In [21]:

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
import statsmodels.formula.api as smf
```

In [5]:

```
import warnings
warnings.filterwarnings("ignore")
```

In [6]:

```
car=pd.read_csv(r"C:\Users\arnak\Downloads\Cars.csv")
```

In [7]:

```
car
```

Out[7]:

	HP	MPG	VOL	SP	WT
0	49	53.700681	89	104.185353	28.762059
1	55	50.013401	92	105.461264	30.466833
2	55	50.013401	92	105.461264	30.193597
3	70	45.696322	92	113.461264	30.632114
4	53	50.504232	92	104.461264	29.889149
...	...	...	...	...	...
76	322	36.900000	50	169.598513	16.132947
77	238	19.197888	115	150.576579	37.923113
78	263	34.000000	50	151.598513	15.769625
79	295	19.833733	119	167.944460	39.423099
80	236	12.101263	107	139.840817	34.948615

81 rows × 5 columns

In [8]:

```
car.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 81 entries, 0 to 80
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   HP      81 non-null     int64  
 1   MPG     81 non-null     float64
 2   VOL     81 non-null     int64  
 3   SP      81 non-null     float64
 4   WT      81 non-null     float64
dtypes: float64(3), int64(2)
memory usage: 3.3 KB
```

In [9]:

```
car.isnull().sum()
```

```
Out[9]:  HP      0
         MPG     0
         VOL     0
         SP      0
         WT      0
         dtype: int64
```

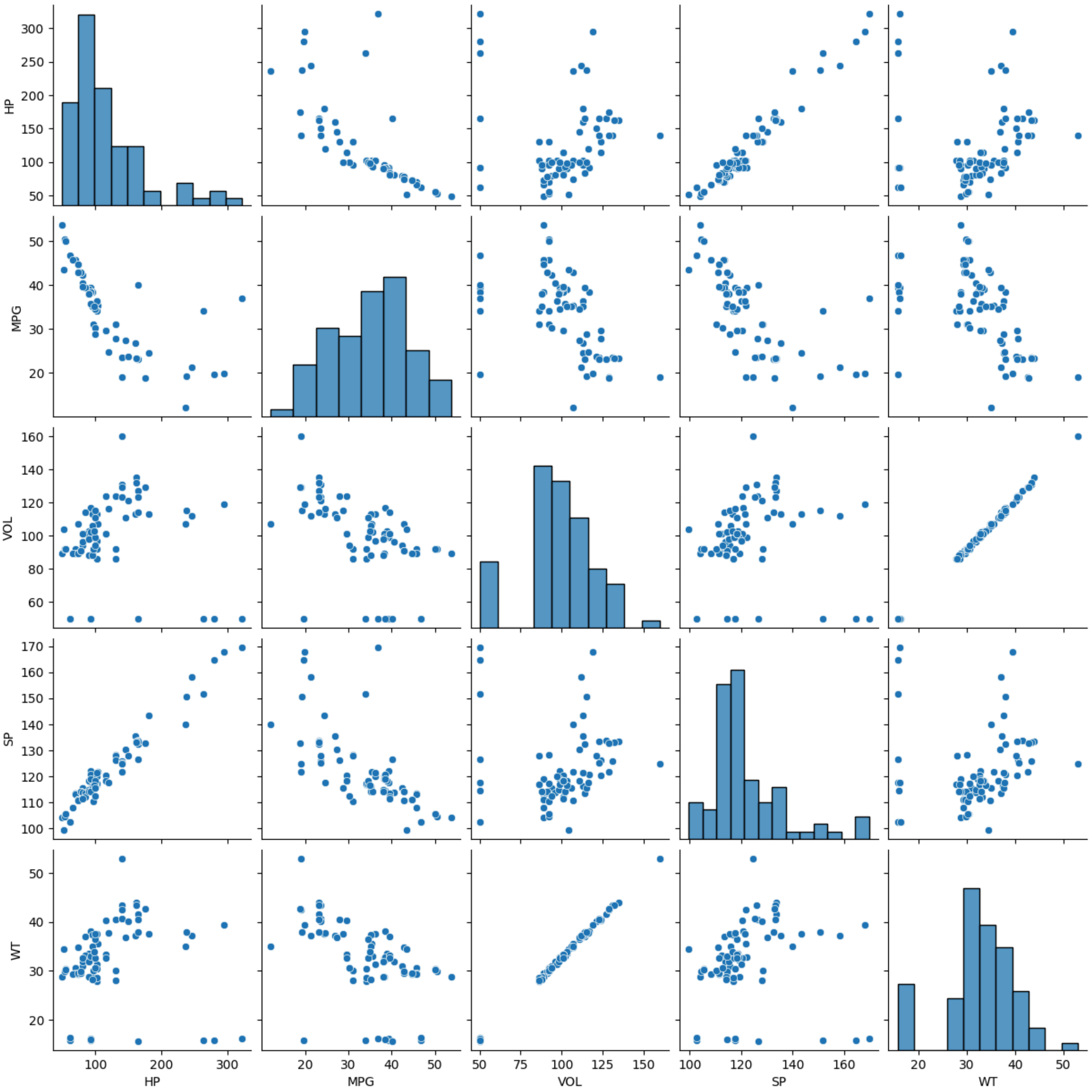
```
In [10]: car.corr()
```

Out[10]:

	HP	MPG	VOL	SP	WT
HP	1.000000	-0.725038	0.077459	0.973848	0.076513
MPG	-0.725038	1.000000	-0.529057	-0.687125	-0.526759
VOL	0.077459	-0.529057	1.000000	0.102170	0.999203
SP	0.973848	-0.687125	0.102170	1.000000	0.102439
WT	0.076513	-0.526759	0.999203	0.102439	1.000000

```
In [11]: sns.pairplot(car)
```

```
Out[11]: <seaborn.axisgrid.PairGrid at 0x2186f9d31d0>
```



```
In [12]: import statsmodels.formula.api as smf
```

```
In [13]: car.columns
```

```
Out[13]: Index(['HP', 'MPG', 'VOL', 'SP', 'WT'], dtype='object')
```

```
In [14]: modal=smf.ols('MPG~WT+VOL+SP+HP',data=car).fit()
```

```
In [15]: modal.params
```

```
Out[15]: Intercept    30.677336
         WT            0.400574
         VOL          -0.336051
         SP            0.395627
         HP           -0.205444
         dtype: float64
```

```
In [16]: print(modal.tvalues,modal.pvalues)
```

```
Intercept    2.058841
WT           0.236541
VOL         -0.590970
SP           2.499880
HP          -5.238735
dtype: float64 Intercept    0.042936
WT           0.813649
VOL          0.556294
SP           0.014579
HP           0.000001
dtype: float64
```

```
In [17]: print(modal.rsquared,modal.rsquared_adj)
```

```
0.7705372737359844 0.7584602881431415
```

```
In [18]: modal.summary()
```

Out[18]:

OLS Regression Results

Dep. Variable:	MPG	**R-squared:**	0.771
Model:	OLS	**Adj. R-squared:**	0.758
Method:	Least Squares	**F-statistic:**	63.80
Date:	Tue, 29 Oct 2024	**Prob (F-statistic):**	1.54e-23
Time:	02:23:40	**Log-Likelihood:**	-233.96
No. Observations:	81	**AIC:**	477.9
Df Residuals:	76	**BIC:**	489.9
Df Model:	4		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	30.6773	14.900	2.059	0.043	1.001	60.354
WT	0.4006	1.693	0.237	0.814	-2.972	3.773
VOL	-0.3361	0.569	-0.591	0.556	-1.469	0.796
SP	0.3956	0.158	2.500	0.015	0.080	0.711
HP	-0.2054	0.039	-5.239	0.000	-0.284	-0.127

Omnibus:	10.780	**Durbin-Watson:**	1.403
Prob(Omnibus):	0.005	**Jarque-Bera (JB):**	11.722
Skew:	0.707	**Prob(JB):**	0.00285
Kurtosis:	4.215	**Cond. No.**	6.09e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.09e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

IRM

In [24]:

```
ml_v=smf.ols('MPG~VOL',data=car).fit()
```

In [26]:

```
print(ml_v.tvalues,ml_v.pvalues)
```

```
Intercept    14.106056
VOL          -5.541400
dtype: float64 Intercept    2.753815e-23
VOL          3.822819e-07
dtype: float64
```

```
In [28]: ml_W=smf.ols('MPG~WT',data=car).fit()
```

```
In [30]: print(ml_W.tvalues,ml_W.pvalues)
```

```
Intercept    14.248923
WT           -5.508067
dtype: float64 Intercept    1.550788e-23
WT           4.383467e-07
dtype: float64
```

```
In [31]: ml_vw=smf.ols('MPG~WT+VOL',data=car).fit()
```

```
In [32]: print(ml_vw.tvalues,ml_vw.pvalues)
```

```
Intercept    12.545736
WT            0.489876
VOL          -0.709604
dtype: float64 Intercept    2.141975e-20
WT           6.255966e-01
VOL          4.800657e-01
dtype: float64
```

Calculating VIF

```
In [33]: car.columns
```

```
Out[33]: Index(['HP', 'MPG', 'VOL', 'SP', 'WT'], dtype='object')
```

```
In [36]: rsq_hp=smf.ols("HP~VOL+SP+WT",data=car).fit().rsquared
         vif_hp=1/(1-rsq_hp)
```

```
In [37]: print(rsq_hp,vif_hp)
```

```
0.9498157963084058 19.92658897499852
```

```
In [38]: rsq_VOL=smf.ols("VOL~HP+SP+WT",data=car).fit().rsquared
         vif_VOL=1/(1-rsq_VOL)
```

```
In [39]: print(rsq_VOL,vif_VOL)
```

```
0.9984345797174133 638.8060836592878
```

```
In [42]: rsq_WT=smf.ols("WT~HP+SP+VOL",data=car).fit().rsquared
         vif_WT=1/(1-rsq_WT)
```

```
In [43]: rsq_SP=smf.ols("SP~HP+VOL+WT",data=car).fit().rsquared
         vif_SP=1/(1-rsq_SP)
```

```
In [44]: print(rsq_WT,vif_WT)
         print(rsq_SP,vif_SP)
```

```
0.9984363610296332 639.5338175572624
0.9500190896665341 20.00763878305008
```

```
In [46]: d1={'Variables':['SP','HP','VOL','WT'],'VIF':[vif_hp,vif_SP,vif_VOL,vif_WT]}
```

```
In [47]: d1
```

Out[47]:
```
{'Variables': ['SP', 'HP', 'VOL', 'WT'],
 'VIF': [19.92658897499852,
  20.00763878305008,
  638.8060836592878,
  639.5338175572624]}
```

In [49]:
```
vif_frame=pd.DataFrame(d1)
vif_frame
```

Out[49]:

	Variables	VIF
0	SP	19.926589
1	HP	20.007639
2	VOL	638.806084
3	WT	639.533818

Subset Selection

ALC VALUE

In [50]:
```
modal=smf.ols('MPG~WT+SP+HP',data=car).fit()
print(modal.rsquared,modal.aic)
```

```
0.7694828139983461 476.2992750152976
```

In [65]:
```
modal=smf.ols('MPG~VOL+SP+HP',data=car).fit()
print(modal.rsquared,modal.aic)
```

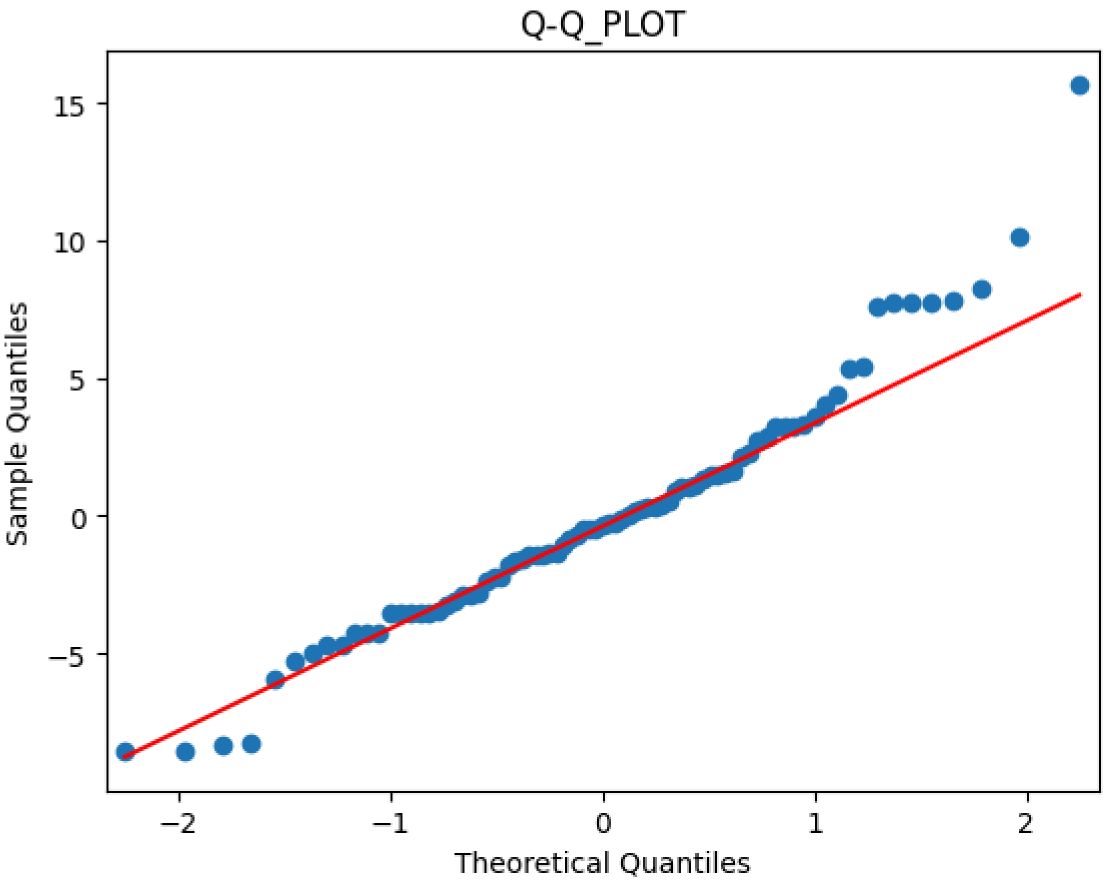
```
0.770368341321302 475.9875158854609
```

In [52]:
```
import statsmodels.api as sm
```

In [ ]:
```
modal=smf.ols('MPG~VOL+SP+HP',data=car).fit()
```

In [66]:
```
qqplot=sm.qqplot(modal.resid,line='q')
plt.title("Q-Q_PLOT")
```

Out[66]:
```
Text(0.5, 1.0, 'Q-Q_PLOT')
```



```
In [ ]:
```

```
In [63]: list(np.where(modal.resid>10))
```

```
Out[63]: [array([ 0, 76], dtype=int64)]
```

```
In [67]: print(modal.rsquared,modal.rsquared_adj)
```

```
0.770368341321302 0.7614216533208333
```

MODEL DELETION TECHNIQUES

```
In [69]: modal
```

```
Out[69]: <statsmodels.regression.linear_model.RegressionResultsWrapper at 0x2187616dcd0>
```

```
In [71]: inf=modal.get_influence()
```

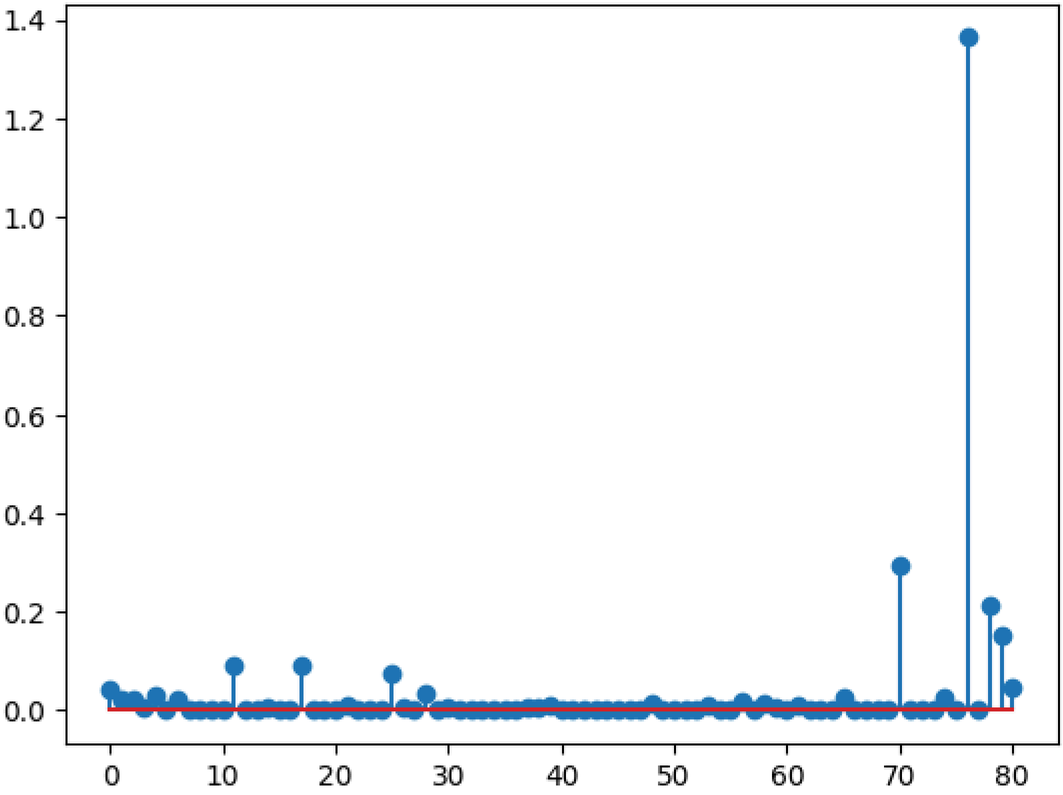
```
In [72]: c,p=inf.cooks_distance
```

```
In [73]: c
```

```
Out[73]: array([4.43781421e-02, 2.31439849e-02, 2.31439849e-02, 5.50743307e-03,
               2.84029117e-02, 3.89961849e-03, 2.31439849e-02, 3.39659293e-03,
               3.39659293e-03, 9.67532550e-04, 3.00465895e-03, 9.32152031e-02,
               2.65177317e-04, 3.00465895e-03, 5.41784561e-03, 3.00465895e-03,
               8.22731925e-04, 9.32152031e-02, 8.22731925e-04, 3.21833541e-04,
               2.70620733e-05, 8.34770054e-03, 1.83348025e-05, 2.72860299e-04,
               1.23307010e-03, 7.41312614e-02, 5.71759163e-03, 2.70620733e-05,
               3.33387970e-02, 9.21393948e-05, 5.93913831e-03, 1.23367282e-03,
               2.67864467e-04, 1.10555542e-03, 1.21312479e-03, 2.01774924e-05,
               4.19374936e-04, 4.18657710e-03, 4.18657710e-03, 1.12077324e-02,
               2.70424384e-04, 3.80042521e-06, 1.99290460e-03, 2.64674273e-03,
               2.22196543e-05, 4.76293133e-04, 2.58868946e-05, 4.92204073e-05,
               1.55796817e-02, 3.53143210e-03, 3.53143210e-03, 3.53143210e-03,
               2.30754944e-05, 1.04055062e-02, 1.07199598e-03, 2.89811901e-04,
               1.71984015e-02, 3.53143210e-03, 1.52234712e-02, 4.17948710e-03,
               1.68557542e-03, 1.20082441e-02, 1.39544913e-03, 3.73211476e-03,
               2.10575197e-03, 2.58577636e-02, 2.77621422e-03, 2.35657828e-04,
               8.06320733e-04, 2.51977732e-03, 2.93197253e-01, 2.03749246e-04,
               1.03287635e-06, 3.08782933e-04, 2.44806249e-02, 3.93356096e-03,
               1.36417439e+00, 1.86353051e-03, 2.14967561e-01, 1.52427254e-01,
               4.78231303e-02])
```

```
In [75]: plt.stem(c)
```

```
Out[75]: <StemContainer object of 3 artists>
```



```
In [76]: np.argmax(c),np.max(c)
```

```
Out[76]: (76, 1.364174386910332)
```

```
In [78]: car.iloc[76]
```

Out[78]:
```
HP      322.000000
MPG      36.900000
VOL      50.000000
SP      169.598513
WT       16.132947
Name: 76, dtype: float64
```

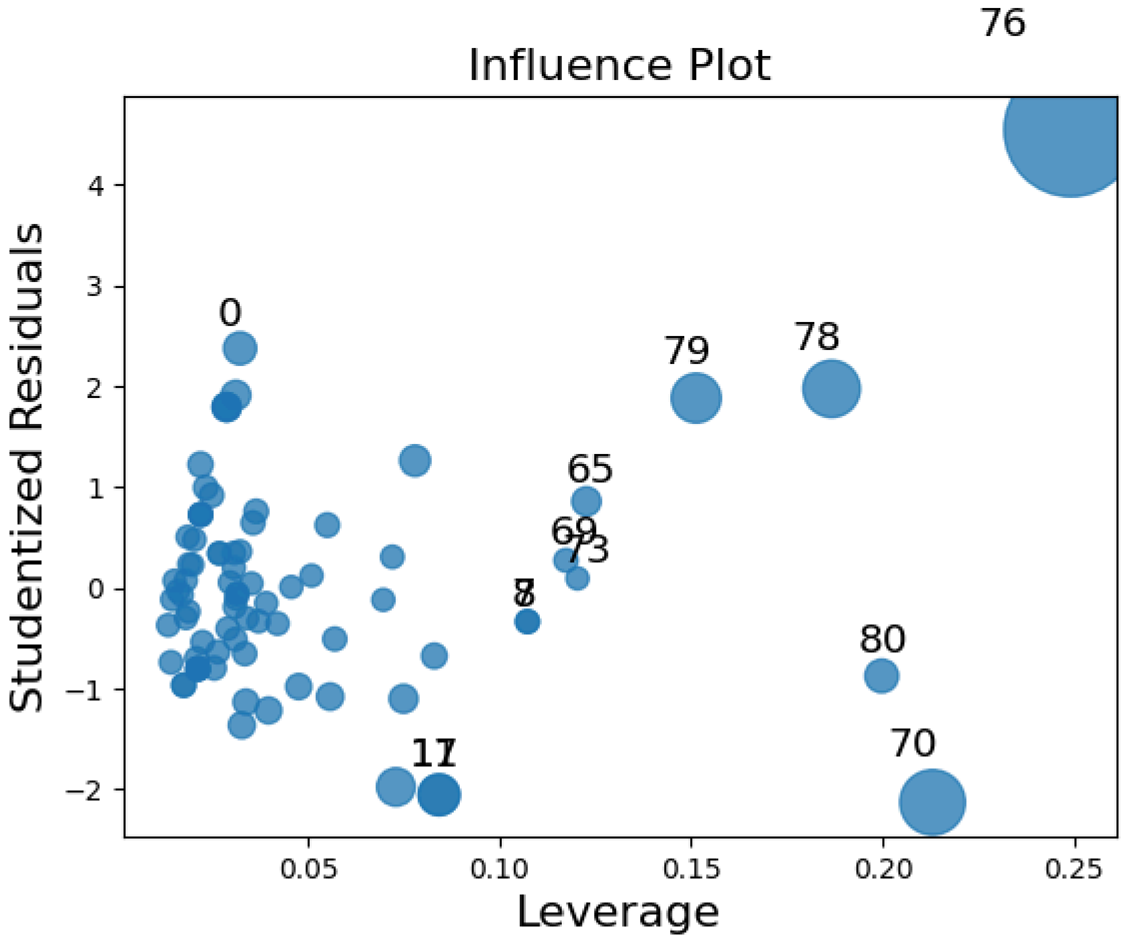
In [79]:
```
car.head()
```

Out[79]:

	HP	MPG	VOL	SP	WT
0	49	53.700681	89	104.185353	28.762059
1	55	50.013401	92	105.461264	30.466833
2	55	50.013401	92	105.461264	30.193597
3	70	45.696322	92	113.461264	30.632114
4	53	50.504232	92	104.461264	29.889149

In [80]:
```
from statsmodels.graphics.regressionplots import influence_plot
```

In [83]:
```
influence_plot(modal);
```



In [84]:
```
car
```

Out[84]:

	HP	MPG	VOL	SP	WT
0	49	53.700681	89	104.185353	28.762059
1	55	50.013401	92	105.461264	30.466833
2	55	50.013401	92	105.461264	30.193597
3	70	45.696322	92	113.461264	30.632114
4	53	50.504232	92	104.461264	29.889149
...	...	...	...	...	...
76	322	36.900000	50	169.598513	16.132947
77	238	19.197888	115	150.576579	37.923113
78	263	34.000000	50	151.598513	15.769625
79	295	19.833733	119	167.944460	39.423099
80	236	12.101263	107	139.840817	34.948615

81 rows × 5 columns

In [87]:

```
car.drop(76,inplace=True)
```

In [89]:

```
final_model=smf.ols('MPG~HP+SP+VOL',data=car).fit()
```

In [90]:

```
final_model.rsquared
```

Out[90]:

```
0.8192122305013385
```

In [ ]:

```

```