In [1]:

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

In [2]:

```
data = pd.read_csv('50_Startups.csv')
data
```

Loading [MathJax]/extensions/Safe.js

Out[2]:

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94
5	131876.90	99814.71	362861.36	New York	156991.12
6	134615.46	147198.87	127716.82	California	156122.51
7	130298.13	145530.06	323876.68	Florida	155752.60
8	120542.52	148718.95	311613.29	New York	152211.77
9	123334.88	108679.17	304981.62	California	149759.96
10	101913.08	110594.11	229160.95	Florida	146121.95
11	100671.96	91790.61	249744.55	California	144259.40
12	93863.75	127320.38	249839.44	Florida	141585.52
13	91992.39	135495.07	252664.93	California	134307.35
14	119943.24	156547.42	256512.92	Florida	132602.65
15	114523.61	122616.84	261776.23	New York	129917.04
16	78013.11	121597.55	264346.06	California	126992.93
17	94657.16	145077.58	282574.31	New York	125370.37
18	91749.16	114175.79	294919.57	Florida	124266.90
19	86419.70	153514.11	0.00	New York	122776.86
20	76253.86	113867.30	298664.47	California	118474.03
21	78389.47	153773.43	299737.29	New York	111313.02
22	73994.56	122782.75	303319.26	Florida	110352.25
23	67532.53	105751.03	304768.73	Florida	108733.99
24	77044.01	99281.34	140574.81	New York	108552.04
25	64664.71	139553.16	137962.62	California	107404.34
26	75328.87	144135.98	134050.07	Florida	105733.54
27	72107.60	127864.55	353183.81	New York	105008.31
28	66051.52	182645.56	118148.20	Florida	103282.38
29	65605.48	153032.06	107138.38	New York	101004.64
30	61994.48	115641.28	91131.24	Florida	99937.59
31	61136.38	152701.92	88218.23	New York	97483.56
32	63408.86	129219.61	46085.25	California	97427.84
33	55493.95	103057.49	214634.81	Florida	96778.92
34	46426.07	157693.92	210797.67	California	96712.80
35	46014.02	85047.44	205517.64	New York	96479.51
36	28663.76	127056.21	201126.82	Florida	90708.19
37	44069.95	51283.14	197029.42	California	89949.14
38	20229.59	65947.93	185265.10	New York	81229.06

Loading [MathJax]/extensions/Safe.js

	R&D Spend	Administration	Marketing Spend	State	Profit
39	38558.51	82982.09	174999.30	California	81005.76
40	28754.33	118546.05	172795.67	California	78239.91
41	27892.92	84710.77	164470.71	Florida	77798.83
42	23640.93	96189.63	148001.11	California	71498.49
43	15505.73	127382.30	35534.17	New York	69758.98
44	22177.74	154806.14	28334.72	California	65200.33
45	1000.23	124153.04	1903.93	New York	64926.08
46	1315.46	115816.21	297114.46	Florida	49490.75
47	0.00	135426.92	0.00	California	42559.73
48	542.05	51743.15	0.00	New York	35673.41
49	0.00	116983.80	45173.06	California	14681.40

In [3]:

```
data.describe()
```

Out[3]:

	R&D Spend	Administration	Marketing Spend	Profit
count	50.000000	50.000000	50.000000	50.000000
mean	73721.615600	121344.639600	211025.097800	112012.639200
std	45902.256482	28017.802755	122290.310726	40306.180338
min	0.000000	51283.140000	0.000000	14681.400000
25%	39936.370000	103730.875000	129300.132500	90138.902500
50%	73051.080000	122699.795000	212716.240000	107978.190000
75%	101602.800000	144842.180000	299469.085000	139765.977500
max	165349.200000	182645.560000	471784.100000	192261.830000

In [4]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   R&D Spend        50 non-null     float64
 1   Administration   50 non-null     float64
 2   Marketing Spend  50 non-null     float64
 3   State            50 non-null     object 
 4   Profit           50 non-null     float64
dtypes: float64(4), object(1)
memory usage: 2.1+ KB
```

In [5]:

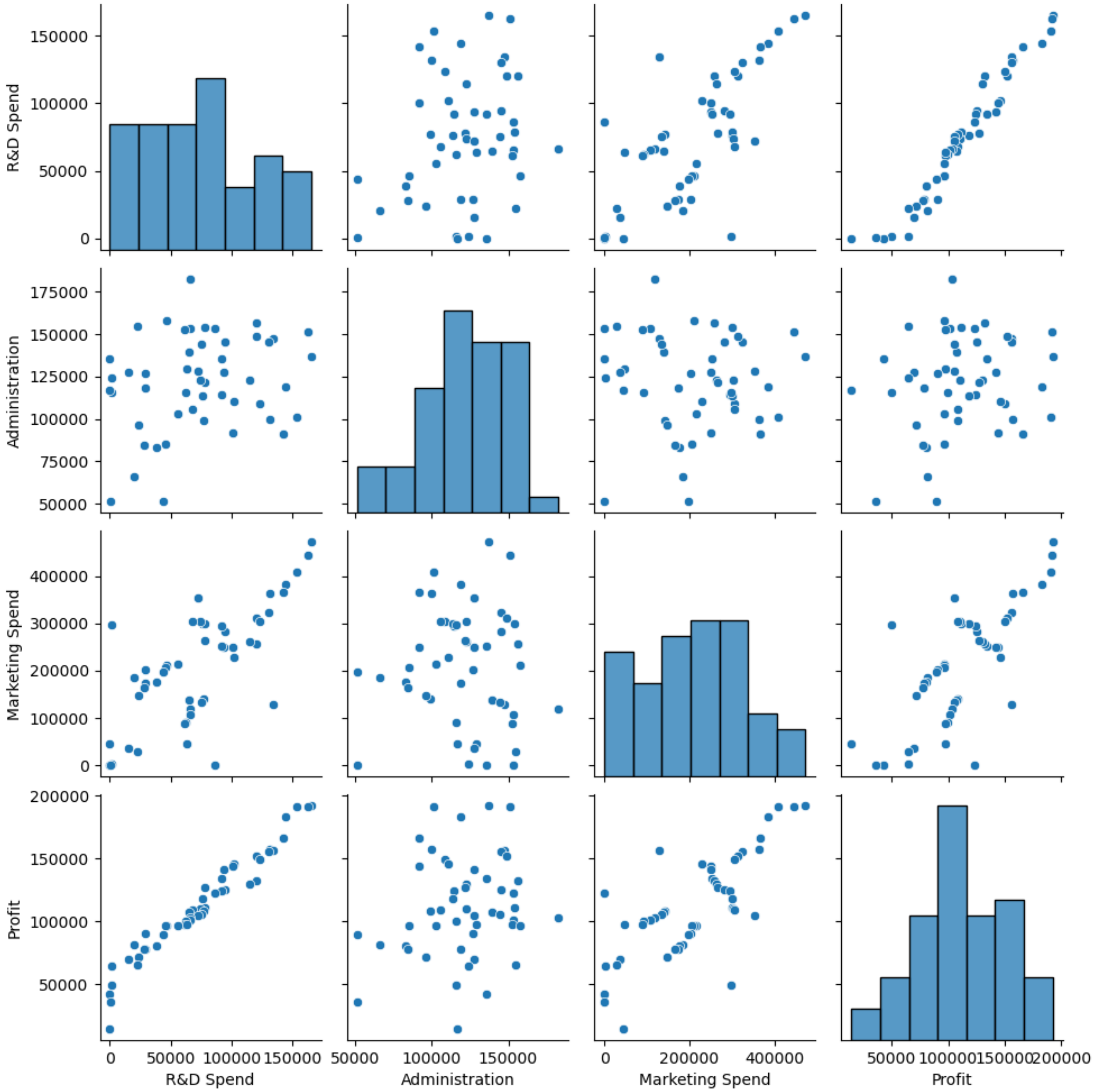
```
data.corr()
```

Out[5]:

	R&D Spend	Administration	Marketing Spend	Profit
R&D Spend	1.000000	0.241955	0.724248	0.972900
Administration	0.241955	1.000000	-0.032154	0.200717
Marketing Spend	0.724248	-0.032154	1.000000	0.747766
Profit	0.972900	0.200717	0.747766	1.000000

In [6]:
```
sns.pairplot(data)
```

Out[6]:
```
<seaborn.axisgrid.PairGrid at 0x1725631b670>
```



In [7]:
```
sns.distplot(data['Profit'])
```

```
C:\Users\ROHIT\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[7]:
```
<AxesSubplot:xlabel='Profit', ylabel='Density'>
```

In [8]:
```
data = data.rename({'R&D Spend':'RD_spend','Marketing Spend':'Marketing_Spend'},axis=1)
data
```

Loading [MathJax]/extensions/Safe.js

Out[8]:

	RD_spend	Administration	Marketing_Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94
5	131876.90	99814.71	362861.36	New York	156991.12
6	134615.46	147198.87	127716.82	California	156122.51
7	130298.13	145530.06	323876.68	Florida	155752.60
8	120542.52	148718.95	311613.29	New York	152211.77
9	123334.88	108679.17	304981.62	California	149759.96
10	101913.08	110594.11	229160.95	Florida	146121.95
11	100671.96	91790.61	249744.55	California	144259.40
12	93863.75	127320.38	249839.44	Florida	141585.52
13	91992.39	135495.07	252664.93	California	134307.35
14	119943.24	156547.42	256512.92	Florida	132602.65
15	114523.61	122616.84	261776.23	New York	129917.04
16	78013.11	121597.55	264346.06	California	126992.93
17	94657.16	145077.58	282574.31	New York	125370.37
18	91749.16	114175.79	294919.57	Florida	124266.90
19	86419.70	153514.11	0.00	New York	122776.86
20	76253.86	113867.30	298664.47	California	118474.03
21	78389.47	153773.43	299737.29	New York	111313.02
22	73994.56	122782.75	303319.26	Florida	110352.25
23	67532.53	105751.03	304768.73	Florida	108733.99
24	77044.01	99281.34	140574.81	New York	108552.04
25	64664.71	139553.16	137962.62	California	107404.34
26	75328.87	144135.98	134050.07	Florida	105733.54
27	72107.60	127864.55	353183.81	New York	105008.31
28	66051.52	182645.56	118148.20	Florida	103282.38
29	65605.48	153032.06	107138.38	New York	101004.64
30	61994.48	115641.28	91131.24	Florida	99937.59
31	61136.38	152701.92	88218.23	New York	97483.56
32	63408.86	129219.61	46085.25	California	97427.84
33	55493.95	103057.49	214634.81	Florida	96778.92
34	46426.07	157693.92	210797.67	California	96712.80
35	46014.02	85047.44	205517.64	New York	96479.51
36	28663.76	127056.21	201126.82	Florida	90708.19
37	44069.95	51283.14	197029.42	California	89949.14
38	20229.59	65947.93	185265.10	New York	81229.06

Loading [MathJax]/extensions/Safe.js

	RD_spend	Administration	Marketing_Spend	State	Profit
39	38558.51	82982.09	174999.30	California	81005.76
40	28754.33	118546.05	172795.67	California	78239.91
41	27892.92	84710.77	164470.71	Florida	77798.83
42	23640.93	96189.63	148001.11	California	71498.49
43	15505.73	127382.30	35534.17	New York	69758.98
44	22177.74	154806.14	28334.72	California	65200.33
45	1000.23	124153.04	1903.93	New York	64926.08
46	1315.46	115816.21	297114.46	Florida	49490.75
47	0.00	135426.92	0.00	California	42559.73
48	542.05	51743.15	0.00	New York	35673.41
49	0.00	116983.80	45173.06	California	14681.40

In [9]:

```
data.drop('State',axis=1)
```

Out[9]:

	RD_spend	Administration	Marketing_Spend	Profit
0	165349.20	136897.80	471784.10	192261.83
1	162597.70	151377.59	443898.53	191792.06
2	153441.51	101145.55	407934.54	191050.39
3	144372.41	118671.85	383199.62	182901.99
4	142107.34	91391.77	366168.42	166187.94
5	131876.90	99814.71	362861.36	156991.12
6	134615.46	147198.87	127716.82	156122.51
7	130298.13	145530.06	323876.68	155752.60
8	120542.52	148718.95	311613.29	152211.77
9	123334.88	108679.17	304981.62	149759.96
10	101913.08	110594.11	229160.95	146121.95
11	100671.96	91790.61	249744.55	144259.40
12	93863.75	127320.38	249839.44	141585.52
13	91992.39	135495.07	252664.93	134307.35
14	119943.24	156547.42	256512.92	132602.65
15	114523.61	122616.84	261776.23	129917.04
16	78013.11	121597.55	264346.06	126992.93
17	94657.16	145077.58	282574.31	125370.37
18	91749.16	114175.79	294919.57	124266.90
19	86419.70	153514.11	0.00	122776.86
20	76253.86	113867.30	298664.47	118474.03
21	78389.47	153773.43	299737.29	111313.02
22	73994.56	122782.75	303319.26	110352.25
23	67532.53	105751.03	304768.73	108733.99
24	77044.01	99281.34	140574.81	108552.04
25	64664.71	139553.16	137962.62	107404.34
26	75328.87	144135.98	134050.07	105733.54
27	72107.60	127864.55	353183.81	105008.31
28	66051.52	182645.56	118148.20	103282.38
29	65605.48	153032.06	107138.38	101004.64
30	61994.48	115641.28	91131.24	99937.59
31	61136.38	152701.92	88218.23	97483.56
32	63408.86	129219.61	46085.25	97427.84
33	55493.95	103057.49	214634.81	96778.92
34	46426.07	157693.92	210797.67	96712.80
35	46014.02	85047.44	205517.64	96479.51
36	28663.76	127056.21	201126.82	90708.19
37	44069.95	51283.14	197029.42	89949.14
38	20229.59	65947.93	185265.10	81229.06

	RD_spend	Administration	Marketing_Spend	Profit
39	38558.51	82982.09	174999.30	81005.76
40	28754.33	118546.05	172795.67	78239.91
41	27892.92	84710.77	164470.71	77798.83
42	23640.93	96189.63	148001.11	71498.49
43	15505.73	127382.30	35534.17	69758.98
44	22177.74	154806.14	28334.72	65200.33
45	1000.23	124153.04	1903.93	64926.08
46	1315.46	115816.21	297114.46	49490.75
47	0.00	135426.92	0.00	42559.73
48	542.05	51743.15	0.00	35673.41
49	0.00	116983.80	45173.06	14681.40

In [10]:

```
model = smf.ols("Profit~RD_spend+Administration+Marketing_Spend+Profit",data=data).fit()
model.summary()
```

Loading [MathJax]/extensions/Safe.js

Out[10]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	1.000
Model:	OLS	**Adj. R-squared:**	1.000
Method:	Least Squares	**F-statistic:**	1.344e+31
Date:	Sun, 28 Jan 2024	**Prob (F-statistic):**	0.00
Time:	21:10:11	**Log-Likelihood:**	1130.7
No. Observations:	50	**AIC:**	-2251.
Df Residuals:	45	**BIC:**	-2242.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	7.276e-11	4.12e-11	1.765	0.084	-1.03e-11	1.56e-10
RD_spend	-1.11e-16	5.3e-16	-0.210	0.835	-1.18e-15	9.56e-16
Administration	-2.776e-17	2.13e-16	-0.130	0.897	-4.57e-16	4.02e-16
Marketing_Spend	8.327e-17	7.06e-17	1.180	0.244	-5.89e-17	2.25e-16
Profit	1.0000	6.14e-16	1.63e+15	0.000	1.000	1.000

Omnibus:	3.482	**Durbin-Watson:**	0.223
Prob(Omnibus):	0.175	**Jarque-Bera (JB):**	2.890
Skew:	0.588	**Prob(JB):**	0.236
Kurtosis:	3.047	**Cond. No.**	2.29e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.29e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

In [11]:

```
model.params
```

Out[11]:

```
Intercept          7.275958e-11
RD_spend          -1.110223e-16
Administration    -2.775558e-17
Marketing_Spend    8.326673e-17
Profit             1.000000e+00
dtype: float64
```

In [12]:

```
print(model.tvalues, '\n', model.pvalues)
```

Loading [MathJax]/extensions/Safe.js

```
Intercept           1.765402e+00
RD_spend           -2.096318e-01
Administration     -1.301306e-01
Marketing_Spend     1.179931e+00
Profit              1.627508e+15
dtype: float64
 Intercept           0.084281
RD_spend            0.834901
Administration      0.897043
Marketing_Spend     0.244228
Profit              0.000000
dtype: float64
```

In [13]:

```
(model.rsquared,model.rsquared_adj)
```

Out[13]:

```
(1.0, 1.0)
```

In [14]:

```
md= smf.ols("Profit~RD_spend",data=data).fit()
print(md.tvalues, '\n' , md.pvalues)
```

```
Intercept    19.320288
RD_spend     29.151139
dtype: float64
 Intercept    2.782697e-24
RD_spend     3.500322e-32
dtype: float64
```

In [15]:

```
md= smf.ols("Profit~Administration",data=data).fit()
print(md.tvalues, '\n' , md.pvalues)
```

```
Intercept         3.040044
Administration    1.419493
dtype: float64
 Intercept         0.003824
Administration    0.162217
dtype: float64
```

In [16]:

```
md= smf.ols("Profit~RD_spend+Administration",data=data).fit()
md.summary()
```

Loading [MathJax]/extensions/Safe.js

Out[16]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.948
Model:	OLS	**Adj. R-squared:**	0.946
Method:	Least Squares	**F-statistic:**	426.8
Date:	Sun, 28 Jan 2024	**Prob (F-statistic):**	7.29e-31
Time:	21:10:52	**Log-Likelihood:**	-526.83
No. Observations:	50	**AIC:**	1060.
Df Residuals:	47	**BIC:**	1065.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	5.489e+04	6016.718	9.122	0.000	4.28e+04	6.7e+04
RD_spend	0.8621	0.030	28.589	0.000	0.801	0.923
Administration	-0.0530	0.049	-1.073	0.289	-0.152	0.046

Omnibus:	14.678	**Durbin-Watson:**	1.189
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	20.449
Skew:	-0.961	**Prob(JB):**	3.63e-05
Kurtosis:	5.474	**Cond. No.**	6.65e+05

Notes:

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

[2] The condition number is large, 6.65e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

In [17]:

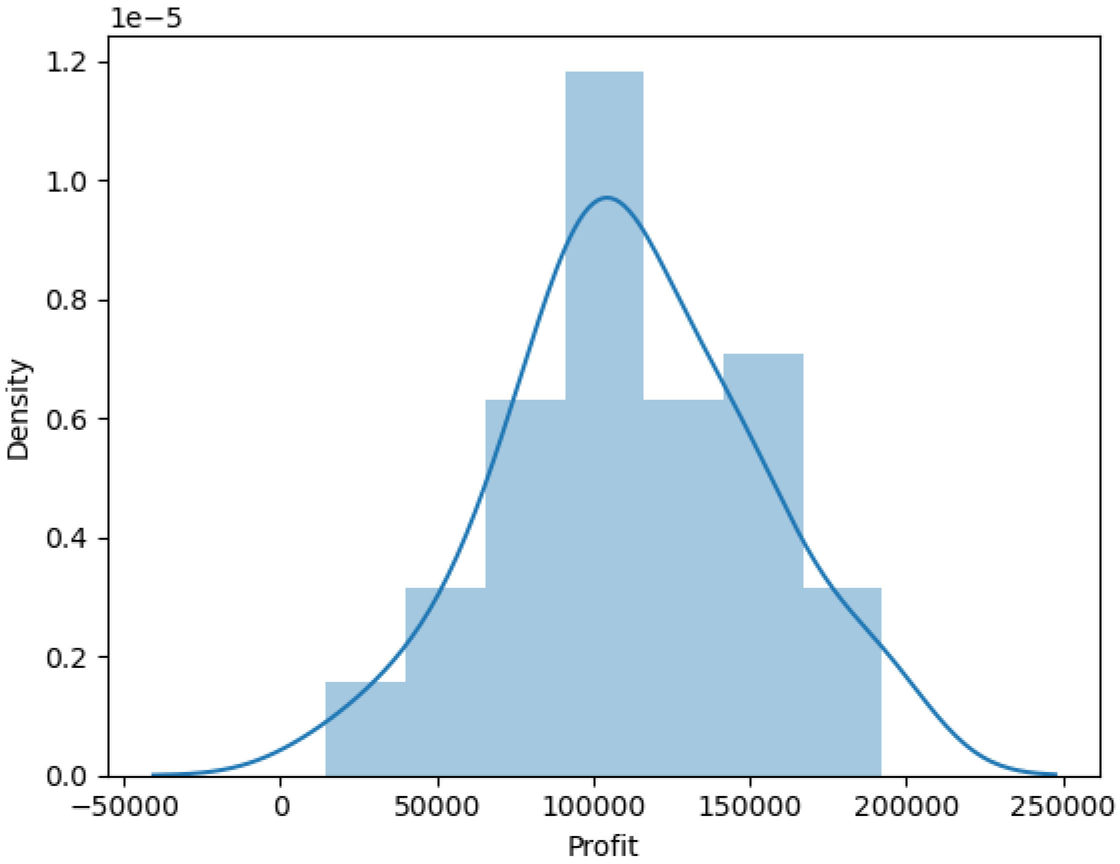
```
rsq_RD = smf.ols("RD_spend~Marketing_Spend+Administration",data=data).fit().rsquared
vif_RD = 1/(1-rsq_RD)
rsq_A = smf.ols("Administration~RD_spend+Marketing_Spend",data=data).fit().rsquared
vif_A= 1/(1-rsq_A)
rsq_M= smf.ols("Marketing_Spend~Administration+RD_spend",data=data).fit().rsquared
vif_M = 1/(1-rsq_M)
d1={'Variables':['Administration','RD_spend','Marketing_Spend'],'VIF':[vif_A,vif_RD,vif_
vif_frame = pd.DataFrame(d1)
vif_frame
```

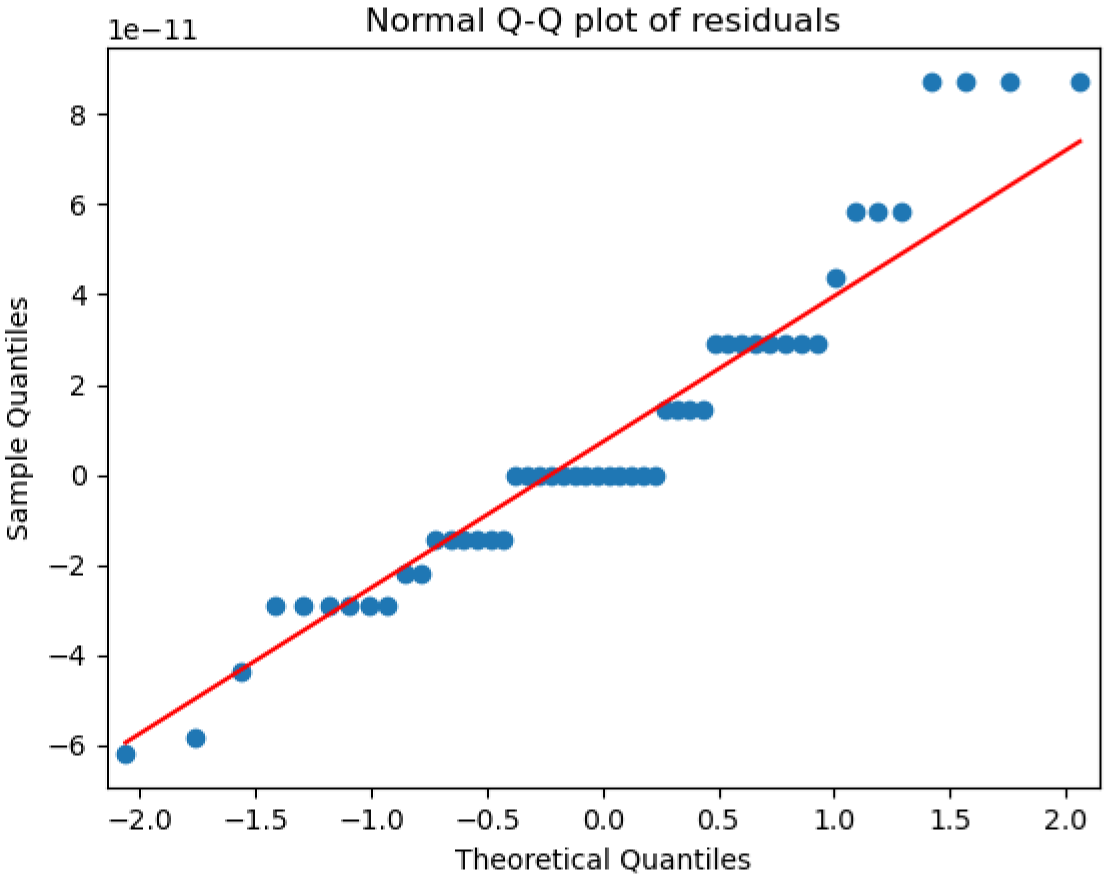
Out[17]:

	Variables	VIF
0	Administration	1.175091
1	RD_spend	2.468903
2	Marketing_Spend	2.326773

In [18]:

```
import statsmodels.api as sm
qqplot=sm.qqplot(model.resid,line='q')
plt.title("Normal Q-Q plot of residuals")
plt.show()
```



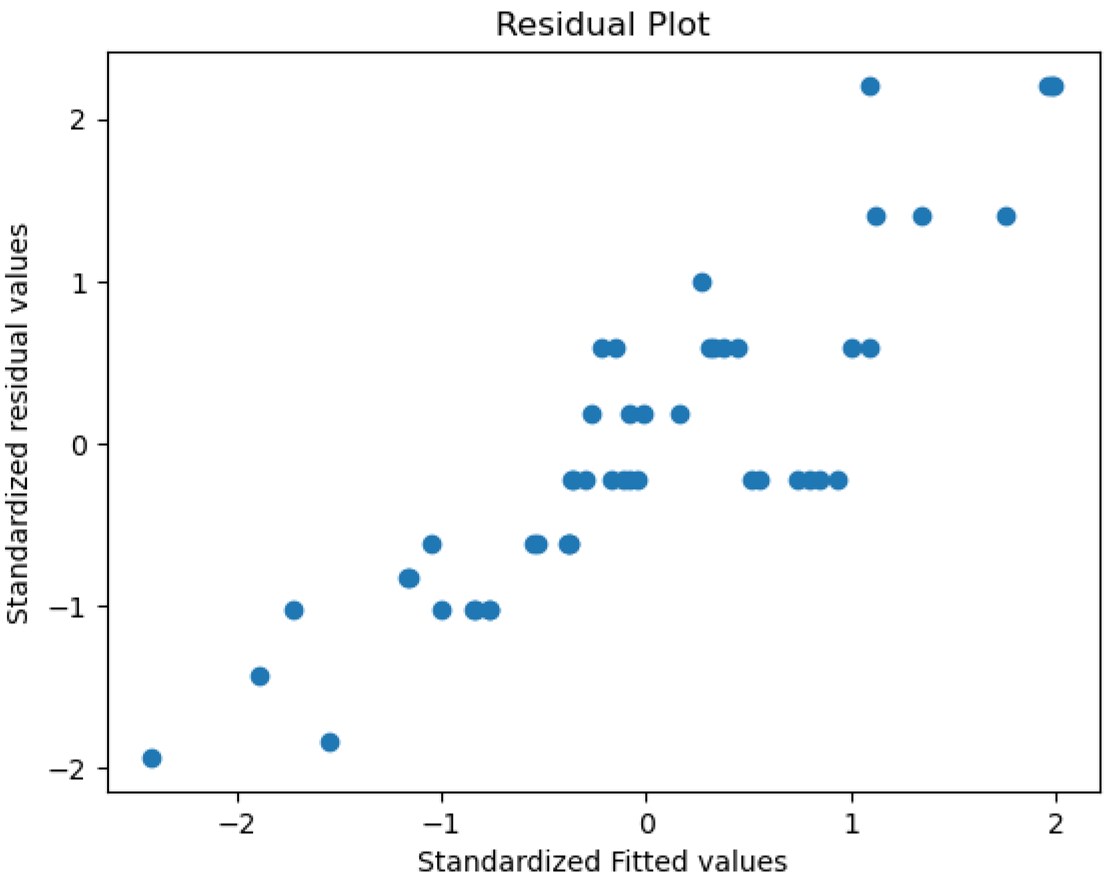


In [19]:
```
def get_standardized_values( vals ):
    return (vals - vals.mean())/vals.std()
```

In [20]:
```
plt.scatter(get_standardized_values(model.fittedvalues),
            get_standardized_values(model.resid))
plt.title('Residual Plot')
plt.xlabel('Standardized Fitted values')
```

Loading [MathJax]/extensions/Safe.js

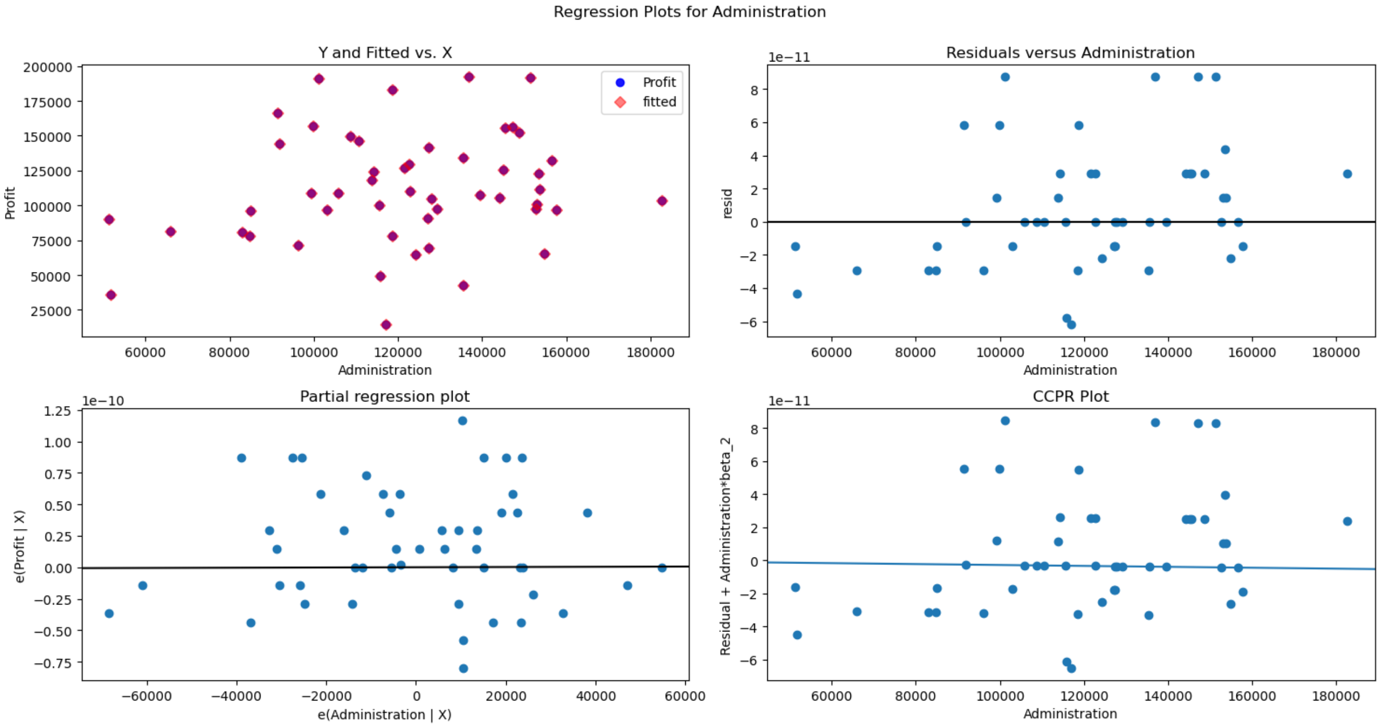
```
plt.ylabel('Standardized residual values')
plt.show()
```



In [21]:

```
fig = plt.figure(figsize=(15,8))
fig = sm.graphics.plot_regress_exog(model, "Administration", fig=fig)
plt.show()
```

```
eval_env: 1
```

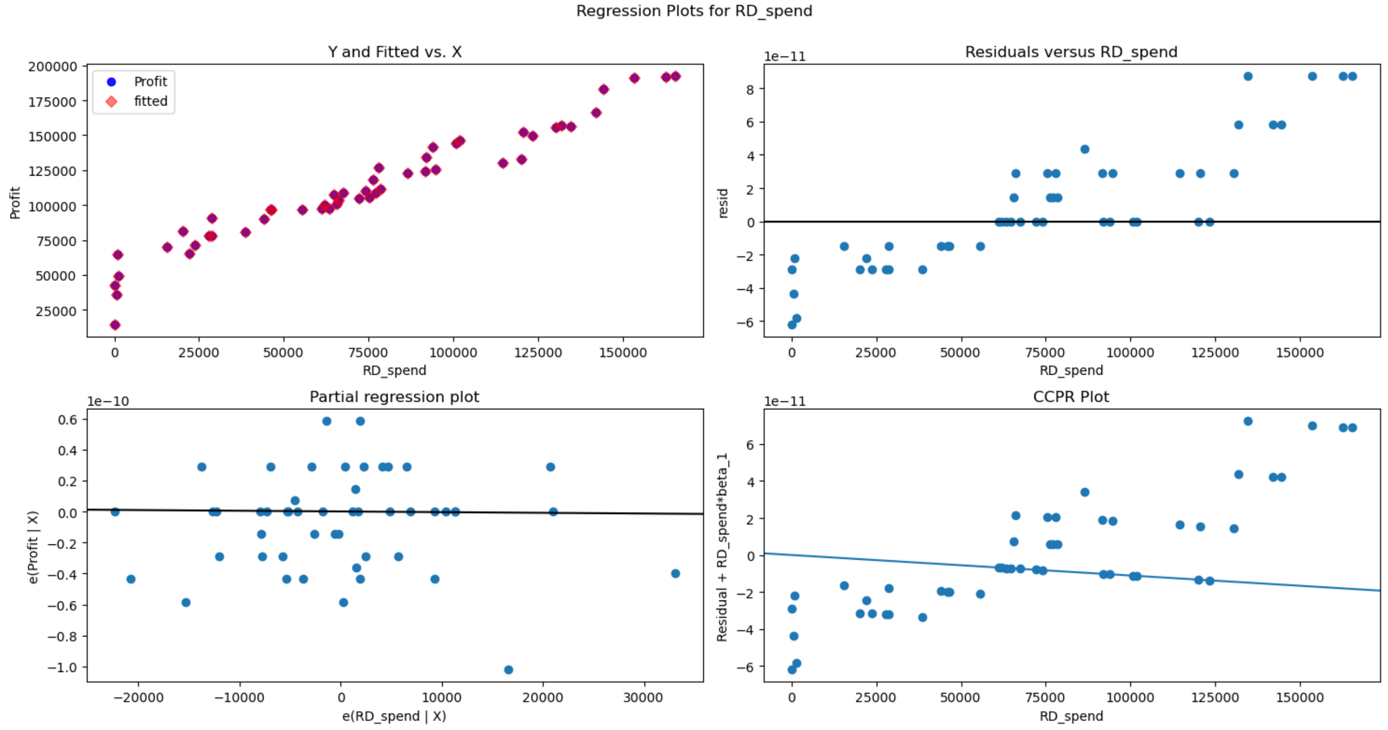


In [22]:

```
fig = plt.figure(figsize=(15,8))
fig = sm.graphics.plot_regress_exog(model, "RD_spend", fig=fig)
plt.show()
```

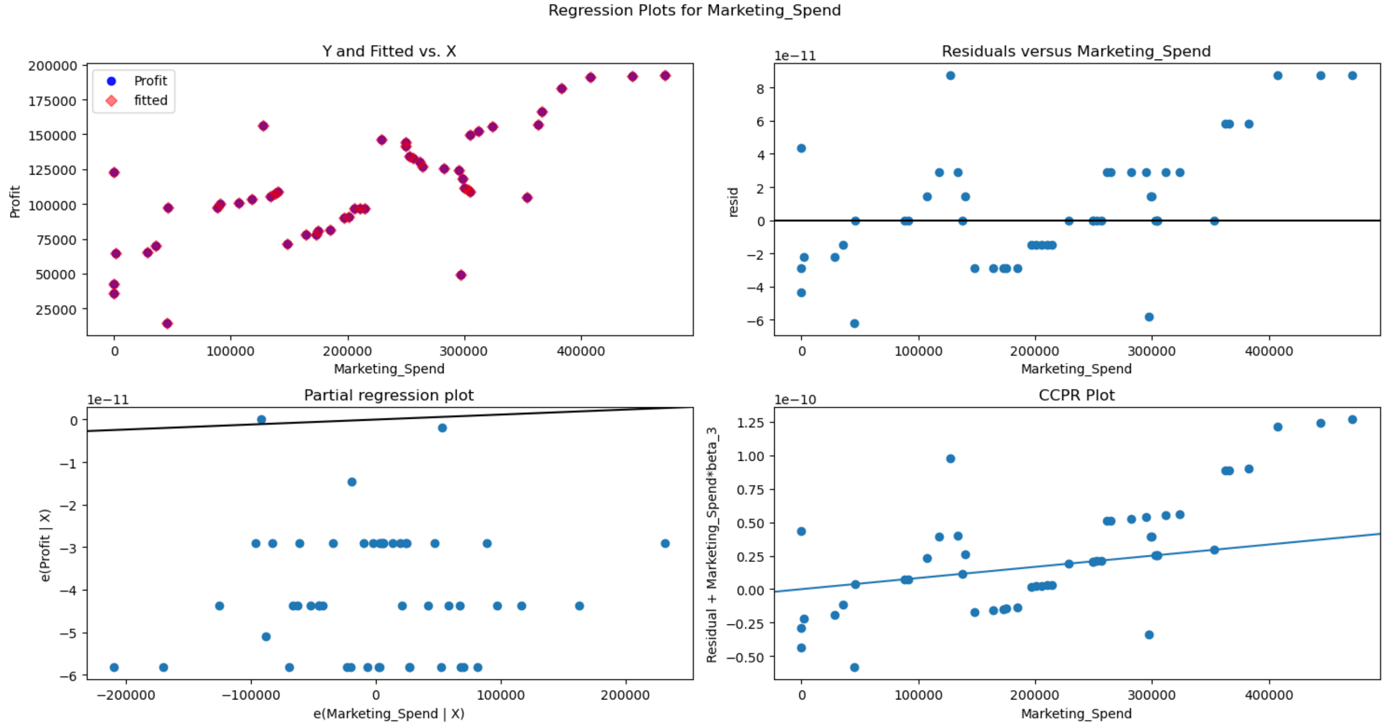
Loading [MathJax]/extensions/Safe.js

```
eval_env: 1
```



In [23]:
```python
fig = plt.figure(figsize=(15,8))
fig = sm.graphics.plot_regress_exog(model, "Marketing_Spend", fig=fig)
plt.show()
```

```
eval_env: 1
```



In [24]:
```python
model_influence = model.get_influence()
(c, _) = model_influence.cooks_distance
c
```

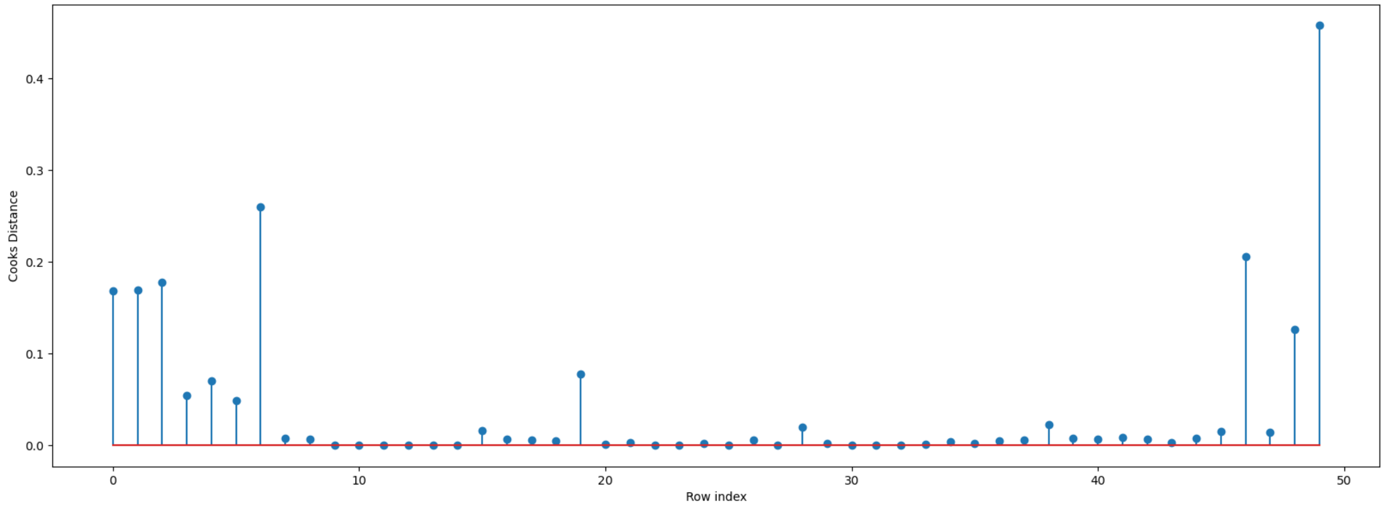
Loading [MathJax]/extensions/Safe.js

Out[24]:

```
array([0.16795827, 0.16870778, 0.17776404, 0.05406131, 0.0704668 ,
       0.04925265, 0.26047563, 0.00831667, 0.00714747, 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.01553896, 0.00668145, 0.00636843, 0.00450749, 0.07759075,
       0.00126136, 0.00335281, 0.        , 0.        , 0.00234966,
       0.        , 0.00647234, 0.        , 0.01963958, 0.00197341,
       0.        , 0.        , 0.        , 0.00097904, 0.00414168,
       0.00216753, 0.00489731, 0.00588332, 0.02315   , 0.00839736,
       0.00657172, 0.00893105, 0.00691511, 0.00310369, 0.00780079,
       0.01488704, 0.20613776, 0.01415728, 0.12589358, 0.45844641])
```

In [25]:

```python
fig = plt.subplots(figsize=(20, 7))
plt.stem(np.arange(len(data)), np.round(c, 3))
plt.xlabel('Row index')
plt.ylabel('Cooks Distance')
plt.show()
```



In [26]:

```python
(np.argmax(c),np.max(c))
```

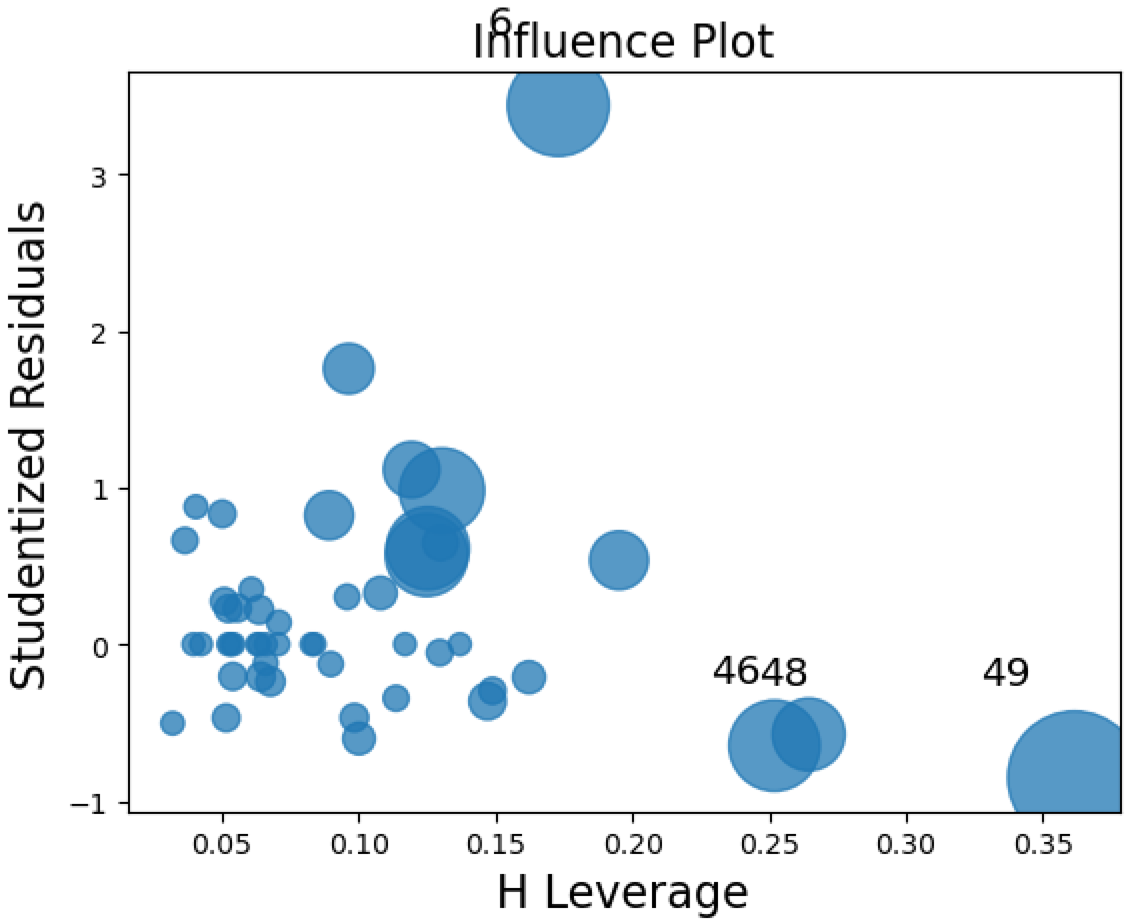
Out[26]:

```
(49, 0.45844641305974987)
```

In [27]:

```python
from statsmodels.graphics.regressionplots import influence_plot
influence_plot(model)
plt.show()
```

Loading [MathJax]/extensions/Safe.js



In [28]:
```
k = data.shape[1]
n = data.shape[0]
leverage_cutoff = 3*((k + 1)/n)
leverage_cutoff
```

Out[28]:
```
0.36
```

In [29]:
```
data[data.index.isin([47, 49])]
```

Out[29]:

	RD_spend	Administration	Marketing_Spend	State	Profit
47	0.0	135426.92	0.00	California	42559.73
49	0.0	116983.80	45173.06	California	14681.40

In [30]:
```
data_new=data.drop(data.index[[47,49]],axis=0).reset_index()
```

In [31]:
```
data_new=data_new.drop(['index'],axis=1)
```

In [32]:
```
data_new
```

Out[32]:

	RD_spend	Administration	Marketing_Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94
5	131876.90	99814.71	362861.36	New York	156991.12
6	134615.46	147198.87	127716.82	California	156122.51
7	130298.13	145530.06	323876.68	Florida	155752.60
8	120542.52	148718.95	311613.29	New York	152211.77
9	123334.88	108679.17	304981.62	California	149759.96
10	101913.08	110594.11	229160.95	Florida	146121.95
11	100671.96	91790.61	249744.55	California	144259.40
12	93863.75	127320.38	249839.44	Florida	141585.52
13	91992.39	135495.07	252664.93	California	134307.35
14	119943.24	156547.42	256512.92	Florida	132602.65
15	114523.61	122616.84	261776.23	New York	129917.04
16	78013.11	121597.55	264346.06	California	126992.93
17	94657.16	145077.58	282574.31	New York	125370.37
18	91749.16	114175.79	294919.57	Florida	124266.90
19	86419.70	153514.11	0.00	New York	122776.86
20	76253.86	113867.30	298664.47	California	118474.03
21	78389.47	153773.43	299737.29	New York	111313.02
22	73994.56	122782.75	303319.26	Florida	110352.25
23	67532.53	105751.03	304768.73	Florida	108733.99
24	77044.01	99281.34	140574.81	New York	108552.04
25	64664.71	139553.16	137962.62	California	107404.34
26	75328.87	144135.98	134050.07	Florida	105733.54
27	72107.60	127864.55	353183.81	New York	105008.31
28	66051.52	182645.56	118148.20	Florida	103282.38
29	65605.48	153032.06	107138.38	New York	101004.64
30	61994.48	115641.28	91131.24	Florida	99937.59
31	61136.38	152701.92	88218.23	New York	97483.56
32	63408.86	129219.61	46085.25	California	97427.84
33	55493.95	103057.49	214634.81	Florida	96778.92
34	46426.07	157693.92	210797.67	California	96712.80
35	46014.02	85047.44	205517.64	New York	96479.51
36	28663.76	127056.21	201126.82	Florida	90708.19
37	44069.95	51283.14	197029.42	California	89949.14
38	20229.59	65947.93	185265.10	New York	81229.06

Loading [MathJax]/extensions/Safe.js

	RD_spend	Administration	Marketing_Spend	State	Profit
39	38558.51	82982.09	174999.30	California	81005.76
40	28754.33	118546.05	172795.67	California	78239.91
41	27892.92	84710.77	164470.71	Florida	77798.83
42	23640.93	96189.63	148001.11	California	71498.49
43	15505.73	127382.30	35534.17	New York	69758.98
44	22177.74	154806.14	28334.72	California	65200.33
45	1000.23	124153.04	1903.93	New York	64926.08
46	1315.46	115816.21	297114.46	Florida	49490.75
47	542.05	51743.15	0.00	New York	35673.41

In [33]:

```
final_Newdata= smf.ols('Profit~Administration+Marketing_Spend',data =data_new).fit()
```

In [34]:

```
(final_Newdata.rsquared,final_Newdata.aic)
```

Out[34]:

```
(0.579904897269647, 1109.6575232827427)
```

In [35]:

```
final_Newdata= smf.ols('Profit~RD_spend+Marketing_Spend',data =data_new).fit()
```

In [36]:

```
(final_Newdata.rsquared,final_Newdata.aic)
```

Out[36]:

```
(0.9588424786144887, 998.1499506151225)
```

In [37]:

```
new_data=pd.DataFrame({'Adiministration':100,'RD_spend':150,'Marketing_Spend':200},index
new_data
```

Out[37]:

	Adiministration	RD_spend	Marketing_Spend
1	100	150	200

In [38]:

```
final_Newdata.predict(new_data)
```

Out[38]:

```
1    50644.293843
dtype: float64
```

In [ ]:

Loading [MathJax]/extensions/Safe.js