Business Problem 1:

Prepare a prediction model for profit of 50_startups data. Do transformations for getting better predictions of profit and make a table containing R^2 value for each prepared model.

- R and D Spend -- Research and develop spend in the past few years
- Administration -- spend on administration in the past few years
- Marketing Spend -- spend on Marketing in the past few years
- State -- states from which data is collected
- Profit -- profit of each state in the past few years)

Solution:

In the given business problem, the independent variables are R and D Spend (**X_1**), Administration spend (**X_2**), Marketing Spend (**X_3**) and the output or dependent variable is Profit **(Y)**. Based on input variables the profit is depended.

So the line equation of the above can be,

$\mathbf{Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3}$

Goal: We need to come up with a best fit model which predicts the output variable.

Codes:

#........load data........

```
import pandas as pd
startup = pd.read_csv("E:\\Data\\Assignments\\i made\\MLR\\50_Startups.csv")
startup.columns
print(startup.shape)
```

Step 1: EDA

1 Business Moments

Mean

R_and_D_Spend	73721.6156
Administration	121344.6396
Marketing_Spend	211025.0978

Profit 112012.6392

Median

R_and_D_Spend	73051.080
Administration	122699.795
Marketing_Spend	212716.240
Profit	107978.190

Since mean and median values of the data are nearby values, we can trust the data.

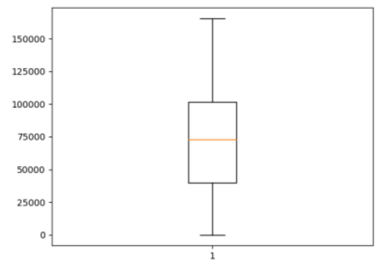
Codes:

```
startup.mean()
startup.median()
```

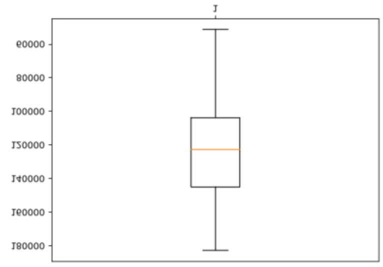
Outlier Presence:

From plotting boxplots we can see presence of outliers.

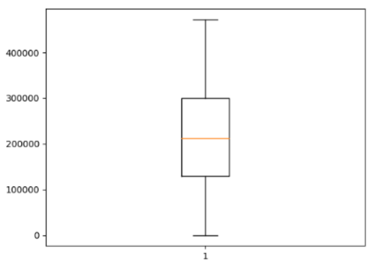
1 R_and_D_Spend



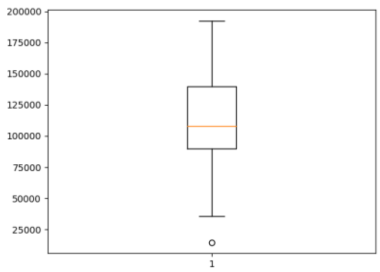
2 Administration



3 Marketing_Spend



4 Profit



From above plots we can observe that only profit variable is having the outlier, as profit is an output variable, there is no need to remove the outliers.

Codes:

```
import matplotlib.pyplot as plt
plt.boxplot(startup["R_and_D_Spend"])
```

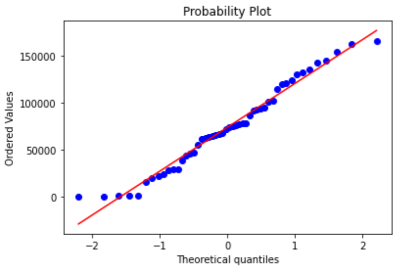
plt.boxplot(startup["Administration"])

plt.boxplot(startup["Marketing_Spend"])

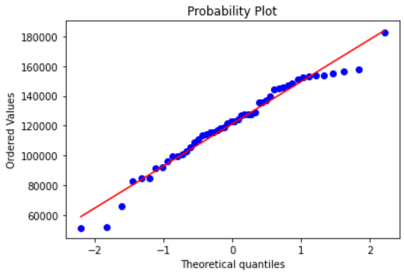
plt.boxplot(startup["Profit"])

2 To check whether the data is normally distributed or not

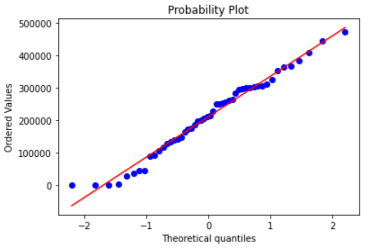
1 R_and_D_Spend



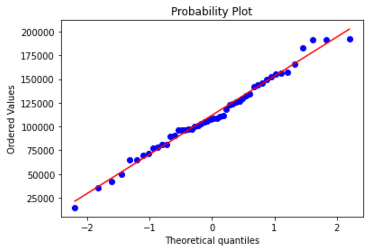
2 Administration



3 Marketing_Spend



4 Profit



Since the maximum number of data points are falling on the straight line, hence we can conclude that the data follows a normal distribution.

Codes:

import pylab

import scipy.stats as st

st.probplot(startup['R_and_D_Spend'], dist="norm",plot=pylab)

st.probplot(startup['Administration'], dist="norm",plot=pylab)

st.probplot(startup['Marketing_Spend'], dist="norm",plot=pylab)

st.probplot(startup['Profit'], dist="norm",plot=pylab)

Data scaling:

To make the data scale free and unit less we need to do normalization or standardisation.

In this business problem we have given a Nominal Data column "State", before normalizing we need to remove that column for analysis. For that we need to create a new Data Frame. Then removing the state column. Later we normalize the data, but here as all the variables are in same unit there is no need to Normalize or Standardize.

Codes:

Removing State column since it is Categorical

Before Removing we need to Crete A Data Frame

```
df = pd.DataFrame(startup)

print(df.shape)


df_drop_state = df.drop(["State"], axis=1 )

print(df_drop_state.shape)
```

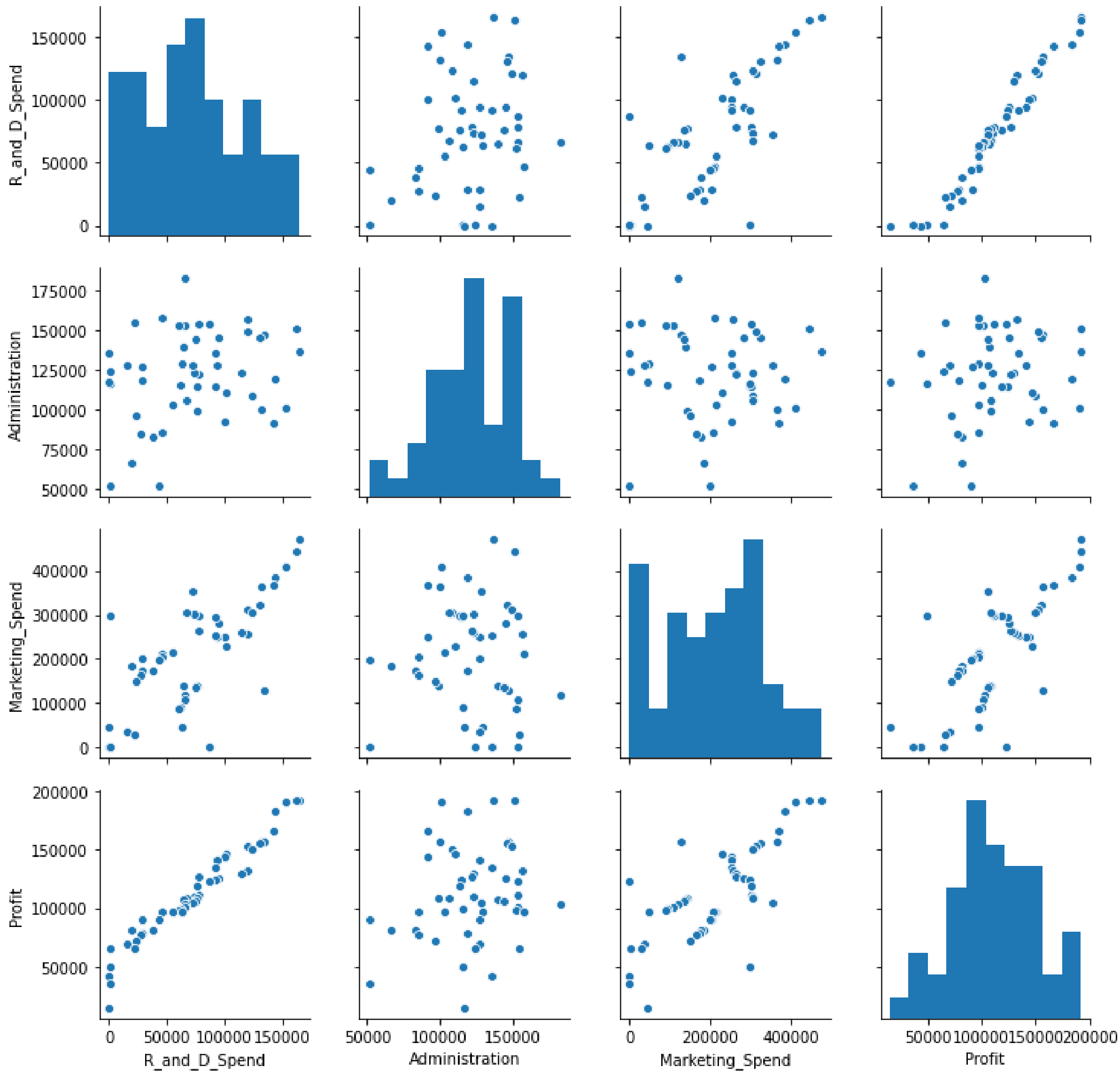
……END OF EDA……

Checking the existence of Collinearity between input variables.

```
import seaborn as sns


sns.pairplot(df_drop_state)

df_drop_state.corr()
```



By observing the correlation coefficients, we can say that there is no collinearity between input and input, but from above plot there exists a collinearity between R_and_D_Spend and Marketing_Spend.

But the correlation coefficient between R_and_D_Spend and Marketing_Spend is 0.725, which indicates a moderate relationship. So this collinearity can be ignored.

Codes:

import seaborn as sns

sns.pairplot(df_drop_state)

Creating dummy column:

As the given data is having a categorical column, we need to create dummies for that column

Codes:

dummy = pd.get_dummies(startup["State"])

Combining with original data:

Data_with_Dummy = pd.concat([df_drop_state, dummy], axis=1)

print(Data_with_Dummy.shape)

<u>Model building:</u>

Codes:

import statsmodels.formula.api as smf # for regression model

ml1=smf.ols('Profit~R_and_D_Spend+Administration+Marketing_Spend',data=Data_with_Dummy).fi
t()

ml1.params

ml1.summary()

df_drop_state.corr()

Summary of the result:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Profit   R-squared:                       0.951
Model:                            OLS   Adj. R-squared:                  0.948
Method:                 Least Squares   F-statistic:                     296.0
Date:                Sun, 26 Apr 2020   Prob (F-statistic):           4.53e-30
Time:                        18:12:57   Log-Likelihood:                -525.39
No. Observations:                  50   AIC:                             1059.
Df Residuals:                      46   BIC:                             1066.
Df Model:                           3
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
Intercept        5.012e+04   6572.353      7.626      0.000    3.69e+04    6.34e+04
R_and_D_Spend       0.8057      0.045     17.846      0.000       0.715       0.897
Administration     -0.0268      0.051     -0.526      0.602      -0.130       0.076
Marketing_Spend     0.0272      0.016      1.655      0.105      -0.006       0.060
==============================================================================
Omnibus:                       14.838   Durbin-Watson:                   1.282
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               21.442
Skew:                          -0.949   Prob(JB):                     2.21e-05
Kurtosis:                       5.586   Cond. No.                     1.40e+06
==============================================================================
```

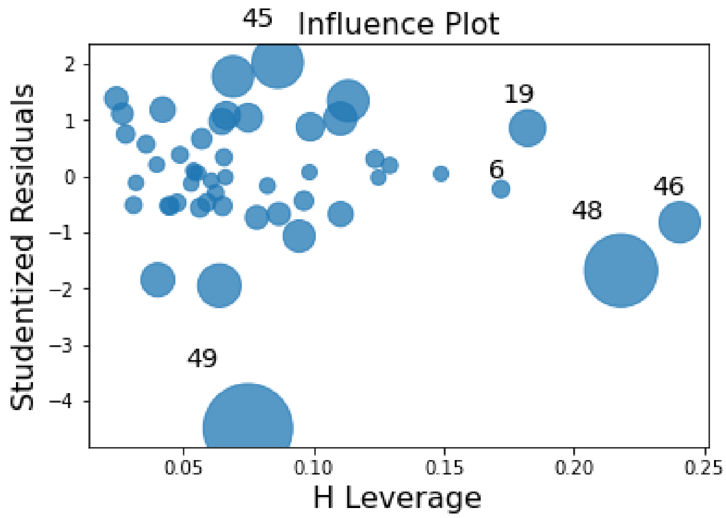
From above summary we can conclude that the R^2 result is satisfactory but the P-value is greater than 0.05 for Administration and Marketing_Spend variable, which is not acceptable.

In order to decrease the P-values, we need to find the influencing variables and we remove those which are influencing more.

Codes:

import statsmodels.api as sm

sm.graphics.influence_plot(ml1)



From above plot e can observe that the Data points (49, 48) influencing more and we are going to remove those data points for analysis.

Codes:

rmv_influenced = df_drop_state.drop(df_drop_state.index[[48,49]],axis=0)

<u>Preparing model with Influenced data set</u>

ml_2=smf.ols('Profit~R_and_D_Spend+Administration+Marketing_Spend',data=rmv_influenced).fit()

ml_2.summary()

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Profit   R-squared:                       0.963
Model:                            OLS   Adj. R-squared:                  0.960
Method:                 Least Squares   F-statistic:                     378.3
Date:                Sun, 26 Apr 2020   Prob (F-statistic):           2.03e-31
Time:                        18:26:47   Log-Likelihood:                -493.33
No. Observations:                  48   AIC:                             994.7
Df Residuals:                      44   BIC:                             1002.
Df Model:                           3
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
Intercept          5.91e+04   5916.711      9.988      0.000    4.72e+04     7.1e+04
R_and_D_Spend        0.7895      0.036     21.718      0.000       0.716       0.863
Administration      -0.0633      0.044     -1.442      0.156      -0.152       0.025
Marketing_Spend      0.0169      0.014      1.249      0.218      -0.010       0.044
==============================================================================
Omnibus:                        0.287   Durbin-Watson:                   1.809
Prob(Omnibus):                  0.866   Jarque-Bera (JB):                0.475
Skew:                           0.057   Prob(JB):                        0.789
Kurtosis:                       2.526   Cond. No.                     1.58e+06
==============================================================================
```

Summary of the model which doesn't have Influencing Data points says that the R^2 value is slightly improved and the P-values also decreased. But still the P-values are not satisfactory, so we need to check the P-values of Administration and Marketing_Spend variable individually with Profit.

Codes:

AD_ml = smf.ols('Profit~ Administration, data = rmv_influenced).fit()

AD_ml.summary()

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Profit   R-squared:                       0.012
Model:                            OLS   Adj. R-squared:                 -0.009
Method:                 Least Squares   F-statistic:                    0.5683
Date:                Wed, 29 Apr 2020   Prob (F-statistic):              0.455
Time:                        17:14:30   Log-Likelihood:                -571.96
No. Observations:                  48   AIC:                             1148.
Df Residuals:                      46   BIC:                             1152.
Df Model:                           1
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept       9.691e+04   2.54e+04      3.816      0.000    4.58e+04    1.48e+05
Administration     0.1523      0.202      0.754      0.455      -0.254       0.559
==============================================================================
Omnibus:                        1.112   Durbin-Watson:                   0.039
Prob(Omnibus):                  0.574   Jarque-Bera (JB):                1.122
Skew:                           0.332   Prob(JB):                        0.571
Kurtosis:                       2.652   Cond. No.                     5.98e+05
==============================================================================
```

For model Profit~ Administration, the P-value is greater than 0.05 i.e., 0.455 which is not Acceptable.

Codes:

MS_ml = smf.ols('Profit~Marketing_Spend',data = rmv_influenced).fit()

MS_ml.summary()

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Profit   R-squared:                       0.516
Model:                            OLS   Adj. R-squared:                  0.505
Method:                 Least Squares   F-statistic:                     49.01
Date:                Wed, 29 Apr 2020   Prob (F-statistic):           9.10e-09
Time:                        14:44:55   Log-Likelihood:                -554.84
No. Observations:                  48   AIC:                             1114.
Df Residuals:                      46   BIC:                             1117.
Df Model:                           1
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
Intercept        6.672e+04   7923.084      8.421      0.000    5.08e+04    8.27e+04
Marketing_Spend     0.2235      0.032      7.000      0.000       0.159       0.288
==============================================================================
Omnibus:                        4.593   Durbin-Watson:                   1.082
Prob(Omnibus):                  0.101   Jarque-Bera (JB):                4.078
Skew:                          -0.348   Prob(JB):                        0.130
Kurtosis:                       4.246   Cond. No.                     5.26e+05
==============================================================================
```

For model Profit~Marketing_Spend, the P-value is lesser than 0.05 i.e., 0.00 which is Acceptable.

Since the results says that the Administration variable has a greater P-value but still we cannot remove the variable so we go for VIF values.

Calculating VIF values:

1 VIF for Market Spend

Codes:

rsq_MS=smf.ols('Marketing_Spend~R_and_D_Spend+Administration',data=rmv_influenced).fit().rsquared

vif_MS = 1/(1-rsq_MS)

vif_MS

VIF for Market Spend: 2.22986

2 VIF for Administration

rsq_ADM=smf.ols('Administration~R_and_D_Spend+Marketing_Spend',data=rmv_influenced).fit().rsquared

vif_ADM = 1/(1-rsq_ADM)

vif_ADM

VIF for Administration: 1.19601

<u>3 VIF for RND</u>

rsq_RND=smf.ols('R_and_D_Spend~Administration+Marketing_Spend',data=rmv_influenced).fit().rsquared

vif_RND = 1/(1-rsq_RND)

vif_RND

VIF for RND: 2.250971

VIF for Administration: 1.19601

VIF for Market Spend: 2.22986

VIF for RND: 2.250971

Based on VIF values we were supposed to remove the least valued variable, but in this case all the VIF values are lesser than 10, so we cannot eliminate any variable.

To get a best fit model we perform transformation techniques.

In these models we concentrate on the AIC and R^2 values, the least AIC with larges R^2 valued model will be considered as best.

<u>Model 1:</u>

<u>Applying Logarithmic function to Profit variable.</u>

Model1=smf.ols('**np.log(Profit)**~R_and_D_Spend+Administration+Marketing_Spend',data=rmv_influenced).fit()

model1.summary()

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         np.log(Profit)   R-squared:                       0.925
Model:                            OLS   Adj. R-squared:                  0.920
Method:                 Least Squares   F-statistic:                     180.7
Date:                Thu, 30 Apr 2020   Prob (F-statistic):           9.50e-25
Time:                        17:31:25   Log-Likelihood:                 46.448
No. Observations:                  48   AIC:                            -84.90
Df Residuals:                      44   BIC:                            -77.41
Df Model:                           3
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
Intercept          11.1241      0.077    143.891      0.000      10.968      11.280
R_and_D_Spend    7.437e-06   4.75e-07     15.658      0.000    6.48e-06    8.39e-06
Administration  -7.299e-07   5.74e-07     -1.272      0.210   -1.89e-06    4.27e-07
Marketing_Spend  -2.05e-09   1.77e-07     -0.012      0.991   -3.58e-07    3.54e-07
==============================================================================
Omnibus:                       23.912   Durbin-Watson:                   1.020
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               47.745
Skew:                          -1.400   Prob(JB):                     4.29e-11
Kurtosis:                       7.004   Cond. No.                     1.58e+06
==============================================================================
```

Model 2:

Applying Logarithmic function to Administration variable.

model2=smf.ols('Profit~R_and_D_Spend+**np.log(Administration)+**Marketing_Spend',data=rmv_influenced).fit()

model2.summary()

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Profit   R-squared:                       0.962
Model:                            OLS   Adj. R-squared:                  0.960
Method:                 Least Squares   F-statistic:                     376.1
Date:                Thu, 30 Apr 2020   Prob (F-statistic):           2.29e-31
Time:                        17:33:05   Log-Likelihood:                -493.47
No. Observations:                  48   AIC:                             994.9
Df Residuals:                      44   BIC:                             1002.
Df Model:                           3
Covariance Type:            nonrobust
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
Intercept               1.269e+05   5.61e+04      2.261      0.029    1.38e+04     2.4e+05
R_and_D_Spend              0.7879      0.036     21.678      0.000       0.715       0.861
np.log(Administration) -6473.0017   4798.323     -1.349      0.184   -1.61e+04    3197.382
Marketing_Spend            0.0179      0.013      1.334      0.189      -0.009       0.045
==============================================================================
Omnibus:                        0.231   Durbin-Watson:                   1.813
Prob(Omnibus):                  0.891   Jarque-Bera (JB):                0.427
Skew:                           0.076   Prob(JB):                        0.808
Kurtosis:                       2.564   Cond. No.                     1.39e+07
==============================================================================
```

Model 3:

Applying Reciprocal function to Administration variable.

model3 = smf.ols('Profit~R_and_D_Spend+**np.reciprocal(Administration)**+Marketing_Spend',data = rmv_influenced).fit()

model3.summary()

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Profit   R-squared:                       0.962
Model:                            OLS   Adj. R-squared:                  0.960
Method:                 Least Squares   F-statistic:                     372.6
Date:                Thu, 30 Apr 2020   Prob (F-statistic):           2.79e-31
Time:                        17:33:15   Log-Likelihood:                -493.68
No. Observations:                  48   AIC:                             995.4
Df Residuals:                      44   BIC:                             1003.
Df Model:                           3
Covariance Type:            nonrobust
===================================================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------
Intercept                    4.634e+04   4768.117      9.719      0.000    3.67e+04     5.6e+04
R_and_D_Spend                   0.7847      0.036     21.684      0.000       0.712       0.858
np.reciprocal(Administration) 5.565e+08   4.68e+08      1.190      0.241   -3.86e+08     1.5e+09
Marketing_Spend                 0.0194      0.013      1.458      0.152      -0.007       0.046
==============================================================================
Omnibus:                        0.182   Durbin-Watson:                   1.797
Prob(Omnibus):                  0.913   Jarque-Bera (JB):                0.378
Skew:                           0.081   Prob(JB):                        0.828
Kurtosis:                       2.597   Cond. No.                     1.15e+11
==============================================================================
```

Model 4:

Applying Square to Administration variable.

rmv_influenced['Administration'] = rmv_influenced.Administration * rmv_influenced.Administration

model4=smf.ols('Profit~R_and_D_Spend+Administration+Marketing_Spend',data=rmv_influenced).fit()

model4.summary()

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Profit   R-squared:                       0.963
Model:                            OLS   Adj. R-squared:                  0.960
Method:                 Least Squares   F-statistic:                     379.2
Date:                Thu, 30 Apr 2020   Prob (F-statistic):           1.92e-31
Time:                        17:37:09   Log-Likelihood:                -493.28
No. Observations:                  48   AIC:                             994.6
Df Residuals:                      44   BIC:                             1002.
Df Model:                           3
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
Intercept          5.57e+04   3808.269     14.627      0.000     4.8e+04    6.34e+04
R_and_D_Spend        0.7898      0.036     21.782      0.000       0.717       0.863
Administration   -2.725e-07   1.84e-07     -1.480      0.146   -6.44e-07    9.87e-08
Marketing_Spend      0.0164      0.014      1.206      0.234      -0.011       0.044
==============================================================================
Omnibus:                        0.352   Durbin-Watson:                   1.793
Prob(Omnibus):                  0.839   Jarque-Bera (JB):                0.521
Skew:                           0.032   Prob(JB):                        0.771
Kurtosis:                       2.494   Cond. No.                     6.11e+10
==============================================================================
```

Model	R squared value	AIC value
Model with original data	0.951	1059
Model with removed influenced data points data	0.963	994.7
Model with log(Profit)	0.925	-84.90
Model with log(Administration)	0.962	994.9
Model with reciprocal(Administration)	0.962	995.4
Model with Administration^2	0.963	994.6

As we can observe the values in the table, we can conclude that the model with Administration^2 is giving a best R^2 Value and least AIC value, so that we consider this model as Best.

<u>Profit Predictions:</u>

profit_pred_model4 = model4.predict(rmv_influenced)

0	188926.001571	1	185158.031645	2	180793.677258	3	172175.412460
4	171668.902385	5	163095.827024	6	158212.874084	7	158152.979714
8	149991.238862	9	154896.562249	10	136619.964771	11	137014.385808
12	129517.264556	13	127500.108797	14	147963.083213	15	146350.417021
16	117624.635471	17	129362.495713	18	129451.526428	19	117536.227703
20	117294.020043	21	116087.600619	22	115010.987535	23	110991.841582
24	116172.877900	25	103731.722388	26	111735.738997	27	113991.268603
28	100718.262985	29	102894.494141	30	102517.811542	31	99082.067910
32	101990.091135	33	100159.041778	34	89051.830395	35	93445.597074
36	77242.212722	37	93025.447860	38	73535.046023	39	87151.397087
40	77418.757368	41	78476.124478	42	74281.942787	43	64111.961601
44	67154.425205	45	52325.480512	46	57960.736732	47	50706.746334

Error:

Actual = rmv_influenced['Profit']

Prediction_model4 = profit_pred_model4

RMSE:

from sklearn.metrics import mean_squared_error

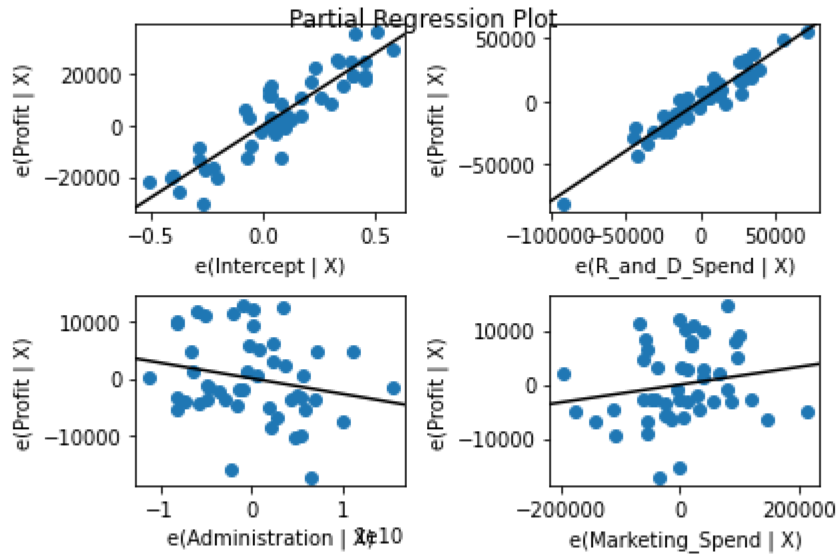
from math import sqrt

rmse_model4 = sqrt(mean_squared_error(Actual, Prediction_model4))

print(rmse_model4)

RMSE: 7028.168934502867

Partial Regression Plot:



The MLR equation with 3 input variables becomes,

$$\mathbf{Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3}$$

Where,

$\boldsymbol{\beta_0}$	**5.57e+04**
$\boldsymbol{\beta_1}$	**0.7898**
$\boldsymbol{\beta_2}$	**-2.725e-07**
$\boldsymbol{\beta_3}$	**0.0164**

Business Problem 2:

Predict Price of the computer, A dataframe containing :

price: price in US dollars of 486 PCs

speed: clock speed in MHz

hd : size of hard drive in MB

ram : size of Ram in in MB

screen : size of screen in inches

cd : is a CD-ROM present ?

multi : is a multimedia kit (speakers, sound card) included ?

premium : is the manufacturer was a "premium" firm (IBM, COMPAQ) ?

ads : number of 486 price listings for each month

trend : time trend indicating month starting from January of 1993 to November of 1995.

Solution:

In the given business problem, 'sale' variable is output and remaining all the variables are inputs.

The independent variables are **speed (X_1), hd (X_2), ram (X_3), screen (X_4), cd (X_5), multi (X_6), premium (X_7), ads (X_8), trend(X_9)** and the output or dependent variable is **price (Y).** Based on the features opted the price of the computer is depended.

So the line equation of the above can be,

$$\mathbf{Y = \beta_0 + \beta_1X_1 + \beta_2X_2 + \beta_3X_3 + \beta_4X_4 + \beta_5X_5 + \beta_6X_6 + \beta_7X_7 + \beta_8X_8 + \beta_9X_9}$$

Goal: We need to come up with a best fit model which predicts the output variable.

Codes:

Load Data:

```
import pandas as pd

computer = pd.read_csv("E:\Data\Assignments\i made\MLR\Computer_Data.csv")

computer.columns
```

Removing sl_no by creating Data Frame

```
df = pd.DataFrame(computer, columns=['sl_no', 'price', 'speed', 'hd', 'ram', 'screen', 'cd',
'multi', 'premium', 'ads', 'trend'])

print(df.shape)
```

Deleting the sl_no column as it doesn’t make any changes.

```
computer_new = df.drop(['sl_no'] , axis='columns')

print(computer_new.shape)

computer_new.corr()
```

From the correlation, as all the values lies lesser than 0.85, we can conclude that there is no any collinearity lies among the inputs. So that we can proceed with the model building.

Creating dummies for the column cd, multi, premium

```
from sklearn import preprocessing

le = preprocessing.LabelEncoder()


computer_new['cd'] = le.fit_transform(computer_new['cd'])

computer_new['multi'] = le.fit_transform(computer_new['multi'])
```

computer_new['premium'] = le.fit_transform(computer_new ['premium'])

After performing above codes, we will get 0's and 1's in place of 'no' and 'yes' respectively.

Creating model:

Model 1:

import statsmodels.formula.api as smf

model1=smf.ols('price~speed+hd+ram+screen+cd+multi+premium+ads+trend',data=computer_new).fit()

model1.summary()

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.776
Model:                            OLS   Adj. R-squared:                  0.775
Method:                 Least Squares   F-statistic:                     2399.
Date:                Fri, 01 May 2020   Prob (F-statistic):               0.00
Time:                        14:05:57   Log-Likelihood:                -44039.
No. Observations:                6259   AIC:                         8.810e+04
Df Residuals:                    6249   BIC:                         8.817e+04
Df Model:                           9
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept        307.9880     60.353      5.103      0.000     189.675     426.301
cd[T.yes]         60.9167      9.516      6.402      0.000      42.263      79.571
multi[T.yes]     104.3238     11.413      9.141      0.000      81.951     126.697
premium[T.yes]  -509.2247     12.342    -41.259      0.000    -533.420    -485.030
speed              9.3203      0.185     50.364      0.000       8.958       9.683
hd                 0.7818      0.028     28.311      0.000       0.728       0.836
ram               48.2560      1.066     45.265      0.000      46.166      50.346
screen           123.0890      3.999     30.776      0.000     115.249     130.929
ads                0.6573      0.051     12.809      0.000       0.557       0.758
trend            -51.8496      0.629    -82.470      0.000     -53.082     -50.617
==============================================================================
```

The P- values of the first model is satisfied but the R^2 value is not acceptable since it shows a moderate relationship between the inputs and output in the model. So we need improve the R squared value by performing some transformation techniques or by removing the influencing data points.

Codes:

import statsmodels.api as sm

sm.graphics.influence_plot(model1)

From influence plot we can see 1700 and 5960 data points are influencing more, so we remove those data points and build a model and can check the corresponding values.

Removing influencing Data points:

Codes:

comp_influenced = computer_new.drop(computer_new.index[[1700,5960]],axis=0)

print(comp_influenced.shape)

Initially the shape of data was 6259 rows, 10 columns, and now it is 6257 rows, 10 columns.

Model 2:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.777
Model:                            OLS   Adj. R-squared:                  0.776
Method:                 Least Squares   F-statistic:                     2414.
Date:                Fri, 01 May 2020   Prob (F-statistic):               0.00
Time:                        15:06:04   Log-Likelihood:                -43999.
No. Observations:                6257   AIC:                         8.802e+04
Df Residuals:                    6247   BIC:                         8.808e+04
Df Model:                           9
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept        322.3846     60.127      5.362      0.000     204.514     440.255
cd[T.yes]         59.0419      9.480      6.228      0.000      40.457      77.627
multi[T.yes]     106.1311     11.375      9.330      0.000      83.833     128.429
premium[T.yes] -509.6628     12.290    -41.469      0.000    -533.756    -485.570
speed              9.3140      0.184     50.544      0.000       8.953       9.675
hd                 0.7861      0.028     28.352      0.000       0.732       0.840
ram               48.2220      1.066     45.255      0.000      46.133      50.311
screen           122.0655      3.985     30.633      0.000     114.254     129.877
ads                0.6555      0.051     12.827      0.000       0.555       0.756
trend            -51.8145      0.627    -82.609      0.000     -53.044     -50.585
==============================================================================
```

By using the data which doesn't contain the influencing data points in the model, the R^2 value is raised slightly but the P-value remained as same previous model.

So we try the transformations to expect the better results.

Model 3:

Building a model based on the variable applied with logarithmic transformation on Price variable.

Codes:

Import numpy as np

model3=smf.ols('np.log(price)~speed+hd+ram+screen+cd+multi+premium+ads+trend',data =comp_influenced).fit()

model3.summary()

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          np.log(price)   R-squared:                       0.784
Model:                            OLS   Adj. R-squared:                  0.783
Method:                 Least Squares   F-statistic:                     2512.
Date:                Fri, 01 May 2020   Prob (F-statistic):               0.00
Time:                        15:21:33   Log-Likelihood:                 4391.7
No. Observations:                6257   AIC:                            -8763.
Df Residuals:                    6247   BIC:                            -8696.
Df Model:                           9
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept          6.8386      0.026    259.797      0.000       6.787       6.890
cd[T.yes]          0.0489      0.004     11.775      0.000       0.041       0.057
multi[T.yes]       0.0481      0.005      9.656      0.000       0.038       0.058
premium[T.yes]    -0.2273      0.005    -42.237      0.000      -0.238      -0.217
speed              0.0043   8.07e-05     52.685      0.000       0.004       0.004
hd                 0.0003   1.21e-05     28.280      0.000       0.000       0.000
ram                0.0208      0.000     44.594      0.000       0.020       0.022
screen             0.0540      0.002     30.935      0.000       0.051       0.057
ads                0.0003   2.24e-05     12.117      0.000       0.000       0.000
trend             -0.0236      0.000    -85.987      0.000      -0.024      -0.023
==============================================================================
```

By applying the Logarithmic Transformation on price variable in the model, the R^2 value is raised slightly but not up to the sufficient level, so continue with the further transformation techniques in next model.

<u>Model 4:</u>

Quadratic Model

Applying square for all input variables and adding with original variable.

Codes:

comp_influenced['speed_sq'] = comp_influenced.speed * comp_influenced.speed

comp_influenced['hd_sq'] = comp_influenced.hd * comp_influenced.hd

comp_influenced['ram_sq'] = comp_influenced.ram * comp_influenced.ram

comp_influenced['screen_sq'] = comp_influenced.screen * comp_influenced.screen

comp_influenced['ads_sq'] = comp_influenced.ads * comp_influenced.ads

comp_influenced['trend_sq'] = comp_influenced.trend * comp_influenced.trend

model4=smf.ols('price~speed+speed_sq+hd+hd_sq+ram+ram_sq+screen+screen_sq+cd+multi+premium+ads+ads_sq+trend+trend_sq',data=comp_influenced).fit()

model4.summary()

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.805
Model:                            OLS   Adj. R-squared:                  0.804
Method:                 Least Squares   F-statistic:                     1713.
Date:                Sat, 02 May 2020   Prob (F-statistic):               0.00
Time:                        22:35:35   Log-Likelihood:                -43580.
No. Observations:                6257   AIC:                         8.719e+04
Df Residuals:                    6241   BIC:                         8.730e+04
Df Model:                          15
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept       1.136e+04    892.369     12.728      0.000    9608.344    1.31e+04
cd[T.yes]         31.4774      9.003      3.496      0.000      13.828      49.127
multi[T.yes]     101.4936     10.662      9.519      0.000      80.593     122.394
premium[T.yes]  -527.7829     11.601    -45.496      0.000    -550.524    -505.042
speed             20.3979      0.789     25.857      0.000      18.851      21.944
speed_sq          -0.0945      0.007    -14.484      0.000      -0.107      -0.082
hd                 1.5376      0.065     23.634      0.000       1.410       1.665
hd_sq             -0.0006   4.56e-05    -12.190      0.000      -0.001      -0.000
ram               53.3756      3.124     17.085      0.000      47.251      59.500
ram_sq            -0.2124      0.115     -1.852      0.064      -0.437       0.012
screen         -1354.4283    116.909    -11.585      0.000   -1583.610   -1125.246
screen_sq         47.6530      3.798     12.548      0.000      40.208      55.098
ads               -1.3837      0.414     -3.340      0.001      -2.196      -0.571
ads_sq             0.0027      0.001      3.346      0.001       0.001       0.004
trend            -19.3967      4.147     -4.677      0.000     -27.526     -11.268
trend_sq          -1.0645      0.133     -7.976      0.000      -1.326      -0.803
==============================================================================
```

R^2 value for this model is acceptable, hence the model is considered as final.

Where value of R^2 is 0.805

<u>The prediction for the price of the computer:</u>

Codes:

price_pred = model4.predict(comp_influenced)

And co efficient values are:

$\boldsymbol{\beta_0}$ 1.136e+04

$\boldsymbol{\beta_1}$ 20.3979

$\boldsymbol{\beta_2}$ 1.5376

$\boldsymbol{\beta_3}$ 53.3756

$\boldsymbol{\beta_4}$ -1354.4283

$\boldsymbol{\beta_5}$ 31.4774

$\boldsymbol{\beta_6}$ 101.4936

$\boldsymbol{\beta_7}$ -527.7829

$\boldsymbol{\beta_8}$ -1.3837

$\boldsymbol{\beta_9}$ -19.3967

Business Problem 3:

Consider only the below columns and prepare a prediction model for predicting Price.

Corolla<-
Corolla[c("Price","Age_08_04","KM","HP","cc","Doors","Gears","Quarterly_Tax","Weight")]

Model -- model of the car

Price -- Offer Price in EUROs

Age_08_04 -- Age in months as in August 2004

Mfg_Month -- Manufacturing month (1-12)

Mfg_Year -- Manufacturing Year

KM -- Accumulated Kilometers on odometer

Fuel_Type -- Fuel Type (Petrol, Diesel, CNG)

HP -- Horse Power

Met_Color -- Metallic Color? (Yes=1, No=0)

Color -- Color (Blue, Red, Grey, Silver, Black, etc.)

Automatic -- Automatic ((Yes=1, No=0)

cc -- Cylinder Volume in cubic centimeters

Doors -- Number of doors

Cylinders -- Number of cylinders

Gears -- Number of gear positions

Quarterly_Tax -- Quarterly road tax in EUROs

Weight -- Weight in Kilograms

Mfr_Guarantee -- Within Manufacturer's Guarantee period (Yes=1, No=0)

BOVAG_Guarantee -- BOVAG (Dutch dealer network) Guarantee (Yes=1, No=0)

Guarantee_Period -- Guarantee period in months

ABS -- Anti-Lock Brake System (Yes=1, No=0)

Airbag_1 -- Driver_Airbag (Yes=1, No=0)

Airbag_2 -- Passenger Airbag (Yes=1, No=0)

Airco -- Airconditioning (Yes=1, No=0)

Automatic_airco -- Automatic Airconditioning (Yes=1, No=0)

Boardcomputer -- Boardcomputer (Yes=1, No=0)

CD_Player -- CD Player (Yes=1, No=0)

Central_Lock -- Central Lock (Yes=1, No=0)

Powered_Windows -- Powered Windows (Yes=1, No=0)

Power_Steering -- Power Steering (Yes=1, No=0)

Radio -- Radio (Yes=1, No=0)

Mistlamps   -- Mistlamps (Yes=1, No=0)

Sport_Model -- Sport Model (Yes=1, No=0)

Backseat_Divider -- Backseat Divider (Yes=1, No=0)

Metallic_Rim --Metallic Rim (Yes=1, No=0)

Radio_cassette -- Radio Cassette (Yes=1, No=0)

Tow_Bar -- Tow Bar (Yes=1, No=0)

Solution:

In the given business problem, the independent variables are **Age_08_04 (X_1), KM (X_2), HP (X_3), cc(X_4), Doors (X_5), Gears (X_6), Quarterly_Tax (X_7), weight (X_8)** and the output or dependent variable is **Price (Y)**. Based on input variables the profit is depended.

So the line equation of the above can be,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \beta_6 X_6 + \beta_7 X_7 + \beta_8 X_8 + \beta_9 X_9$$

Goal: We need to come up with a best fit model which predicts the output variable.

Codes:

#........load data........

```
import pandas as pd
toyota_corolla = pd.read_csv('E:\\Data\\Assignments\\i made\\MLR\\ToyotaCorolla.csv' ,
engine = 'python')
toyota_corolla.columns
```

Creating a data frame to keep only required variables

Codes:

toyota = pd.DataFrame(toyota_corolla, columns = ["Price","Age_08_04","KM","HP","cc","Doors","Gears","Quarterly_Tax","Weight"])

toyota.columns

toyota.shape

(1436, 9)

To check the collinearity among the variables can be measured by:

1 Correlation:

Codes:

corr = toyota.corr()

Index	Price	Age_08_04	KM	HP	cc	Doors	Gears	Quarterly_Tax	Weight
Price	1	-0.87659	-0.56996	0.31499	0.126389	0.185326	0.0631039	0.219197	0.581198
Age_08_04	-0.87659	1	0.505672	-0.156622	-0.0980837	-0.148359	-0.00536395	-0.198431	-0.470253
KM	-0.56996	0.505672	1	-0.333538	0.102683	-0.0361966	0.0150233	0.278165	-0.0285985
HP	0.31499	-0.156622	-0.333538	1	0.0358558	0.0924245	0.209477	-0.298432	0.0896141
cc	0.126389	-0.0980837	0.102683	0.0358558	1	0.0799033	0.0146294	0.306996	0.335637
Doors	0.185326	-0.148359	-0.0361966	0.0924245	0.0799033	1	-0.160141	0.109363	0.302618
Gears	0.0631039	-0.00536395	0.0150233	0.209477	0.0146294	-0.160141	1	-0.00545196	0.0206133
Quarterly_Tax	0.219197	-0.198431	0.278165	-0.298432	0.306996	0.109363	-0.00545196	1	0.626134
Weight	0.581198	-0.470253	-0.0285985	0.0896141	0.335637	0.302618	0.0206133	0.626134	1

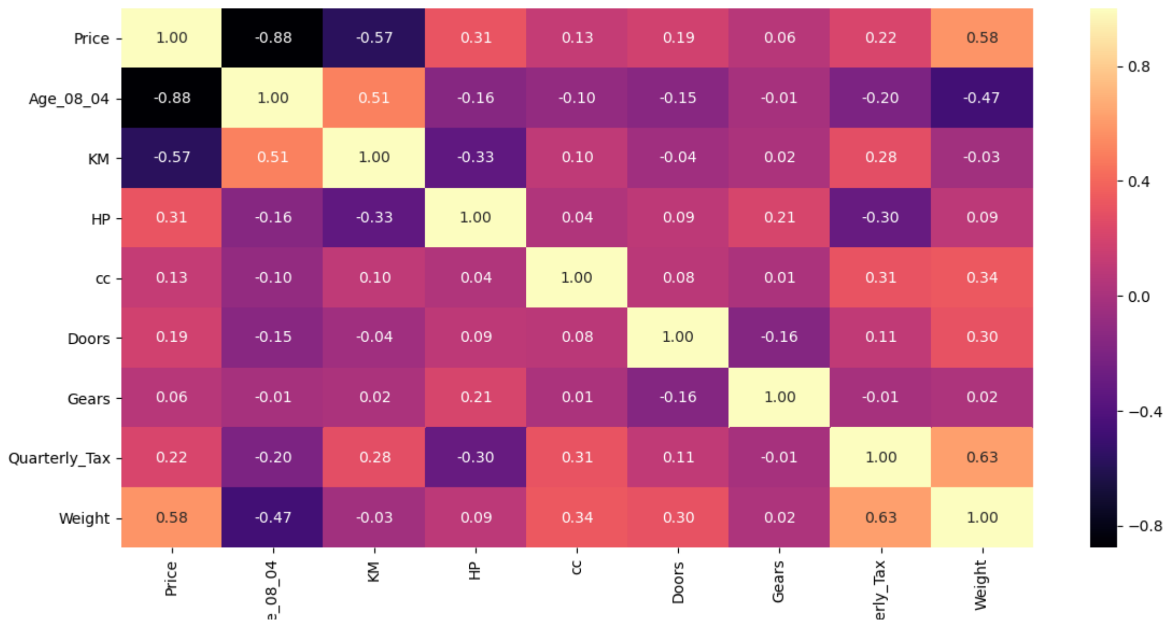
The correlations says that there exists no collinearity among the variables.

2 Plotting a heat map:

Codes:

import seaborn as sns

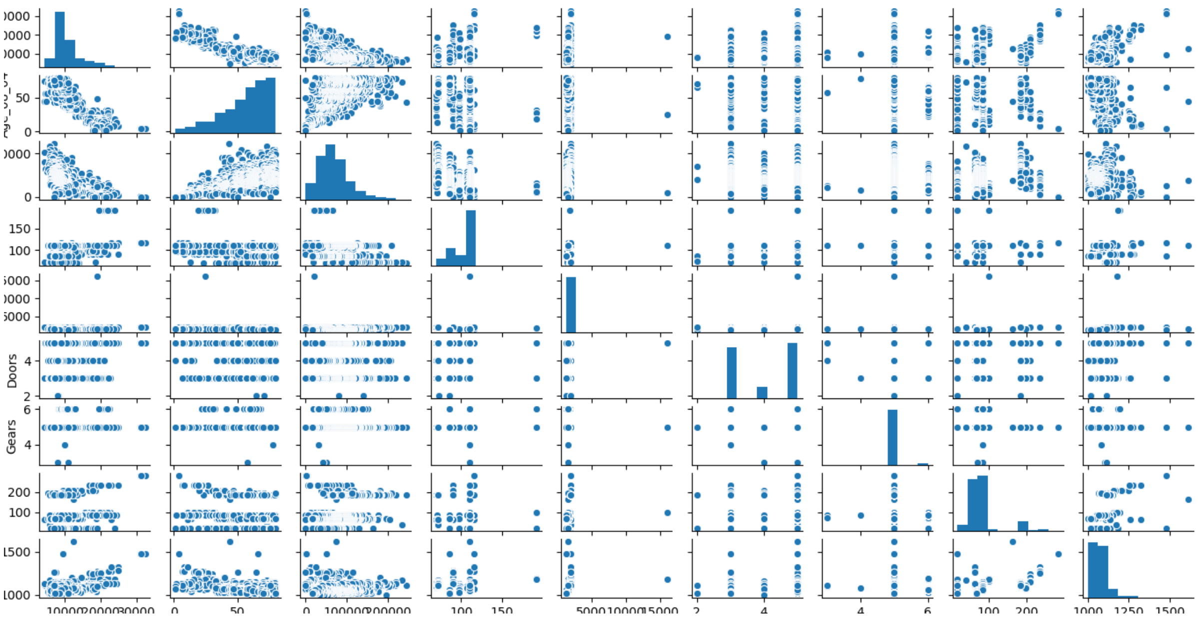
sns.heatmap(corr, cmap='magma', annot=True, fmt=".2f")



3 Pair plots among the variables:

Codes:

```
import seaborn as sns
sns.pairplot(toyota)
```



So far from all above three analysis we can conclude that there doesn’t exists a multicollinearity among the variables. But it is possible in model building if the p values is greater than 0.05 indicates the existence of multicollinearity.

So we can proceed to build the models.

Model Building

import statsmodels.formula.api as smf # for regression model

Model 1

model1 = smf.ols('Price~ Age_08_04+KM+HP+cc+Doors+Gears+Quarterly_Tax+Weight', data=toyota).fit()

model1.summary()

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Price   R-squared:                       0.864
Model:                            OLS   Adj. R-squared:                  0.863
Method:                 Least Squares   F-statistic:                     1131.
Date:                Sun, 03 May 2020   Prob (F-statistic):               0.00
Time:                        10:56:04   Log-Likelihood:                -12376.
No. Observations:                1436   AIC:                         2.477e+04
Df Residuals:                    1427   BIC:                         2.482e+04
Df Model:                           8
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
Intercept     -5573.1064   1411.390     -3.949      0.000   -8341.728   -2804.485
Age_08_04      -121.6584      2.616    -46.512      0.000    -126.789    -116.527
KM               -0.0208      0.001    -16.622      0.000      -0.023      -0.018
HP               31.6809      2.818     11.241      0.000      26.152      37.209
cc               -0.1211      0.090     -1.344      0.179      -0.298       0.056
Doors            -1.6166     40.006     -0.040      0.968     -80.093      76.859
Gears           594.3199    197.055      3.016      0.003     207.771     980.869
Quarterly_Tax     3.9491      1.310      3.015      0.003       1.379       6.519
Weight           16.9586      1.068     15.880      0.000      14.864      19.054
==============================================================================
```

The R^2 value is greater than 0.8, it is acceptable but the p values of the cc and Doors indicating the existence of multicollinearity. To remove collinearity, we build models on the individual basis for cc and Doors.

Model based on 'cc':

Codes:

modelcc = smf.ols('Price~ cc', data=toyota).fit()

modelcc.summary()

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Price   R-squared:                       0.016
Model:                            OLS   Adj. R-squared:                  0.015
Method:                 Least Squares   F-statistic:                     23.28
Date:                Sun, 03 May 2020   Prob (F-statistic):           1.55e-06
Time:                        11:01:02   Log-Likelihood:                -13795.
No. Observations:                1436   AIC:                         2.759e+04
Df Residuals:                    1434   BIC:                         2.760e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   9027.5548    365.576     24.694      0.000    8310.435    9744.675
cc             1.0802      0.224      4.825      0.000       0.641       1.519
==============================================================================
Omnibus:                      465.181   Durbin-Watson:                   0.267
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1390.401
Skew:                           1.649   Prob(JB):                    1.20e-302
Kurtosis:                       6.516   Cond. No.                     6.29e+03
==============================================================================
```

The p value for the model with cc variable is 0, which is significant.

<u>Model based on 'Doors':</u>

Codes:

modelDoors = smf.ols('Price~ Doors', data=toyota).fit()

modelDoors.summary()

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Price   R-squared:                       0.034
Model:                            OLS   Adj. R-squared:                  0.034
Method:                 Least Squares   F-statistic:                     51.00
Date:                Sun, 03 May 2020   Prob (F-statistic):           1.46e-12
Time:                        11:02:38   Log-Likelihood:                -13782.
No. Observations:                1436   AIC:                         2.757e+04
Df Residuals:                    1434   BIC:                         2.758e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   7885.0058    409.438     19.258      0.000    7081.843    8688.168
Doors        705.5586     98.795      7.142      0.000     511.761     899.356
==============================================================================
Omnibus:                      466.779   Durbin-Watson:                   0.287
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1406.209
Skew:                           1.651   Prob(JB):                    4.42e-306
Kurtosis:                       6.549   Cond. No.                         19.0
==============================================================================
```

For model based on Doors also indicates significant p value. So that we can check by building a model based on cc and Doors variable together.

Model based on ‘cc’ and ‘Doors’ together:

Codes:

modelccDoors = smf.ols('Price~ cc+Doors ', data=toyota).fit()

modelccDoors.summary()

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Price   R-squared:                       0.047
Model:                            OLS   Adj. R-squared:                  0.046
Method:                 Least Squares   F-statistic:                     35.24
Date:                Sun, 03 May 2020   Prob (F-statistic):           1.15e-15
Time:                        11:04:13   Log-Likelihood:                -13772.
No. Observations:                1436   AIC:                         2.755e+04
Df Residuals:                    1433   BIC:                         2.757e+04
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   6509.4211    515.773     12.621      0.000    5497.670    7521.173
Doors        671.3973     98.501      6.816      0.000     478.176     864.619
cc             0.9597      0.221      4.340      0.000       0.526       1.393
==============================================================================
Omnibus:                      448.845   Durbin-Watson:                   0.290
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1294.854
Skew:                           1.603   Prob(JB):                    6.70e-282
Kurtosis:                       6.370   Cond. No.                     9.09e+03
==============================================================================
```

Model with on cc and Doors variable together also indicates the significant p value. So we plot influence plot to recognize which data points are influencing more so that we can remove and build model based on that data set.

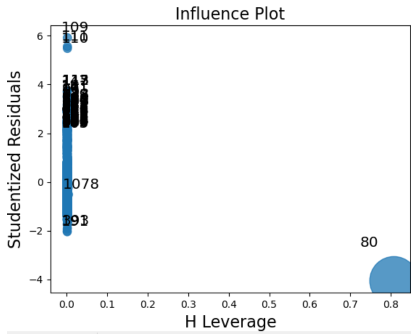
Influence Plot:

Codes:

import statsmodels.api as sm

sm.graphics.influence_plot(modelccDoors)

influence.measures(modelccDoors.toyota)



From the influence plot we can observe the data point with index value 80 is influencing more, we need to remove that data point.

Removing the Data Point:

Codes:

toyota_new = toyota.drop(toyota.index[[80]], axis=0)

print(toyota_new.shape)

(1435, 9)

Building a model based on the Data set with eliminated influencing data point:

Model 2:

model2 = smf.ols('Price~ Age_08_04+KM+HP+cc+Doors+Gears+Quarterly_Tax+Weight', data=toyota_new).fit()

model2.summary()

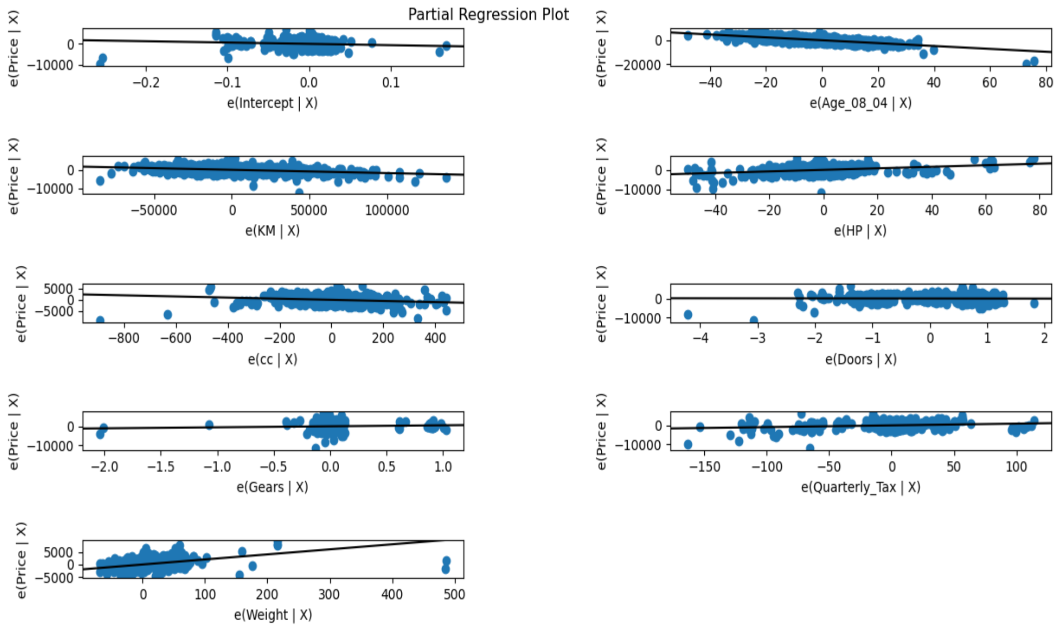
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Price   R-squared:                       0.869
Model:                            OLS   Adj. R-squared:                  0.869
Method:                 Least Squares   F-statistic:                     1186.
Date:                Sun, 03 May 2020   Prob (F-statistic):               0.00
Time:                        11:14:30   Log-Likelihood:                -12335.
No. Observations:                1435   AIC:                         2.469e+04
Df Residuals:                    1426   BIC:                         2.473e+04
Df Model:                           8
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
Intercept     -6284.7401   1382.748     -4.545      0.000   -8997.180   -3572.301
Age_08_04      -120.4550      2.562    -47.021      0.000    -125.480    -115.430
KM               -0.0178      0.001    -13.973      0.000      -0.020      -0.015
HP               39.3463      2.911     13.516      0.000      33.636      45.057
cc               -2.5242      0.307     -8.216      0.000      -3.127      -1.922
Doors           -27.2285     39.241     -0.694      0.488    -104.206      49.749
Gears           523.9416    192.865      2.717      0.007     145.612     902.271
Quarterly_Tax     9.0440      1.425      6.348      0.000       6.249      11.839
Weight           20.1655      1.116     18.076      0.000      17.977      22.354
==============================================================================
```

The R^2 value is satisfactory but the p value of the Doors variable is still higher. For collinearity, contribution of Doors variable is more.

So we need to remove the Doors variable and build a model. By plotting Partial Regression plot the significance can be observed, if the Door variable having most horizontal line then we can surely remove the Doors variable.

Codes:

```
sm.graphics.plot_partregress_grid(model2)
```



By observing the plot, Doors variable having most horizontal line it can be removed.

Model without Doors variable:

Model 3:

Codes:

```
model3=smf.ols('Price~Age_08_04+KM+HP+cc+Gears+Quarterly_Tax+Weight',data=toyota_new).fit()

model3.summary()
```

```
                           OLS Regression Results
==============================================================================
Dep. Variable:                  Price   R-squared:                       0.869
Model:                            OLS   Adj. R-squared:                  0.869
Method:                 Least Squares   F-statistic:                     1356.
Date:                Sun, 03 May 2020   Prob (F-statistic):               0.00
Time:                        11:38:32   Log-Likelihood:                -12335.
No. Observations:                1435   AIC:                         2.469e+04
Df Residuals:                    1427   BIC:                         2.473e+04
Df Model:                           7
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
Intercept     -6313.9396   1381.857     -4.569      0.000   -9024.628   -3603.251
Age_08_04      -120.4577      2.561    -47.031      0.000    -125.482    -115.433
KM               -0.0179      0.001    -14.029      0.000      -0.020      -0.015
HP               39.1593      2.898     13.512      0.000      33.474      44.844
cc               -2.5069      0.306     -8.188      0.000      -3.107      -1.906
Gears           549.7311    189.216      2.905      0.004     178.561     920.902
Quarterly_Tax     9.0759      1.424      6.374      0.000       6.283      11.869
Weight           19.9623      1.076     18.547      0.000      17.851      22.074
==============================================================================
```

The R^2 and p values of all the variables are satisfactory and the model is acceptable.

After removing the Doors variable the new line equation will be:

$Y = \beta_0 + \beta_1X_1 + \beta_2X_2 + \beta_3X_3 + \beta_4X_4 + \beta_5X_5 + \beta_6X_6 + \beta_7X_7$

Where,

β_0 -6313.9396

β_1 -120.4577

β_2 -0.0179

β_3 39.1593

β_4 -2.5069

β_5 549.7311

β_6 9.0759

β_7 19.9623