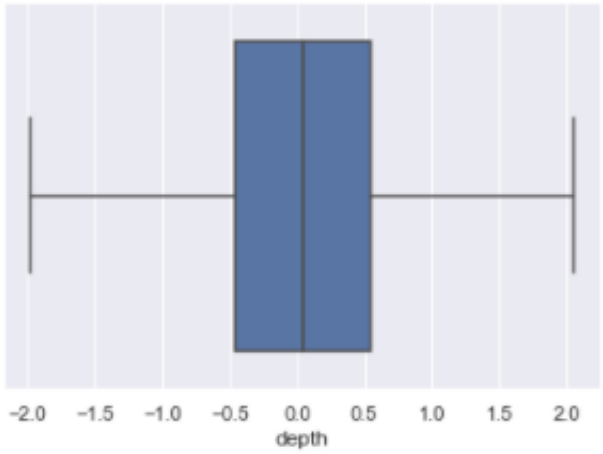
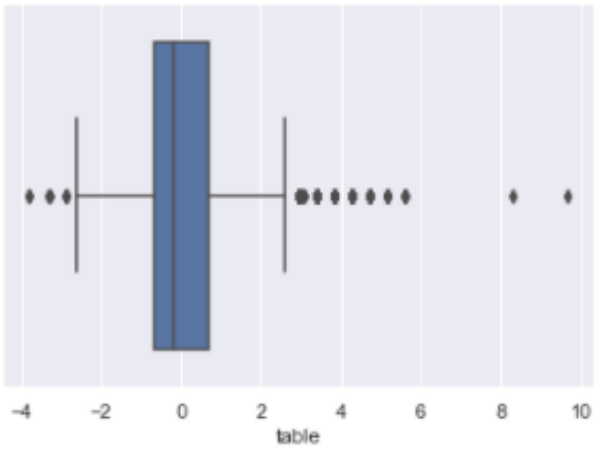
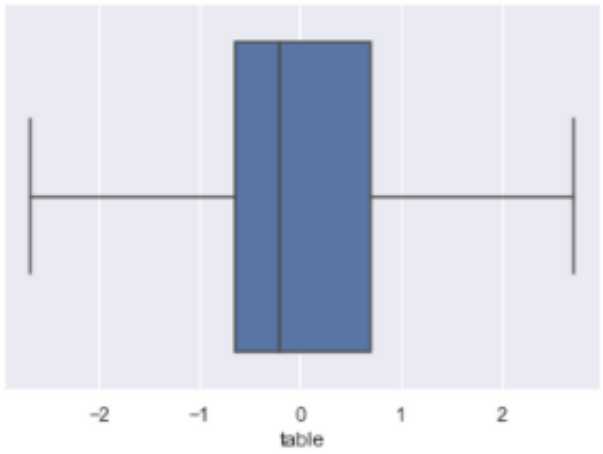
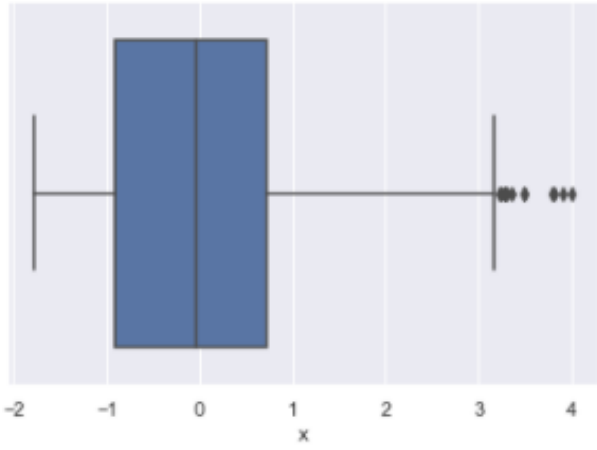
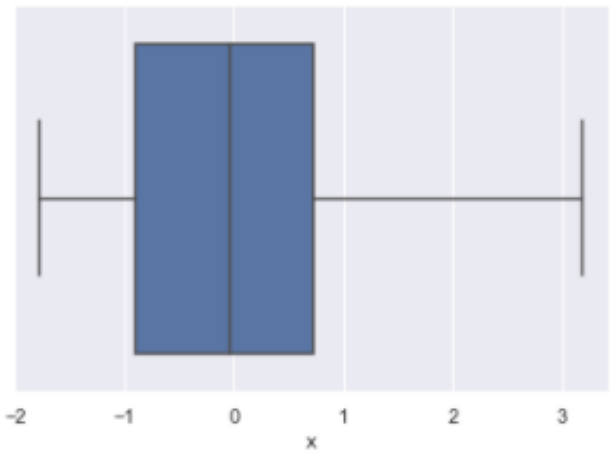
- We have certain rows having values zero, the x, y, z are the dimensions of a diamond so this can’t take into model. As there are very less rows.
- We can drop these rows as don’t have any meaning in model building.
- Yes, we have Null values in depth, since depth being continuous variable .
- Mean 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.

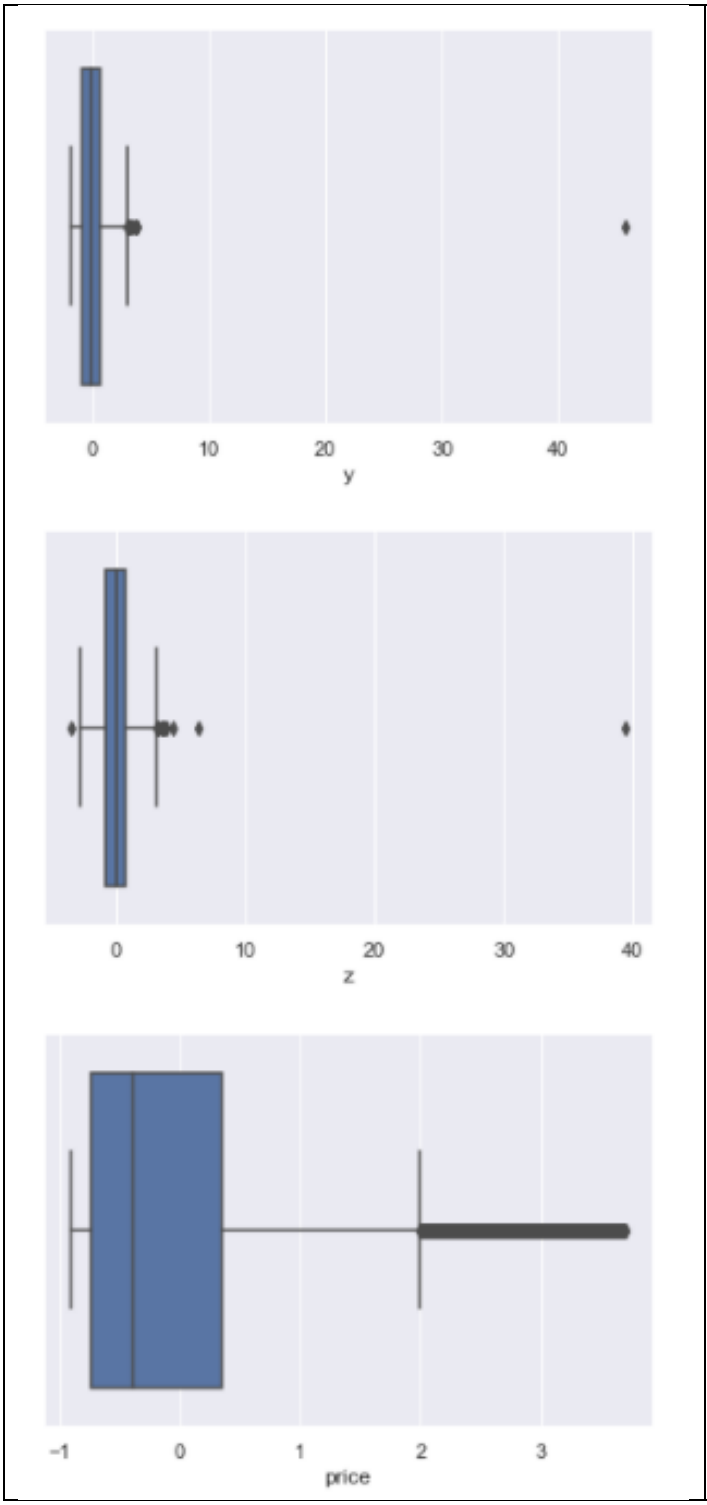
SCALING

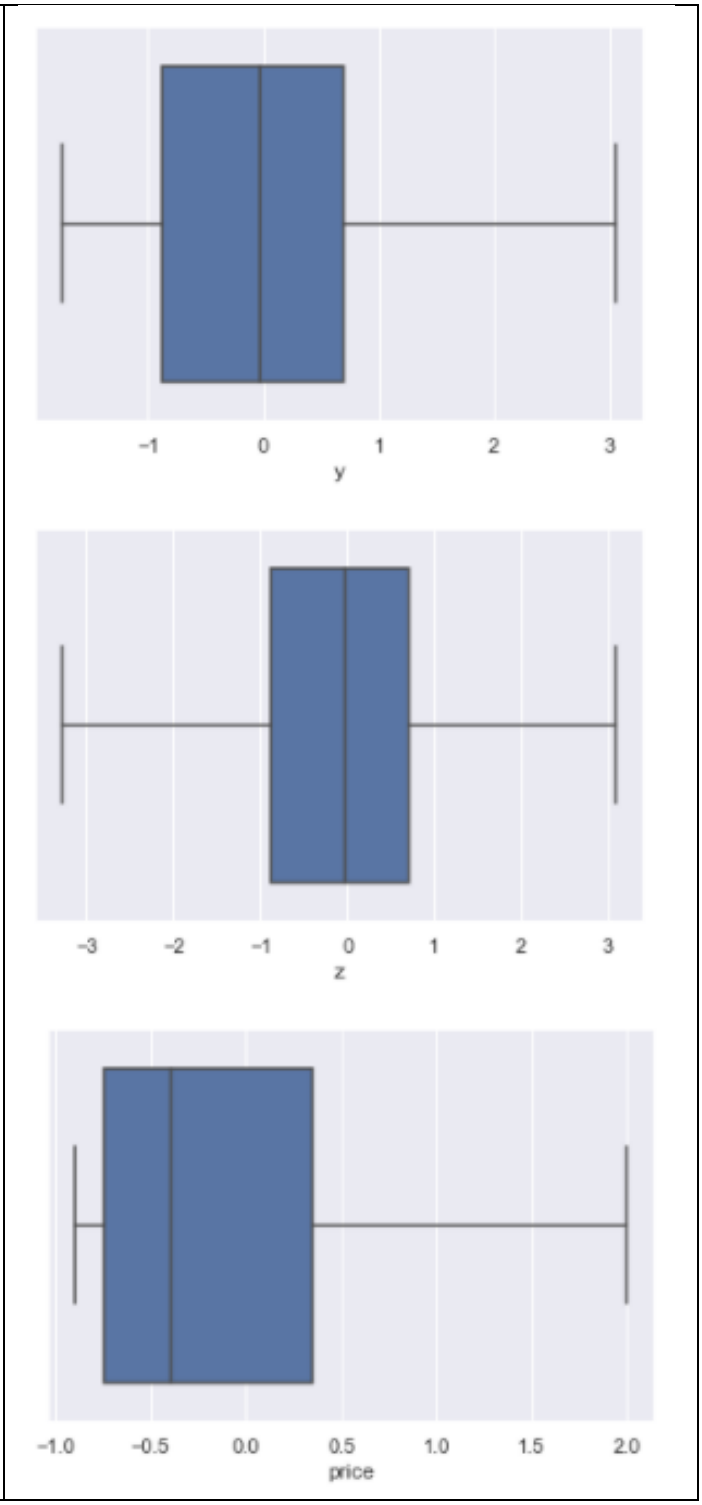
- **Scaling can be useful to reduce or check the multi collinearity in the data, so if scaling is not applied I find the VIF – variance inflation factor values very high. Which indicates presence of multi collinearity**
- ***These values are calculated after building the model of linear regression. To understand the multi collinearity in the model***
- ***The scaling had no impact in model score or coefficients of attributes nor the intercept.***

Scaled Dataset

	carat	cut	color	clarity	depth	table	x	y	z	price
0	-1.043125	Ideal	E	SI1	0.253399	0.244112	-1.295920	-1.240065	-1.224865	-0.854851
1	-0.980310	Premium	G	IF	-0.679158	0.244112	-1.162787	-1.094057	-1.169142	-0.734303
2	0.213173	Very Good	E	VVS2	0.325134	1.140496	0.275049	0.331668	0.335404	0.584271
3	-0.791865	Ideal	F	VS1	-0.105277	-0.652273	-0.807766	-0.802041	-0.806936	-0.709945
4	-1.022187	Ideal	F	VVS1	-0.966099	0.692304	-1.224916	-1.119823	-1.238796	-0.785257

Before Outlier Treatment	After Outlier Treatment
	
	
	
	





As observed, all the outliers have been successfully removed.

VIF Before Scaling	VIF After Scaling
carat ---> 112.29255171268198	*carat ---> 33.35086119845924*
depth ---> 506.27609400991827	*depth ---> 4.573918951598579*
table ---> 501.5989769690286	*table ---> 1.7728852812618963*
x ---> 1077.6155332643164	*x ---> 463.5542785436457*
y ---> 348.2633201055354	*y ---> 462.769821646584*
z ---> 377.8378400515017	*z ---> 238.65819968687333*
price ---> 13.530236240716032	*cut_Good ---> 3.6096181949437143*
	cut_Ideal ---> 14.34812508118844
	cut_Premium ---> 8.623414379121153
	cut_Very Good ---> 7.848451571723688
	color_E ---> 2.371070464762613

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

Converting categorical to dummy variables in data

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

5 rows × 24 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')
```

- **Dummies have been encoded.**
- **Linear regression model does not take categorical values therefore we have encoded categorical values to integer for better results.**

1.3.4 Splitting the data

Linear Regression Model

Invoking the LinearRegression function and finding the bestfit model on training data

```
LinearRegression()

The coefficient for carat is 1.100941784780449
The coefficient for depth is 0.005605143445570782
The coefficient for table is -0.01331950038680386
The coefficient for x is -0.305043498196334
The coefficient for y is 0.3039144895792659
The coefficient for z is -0.13916571567988056
The coefficient for cut_Good is 0.0940340291297785
The coefficient for cut_Ideal is 0.15231074620567447
The coefficient for cut_Premium is 0.14852774839849314
The coefficient for cut_Very Good is 0.12583881878452674
The coefficient for color_E is -0.04705442233369867
The coefficient for color_F is -0.04268437439142842
The coefficient for color_G is -0.10072161838356805
The coefficient for color_H is -0.20767313311661595
The coefficient for color_I is -0.3239541927462746
```

The coefficient for color_J is -0.4685893027501581
The coefficient for clarity_IF is 0.9997691394634906
The coefficient for clarity_SI1 is 0.6389785818271349
The coefficient for clarity_SI2 is 0.42959662348315747
The coefficient for clarity_VS1 is 0.8380875826737575
The coefficient for clarity_VS2 is 0.7660244466083624
The coefficient for clarity_VVS1 is 0.9420769630114086
The coefficient for clarity_VVS2 is 0.9313670288415694

1.3.5 Intercept for the model

The intercept for our model is -0.7567627863049398

1.3.6 R square on training data

The R-squared score on training data is: 0.9419557931252712

1.3.7 R square on testing data

The R-squared score on testing data is: 0.9381643998102491

1.3.8 RMSE on Training data

The RMSE score on training data is: 0.20690072466418796

1.3.9 RMSE on Testing data

The R-squared score on testing data is: 0.21647817772382866

We still find we have multi collinearity in the dataset, to drop these values to a further lower level we can drop columns after performing 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 than 5%.***

1.3.10 STATSMODEL

1.3.11 Best Parameters

Intercept	-0.756763
carat	1.100942
depth	0.005605
table	-0.013320
x	-0.305043
y	0.303914
z	-0.139166
cut_Good	0.094034
cut_Ideal	0.152311
cut_Premium	0.148528
cut_Very_Good	0.125839
color_E	-0.047054
color_F	-0.062684
color_G	-0.100722
color_H	-0.207673
color_I	-0.323954
color_J	-0.468589
clarity_IF	0.999769
clarity_SI1	0.638979
clarity_SI2	0.429597
clarity_VS1	0.838088
clarity_VS2	0.766024
clarity_VVS1	0.942077
clarity_VVS2	0.931367

1.3.12 Inferential Statistics

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, 04 Jun 2021	Prob (F-statistic):	0.00
Time:	15:59:22	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):	17654.853
Skew:	1.208	Prob(JB):	0.00
Kurtosis:	7.076	Cond. No.	57.0

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

As it can be observed above the P-value for depth variable is 0.127 which is greater than our alpha i.e 0.05, depicting multicollinearity present therefore we will drop the variable depth and perform the statsmodel again.

- ***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.***

1.3.13 RMSE Value

Squareroot of mean_sq_error is standard deviation i.e. avg variance between predicted and actual - 0.21647817772382846

After dropping the depth variable

1.3.14 OLS Regression Results

Best Parameters

Intercept	*-0.756657*
carat	*1.101954*
table	*-0.013928*
x	*-0.315617*
y	*0.283420*
z	*-0.108789*
cut_Good	*0.095123*
cut_Ideal	*0.151173*
cut_Premium	*0.147355*
cut_Very_Good	*0.125514*
color_E	*-0.047114*
color_F	*-0.062727*
color_G	*-0.100657*
color_H	*-0.207568*
color_I	*-0.323689*
color_J	*-0.468428*
clarity_IF	*1.000046*
clarity_SI1	*0.639804*
clarity_SI2	*0.430195*
clarity_VS1	*0.838626*
clarity_VS2	*0.766683*
clarity_VVS1	*0.942390*
clarity_VVS2	*0.931898*

1.3.15 *Inferential Statistics*

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, 04 Jun 2021	Prob (F-statistic):	0.00
Time:	15:59:22	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

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

- **The overall P value is less than alpha, so rejecting H0 and accepting Ha that atleast 1 regression co-efficient is not 0. Here all regression co-efficients are not 0.**

1.3.16 Concatenate X and y into a single dataframe

	carat	depth	table	x	y	z	cut_Good	cut_Ideal	cut_Premium	cut_Very Good	...	color_I	color_J	clarity_IF	clarity_SI1
7598	-0.184654	1.114220	0.692304	-0.186479	-0.106356	-0.026801	0	0	0	1	...	0	0	0	0
8882	-1.043125	0.827280	0.244112	-1.295920	-1.222887	-1.169142	0	0	0	1	...	0	0	0	0
22763	-0.205593	1.544631	0.692304	-0.079973	-0.114944	0.070716	1	0	0	0	...	0	0	0	0
6643	-0.917495	-1.109569	-0.652273	-0.958650	-0.956637	-1.043763	0	1	0	0	...	0	0	0	0
18701	1.804485	0.899015	-0.204081	1.526498	1.516910	1.630986	0	0	0	1	...	1	0	0	1

5 rows × 24 columns

The Root Mean Squared Error - RMSE = 0.20690072466418777

1.3.17 The Complete Equation Achieved after Linear Regression

*(-0.76) * Intercept + (1.1) * carat + (0.01) * depth + (-0.01) * table + (-0.31) * x + (0.3) * y + (-0.14) * z + (0.09) * 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 +*

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 **approximately 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

- 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.
- The clarity of the diamond is the next important attributes the more the clear is the stone the profits are more
- The best attributes are

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

2 Problem 2: Logistic Regression and LDA

You are hired by a tour and travel agency which deals in selling holiday packages. You are provided details of 872 employees of a company. Among these employees, some opted for the package and some didn't. You have to help the company in predicting whether an employee will opt for the package or not on the basis of the information given in the data set. Also, find out the important factors on the basis of which the company will focus on particular employees to sell their packages.

2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, write an inference on it. Perform Univariate and Bivariate Analysis. Do exploratory data analysis.

- **Loading all the necessary library for the model building and reading the head and tail of the dataset to check whether data has been properly fed**

2.1.1 Head of the Data

	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
0	no	48412	30	8	1	1	no
1	yes	37207	45	8	0	1	no
2	no	58022	46	9	0	0	no
3	no	66503	31	11	2	0	no
4	no	66734	44	12	0	2	no

Tail of the Data

	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
867	no	40030	24	4	2	1	yes
868	yes	32137	48	8	0	0	yes
869	no	25178	24	6	2	0	yes
870	yes	55958	41	10	0	1	yes
871	no	74659	51	10	0	0	yes

2.1.2 Checking the Shape of the Data

```
The total number of rows present in the dataset above is :   872
The total number of columns/variables present in the dataset above is :   7
```

2.1.3 Checking the info of the Data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 872 entries, 0 to 871
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Holliday_Package   872 non-null    object
 1   Salary             872 non-null    int64 
 2   age                872 non-null    int64 
 3   educ               872 non-null    int64 
 4   no_young_children  872 non-null    int64 
 5   no_older_children  872 non-null    int64 
 6   foreign            872 non-null    object
dtypes: int64(5), object(2)
memory usage: 47.8+ KB
```

```
Holliday_Package     0
Salary               0
age                  0
educ                 0
no_young_children    0
no_older_children    0
foreign              0
dtype: int64
```

- ***No null values in the dataset ,***
- ***We have integer and object data***

2.1.4 Descriptive Analysis

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Holliday_Package	872	2	no	471	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Salary	872	NaN	NaN	NaN	47729.2	23418.7	1322	35324	41903.5	53469.5	236961
age	872	NaN	NaN	NaN	39.9553	10.5517	20	32	39	48	62
educ	872	NaN	NaN	NaN	9.30734	3.03626	1	8	9	12	21
no_young_children	872	NaN	NaN	NaN	0.311927	0.61287	0	0	0	0	3
no_older_children	872	NaN	NaN	NaN	0.982798	1.08679	0	0	1	2	6
foreign	872	2	no	656	NaN	NaN	NaN	NaN	NaN	NaN	NaN

- *We have integer and continuous data,*
- *Holiday package is our target variable*
- *Salary, age, educ and number young children, number older children of employee have the went to foreign, these are the attributes we have to cross examine and help the company predict weather the person will opt for holiday package or not.*

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

2.1.5 Checking unique values for categorical variables

```
HOLLIDAY_PACKAGE :  2
yes     401
no      471
Name: Holliday_Package, dtype: int64
```

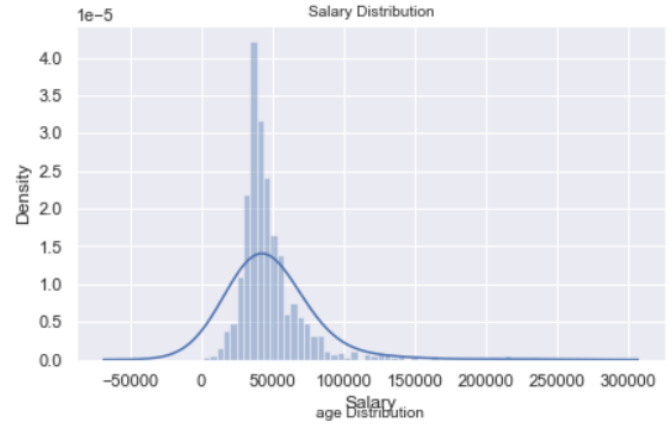
```
FOREIGN :  2
yes     216
no      656
Name: foreign, dtype: int64
```

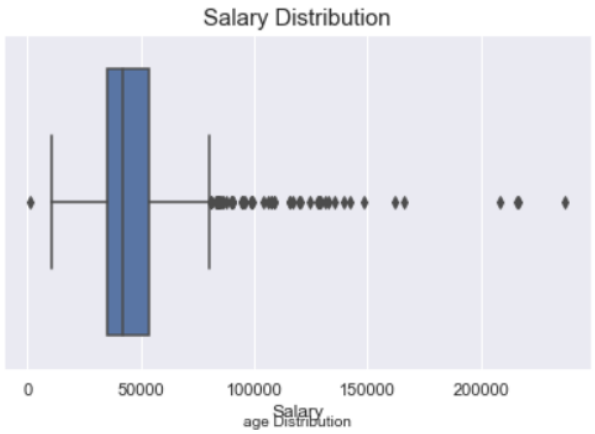
2.1.6 Percentage of target :

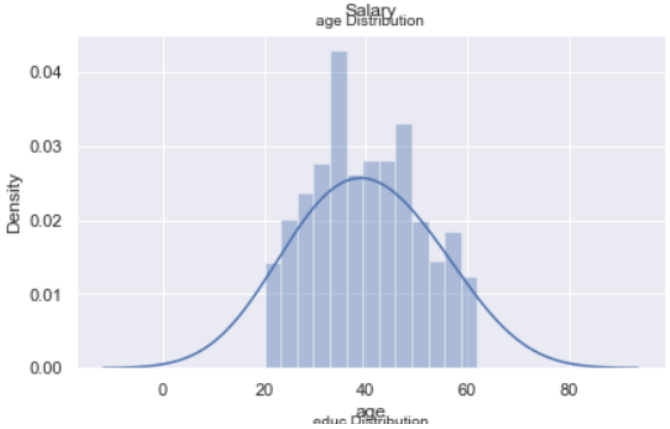
```
no      0.540138
yes     0.459862
Name: Holliday_Package, dtype: float64
```

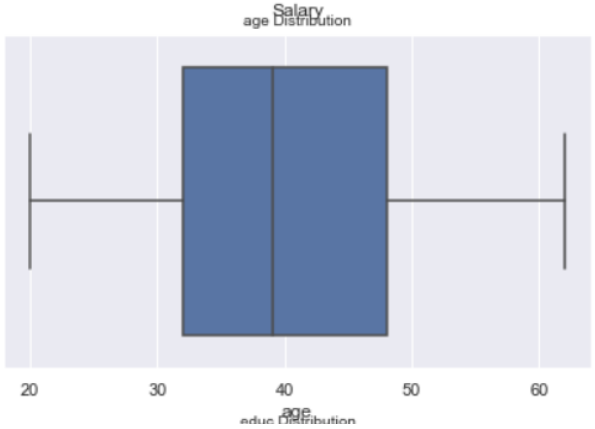
This split indicates that <u>46%</u> of employees are interested in the holiday package.

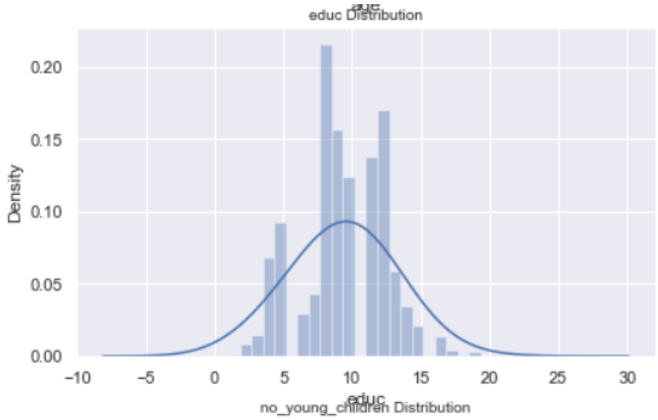
2.1.7 Univariate / Bivariate analysis

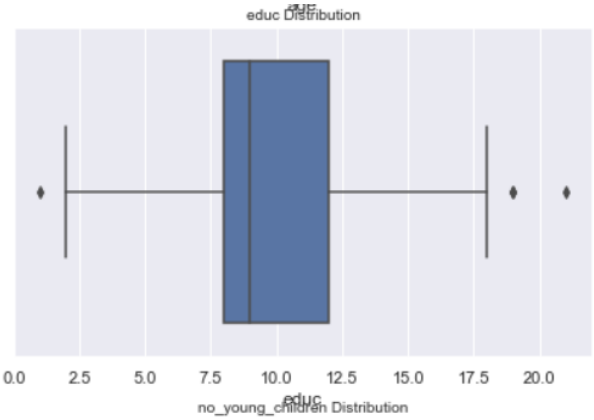


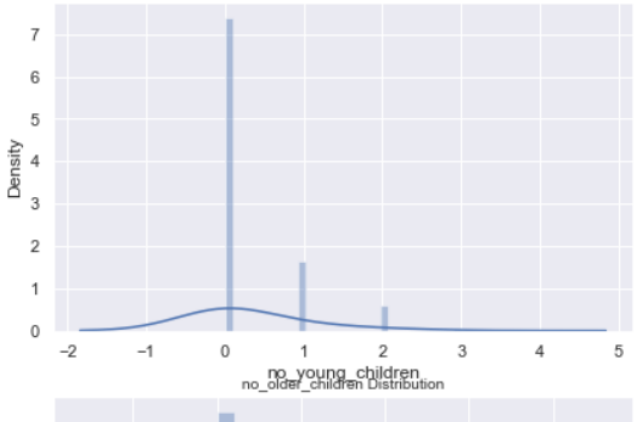


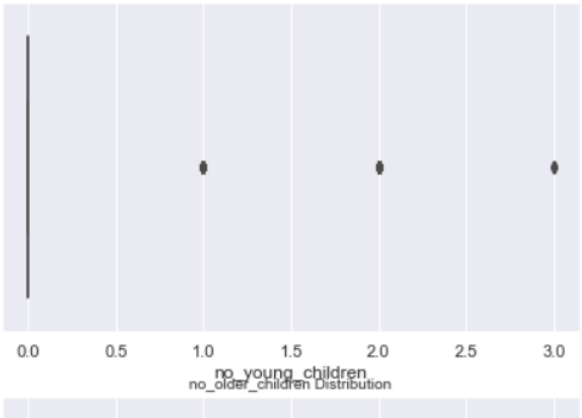


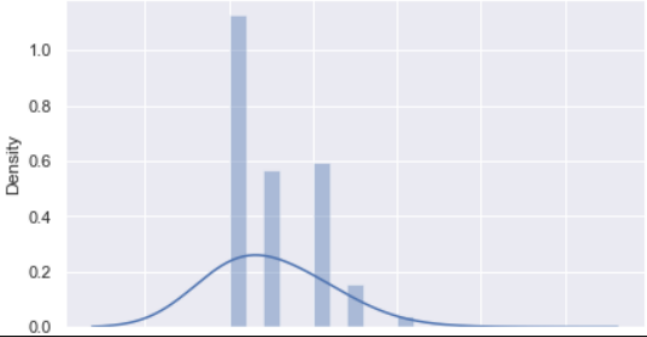


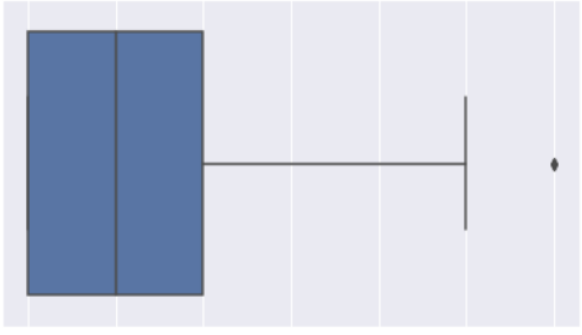












```
Index(['Holliday_Package', 'Salary', 'age', 'educ', 'no_young_children',
       'no_older_children', 'foreign'],
      dtype='object')
```

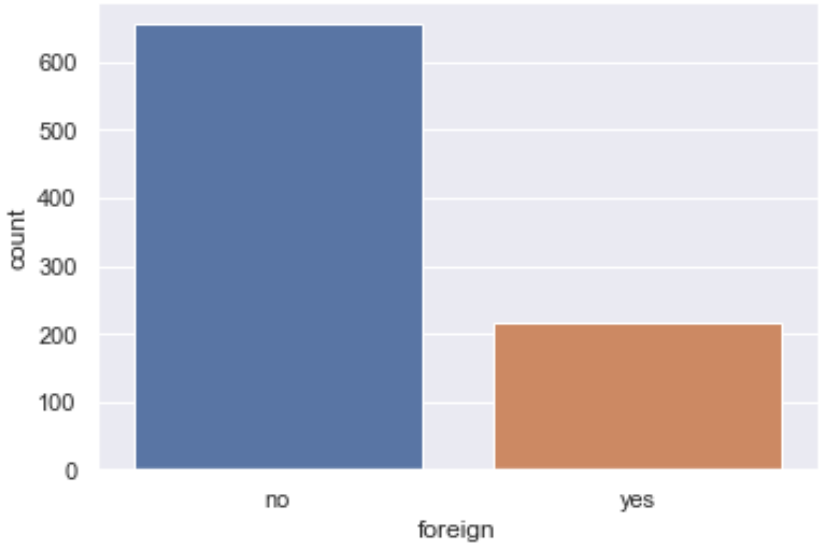
- **We can observe age seems to be normally distributed while there is skewness present in other variables.**
- ***Moreover, outliers are also present in the data. I will not be removing them as they appear to be genuine as a person can have extremely high salary and similarly for other variables.***

2.1.8 Skewness

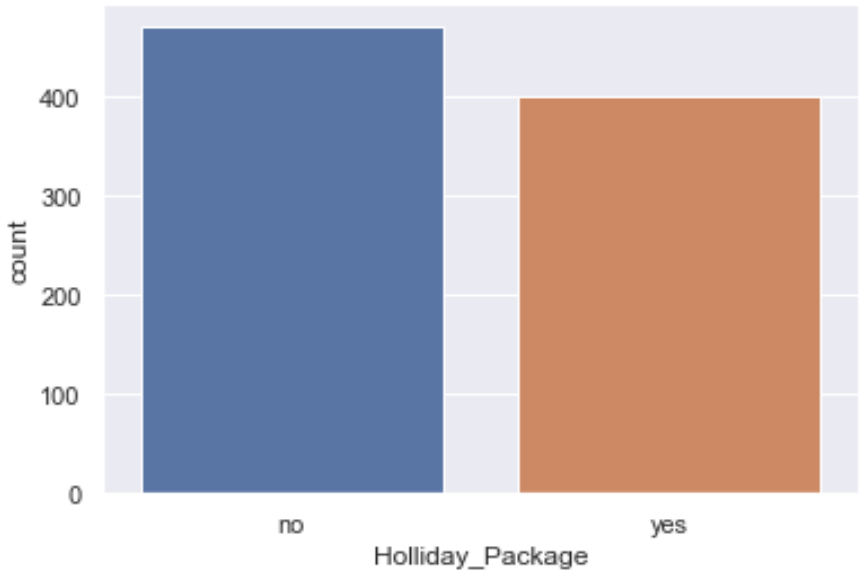
Salary	*3.103216*
age	*0.146412*
educ	*-0.045501*
no_young_children	*1.946515*
no_older_children	*0.953951*

2.1.9 Univariate Analysis for Categorical Variables

2.1.10 FOREIGN



2.1.11 HOLLIDAY PACKAGE

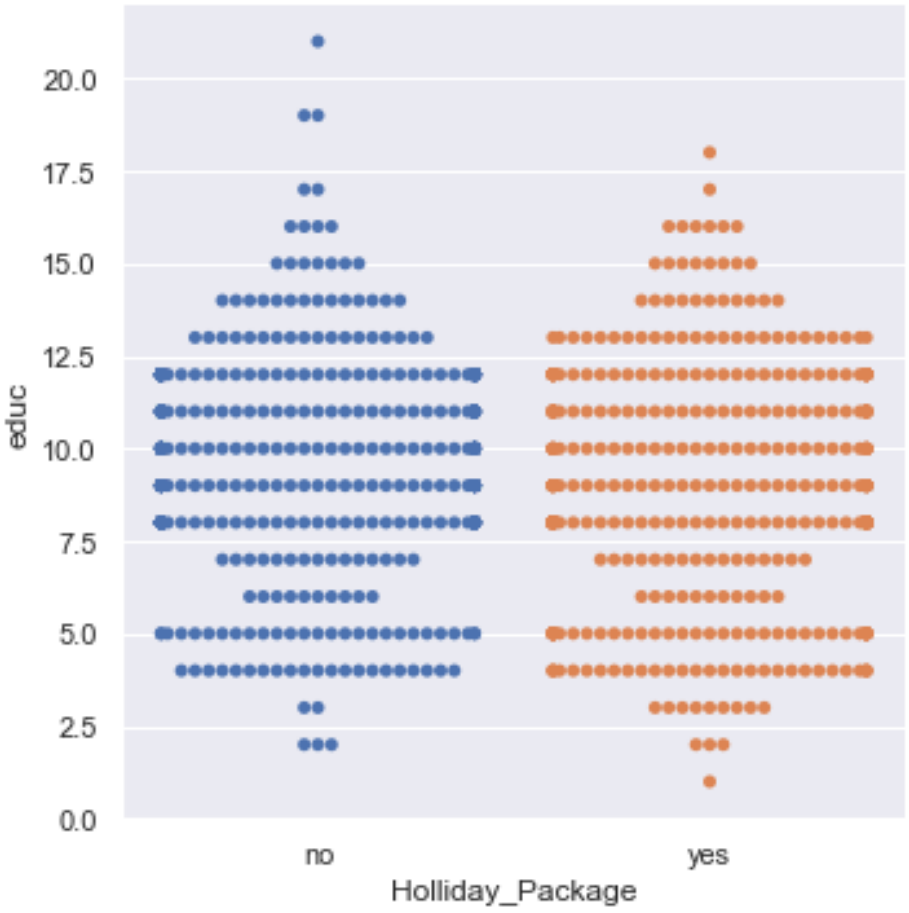


2.1.12 HOLLIDAY PACKAGE v/s SALARY

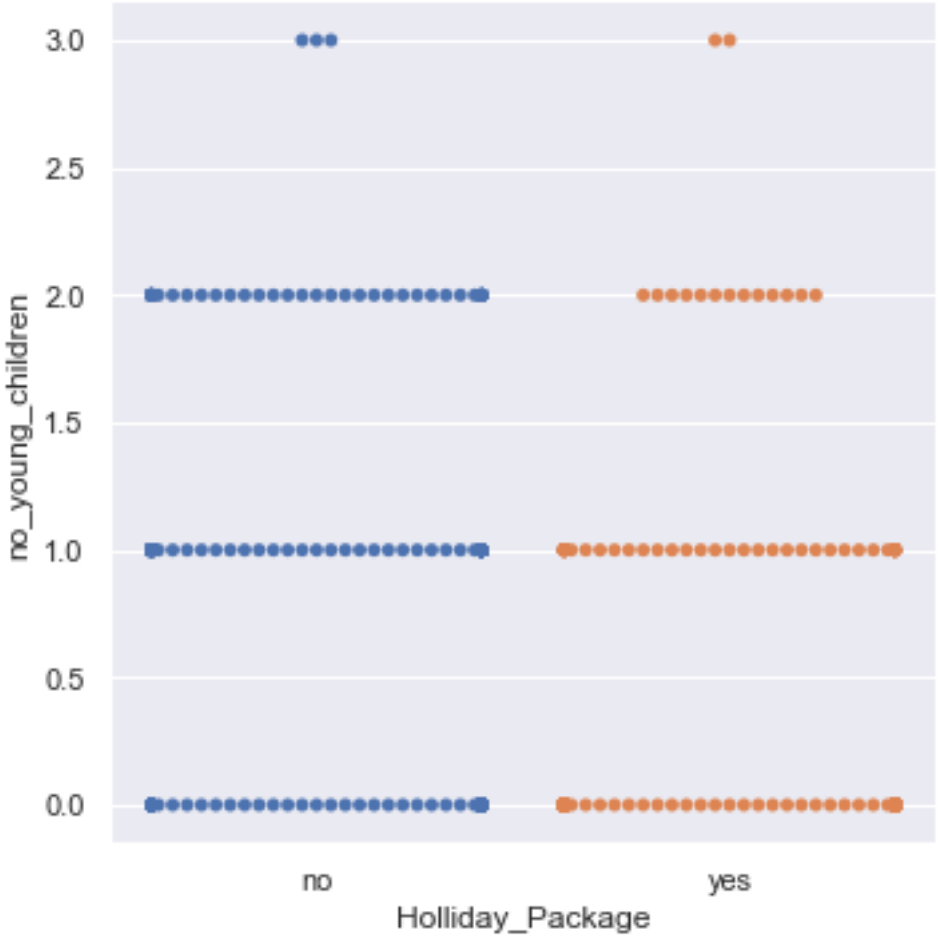
- ***We can see employee below salary 150000 have always opted for holiday package***

2.1.13 HOLLIDAY PACKAGE v/s AGE

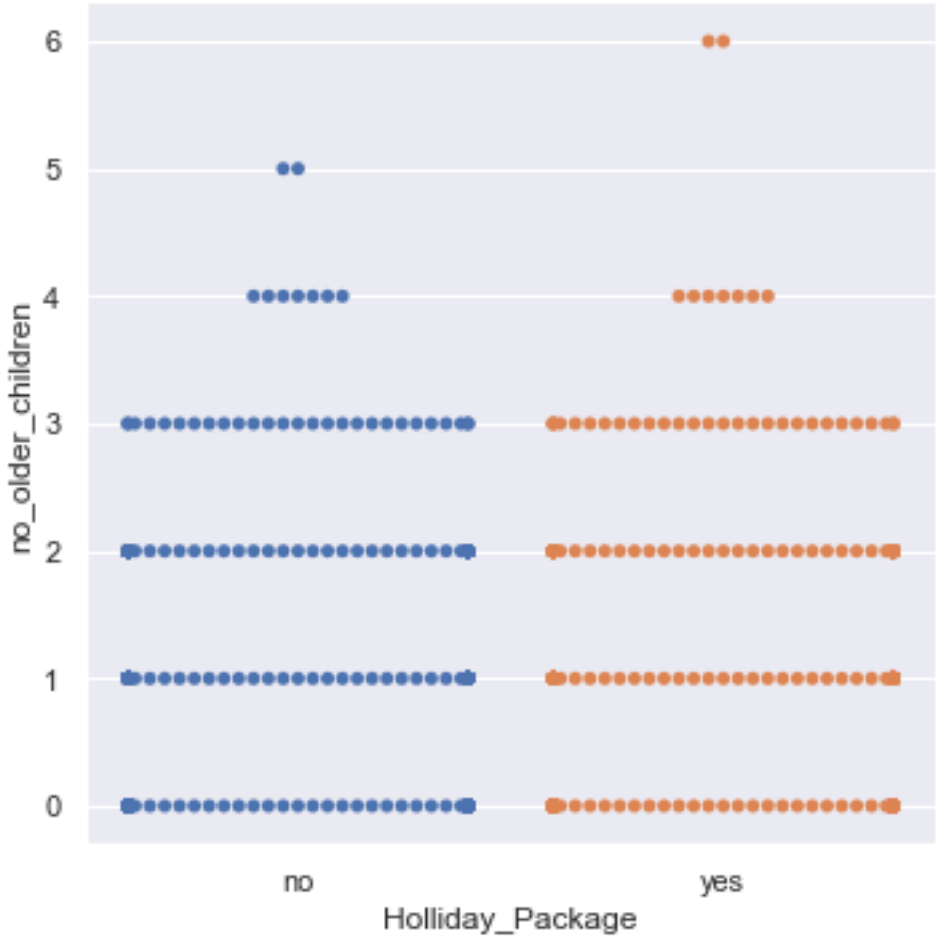
2.1.14 HOLLIDAY PACKAGE v/s EDUCATION



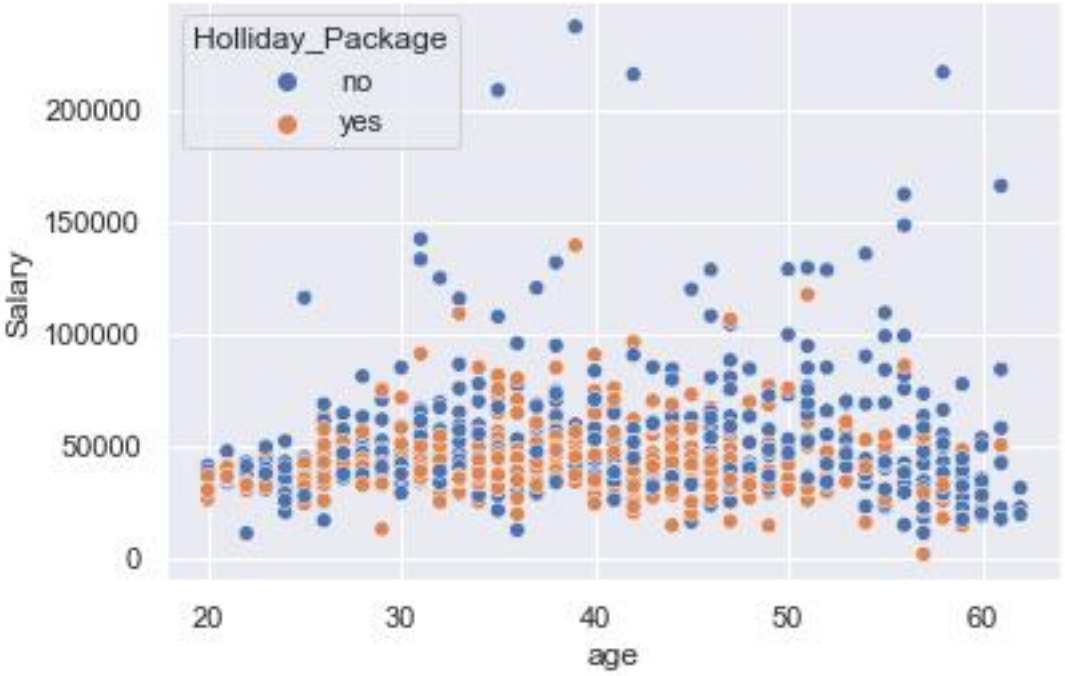
2.1.15 HOLLIDAY PACKAGE v/s YOUNG CHILDREN



2.1.16 HOLLIDAY PACKAGE v/s OLDER CHILDREN

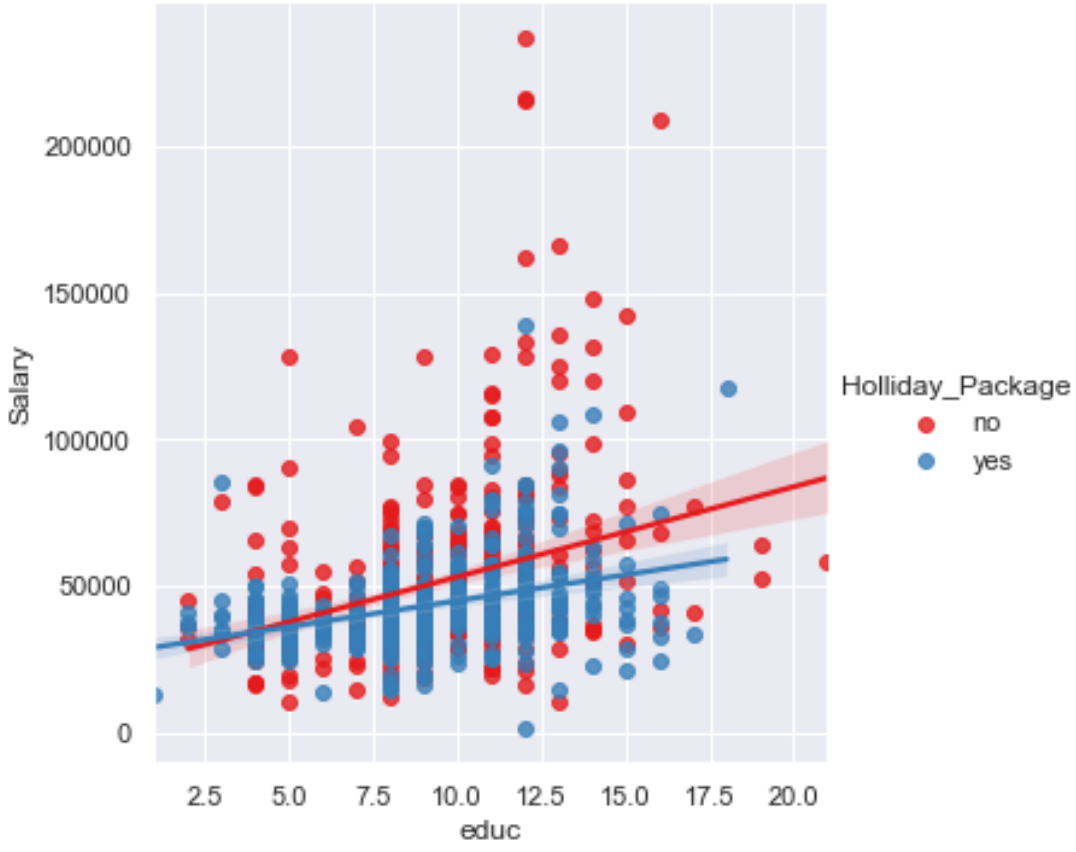


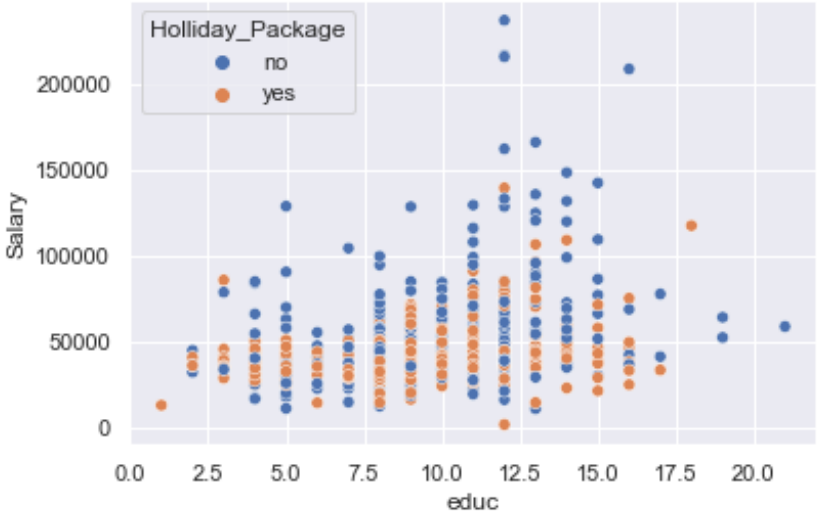
2.1.17 AGE VS SALARY w.r.t. HOLLIDAY PACKAGE



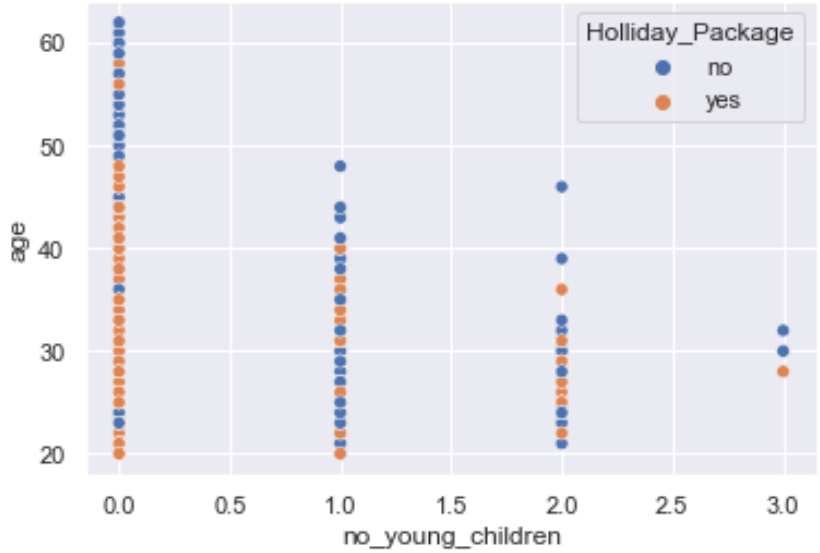
- ***Employee age over 50 to 60 have seems to be not taking the holiday package, whereas in the age 30 to 50 and salary less than 50000 people have opted more for holiday package.***

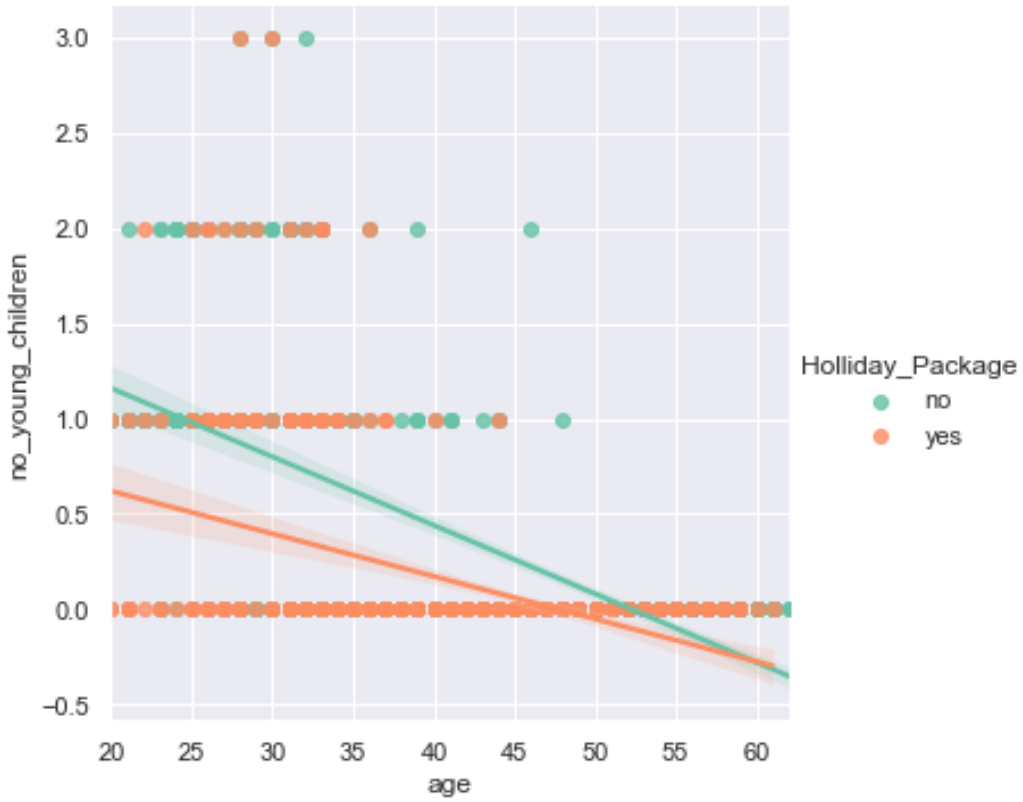
2.1.18 EDUC VS SALARY w.r.t HOLLIDAY PACKAGE



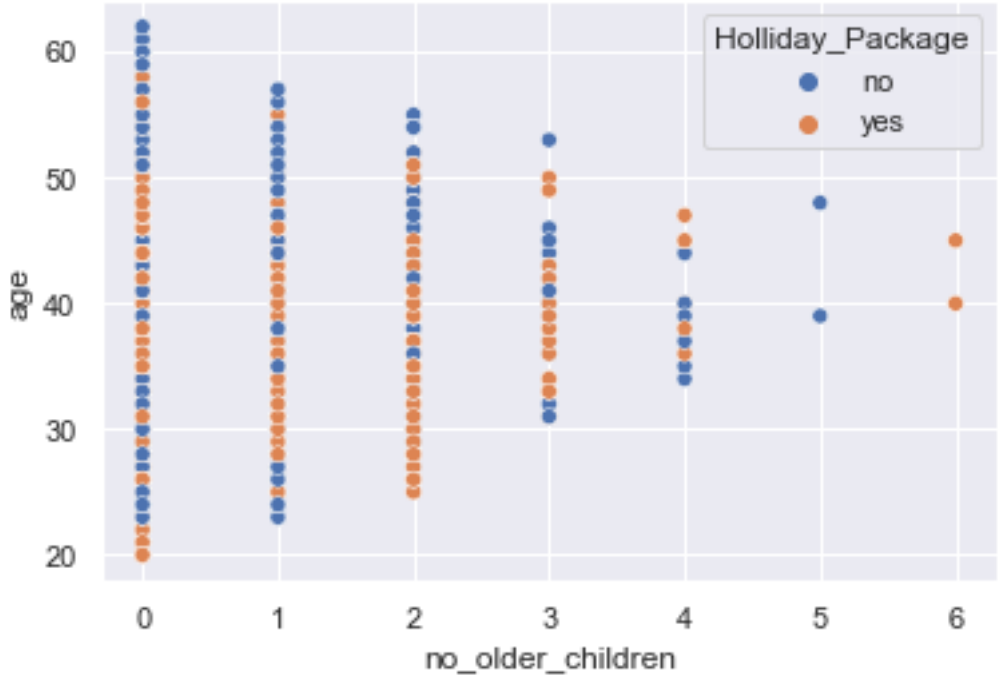


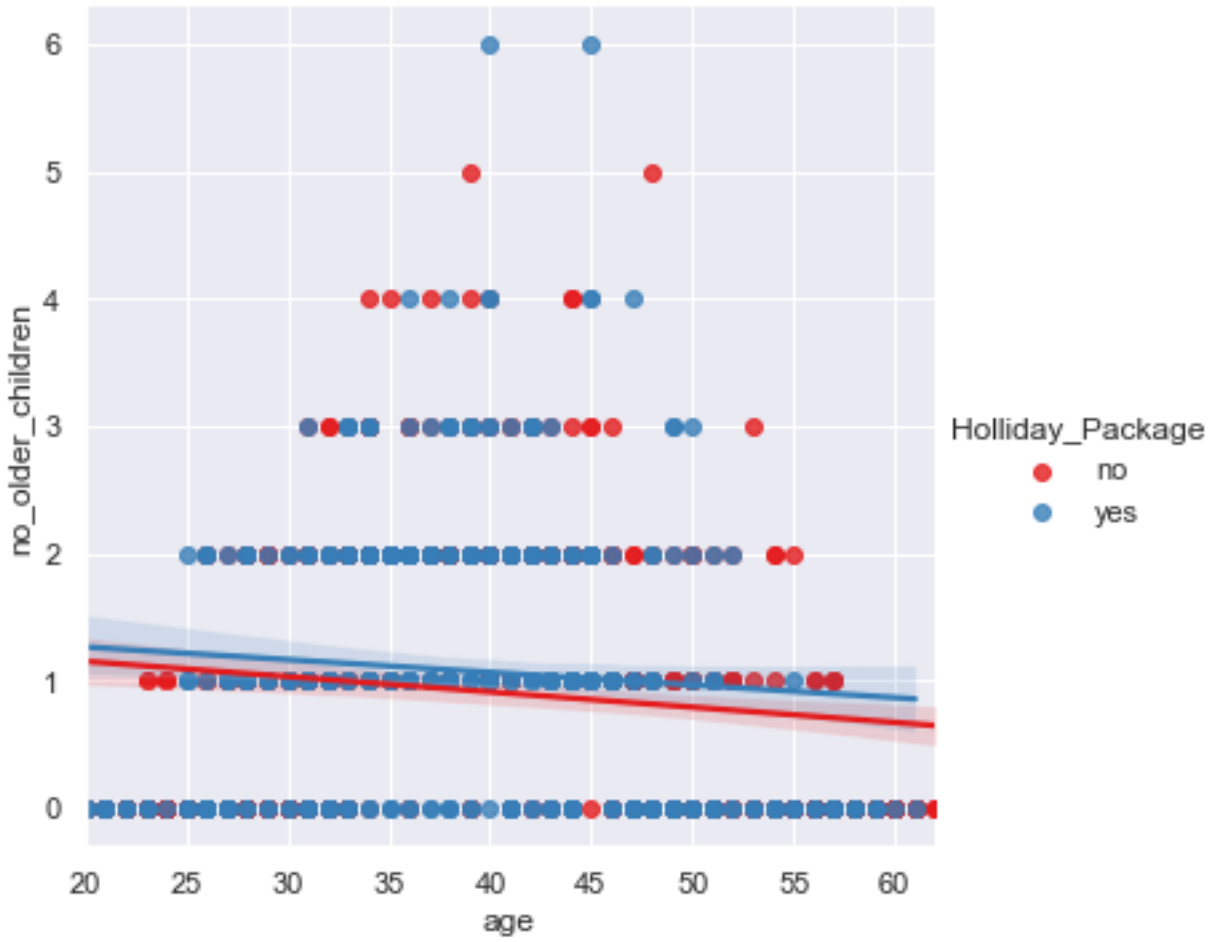
2.1.19 YOUNG CHILDREN VS AGE w.r.t HOLLIDAY PACKAGE





2.1.20 OLDER CHILDREN VS AGE w.r.t HOLLIDAY_PACKAGE

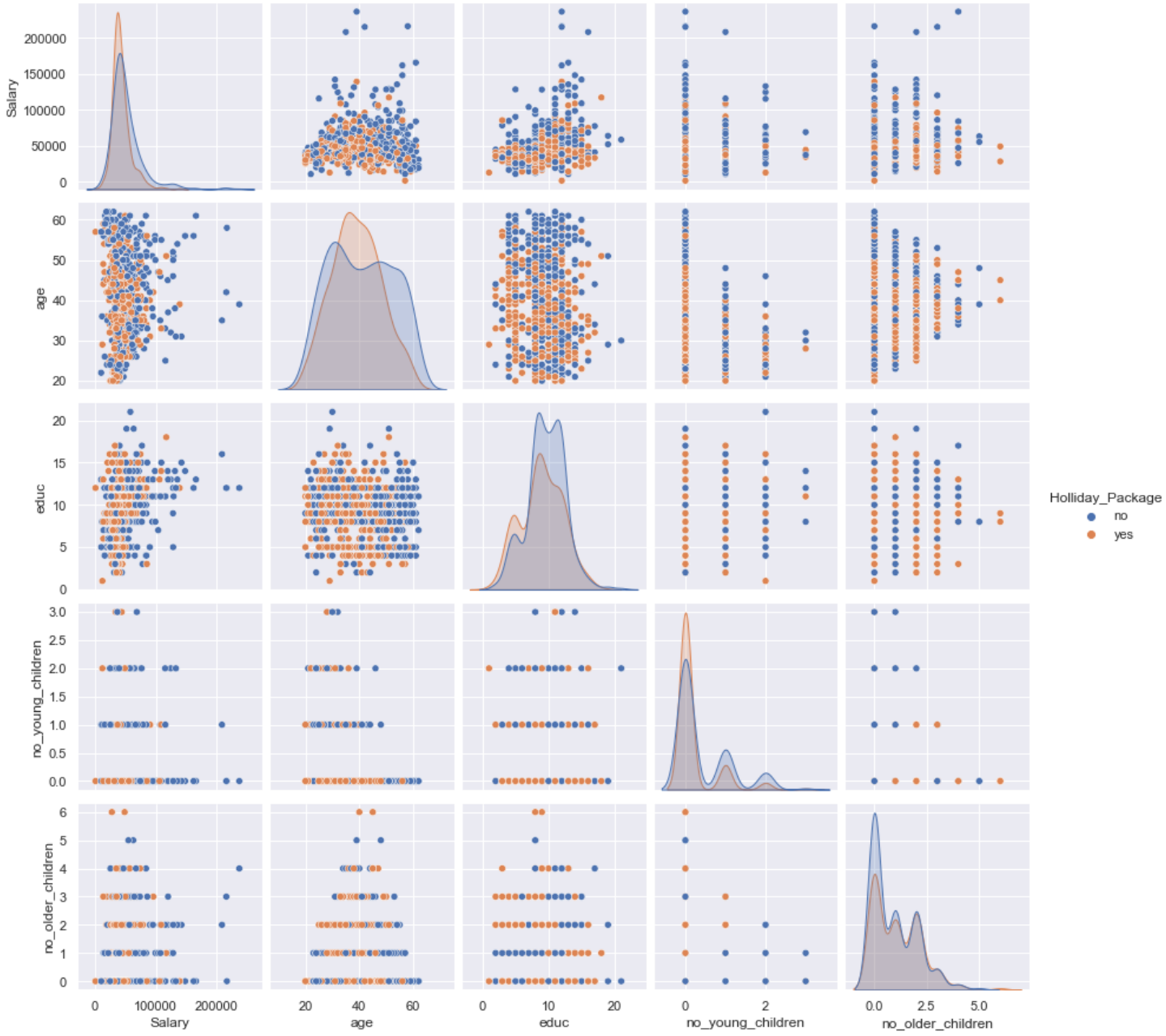




2.1.21 Bivariate Analysis

2.1.22 Pairplot

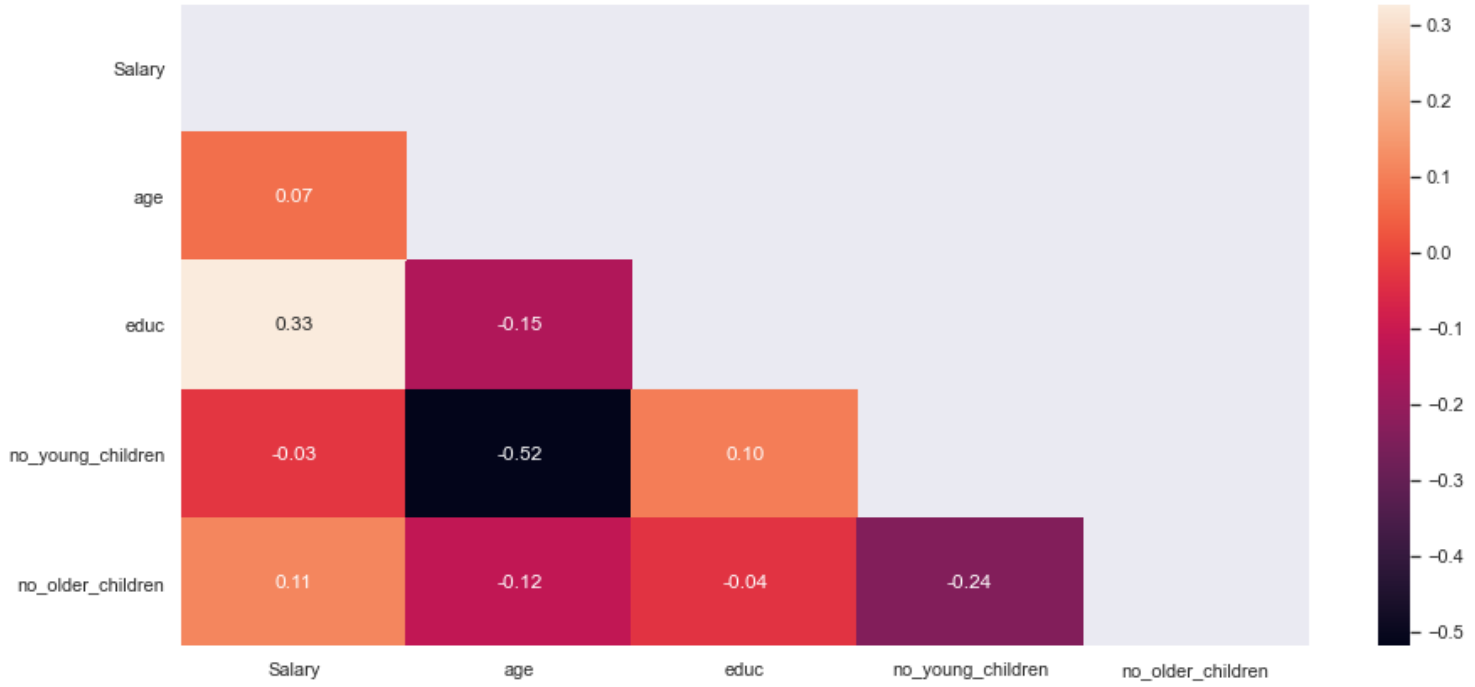
A pair plot plots the relationships between all numeric variables in a dataset. The diagonal below is the histogram for each variable and shows the distribution. From the below plot, we can observe if there are relationships between every two pair of variables.



- **We can observe age seems to be normally distributed while there is skewness present in other variables.**

2.1.23 Correlation Heat Map

The correlation coefficient shown in the table below shows the degree of correlation between the two variables represented in X axis and Y axis. It varies between -1 (maximum negative correlation) to +1 (maximum positive correlation).



- *No Multicollinearity present in the data*

2.2 Do not scale the data. Encode the data (having string values) for Modelling. Data Split: Split the data into train and test (70:30). Apply Logistic Regression and LDA (linear discriminant analysis).

2.2.1 Converting categorical to dummy variables in data

	Salary	age	educ	no_young_children	no_older_children	Holliday_Package_yes	foreign_yes
0	48412	30	8	1	1	0	0
1	37207	45	8	0	1	1	0
2	58022	46	9	0	0	0	0
3	66503	31	11	2	0	0	0
4	66734	44	12	0	2	0	0

```
Index(['Salary', 'age', 'educ', 'no_young_children', 'no_older_children'
,
       'Holliday_Package_yes', 'foreign_yes'],
      dtype='object')
```

2.2.2 Train/Test Split

Here we will split our Data into training and testing in the ration of 70/30

Observing the ratio of employees who chose the holliday_package to those who didn't

```
0    0.539344
1    0.460656
Name: Holliday_Package_yes, dtype: float64
```

- ***This split indicates that 46% of employees are interested in the holiday package***

2.2.3 Applying GridSearchCV for Logistic Regression

```
GridSearchCV(cv=10, estimator=LogisticRegression(max_iter=2000, random_s
tate=1),
             param_grid={'penalty': ['l1', 'l2', 'none'],
                         'solver': ['lbfgs', 'liblinear', 'newton-cg'],
                         'tol': [0.0001, 1e-06]})
Best Parameters:  {'penalty': 'l2', 'solver': 'liblinear', 'tol': 1e-06}


Best Estimator:  LogisticRegression(max_iter=2000, random_state=1, solve
r='liblinear', tol=1e-06)
```

2.2.4 Prediction on the training set

```
array([0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0,
       0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0,
       0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0,
       1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0,
       1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      dtype=uint8)
```

2.2.5 Getting the probabilities on the test set

	0	1
0	0.672537	0.327463
1	0.581248	0.418752
2	0.684005	0.315995
3	0.536270	0.463730
4	0.545159	0.454841

2.2.6 LDA Model

2.2.6.1Using the original data without dummies

	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
0	no	48412	30	8	1	1	no
1	yes	37207	45	8	0	1	no
2	no	58022	46	9	0	0	no
3	no	66503	31	11	2	0	no
4	no	66734	44	12	0	2	no

2.2.7 Converting Categorical features to codes

```
feature: Holliday_Package
[no, yes]
Categories (2, object): [no, yes]
[0 1]


feature: foreign
[no, yes]
Categories (2, object): [no, yes]
[0 1]
```

2.2.8 Splitting the Data into Training and Testing

2.2.9 Building LDA Model

```
LinearDiscriminantAnalysis()
```

2.2.10 Data Prediction

```
array([0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0,
       0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0,
       0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0,
       1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0,
       1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      dtype=int8)
```

2.2.11 Training Data Probability Prediction

	0	1
0	0.261849	0.738151
1	0.710383	0.289617
2	0.617657	0.382343
3	0.235165	0.764835
4	0.533171	0.466829

Testing Data Probability Prediction

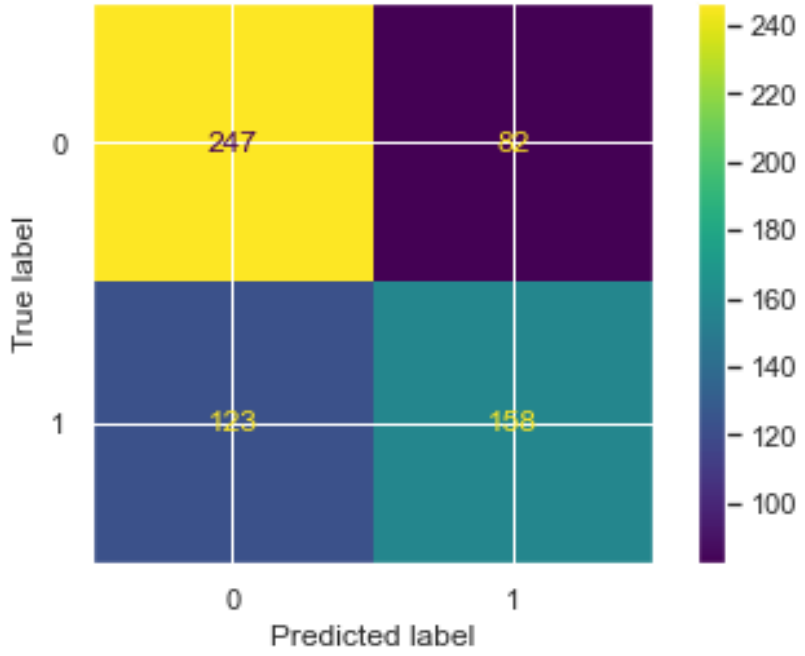
	0	1
0	0.708475	0.291525
1	0.533448	0.466552
2	0.717871	0.282129
3	0.504865	0.495135
4	0.555863	0.444137

2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model Final Model: Compare both the models and write inference which model is best/optimized.

2.3.1 Logistic Regression Metrics

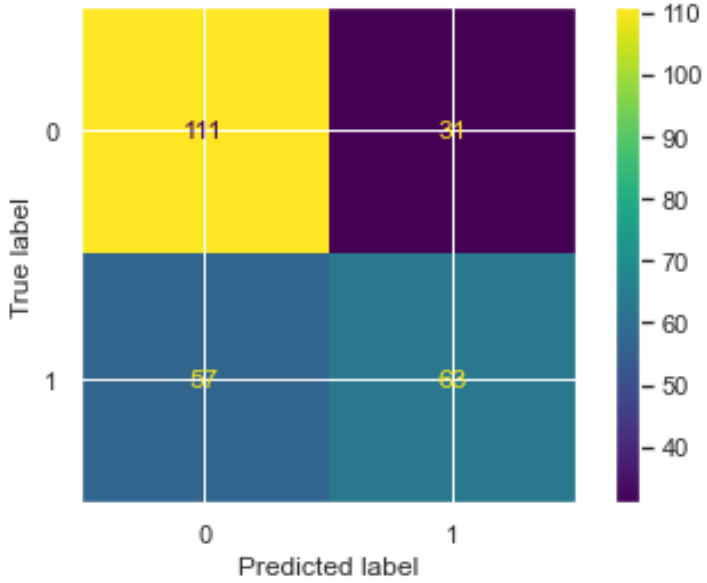
- **Confusion matrix and Classification Report on the training data**

	precision	recall	f1-score	support
0	0.67	0.75	0.71	329
1	0.66	0.56	0.61	281
accuracy			0.66	610
macro avg	0.66	0.66	0.66	610
weighted avg	0.66	0.66	0.66	610



2.3.2 Confusion matrix and Classification Report on the test data

	precision	recall	f1-score	support
0	0.66	0.78	0.72	142
1	0.67	0.53	0.59	120
accuracy			0.66	262
macro avg	0.67	0.65	0.65	262
weighted avg	0.67	0.66	0.66	262

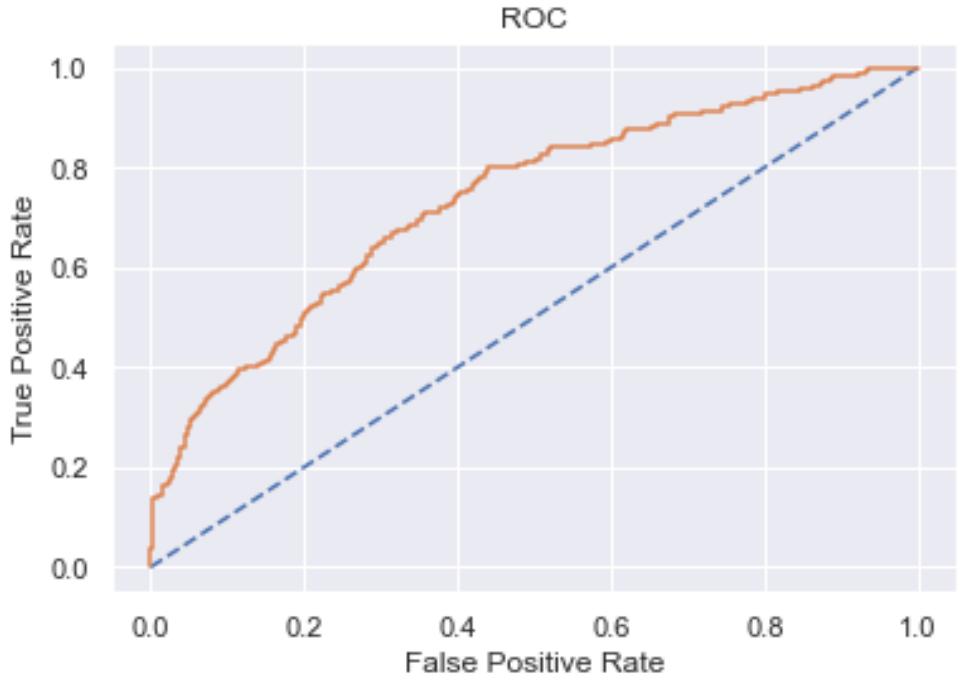


2.3.3 Accuracy - Training Data

The Accuracy of the Training Data is: *0.6639344262295082*

2.3.4 AUC and ROC for the training data

The AUC score is: 0.734

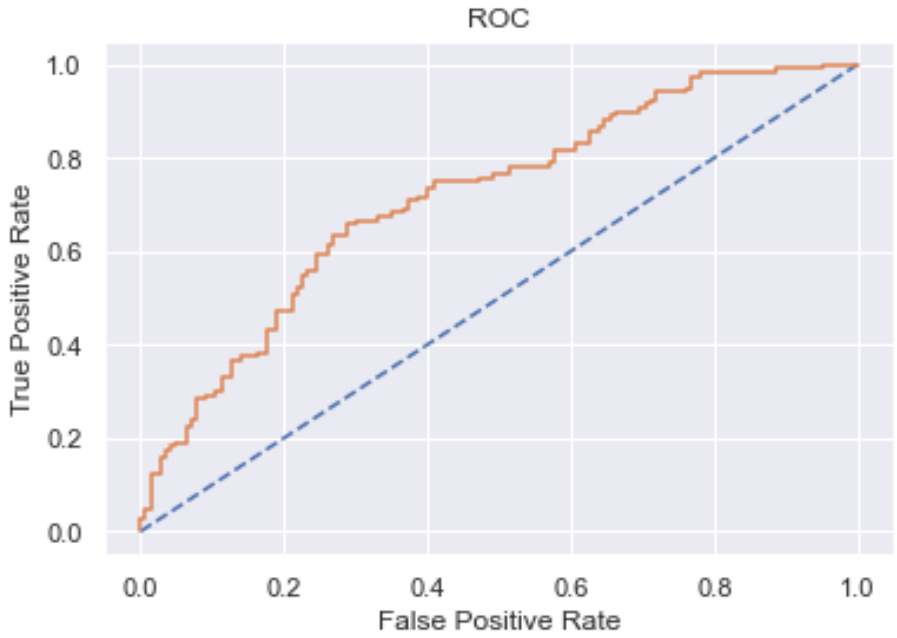


2.3.5 Accuracy - Test Data

The Accuracy of the Testing Data is: 0.6641221374045801

2.3.6 AUC and ROC for the testing data

The AUC Score is: 0.717



2.3.7 Logistic Regression Training Metrics

```
lr_train_precision  0.66
lr_train_recall  0.56
lr_train_f1  0.61
```

2.3.8 Logistic Regression Testing Metrics

```
lr_test_precision  0.67
lr_test_recall  0.52
lr_test_f1  0.59
```

2.3.9 Linear Discriminanat Analysis Metrics

2.3.10 Accuarcy - Train Data

```
The Accuracy of the Training Data is:  0.6639344262295082
```

2.3.11 Classification Report on the training data

	precision	recall	f1-score	support
0	0.67	0.74	0.70	329
1	0.65	0.58	0.61	281
accuracy			0.66	610
macro avg	0.66	0.66	0.66	610
weighted avg	0.66	0.66	0.66	610

2.3.12 Confusion matrix on the training data

```
array([[243,  86],
       [119, 162]], dtype=int64)
```

2.3.13 Accuracy - Testing Data

```
The Accuracy of the Testing Data is:  0.6412213740458015
```

2.3.14 Classification Report on the Testing data

	precision	recall	f1-score	support
0	0.64	0.77	0.70	142
1	0.64	0.49	0.56	120
accuracy			0.64	262
macro avg	0.64	0.63	0.63	262
weighted avg	0.64	0.64	0.63	262

2.3.15 Confusion Matrix on the Testing data

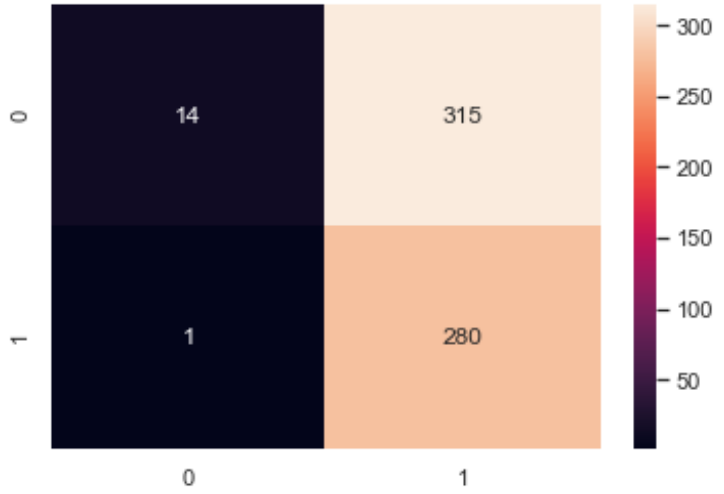
```
array([[109,  33],
       [ 61,  59]], dtype=int64)
```

2.3.16 CHANGING THE CUT-OFF VALUE TO CHECK OPTIMAL VALUE THAT GIVES BETTER ACCURACY AND F1-SCORE

0.1

Accuracy Score 0.482
F1 Score 0.6393

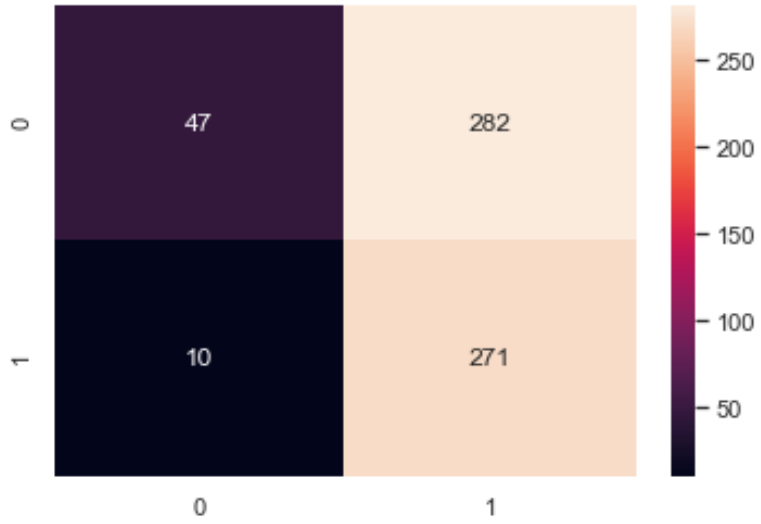
Confusion Matrix



0.2

Accuracy Score 0.5213
F1 Score 0.6499

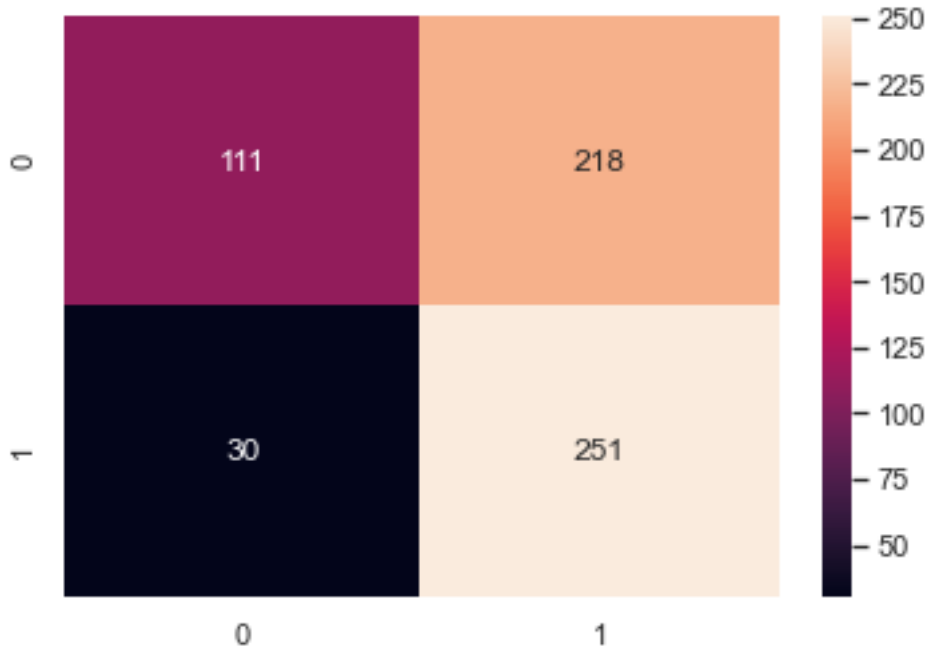
Confusion Matrix



0.3

Accuracy Score 0.5934
F1 Score 0.6693

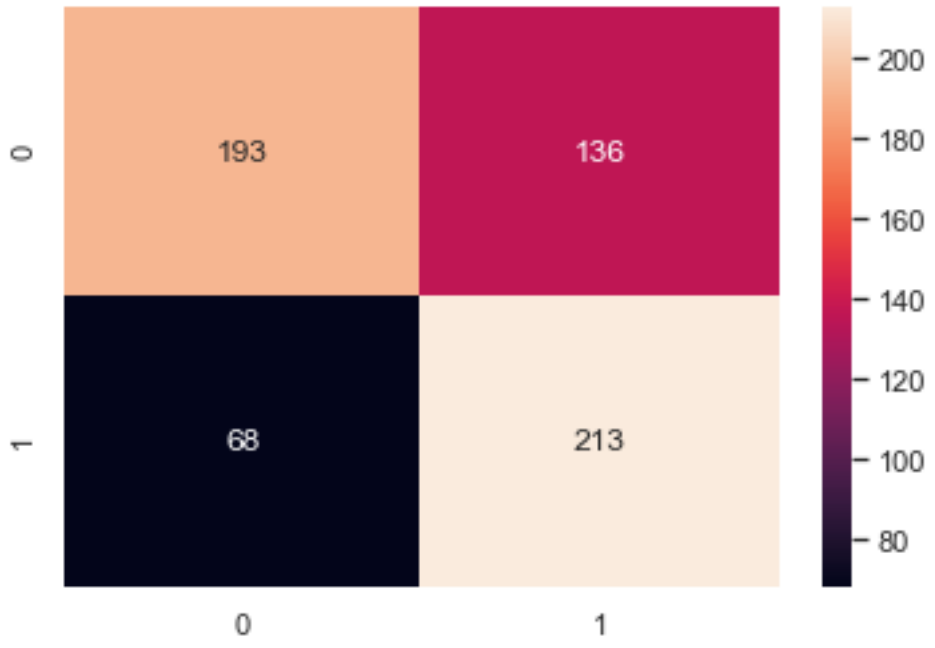
Confusion Matrix



0.4

Accuracy Score 0.6656
F1 Score 0.6762

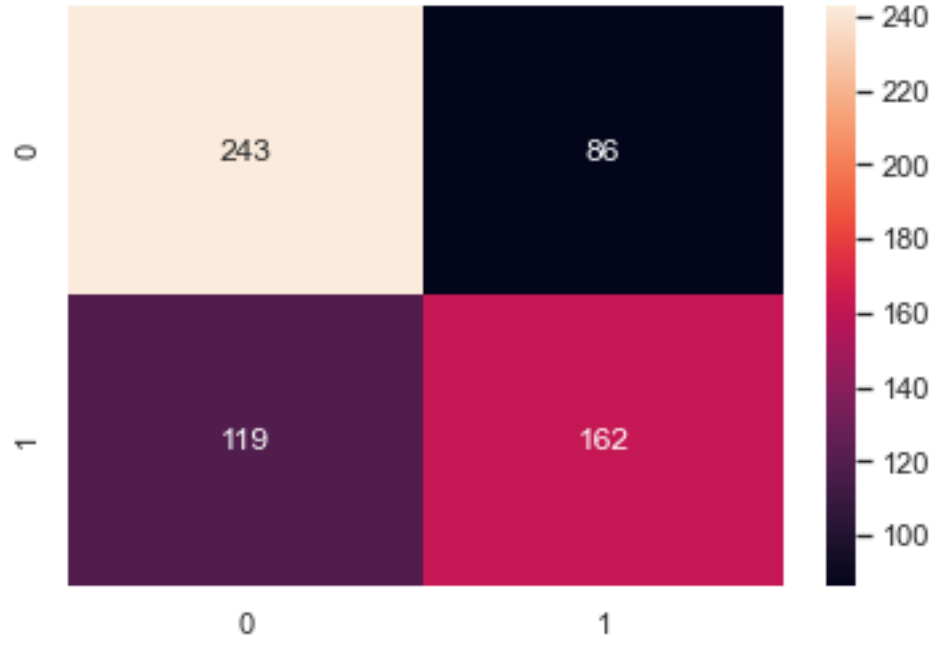
Confusion Matrix



0.5

Accuracy Score 0.6639
F1 Score 0.6125

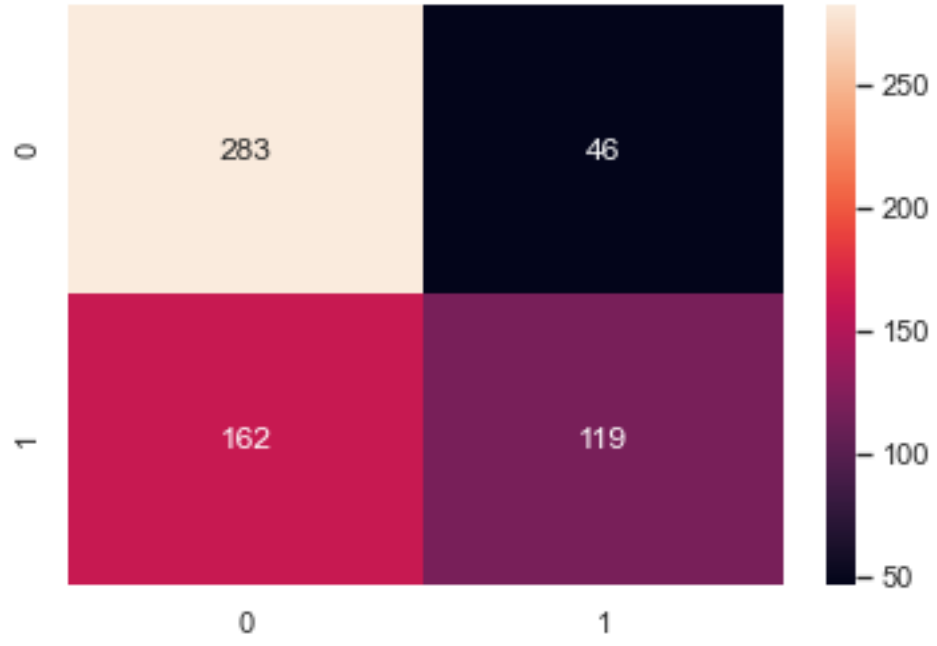
Confusion Matrix



0.6

Accuracy Score 0.659
F1 Score 0.5336

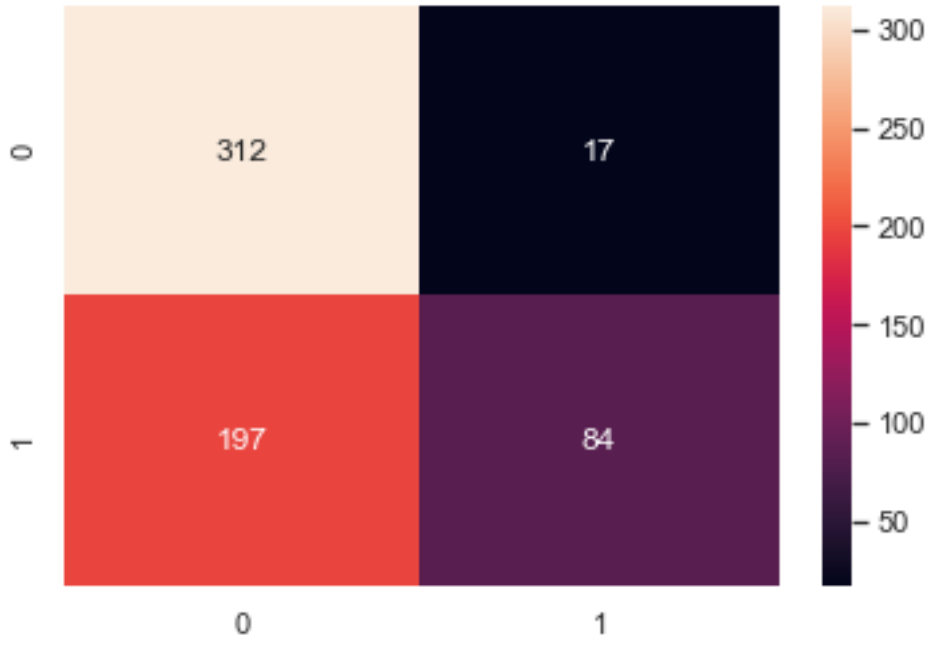
Confusion Matrix



0.7

Accuracy Score 0.6492
F1 Score 0.4398

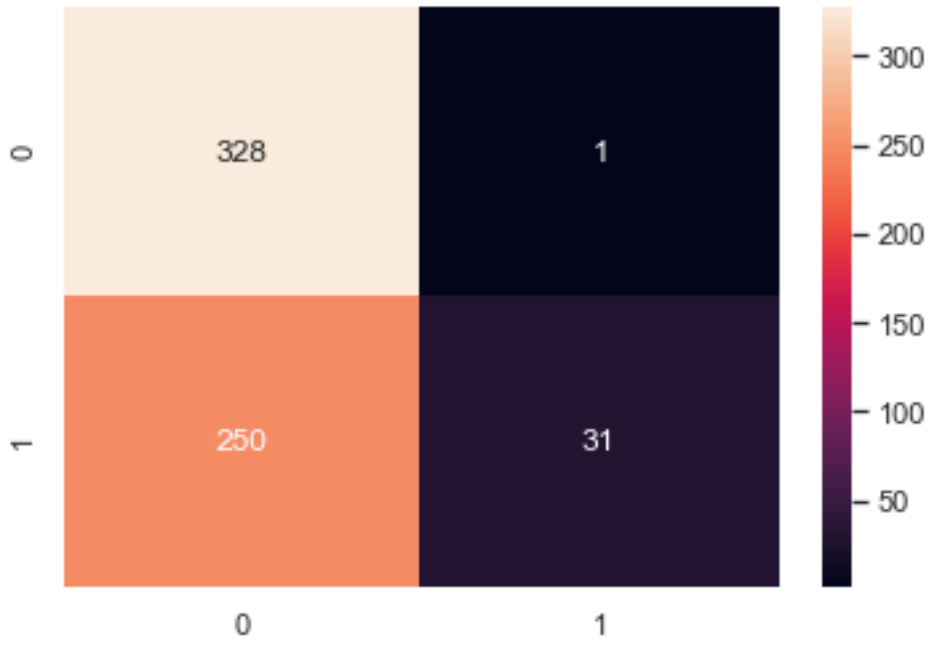
Confusion Matrix



0.8

Accuracy Score 0.5885
F1 Score 0.1981

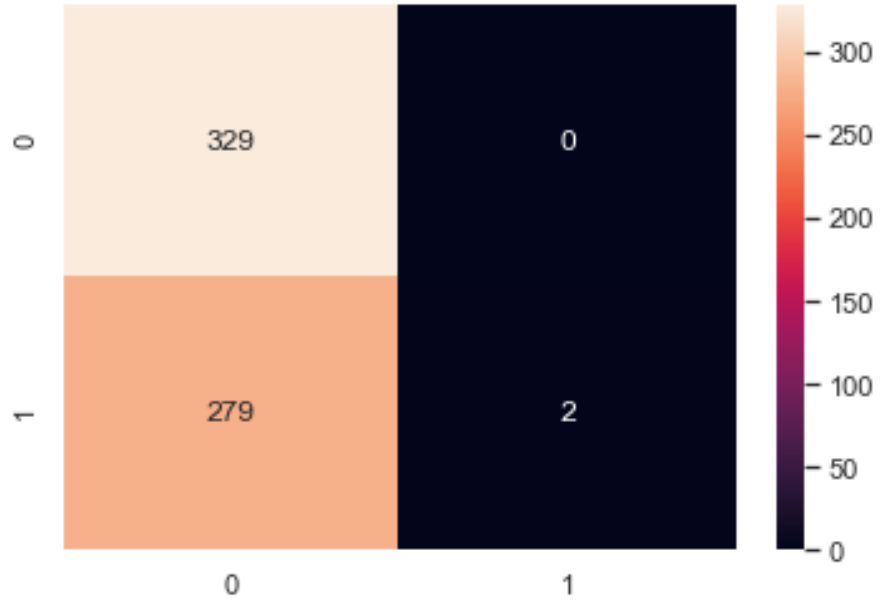
Confusion Matrix



0.9

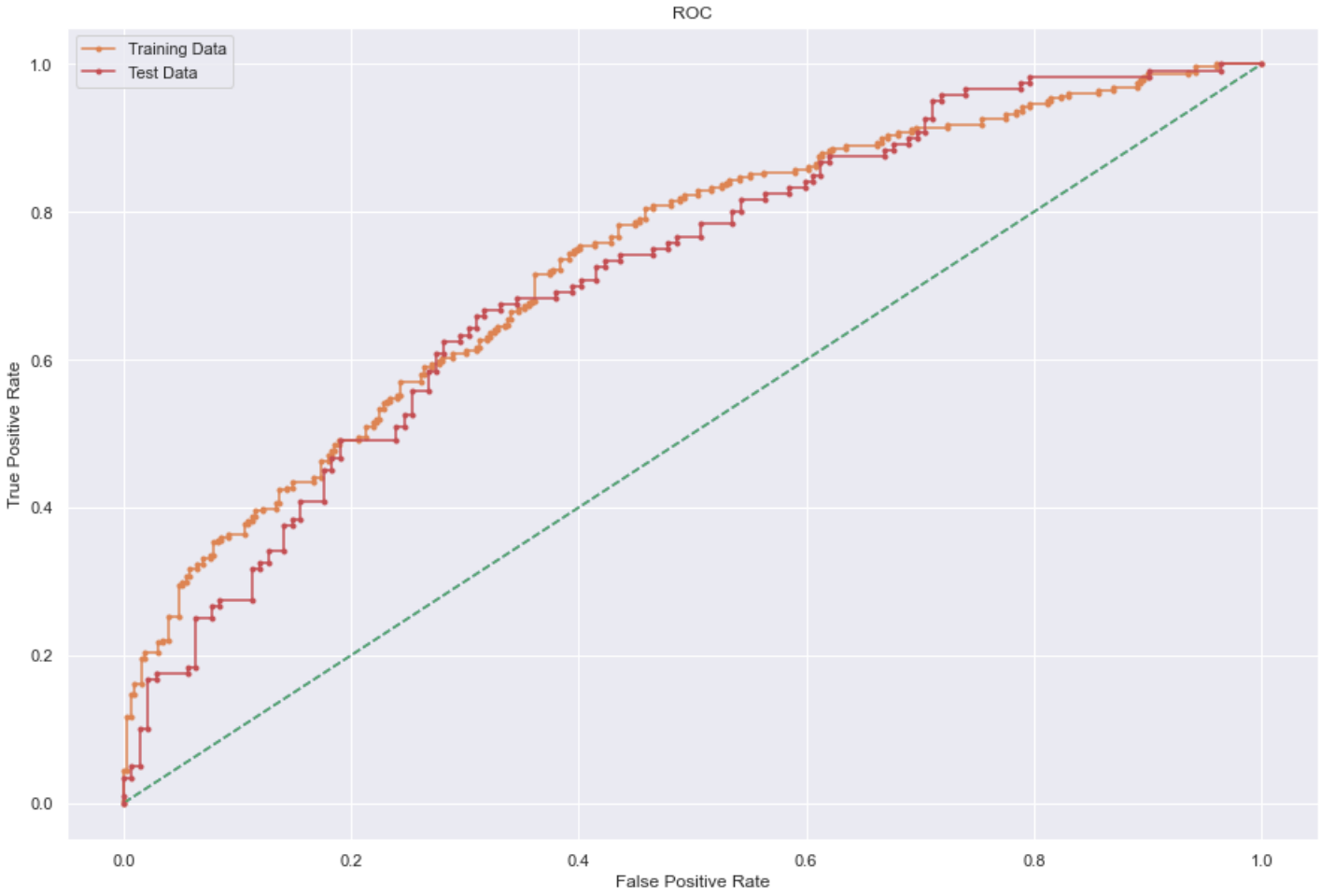
Accuracy Score 0.5426
F1 Score 0.0141

Confusion Matrix



2.3.17 AUC and ROC for Training and Testing Data

AUC for the Training Data: 0.733
AUC for the Test Data: 0.714



2.3.18 LDA Training Metrics

```
lda_train_precision  0.65
lda_train_recall  0.58
lda_train_f1  0.61
```

2.3.19 LDA Testing Metrics

```
lda_test_precision  0.64
lda_test_recall  0.49
lda_test_f1  0.56
```

2.3.20 Combined Metrics for Both LDA and LR Model

	LR Train	LR Test	LDA Train	LDA Test
Accuracy	0.66	0.66	0.66	0.64
AUC	0.73	0.72	0.73	0.71
Recall	0.56	0.52	0.58	0.49
Precision	0.66	0.67	0.65	0.64
F1 Score	0.61	0.59	0.61	0.56

2.3.21 Observations

- ***Both Models are performing well as the metrics for both Training and Testing Data are almost Similar***
- ***There is hardly much difference in both models but the Recall, Precision and F1-Score Variation is less for LR as compared to LDA model.***
- ***Therefore LR Model would be a suitable choice. Moreover LDA mostly asume normal distribution but here thats not the case therefore Logistic Regression would be better.***
- ***If we were to scale the data then LDA would have been the better option.***

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

- *We had a business problem where we need predict whether an employee would opt for a holiday package or not, for this problem we had done predictions both logistic regression and linear discriminant analysis. Since both are results are same.*

- *The EDA analysis clearly indicates certain criteria where we could find people aged above 50 are not interested much in holiday packages.*
  - *So this is one of the we find aged people not opting for holiday packages.*
  - *People ranging from the age 30 to 50 generally opt for holiday packages.*
  - *Employee age over 50 to 60 have seems to be not taking the holiday package, whereas in the age 30 to 50 and salary less than 50000 people have opted more for holiday package.*

- *The important factors deciding the predictions are salary, age and educ.*

- **Recommendations**

1. **To improve holiday packages over the age above 50 we can provide religious destination places as peaceful locations are preferred by people of that age.**
2. **For people earning more than 150000 we can provide vacation holiday packages.**
3. **For employee having more than 3 number of older children we can provide packages in holiday vacation places.**

THE END