import pandas as pd import numpy as np import matplotlib.pyplot as plt %matplotlib inline from statsmodels.graphics.regressionplots import influence_plot

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.graphics.regressionplots import influence_plot
import statsmodels.formula.api as smf
import numpy as np
```

In [2]:
```
data = pd.read_csv("Toyoto_Corrola.csv")
```

In [3]:
```
data.head()
```

Out[3]:

	Id	Model	Price	Age_08_04	KM	HP	Doors	Cylinders	Gears	Weight
0	1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	46986	90	3	4	5	1165
1	2	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	72937	90	3	4	5	1165
2	3	ÊTOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	41711	90	3	4	5	1165
3	4	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	48000	90	3	4	5	1165
4	5	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	13750	30	38500	90	3	4	5	1170

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1436 entries, 0 to 1435
Data columns (total 10 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Id         1436 non-null   int64
 1   Model      1436 non-null   object
 2   Price      1436 non-null   int64
 3   Age_08_04  1436 non-null   int64
 4   KM         1436 non-null   int64
 5   HP         1436 non-null   int64
 6   Doors      1436 non-null   int64
 7   Cylinders  1436 non-null   int64
 8   Gears      1436 non-null   int64
 9   Weight     1436 non-null   int64
dtypes: int64(9), object(1)
memory usage: 112.3+ KB
```

In [5]:
```
# drop the id coloum
```

In [6]:
```
new = data.drop("Id",axis=1)
```

In [7]:
```
new
```

Out[7]:

	Model	Price	Age_08_04	KM	HP	Doors	Cylinders	Gears	Weight
0	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	46986	90	3	4	5	1165
1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	72937	90	3	4	5	1165
2	ÊTOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	41711	90	3	4	5	1165
3	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	48000	90	3	4	5	1165
4	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	13750	30	38500	90	3	4	5	1170
...	...	...	...	...	...	...	...	...	...
1431	TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors	7500	69	20544	86	3	4	5	1025
1432	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	10845	72	19000	86	3	4	5	1015
1433	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	8500	71	17016	86	3	4	5	1015
1434	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	7250	70	16916	86	3	4	5	1015
1435	TOYOTA Corolla 1.6 LB LINEA TERRA 4/5-Doors	6950	76	1	110	5	4	5	1114

1436 rows × 9 columns

In [8]:
```
new.isnull().sum()
```

Out[8]:
```
Model        0
Price        0
Age_08_04    0
KM           0
HP           0
Doors        0
Cylinders    0
Gears        0
Weight       0
dtype: int64
```

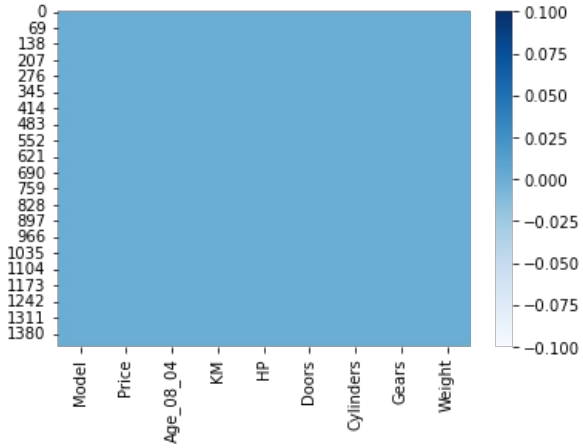
In [9]:
```
new.dtypes
```

Out[9]:
```
Model        object
Price         int64
Age_08_04     int64
KM            int64
HP            int64
Doors         int64
Cylinders     int64
Gears         int64
Weight        int64
dtype: object
```

In [10]:
```
import seaborn as sns
```

In [11]:
```
sns.heatmap(new.isnull(),cmap='Blues')
```

Out[11]:
```
<AxesSubplot:>
```



In [12]:
```
#by heat map there are no null values
```

In [13]:
```
new.corr()
```

Out[13]:

	Price	Age_08_04	KM	HP	Doors	Cylinders	Gears	Weight
Price	1.000000	-0.876590	-0.569960	0.314990	0.185326	NaN	0.063104	0.581198
Age_08_04	-0.876590	1.000000	0.505672	-0.156622	-0.148359	NaN	-0.005364	-0.470253
KM	-0.569960	0.505672	1.000000	-0.333538	-0.036197	NaN	0.015023	-0.028598
HP	0.314990	-0.156622	-0.333538	1.000000	0.092424	NaN	0.209477	0.089614
Doors	0.185326	-0.148359	-0.036197	0.092424	1.000000	NaN	-0.160141	0.302618
Cylinders	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Gears	0.063104	-0.005364	0.015023	0.209477	-0.160141	NaN	1.000000	0.020613
Weight	0.581198	-0.470253	-0.028598	0.089614	0.302618	NaN	0.020613	1.000000

In [14]:
```
new
```

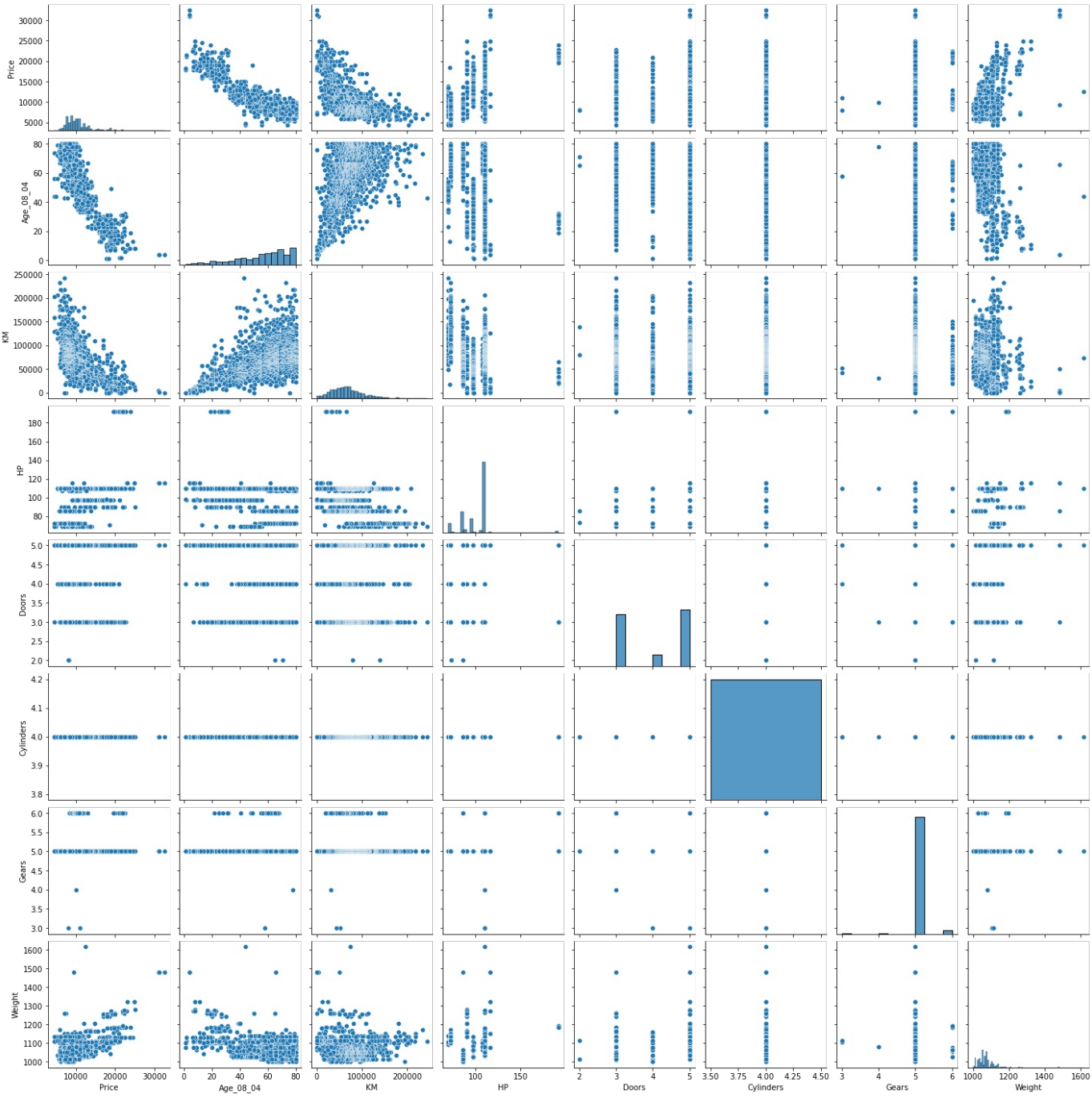
Out[14]:

	Model	Price	Age_08_04	KM	HP	Doors	Cylinders	Gears	Weight
0	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	46986	90	3	4	5	1165
1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	72937	90	3	4	5	1165
2	ÊTOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	41711	90	3	4	5	1165
3	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	48000	90	3	4	5	1165
4	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	13750	30	38500	90	3	4	5	1170
...	...	...	...	...	...	...	...	...	...
1431	TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors	7500	69	20544	86	3	4	5	1025
1432	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	10845	72	19000	86	3	4	5	1015
1433	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	8500	71	17016	86	3	4	5	1015
1434	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	7250	70	16916	86	3	4	5	1015
1435	TOYOTA Corolla 1.6 LB LINEA TERRA 4/5-Doors	6950	76	1	110	5	4	5	1114

1436 rows × 9 columns

In [15]:
```
sns.pairplot(new)
```

Out[15]:
```
<seaborn.axisgrid.PairGrid at 0x2b63877cee0>
```



In [16]:
```
new.duplicated()
```

Out[16]:
```
0       False
1       False
2       False
3       False
4       False
        ...
1431    False
1432    False
1433    False
1434    False
1435    False
Length: 1436, dtype: bool
```

In [17]:
```
# no duplicates
```

In [18]:
```
# we can use dummies
new = pd.get_dummies(new,columns=["Cylinders"])
```

In [19]:
```
new
```

Out[19]:

	Model	Price	Age_08_04	KM	HP	Doors	Gears	Weight	Cylinders_4
0	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	46986	90	3	5	1165	1
1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	72937	90	3	5	1165	1
2	ÊTOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	41711	90	3	5	1165	1
3	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	48000	90	3	5	1165	1
4	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	13750	30	38500	90	3	5	1170	1
...	...	...	...	...	...	...	...	...	...
1431	TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors	7500	69	20544	86	3	5	1025	1
1432	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	10845	72	19000	86	3	5	1015	1
1433	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	8500	71	17016	86	3	5	1015	1
1434	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	7250	70	16916	86	3	5	1015	1
1435	TOYOTA Corolla 1.6 LB LINEA TERRA 4/5-Doors	6950	76	1	110	5	5	1114	1

1436 rows × 9 columns

In [20]:
```
new.head()
```

Out[20]:

	Model	Price	Age_08_04	KM	HP	Doors	Gears	Weight	Cylinders_4
0	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	46986	90	3	5	1165	1
1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	72937	90	3	5	1165	1
2	ÊTOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	41711	90	3	5	1165	1
3	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	48000	90	3	5	1165	1
4	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	13750	30	38500	90	3	5	1170	1

In [21]:
```
import statsmodels.formula.api as smf
```

In [22]:
```
model = smf.ols("Price~Age_08_04+KM+HP+Doors+Gears+Weight+Cylinders_4",data=new).fit()
model.summary()
```

Out[22]:

OLS Regression Results

Dep. Variable:	Price	**R-squared:**	0.863
Model:	OLS	**Adj. R-squared:**	0.862
Method:	Least Squares	**F-statistic:**	1498.
Date:	Tue, 08 Mar 2022	**Prob (F-statistic):**	0.00
Time:	12:27:57	**Log-Likelihood:**	-12381.

No. Observations:	1436	**AIC:**	2.478e+04
Df Residuals:	1429	**BIC:**	2.481e+04
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-3492.1868	656.018	-5.323	0.000	-4779.048	-2205.325
Age_08_04	-122.2422	2.616	-46.729	0.000	-127.374	-117.111
KM	-0.0200	0.001	-16.543	0.000	-0.022	-0.018
HP	28.3501	2.615	10.842	0.000	23.221	33.479
Doors	-9.6802	39.992	-0.242	0.809	-88.129	68.769
Gears	622.2829	197.410	3.152	0.002	235.038	1009.528
Weight	18.6097	0.829	22.447	0.000	16.983	20.236
Cylinders_4	-3492.1868	656.018	-5.323	0.000	-4779.048	-2205.325

Omnibus:	199.596	**Durbin-Watson:**	1.564
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1569.510
Skew:	-0.381	**Prob(JB):**	0.00
Kurtosis:	8.065	**Cond. No.**	6.67e+20

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.97e-29. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

In [23]:
```
# here rsquared is > than 0.75
#but probaility of doors >0.05
```

simple linear regression

In [24]:
```
model_lr = smf.ols("Price~Doors",data = new).fit()
model_lr.summary()
```

Out[24]:

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:	Tue, 08 Mar 2022	**Prob (F-statistic):**	1.46e-12
Time:	12:27:57	**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

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

In [25]:
```
# here probability s less but r_sqr is not good
```

In [26]:
```
# remove the Doors
```

In [27]:
```
final = new.drop("Doors",axis=1)
```

In [28]:
```
final
```

Out[28]:

	Model	Price	Age_08_04	KM	HP	Gears	Weight	Cylinders_4
0	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	46986	90	5	1165	1
1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	72937	90	5	1165	1
2	ÊTOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	41711	90	5	1165	1
3	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	48000	90	5	1165	1
4	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	13750	30	38500	90	5	1170	1
...	...	...	...	...	...	...	...	...
1431	TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors	7500	69	20544	86	5	1025	1
1432	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	10845	72	19000	86	5	1015	1
1433	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	8500	71	17016	86	5	1015	1
1434	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	7250	70	16916	86	5	1015	1
1435	TOYOTA Corolla 1.6 LB LINEA TERRA 4/5-Doors	6950	76	1	110	5	1114	1

1436 rows × 8 columns

In [29]:
```
# then build the model without doors
```

In [30]:
```
model1 = smf.ols("Price~Age_08_04+KM+HP+Gears+Weight+Cylinders_4",data=new).fit()
model1.summary()
```

Out[30]:

OLS Regression Results

Dep. Variable:	Price	**R-squared:**	0.863
Model:	OLS	**Adj. R-squared:**	0.862
Method:	Least Squares	**F-statistic:**	1798.
Date:	Tue, 08 Mar 2022	**Prob (F-statistic):**	0.00
Time:	12:27:58	**Log-Likelihood:**	-12381.
No. Observations:	1436	**AIC:**	2.477e+04
Df Residuals:	1430	**BIC:**	2.480e+04
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-3503.4136	654.161	-5.356	0.000	-4786.632	-2220.196
Age_08_04	-122.2437	2.615	-46.745	0.000	-127.374	-117.114
KM	-0.0200	0.001	-16.551	0.000	-0.022	-0.018
HP	28.2829	2.599	10.882	0.000	23.184	33.381
Gears	631.5435	193.604	3.262	0.001	251.766	1011.321
Weight	18.5574	0.800	23.192	0.000	16.988	20.127
Cylinders_4	-3503.4136	654.161	-5.356	0.000	-4786.632	-2220.196

Omnibus:	198.155	**Durbin-Watson:**	1.564
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1550.366
Skew:	-0.376	**Prob(JB):**	0.00
Kurtosis:	8.034	**Cond. No.**	4.16e+19

Notes:

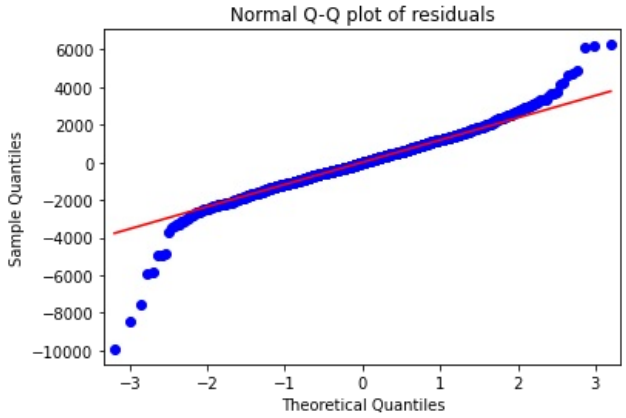
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 5.06e-27. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Residual Analysis

Test for normality of Residuals(Q-Q)plot

In [31]:
```
import statsmodels.api as sm
qqplot = sm.qqplot(model1.resid,line='q')
plt.title("Normal Q-Q plot of residuals")
plt.show()
```

```
C:\Users\rajesh\anaconda3\lib\site-packages\statsmodels\graphics\gofplots.py:993: UserWarning: marker is redundan
tly defined by the 'marker' keyword argument and the fmt string "bo" (-> marker='o'). The keyword argument will t
ake precedence.
  ax.plot(x, y, fmt, **plot_style)
```

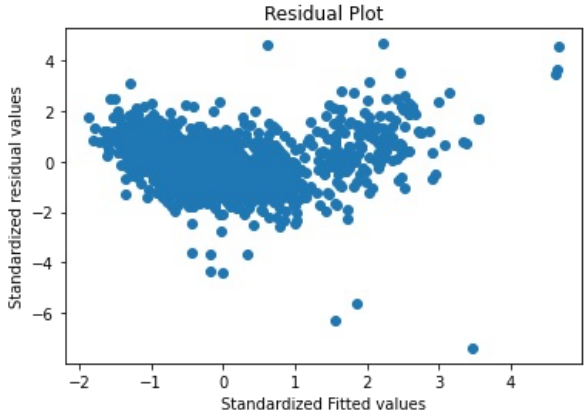


Residual Plot for Homoscedasticity

In [32]:
```
def get_standardized_values(vals):
    return (vals - vals.mean())/vals.std()
```

In [33]:
```
plt.scatter(get_standardized_values(model1.fittedvalues),
            get_standardized_values(model1.resid))

plt.title('Residual Plot')
plt.xlabel('Standardized Fitted values')
plt.ylabel('Standardized residual values')
plt.show()
```



Model Deletion Diagnostics

Detecting Influencers/Outliers

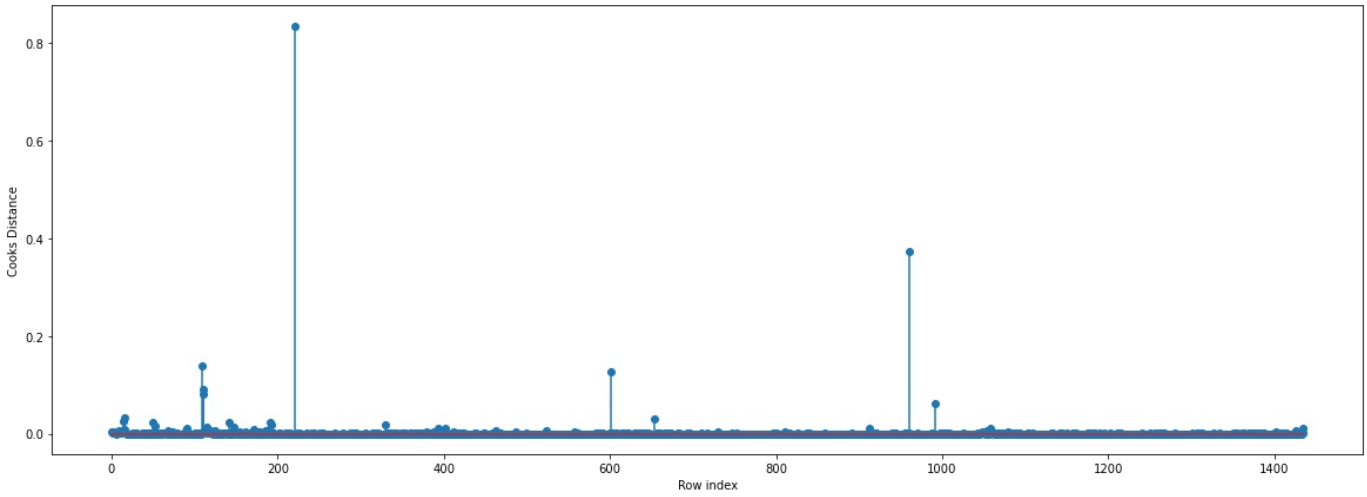
Cook's Distance

In [34]:
```
model_influence = model.get_influence()
(c, _) = model_influence.cooks_distance
```

In [35]:
```
K=8
N=1436
3*(K+1)/N
```

Out[35]:
```
0.018802228412256268
```

In [36]:
```
#Plot the influencers values using stem plot
fig = plt.figure(figsize=(20, 7))
plt.stem(np.arange(len(final)), np.round(c, 3))
plt.xlabel('Row index')
plt.ylabel('Cooks Distance')
plt.show()
```

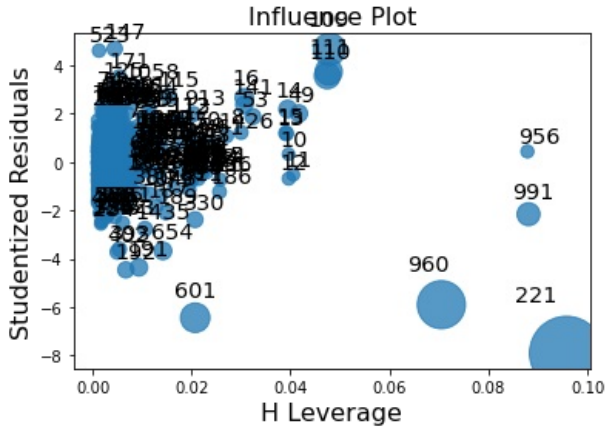


In [37]:
```
(np.argmax(c),np.max(c))
```

Out[37]:
```
(221, 0.8359007508851538)
```

High Influence points

In [38]:
```
from statsmodels.graphics.regressionplots import influence_plot
influence_plot(model1)
plt.show()
```



In [39]:
```
k = final.shape[1]
n = final.shape[0]
leverage_cutoff = 3*((k + 1)/n)
```

In [40]:
```
# from the above 221and 111 are high influencers
```

In [41]:
```
final[final.index.isin([221, 111])]
```

Out[41]:

	Model	Price	Age_08_04	KM	HP	Gears	Weight	Cylinders_4
111	TOYOTA Corolla VERSO 2.0 D4D SOL (7) MPV	31275	4	1500	116	5	1480	1
221	TOYOTA Corolla 1.6 HB LINEA SOL 4/5-Doors	12450	44	74172	110	5	1615	1

In [42]:
```
final.head()
```

Out[42]:

	Model	Price	Age_08_04	KM	HP	Gears	Weight	Cylinders_4
0	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	46986	90	5	1165	1
1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	72937	90	5	1165	1
2	ÊTOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	41711	90	5	1165	1
3	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	48000	90	5	1165	1
4	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	13750	30	38500	90	5	1170	1

Improving the model

In [43]:
```
toyota = pd.read_csv("Toyoto_Corrola.csv")
```

In [44]:
```
#Discard the data points which are influencers and reasign the row number (reset_index())
toyota1=toyota.drop(toyota.index[[221,111]],axis=0).reset_index()
```

In [45]:
```
#Drop the original index
toyota1=toyota1.drop(['index'],axis=1)
```

In [46]:
```
toyota1
```

Out[46]:

	Id	Model	Price	Age_08_04	KM	HP	Doors	Cylinders	Gears	Weight
0	1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	46986	90	3	4	5	1165
1	2	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	72937	90	3	4	5	1165
2	3	ÊTOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	41711	90	3	4	5	1165
3	4	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	48000	90	3	4	5	1165
4	5	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	13750	30	38500	90	3	4	5	1170
...	...	...	...	...	...	...	...	...	...	...
1429	1438	TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors	7500	69	20544	86	3	4	5	1025
1430	1439	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	10845	72	19000	86	3	4	5	1015
1431	1440	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	8500	71	17016	86	3	4	5	1015
1432	1441	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	7250	70	16916	86	3	4	5	1015
1433	1442	TOYOTA Corolla 1.6 LB LINEA TERRA 4/5-Doors	6950	76	1	110	5	4	5	1114

1434 rows × 10 columns

In [47]:
```
# drop the id
```

In [48]:
```
t1 = toyota1.drop("Id",axis=1)
```

In [49]:
```
t1
```

Out[49]:

	Model	Price	Age_08_04	KM	HP	Doors	Cylinders	Gears	Weight
0	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	46986	90	3	4	5	1165
1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	72937	90	3	4	5	1165
2	ÊTOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	41711	90	3	4	5	1165
3	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	48000	90	3	4	5	1165
4	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	13750	30	38500	90	3	4	5	1170
...	...	...	...	...	...	...	...	...	...
1429	TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors	7500	69	20544	86	3	4	5	1025
1430	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	10845	72	19000	86	3	4	5	1015
1431	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	8500	71	17016	86	3	4	5	1015
1432	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	7250	70	16916	86	3	4	5	1015
1433	TOYOTA Corolla 1.6 LB LINEA TERRA 4/5-Doors	6950	76	1	110	5	4	5	1114

1434 rows × 9 columns

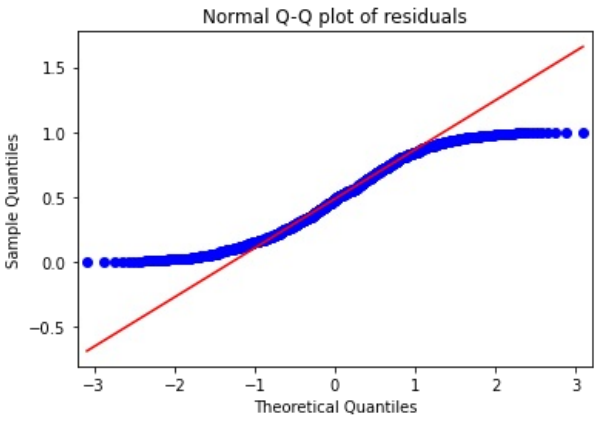
Predicting the new data

In [50]:
```
new_data=pd.DataFrame({'Age_08_04':40,'KM':95,'Doors':90,'Cylinders':35,'Gears':15,'Weight':35},index=[1])
```

In [51]:
```
data=np.random.uniform(0,1,1000)

import statsmodels.api as sm
qqplot=sm.qqplot(data,line='q') # line = 45 to draw the diagnoal line
plt.title("Normal Q-Q plot of residuals")
plt.show()
```

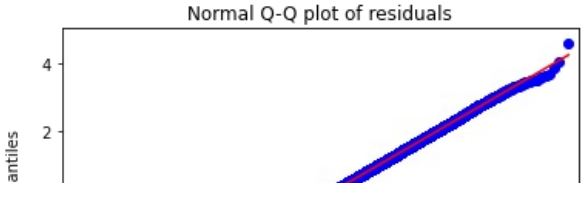
```
C:\Users\rajesh\anaconda3\lib\site-packages\statsmodels\graphics\gofplots.py:993: UserWarning: marker is redundan
tly defined by the 'marker' keyword argument and the fmt string "bo" (-> marker='o'). The keyword argument will t
ake precedence.
  ax.plot(x, y, fmt, **plot_style)
```

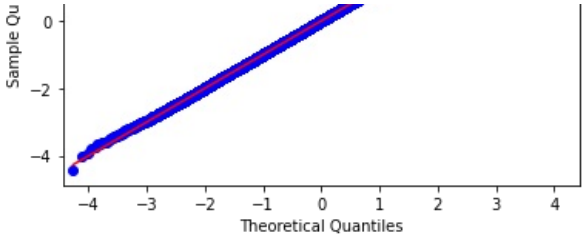


In [52]:
```
data=np.random.normal(0,1,100000)

import statsmodels.api as sm
qqplot=sm.qqplot(data,line='q') # line = 45 to draw the diagnoal line
plt.title("Normal Q-Q plot of residuals")
plt.show()
```

```
C:\Users\rajesh\anaconda3\lib\site-packages\statsmodels\graphics\gofplots.py:993: UserWarning: marker is redundan
tly defined by the 'marker' keyword argument and the fmt string "bo" (-> marker='o'). The keyword argument will t
ake precedence.
  ax.plot(x, y, fmt, **plot_style)
```





In [ ]: