Import neccessery libraries

In [55]:
```
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from numpy.polynomial.polynomial import polyfit
from sklearn.linear_model import LinearRegression
import seaborn as sns
import statsmodels.stats.tests.test_influence
from sklearn.feature_selection import RFE
from statsmodels.stats.outliers_influence import variance_inflation_factor
import warnings
warnings.filterwarnings('ignore')
```

Problem

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","QuarterlyTax","

Import data

In [2]:
```
import os
```

In [3]:
```
os.getcwd()
```

Out[3]:
```
'C:\\Users\\Akarsh\\assignment-5'
```

In [4]:
```
os.chdir('C:\\Users\\Akarsh\\Desktop\\assignments\\multiple linear regressi
```

In [5]:
```
os.getcwd()
```

Out[5]:
```
'C:\\Users\\Akarsh\\Desktop\\assignments\\multiple linear regression'
```

In [6]:
```
toyota_data = pd.read_csv('ToyotaCorolla.csv',encoding='latin1')
toyota_data
```

Out[6]:

	Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Colo
0	1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	10	2002	46986	Diesel	90	
1	2	TOYOTA Corolla 2.0 D4D HATCHB TERRA	13750	23	10	2002	72937	Diesel	90	

	Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Colo
		2/3-Doors								
2	3	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	9	2002	41711	Diesel	90	
3	4	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	7	2002	48000	Diesel	90	
4	5	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	13750	30	3	2002	38500	Diesel	90	
...	...	...	...	...	...	...	...	...	...	
1431	1438	TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors	7500	69	12	1998	20544	Petrol	86	
1432	1439	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	10845	72	9	1998	19000	Petrol	86	
1433	1440	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	8500	71	10	1998	17016	Petrol	86	
1434	1441	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	7250	70	11	1998	16916	Petrol	86	
1435	1442	TOYOTA Corolla 1.6 LB LINEA	6950	76	5	1998	1	Petrol	110	

Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Colc
	TERRA 4/5-								

In [7]:
```
toyota_data_2 = toyota_data [['Price','Age_08_04','KM','HP','cc','Doors','(
toyota_data_2
```

Out[7]:

	Price	Age_08_04	KM	HP	cc	Doors	Gears	Quarterly_Tax	Weight
0	13500	23	46986	90	2000	3	5	210	1165
1	13750	23	72937	90	2000	3	5	210	1165
2	13950	24	41711	90	2000	3	5	210	1165
3	14950	26	48000	90	2000	3	5	210	1165
4	13750	30	38500	90	2000	3	5	210	1170
...	...	...	...	...	...	...	...	...	...
1431	7500	69	20544	86	1300	3	5	69	1025
1432	10845	72	19000	86	1300	3	5	69	1015
1433	8500	71	17016	86	1300	3	5	69	1015
1434	7250	70	16916	86	1300	3	5	69	1015
1435	6950	76	1	110	1600	5	5	19	1114

1436 rows × 9 columns

In [ ]:
```
```

Data understanding

In [8]:
```
toyota_data_2.shape
```

Out[8]:
```
(1436, 9)
```

In [9]:
```
toyota_data_2.isna().sum()
```

Out[9]:
```
Price           0
Age_08_04       0
KM              0
HP              0
cc              0
Doors           0
Gears           0
Quarterly_Tax   0
Weight          0
dtype: int64
```

In [10]:
```
toyota_data_2.dtypes
```

Out[10]:
```
Price           int64
Age_08_04       int64
```

```
KM              int64
HP              int64
cc              int64
Doors           int64
Gears           int64
Quarterly_Tax   int64
Weight          int64
```

In [11]:

```
toyota_data_2.describe
```

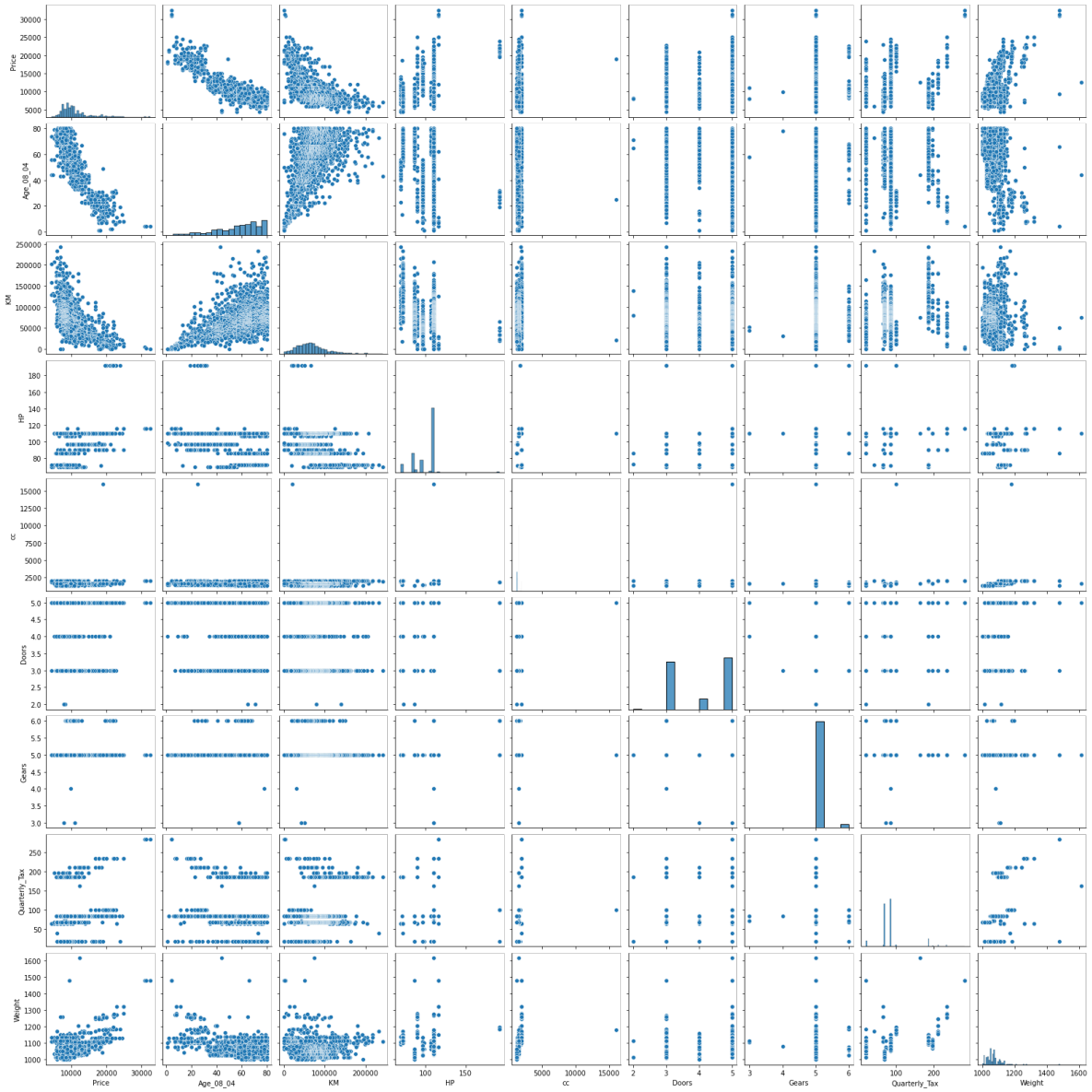
Out[11]:

```
<bound method NDFrame.describe of       Price  Age_08_04     KM   HP     cc
Doors  Gears  Quarterly_Tax  Weight
0     13500         23  46986   90  2000      3      5            210    11
65
1     13750         23  72937   90  2000      3      5            210    11
65
2     13950         24  41711   90  2000      3      5            210    11
65
3     14950         26  48000   90  2000      3      5            210    11
65
4     13750         30  38500   90  2000      3      5            210    11
70
...     ...        ...    ...  ...   ...    ...    ...            ...
...
1431   7500         69  20544   86  1300      3      5             69    10
25
1432  10845         72  19000   86  1300      3      5             69    10
15
1433   8500         71  17016   86  1300      3      5             69    10
15
1434   7250         70  16916   86  1300      3      5             69    10
15
1435   6950         76      1  110  1600      5      5             19    11
14

[1436 rows x 9 columns]>
```

In [16]:

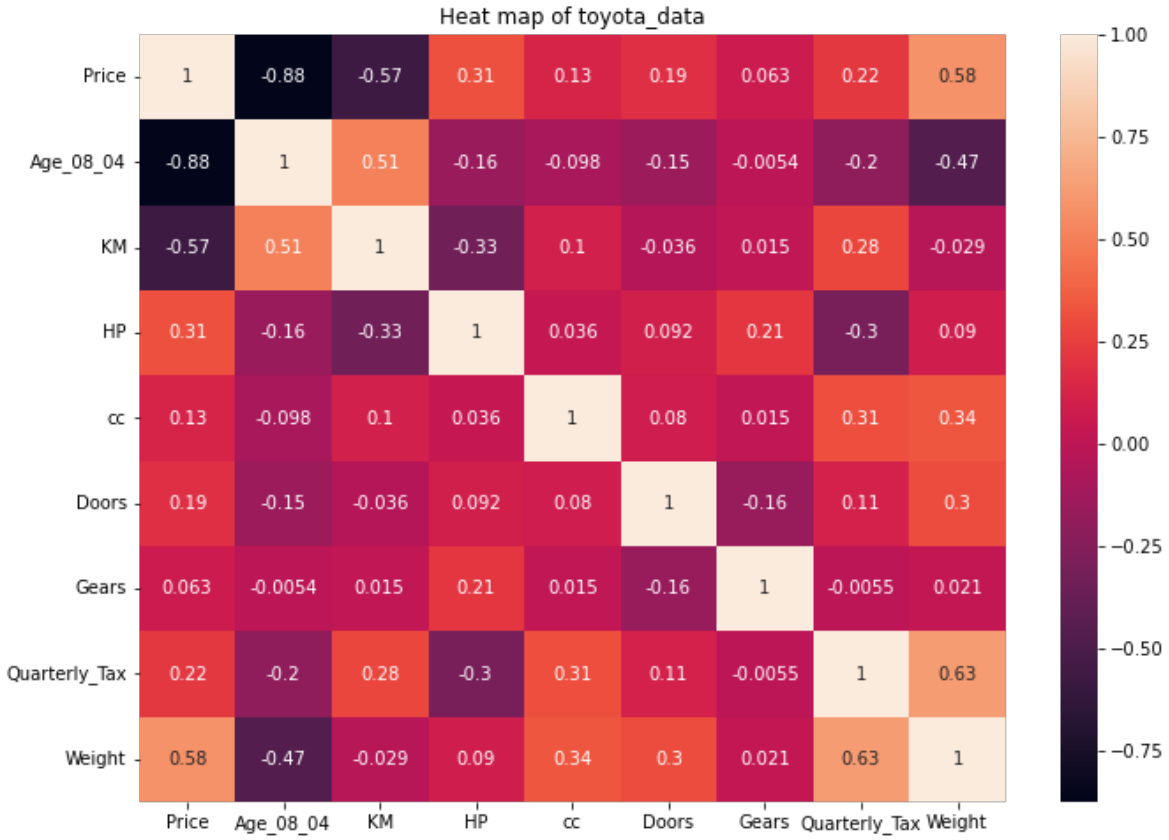
```
sns.pairplot(toyota_data_2)
plt.show()
```



Correlation matrix

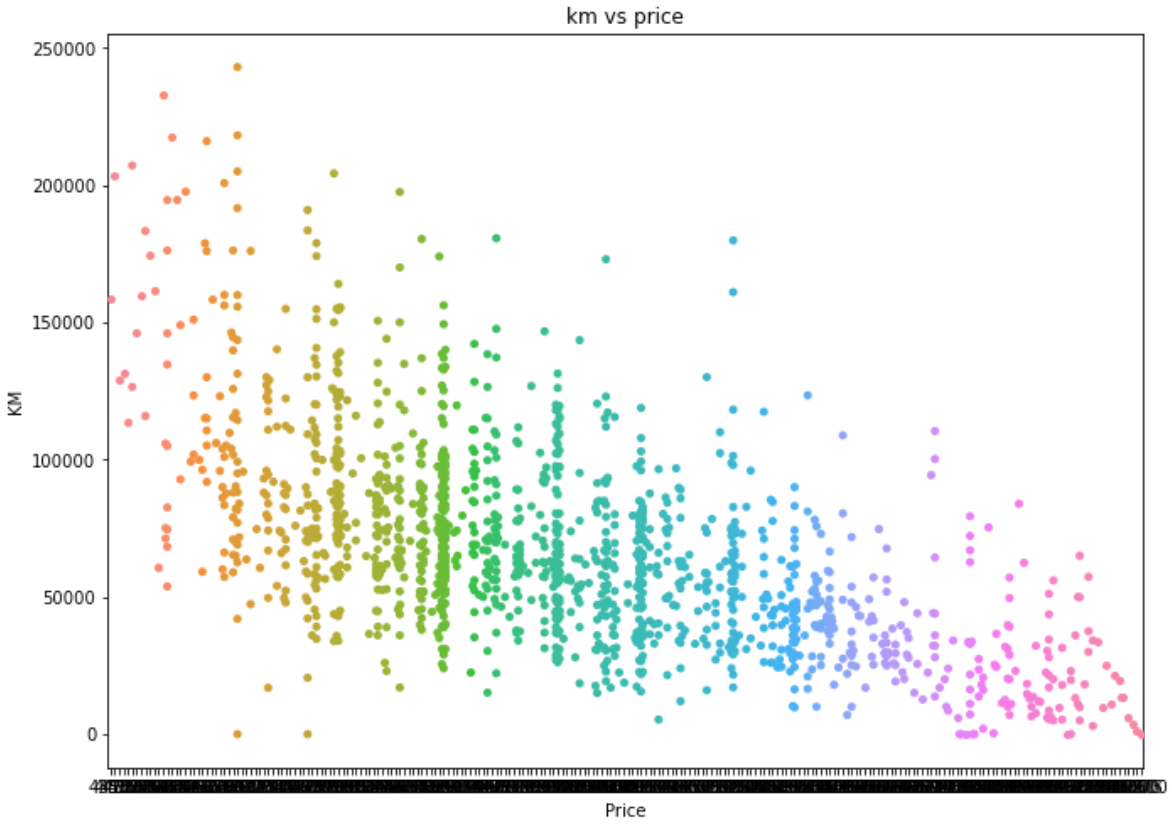
In [14]:
```
corrMatrix = toyota_data_2.corr()
```

In [15]:
```
plt.figure(figsize=(11,8))
plt.title('Heat map of toyota_data')
sns.heatmap(corrMatrix, annot=True)
plt.show()
```



In [57]:

```
plt.figure(figsize=(11,8))
plt.title('km vs price')
sns.swarmplot(x ='Price',y='KM',data = toyota_data_2,size = 5)
plt.show()
```



Regression model

In [19]:
```
X = toyota_data_2[['Age_08_04','KM','HP','cc','Doors','Gears','Quarterly_Ta
Y = toyota_data_2[['Price']]
```

In [20]:
```
model = sm.OLS(Y, X).fit()
predictions = model.predict(X)
model.summary()
```

Out[20]:

OLS Regression Results

Dep. Variable:	Price	**R-squared (uncentered):**	0.986
Model:	OLS	**Adj. R-squared (uncentered):**	0.986
Method:	Least Squares	**F-statistic:**	1.247e+04
Date:	Thu, 17 Feb 2022	**Prob (F-statistic):**	0.00
Time:	10:51:41	**Log-Likelihood:**	-12383.
No. Observations:	1436	**AIC:**	2.478e+04
Df Residuals:	1428	**BIC:**	2.482e+04
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Age_08_04	-125.4510	2.445	-51.303	0.000	-130.248	-120.654
KM	-0.0205	0.001	-16.305	0.000	-0.023	-0.018
HP	33.4737	2.796	11.973	0.000	27.990	38.958
cc	-0.1032	0.090	-1.141	0.254	-0.281	0.074
Doors	-7.2494	40.184	-0.180	0.857	-86.075	71.576
Gears	78.3780	148.258	0.529	0.597	-212.449	369.205
Quarterly_Tax	5.8258	1.227	4.748	0.000	3.419	8.233
Weight	14.0322	0.773	18.157	0.000	12.516	15.548

Omnibus:	108.641	**Durbin-Watson:**	1.509
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	562.996
Skew:	0.019	**Prob(JB):**	5.59e-123
Kurtosis:	6.067	**Cond. No.**	3.26e+05

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 3.26e+05. This might indicate that there are strong multicollinearity or other numerical problems.

R sq and p Value of the Model is Good and the model can be accepted. However as you can see not all variables have acceptable p value. Thus we have Multicollinearitv issue in our Data Frame

Multicollinearity

Finding Cook's Distance

In [21]:
```
infl = model.get_influence()
```

In [22]:
```
summ_df = infl.summary_frame()
```

In [23]:
```
summ_df.sort_values('cooks_d', ascending=False)
```

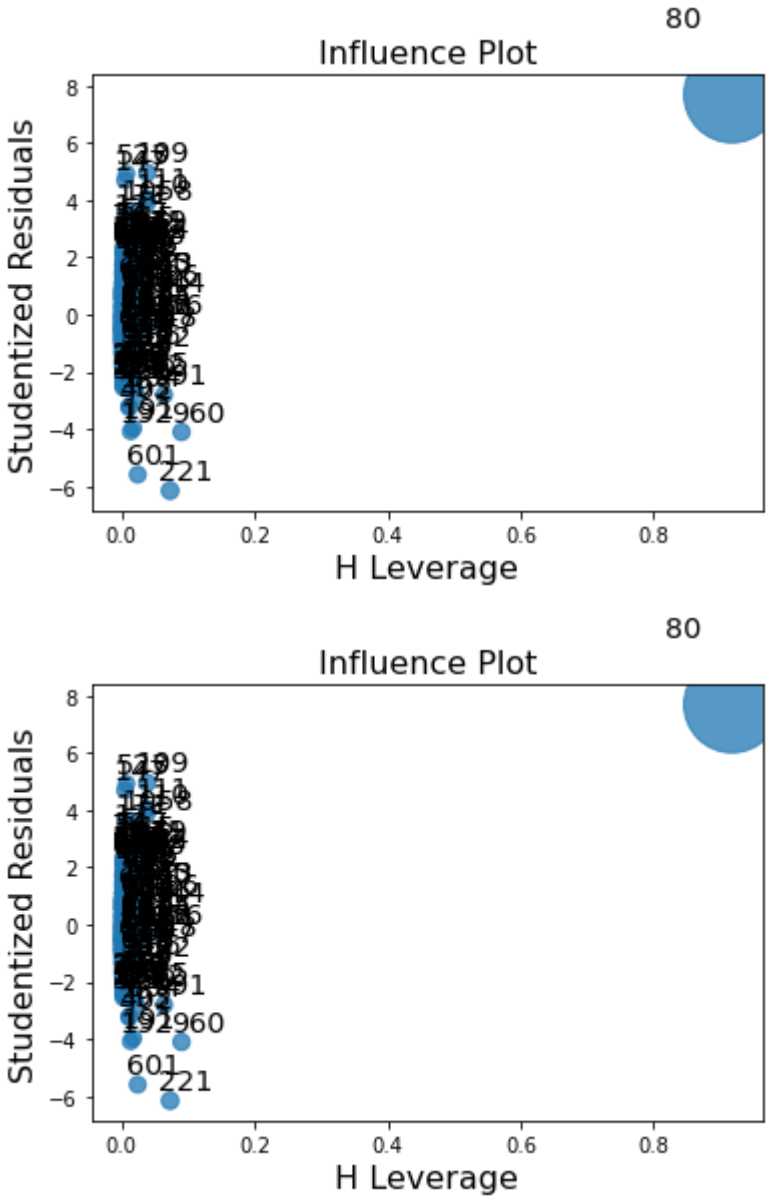
Out[23]:

	dfb_Age_08_04	dfb_KM	dfb_HP	dfb_cc	dfb_Doors	dfb_Gears	dfb_Quarte
80	-0.289686	-2.363023e+00	-2.781734	2.615263e+01	0.645518	0.905398	-4.
221	-0.255051	-2.607849e-02	0.174427	4.179288e-01	0.553762	1.572311	0.
960	-0.208597	3.740370e-02	0.443719	1.977218e-01	0.539659	0.995730	0.
109	0.104572	-2.934313e-01	0.114571	-1.902240e-01	-0.220145	-0.606577	0.
601	-0.213100	3.186680e-01	0.385097	-1.107466e-01	0.404397	0.551158	0.
...	...	...	...	...	...	...	
1167	-0.000069	-5.512316e-05	0.000142	6.568459e-05	0.000134	-0.000053	0.
482	-0.000005	2.712115e-08	-0.000085	2.916558e-07	-0.000162	0.000013	-0.
1433	-0.000136	1.856487e-04	0.000104	1.209905e-05	0.000052	-0.000037	-0.
397	-0.000015	4.001993e-05	0.000028	1.344847e-06	-0.000098	-0.000034	0.
922	-0.000027	1.632787e-05	-0.000027	-8.283993e-06	0.000052	0.000017	0.

1436 rows × 14 columns

In [24]:
```
infl.plot_influence()
```

Out[24]:



Index 80 has highest Cook's Distance

Finding Variance Inflation Factor (VIF)

In [25]:
```
vif = pd.DataFrame()
vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range
```

In [26]:
```
vif["features"] = X.columns
```

In [27]:
```
vif.round(1)
```

Out[27]:

	VIF Factor	features
0	16.4	Age_08_04
1	7.6	KM
2	64.9	HP
3	17.2	cc

	VIF Factor	features
4	21.9	Doors
5	438.6	Gears
6	11.0	Quarterly_Tax
7	543.1	Weight

As expected, Geras and Weight have a high variance inflation factor because they "explain" the same variance within this dataset. We would need to discard one of these variables to improve model and try to solve multicolinearity.

In [29]:
```
# Removed Weight from the dataframe and Tested the model
new_X = toyota_data_2[['Age_08_04','KM','HP','cc','Doors','Gears','Quarterl
```

In [30]:
```
new_model = sm.OLS(Y, new_X).fit()
new_predictions = new_model.predict(new_X)
new_model.summary()
```

Out[30]:

OLS Regression Results

Dep. Variable:	Price	**R-squared (uncentered):**	0.983
Model:	OLS	**Adj. R-squared (uncentered):**	0.983
Method:	Least Squares	**F-statistic:**	1.155e+04
Date:	Thu, 17 Feb 2022	**Prob (F-statistic):**	0.00
Time:	10:59:40	**Log-Likelihood:**	-12532.
No. Observations:	1436	**AIC:**	2.508e+04
Df Residuals:	1429	**BIC:**	2.512e+04
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Age_08_04	-132.0628	2.682	-49.245	0.000	-137.323	-126.802
KM	-0.0208	0.001	-14.947	0.000	-0.024	-0.018
HP	44.4711	3.027	14.692	0.000	38.533	50.409
cc	0.1805	0.099	1.827	0.068	-0.013	0.374
Doors	272.5298	41.159	6.621	0.000	191.791	353.269
Gears	2417.9083	81.331	29.729	0.000	2258.368	2577.449
Quarterly_Tax	17.0169	1.177	14.462	0.000	14.709	19.325

Omnibus:	184.883	**Durbin-Watson:**	1.396
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	583.580

Skew:	0.640	**Prob(JB):**	1.89e-127
Kurtosis:	5.849	**Cond. No.**	1.63e+05

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 1.63e+05. This might indicate that there are

As you can see, once we remove "Weight" from input variables and run the model again, all the variables are significant.

Final Model

In [34]:
```
# Removed Index with highest Cook's distance to remove the hightest influe
new_df = toyota_data_2.drop(toyota_data_2.index[80])
```

In [35]:
```
new_X = new_df[['Age_08_04','KM','HP','cc','Doors','Gears','Quarterly_Tax']
new_Y = new_df[['Price']]
```

In [36]:
```
final_model = sm.OLS(new_Y, new_X).fit()
predictions = final_model.predict(new_X)
final_model.summary()
```

Out[36]:

OLS Regression Results

Dep. Variable:	Price	**R-squared (uncentered):**	0.983
Model:	OLS	**Adj. R-squared (uncentered):**	0.983
Method:	Least Squares	**F-statistic:**	1.152e+04
Date:	Thu, 17 Feb 2022	**Prob (F-statistic):**	0.00
Time:	11:03:01	**Log-Likelihood:**	-12524.
No. Observations:	1435	**AIC:**	2.506e+04
Df Residuals:	1428	**BIC:**	2.510e+04
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Age_08_04	-132.0191	2.684	-49.196	0.000	-137.283	-126.755
KM	-0.0210	0.001	-14.581	0.000	-0.024	-0.018
HP	43.7530	3.287	13.310	0.000	37.304	50.202
cc	0.3468	0.313	1.109	0.268	-0.267	0.960

Doors	270.3889	41.346	6.540	0.000	189.284	351.494
Gears	2394.0486	91.807	26.077	0.000	2213.957	2574.140
Quarterly_Tax	16.4778	1.520	10.843	0.000	13.497	19.459

Omnibus:	183.937	**Durbin-Watson:**	1.393
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	582.178
Skew:	0.636	**Prob(JB):**	3.82e-127
Kurtosis:	5.849	**Cond. No.**	1.82e+05

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 1.82e+05. This might indicate that there are strong multicollinearity or other numerical problems.

1 - p-value < 0.01

Thus the model is accepted

2 - coefficient == -132.01 Thus if the value of x increased by 1, the predicted value of Price will decrease by 132.01

3 - Adj. R-sqared == 0.983 Thus the model explains 98.3% of the variance in dependent variable

In [ ]: