Linear Regression – Diamonds Prices

Given Data Base:

Raw table (CSV Format) that contains an data on Diamonds. The variables are as following:

- Cut
- Color
- Clarity
- Depth
- Table
- X
- Y
- Z
- Price

The task:

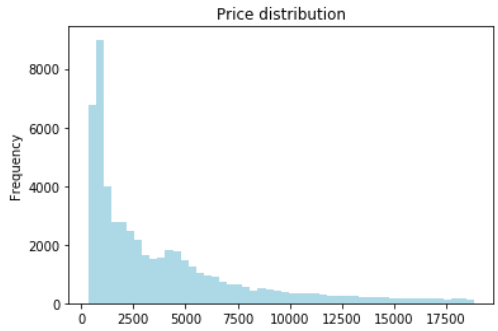
Predict Prices of diamonds based on its characteristics

.

Part I – 2D Linear Regression Model

Explanatory analysis:

Price analysis –

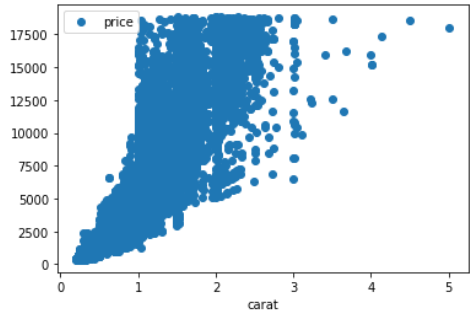


Aggregations:

```
max       18823.000000
mean       3932.799722
median     2401.000000
min         326.000000
```

```
df['price'].agg(['max','mean','median','min'])
df['price'].plot(kind='hist', bins=50, color='lightblue', title='Price distribution')
```

Correlation between Carat & Price:



We can see that the correlation between Carat and Price is positive.

```
df.plot(x='carat',y='price', style='o')
```

2D Linear Regression:

Defining variables & splitting the data -

Independent Variable (Input) - Carat

Dependent Variable (Output) – Price

```
X = df['carat'].values.reshape(-1,1) # Independed Variable - Input
y = df['price'].values.reshape(-1,1) # Depended Variable – Output

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

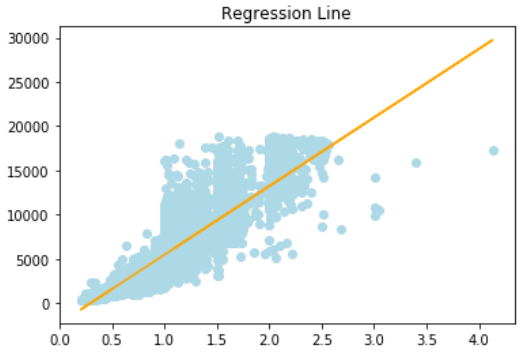
Fitting Regressor & Predicting -

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

y_pred = regressor.predict(X_test)
```

Results -

```
R-Squared:  0.8515758113126248
Coefficient:  7745.256582433882
Intercept:  -2248.460057551038
```

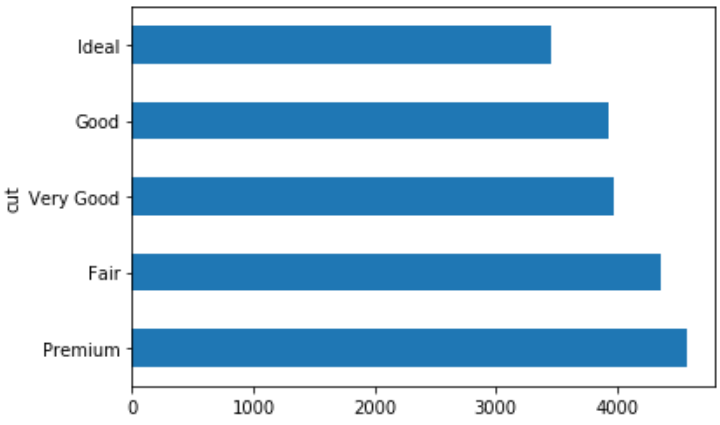



```
print('R-Squared: ', metrics.r2_score(y_test,y_pred))
print('Coefficient: ',regressor.coef_[0][0])
print('Intercept: ',regressor.intercept_[0])
plt.scatter(X_test,y_test, color='lightblue')
plt.plot(X_test,y_pred, color='orange')
plt.title('Regression Line')
plt.show()
```

Part II – Multiple Linear Regression Model

Converting 'Cut' into Ordinal Variable –

We can see that the price is affected by the cut level. Therefore, we converted Cut from Categorial into Ordinal variable (level 1 to 5).




```
df.groupby('cut').mean()['price'].sort_values(ascending=False).plot(kind='barh')

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

Creating Additional Variable – Size

With multiplying X, Y and Z measures, we can create a new variable; Size.

```
df['size'] = df['x']*df['y']*df['z']
```

Correlation Table –

	carat	cut	depth	x	y	z	size	price
carat	1.000000	-0.114426	0.028224	0.975094	0.951722	0.953387	0.976308	0.921591
cut	-0.114426	1.000000	-0.169916	-0.105361	-0.105319	-0.126726	-0.101119	-0.049421
depth	0.028224	-0.169916	1.000000	-0.025289	-0.029341	0.094924	0.009157	-0.010647
x	0.975094	-0.105361	-0.025289	1.000000	0.974701	0.970772	0.956564	0.884435
y	0.951722	-0.105319	-0.029341	0.974701	1.000000	0.952006	0.975143	0.865421
z	0.953387	-0.126726	0.094924	0.970772	0.952006	1.000000	0.950065	0.861249
size	0.976308	-0.101119	0.009157	0.956564	0.975143	0.950065	1.000000	0.902385
price	0.921591	-0.049421	-0.010647	0.884435	0.865421	0.861249	0.902385	1.000000

```
df[['carat','cut','depth','x','y','z','size','price']].corr()
```

Multiple Variables Linear Regression:

Defining variables & splitting the data -

Independent Variable (Input) – Carat, Cut, Size, X, Y, Z

Dependent Variable (Output) – Price

```
X = df[['cut','carat','size','x','y','z']].values
y = df['price'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

Coefficient Table:

	0
cut	170.735240
carat	8418.080310
size	13.083272
x	-154.892973
y	-601.625678
z	-701.238426

```
v_def = df[['cut','carat','size','x','y','z']]
coeff_df = pd.DataFrame(regressor.coef_,v_def.columns)
coeff_df
```

Fitting Regressor & Predicting -

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

y_pred = regressor.predict(X_test)
```

Results -

```
R-Squared:  0. 0.8607280804681006
```

```
print('R-Squared: ', metrics.r2_score(y_test,y_pred))
```