In [86]:

```
import pandas as pd
import numpy as np
from scipy import stats
from matplotlib import pyplot as plt
from scipy import stats
import seaborn as sns
from statsmodels.graphics.regressionplots import influence_plot
import statsmodels.formula.api  as smf
import statsmodels.api as sm
import warnings
warnings.filterwarnings('ignore')
```

In [87]:

```
# Import Data
```

In [88]:

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

Out[88]:

	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

	R&D Spend	Administration	Marketing Spend	State	Profit
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
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 [89]:

```
dt.head()
```

Out[89]:

	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

In [90]:

```
# Data Understanding
```

In [91]:

```
dt.shape
```

Out[91]:

```
(50, 5)
```

In [92]:

```
dt.isna().sum()
```

Out[92]:

```
R&D Spend          0
Administration     0
Marketing Spend    0
State              0
Profit             0
dtype: int64
```

In [93]:

```
dt .dtypes
```

Out[93]:

```
R&D Spend          float64
Administration     float64
Marketing Spend    float64
State               object
Profit             float64
dtype: object
```

In [94]:

```
dt.describe
```

Out[94]:

```
<bound method NDFrame.describe of     R&D Spend  Administration  Marketing S
pend       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
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 [95]:

```
dt = dt.rename({'R&D Spend':'RDS','Administration':'ADS','Marketing Spend':'MKTS'},axis=1)
dt
```

Out[95]:

	RDS	ADS	MKTS	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

	RDS	ADS	MKTS	State	Profit
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
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 [96]:

```
dt[dt.duplicated()] #no duplicated data
```

Out[96]:

RDS	ADS	MKTS	State	Profit

In [97]:

```
dt.describe
```

Out[97]:

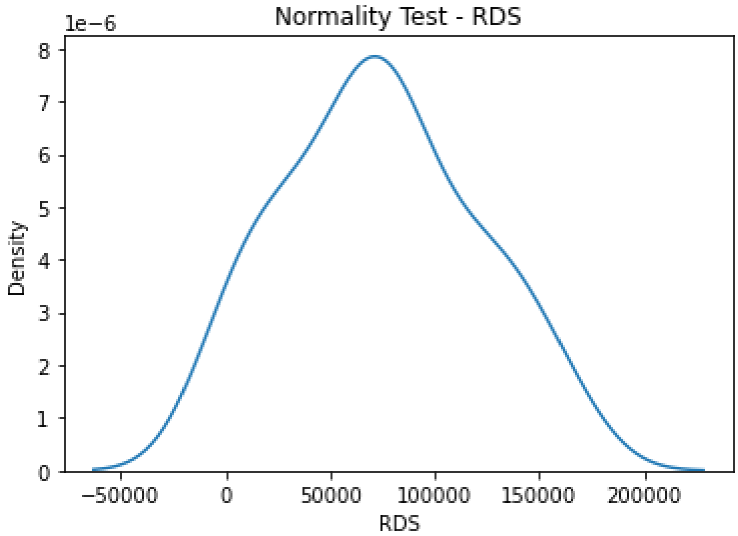
```
<bound method NDFrame.describe of             RDS          ADS        MKTS
State      Profit
0   165349.20  136897.80  471784.10    New York  192261.83
1   162597.70  151377.59  443898.53  California  191792.06
2   153441.51  101145.55  407934.54     Florida  191050.39
3   144372.41  118671.85  383199.62    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
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 [98]:

```
# Normality test
```

In [99]:

```
sns.distplot(a=dt['RDS'],hist=False)
plt.title('Normality Test - RDS')
plt.show()
```



In [100]:

```
dt['RDS'].skew()
```

Out[100]:

0.164002172321177

In [101]:

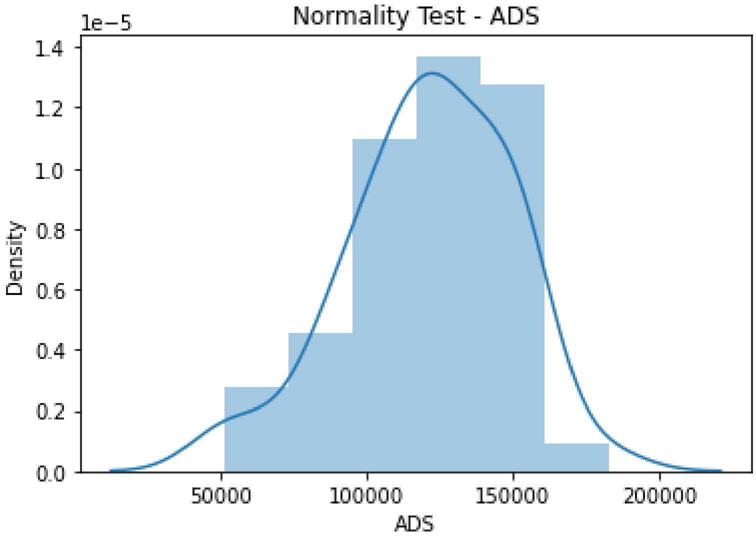
```
dt['RDS'].kurtosis()
```

Out[101]:

-0.7614645568424674

In [102]:

```
sns.distplot(a=dt['ADS'],hist=True)
plt.title('Normality Test - ADS')
plt.show()
```



In [103]:

```
dt['ADS'].skew()
```

Out[103]:

```
-0.4890248099671768
```

In [104]:

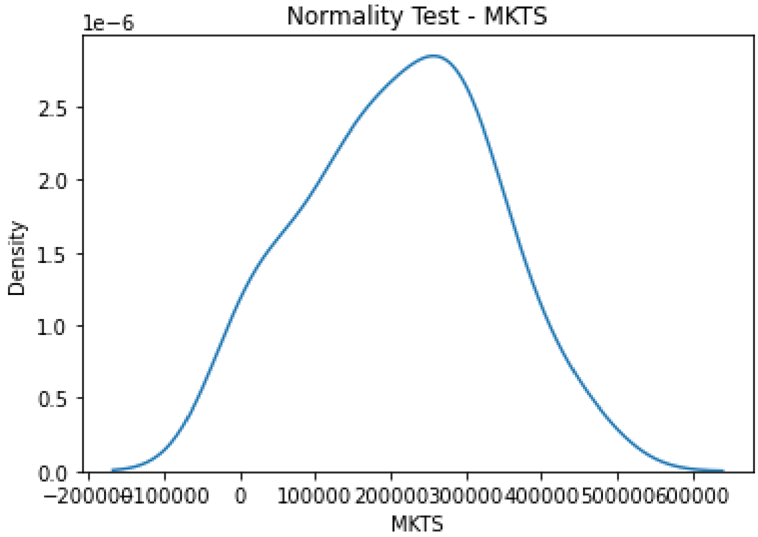
```
dt['ADS'].kurtosis()
```

Out[104]:

```
0.22507113536865386
```

In [105]:

```
sns.distplot(a=dt['MKTS'],hist=False)
plt.title('Normality Test - MKTS')
plt.show()
```



In [106]:

```
dt['MKTS'].skew()
```

Out[106]:

-0.04647226758360412

In [107]:

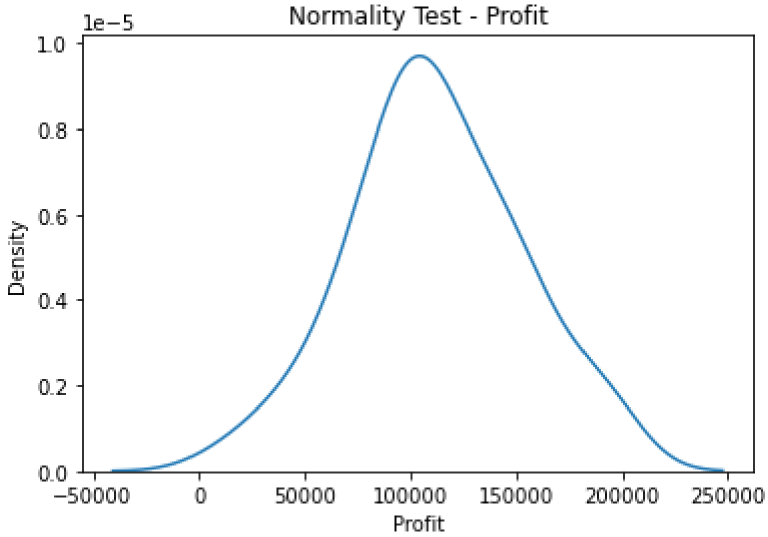
```
dt['MKTS'].kurtosis()
```

Out[107]:

-0.6717011281297514

In [108]:

```
sns.distplot(a=dt['Profit'],hist= False)
plt.title('Normality Test - Profit')
plt.show()
```



In [109]:

```
dt['Profit'].skew()
```

Out[109]:

0.023291019769116614

In [110]:

```
dt['Profit'].kurtosis()
```

Out[110]:

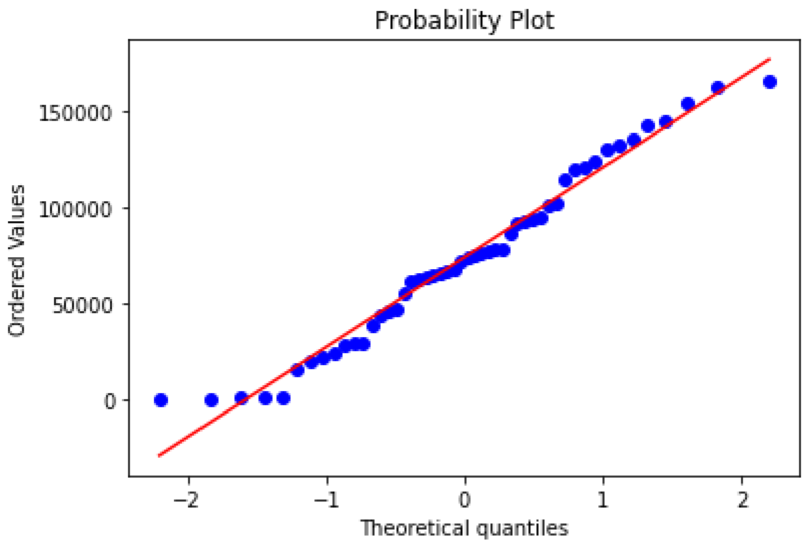
-0.06385888546853113

In [111]:

```
# Normality test using probplot
```

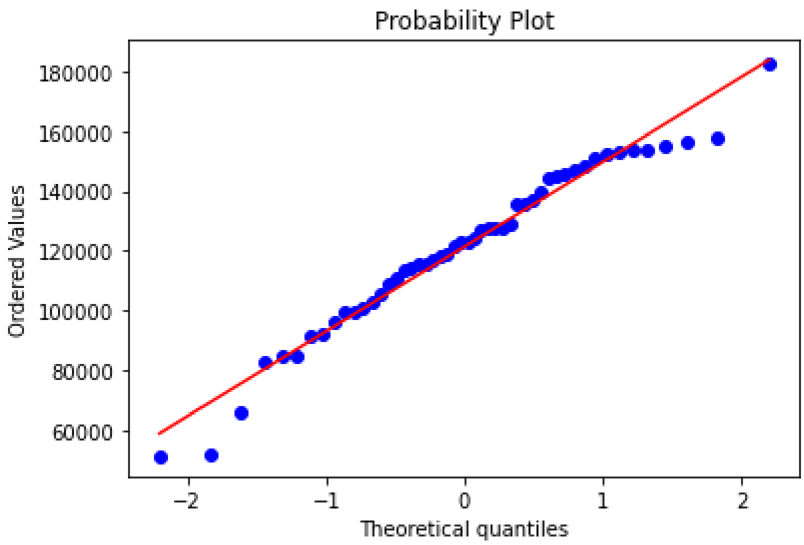
In [112]:

```
stats.probplot(x=dt['RDS'],dist='norm',plot=plt)
plt.show()
```



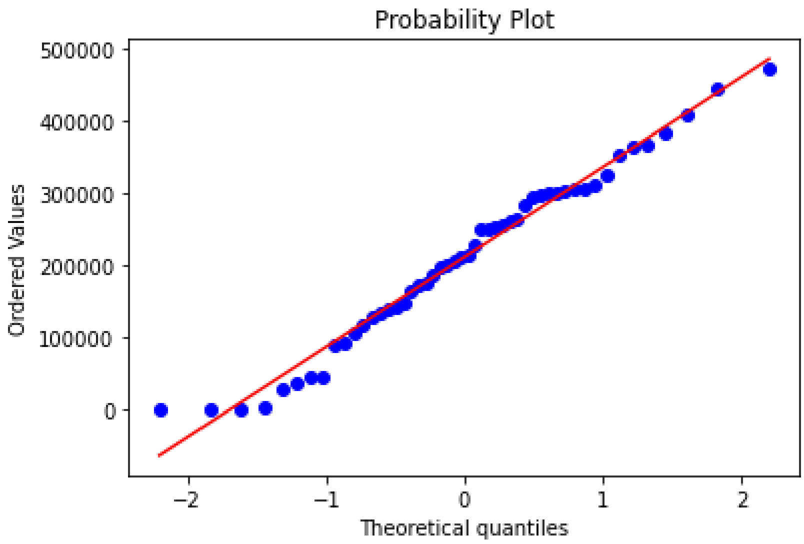
In [113]:

```
stats.probplot(x=dt['ADS'],dist='norm',plot=plt)
plt.show()
```



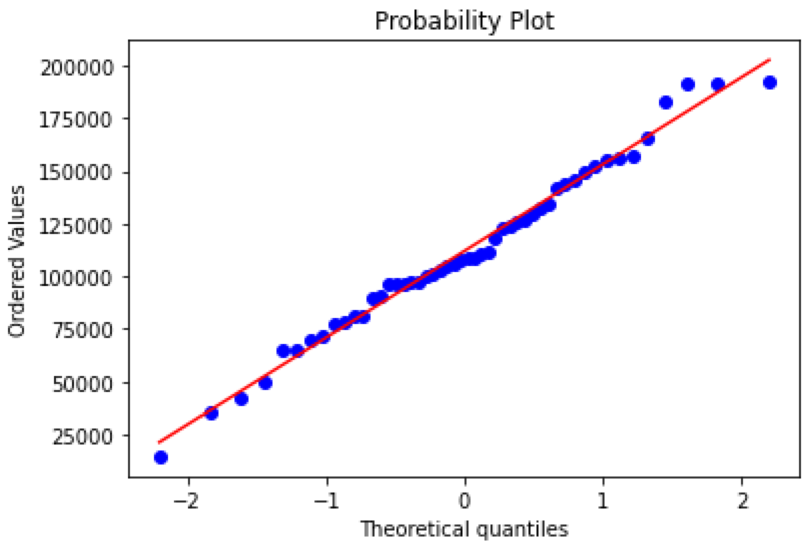
In [114]:

```
stats.probplot(x=dt['MKTS'],dist='norm',plot=plt)
plt.show()
```



In [115]:

```
stats.probplot(x=dt['Profit'],dist='norm',plot=plt)
plt.show()
```

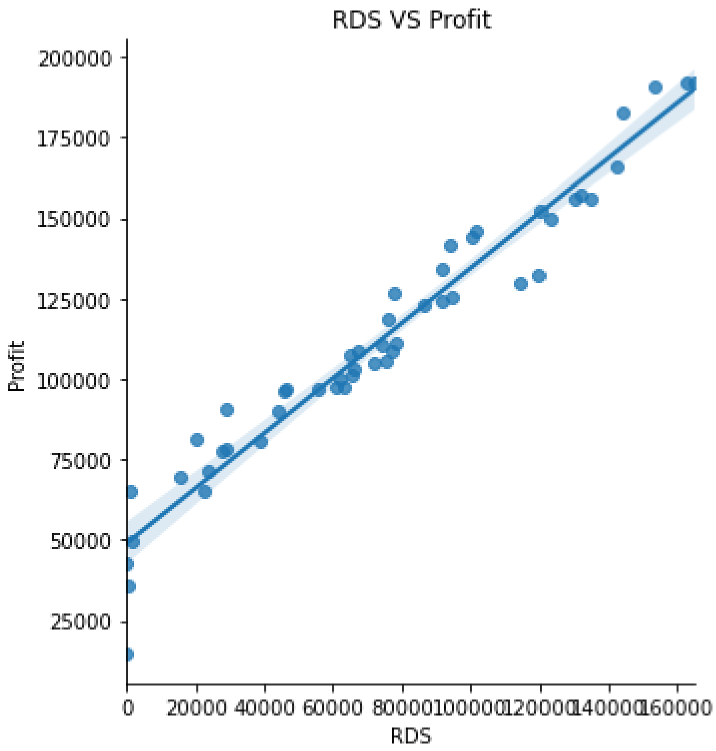


In [116]:

```
## Normality test is failed
#Linearity test
```

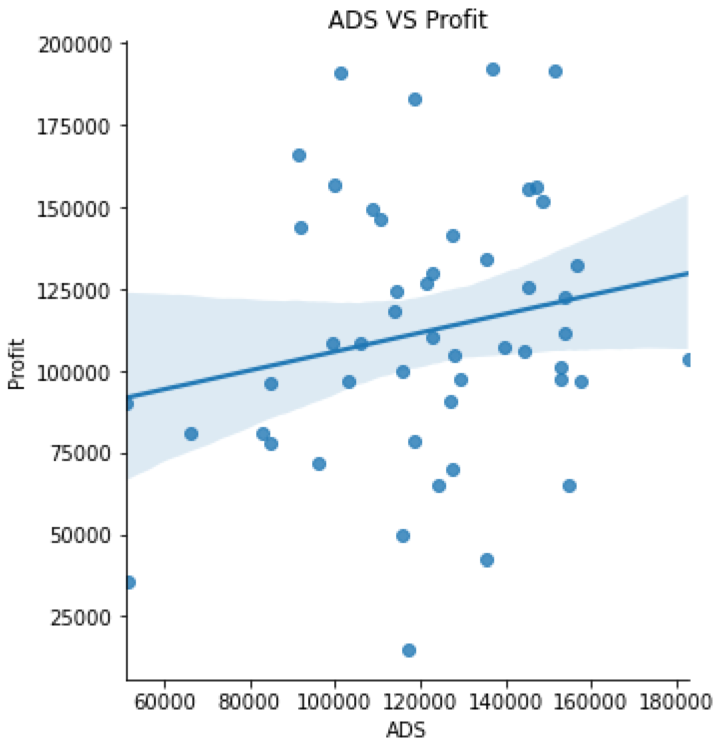
In [117]:

```
sns.lmplot(x='RDS',y='Profit',data=dt)
plt.title('RDS VS Profit')
plt.show()
```



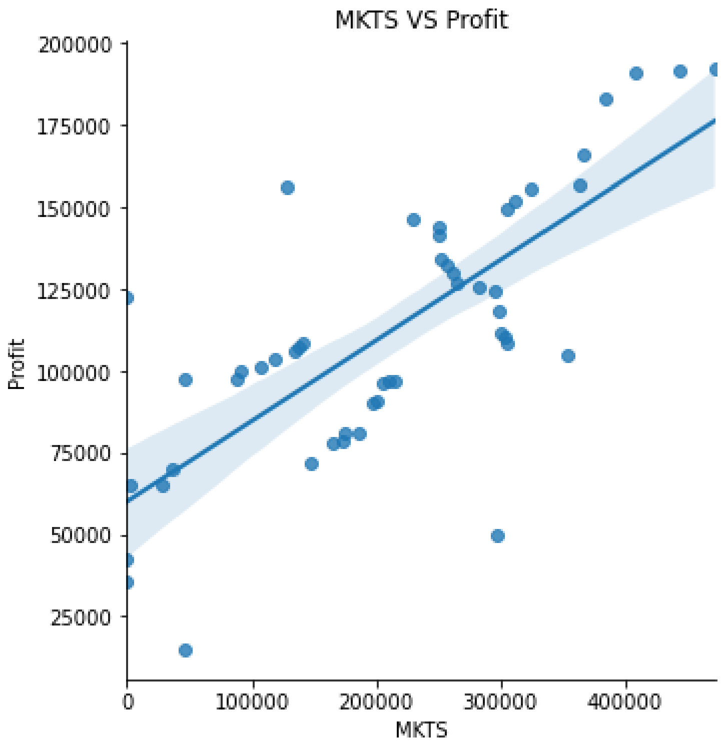
In [118]:

```
sns.lmplot(x='ADS',y='Profit',data=dt)
plt.title('ADS VS Profit')
plt.show()
```



In [119]:

```
sns.lmplot(x='MKTS',y='Profit',data=dt)
plt.title('MKTS VS Profit')
plt.show()
```



In [120]:

```
## Linearity test failed
#Correlation
```

In [121]:

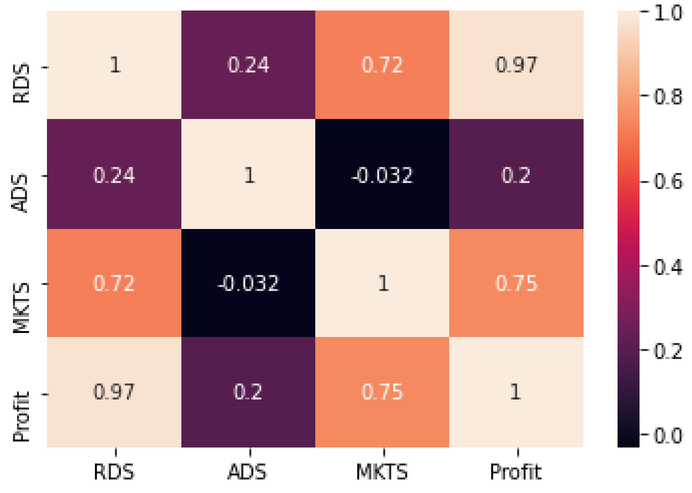
```
data_corr_matrix = dt.corr().round(3)
data_corr_matrix
```

Out[121]:

	RDS	ADS	MKTS	Profit
RDS	1.000	0.242	0.724	0.973
ADS	0.242	1.000	-0.032	0.201
MKTS	0.724	-0.032	1.000	0.748
Profit	0.973	0.201	0.748	1.000

In [122]:

```
sns.heatmap(data=data_corr_matrix,annot=True)
plt.show()
```



create a Reference data to understand how the x features should behave with y axis.

In [123]:

```
dt.shape
```

Out[123]:

```
(50, 5)
```

In [130]:

```
X = np.random.randn(81)
y = 10 * X + np.random.randn(81)*2
```

In [131]:

```
X_df = pd.DataFrame(data=[X,y]).T
X_df.columns= ['X','y']
X_df
```

Out[131]:

	X	y
0	0.384607	8.306832
1	1.715146	19.617157
2	-1.267898	-15.639936
3	-1.374264	-13.596102
4	-1.464365	-14.069694
...	...	...
76	1.548219	14.349363
77	0.226917	-0.725109
78	-0.767602	-6.169057
79	0.190755	2.278621
80	0.686134	8.088709

81 rows × 2 columns

In [132]:

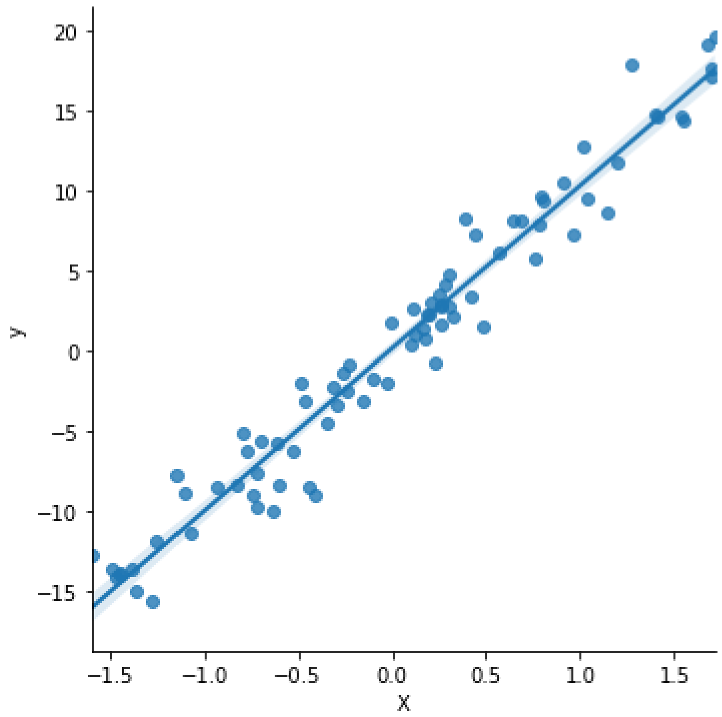
```
# 1. Linearity Test
```

In [133]:

```
sns.lmplot(x='X',y='y',data=X_df)
```

Out[133]:

<seaborn.axisgrid.FacetGrid at 0x2179e9a94c0>

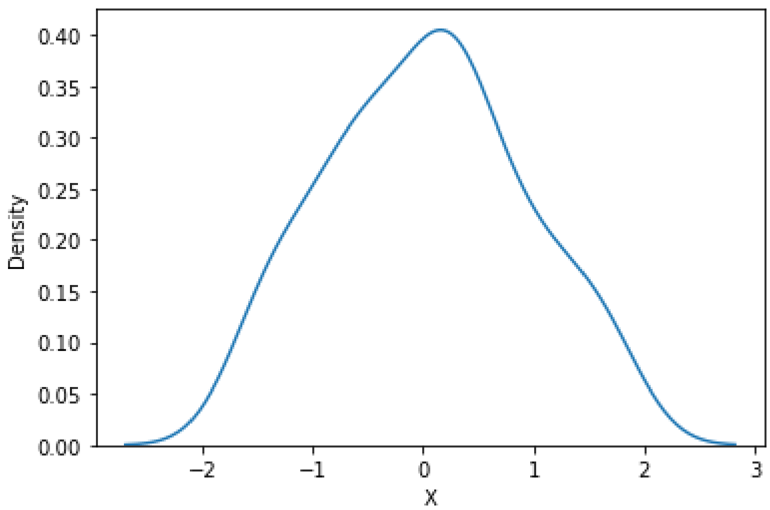


In [134]:

```
# 2. Normality Test
```

In [135]:

```
sns.distplot(a=X_df['X'],hist=False)
plt.show()
```



In [136]:

```
X_df.skew()
```

Out[136]:

```
X    0.082954
y    0.185472
dtype: float64
```

In [137]:

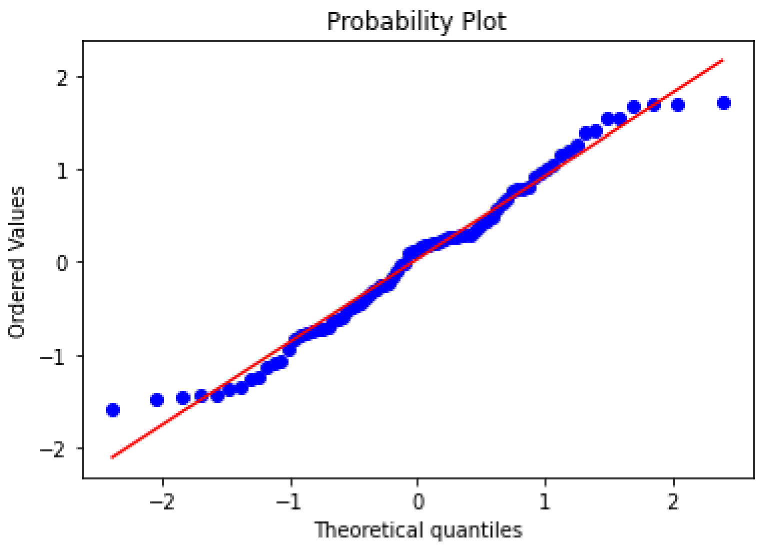
```
X_df.kurtosis()
```

Out[137]:

```
X   -0.747365
y   -0.743023
dtype: float64
```

In [138]:

```
stats.probplot(x = X_df['X'],dist='norm',plot=plt)
plt.show()
```



In [139]:

```
## 3. Multicollinearity Test - Passed.
## 4. AutoRegression Test - Passed.
## 5. Homoscedasticity Test || 6. Zero Residual Mean Test
```

In [140]:

```
# Model Building
```

In [141]:

```
X = X_df[['X']]
y = X_df[['y']]
```

In [142]:

```
# sklearn training
```

In [143]:

```
from sklearn.linear_model import LinearRegression
linear_model = LinearRegression() #Object Creation/Model Initialization
linear_model.fit(X,y)
```

Out[143]:

```
LinearRegression()
```

In [144]:

```
linear_model.intercept_
```

Out[144]:

```
array([0.17592943])
```

In [145]:

```
linear_model.coef_
```

Out[145]:

```
array([[10.14155551]])
```

In [146]:

```
# Model Testing
```

In [147]:

```
y_prediction = linear_model.predict(X)
```

In [148]:

```
# Model Evaluation
```

In [149]:

```
y
```

Out[149]:

	y
0	8.306832
1	19.617157
2	-15.639936
3	-13.596102
4	-14.069694
...	...
76	14.349363
77	-0.725109
78	-6.169057
79	2.278621
80	8.088709

81 rows × 1 columns

In [150]:

```
y_prediction
```

Out[150]:

```
array([[  4.07644088],
       [ 17.57017614],
       [-12.6825248 ],
       [-13.76124778],
       [-14.67500844],
       [ -4.77503559],
       [  5.98338369],
       [  7.90407094],
       [ 17.43011895],
       [  3.07012549],
       [ -6.24285239],
       [  0.08063971],
       [ -2.75127381],
       [ 13.08902277],
       [ -6.88330392],
       [ -2.18122959],
       [-15.93687092],
       [  6.69890792],
       [ 14.49486578],
       [ -1.43758074],
       [ -7.28702606],
       [ 12.32496482],
       [ -4.26453375],
       [ -7.88114027],
       [ -9.26006689],
       [  1.8855998 ],
       [ -3.94467261],
       [  9.40011423],
       [  8.24091466],
       [ -0.14448755],
       [ 14.34636004],
       [-10.97113356],
       [  2.13974587],
       [ -2.46470982],
       [ -7.11606617],
       [  8.06729968],
       [  9.92042582],
       [ -3.03264448],
       [  2.86628751],
       [  2.79450745],
       [-13.59361284],
       [ 11.81064941],
       [ 15.80421696],
       [-12.53819473],
       [  1.26920123],
       [ -3.35558771],
       [ -0.83473953],
       [  5.11237366],
       [ 17.16389008],
       [ 10.68094313],
       [ -5.1794653 ],
       [ -6.04579724],
       [  1.39654638],
       [  8.34598413],
       [-14.51148359],
```

```
       [ -2.3039884 ],
       [  4.477364  ],
       [ 10.44586125],
       [ -7.14381239],
       [ -5.92926769],
       [  3.21879783],
       [-14.51055011],
       [-10.6681231 ],
       [  3.48783058],
       [ 17.44366983],
       [  1.95509722],
       [  4.61659172],
       [-14.8692146 ],
       [  2.82488111],
       [ -8.17537962],
       [  2.23561163],
       [  2.68355246],
       [  1.18178129],
       [-11.39354836],
       [ -4.55546293],
       [  3.20767103],
       [ 15.87727815],
       [  2.47721936],
       [ -7.60874483],
       [  2.11048107],
       [  7.13439212]])
```

In [151]:

```
error = y - y_prediction
error
```

Out[151]:

	y
0	4.230391
1	2.046980
2	-2.957411
3	0.165146
4	0.605315
...	...
76	-1.527915
77	-3.202328
78	1.439688
79	0.168140
80	0.954317

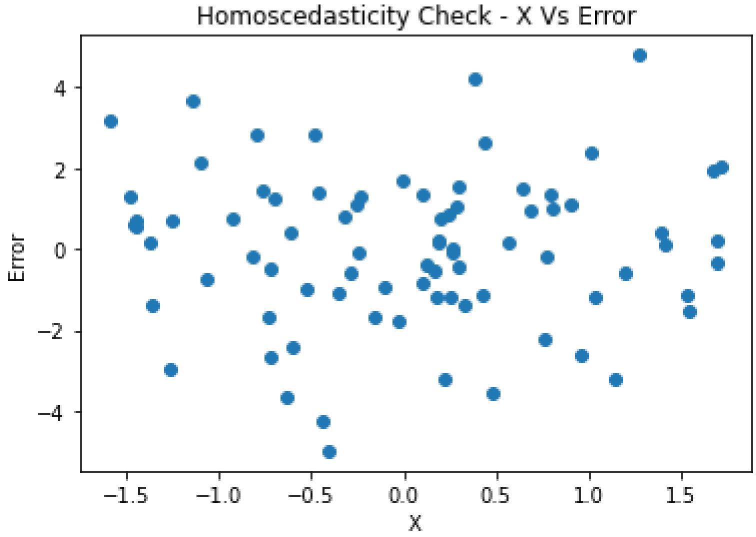
81 rows × 1 columns

In [152]:

```
## 5. Homoscedasticity Check
```

In [153]:

```
plt.scatter(x = X_df['X'],y = error)
plt.title('Homoscedasticity Check - X Vs Error')
plt.xlabel('X')
plt.ylabel('Error')
plt.show()
```

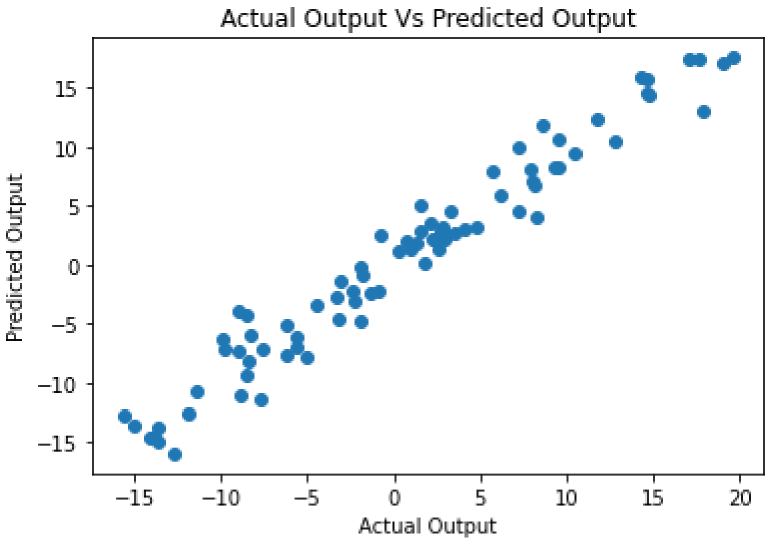


In [154]:

```
## 6. Zero Residual Mean Test
```

In [156]:

```
plt.scatter(x = y,y = y_prediction)
plt.title('Actual Output Vs Predicted Output')
plt.xlabel('Actual Output')
plt.ylabel('Predicted Output')
plt.show()
```



Zero residual Mean Test is Passed.

In [159]:

```
# Back to DATA
```

In [161]:

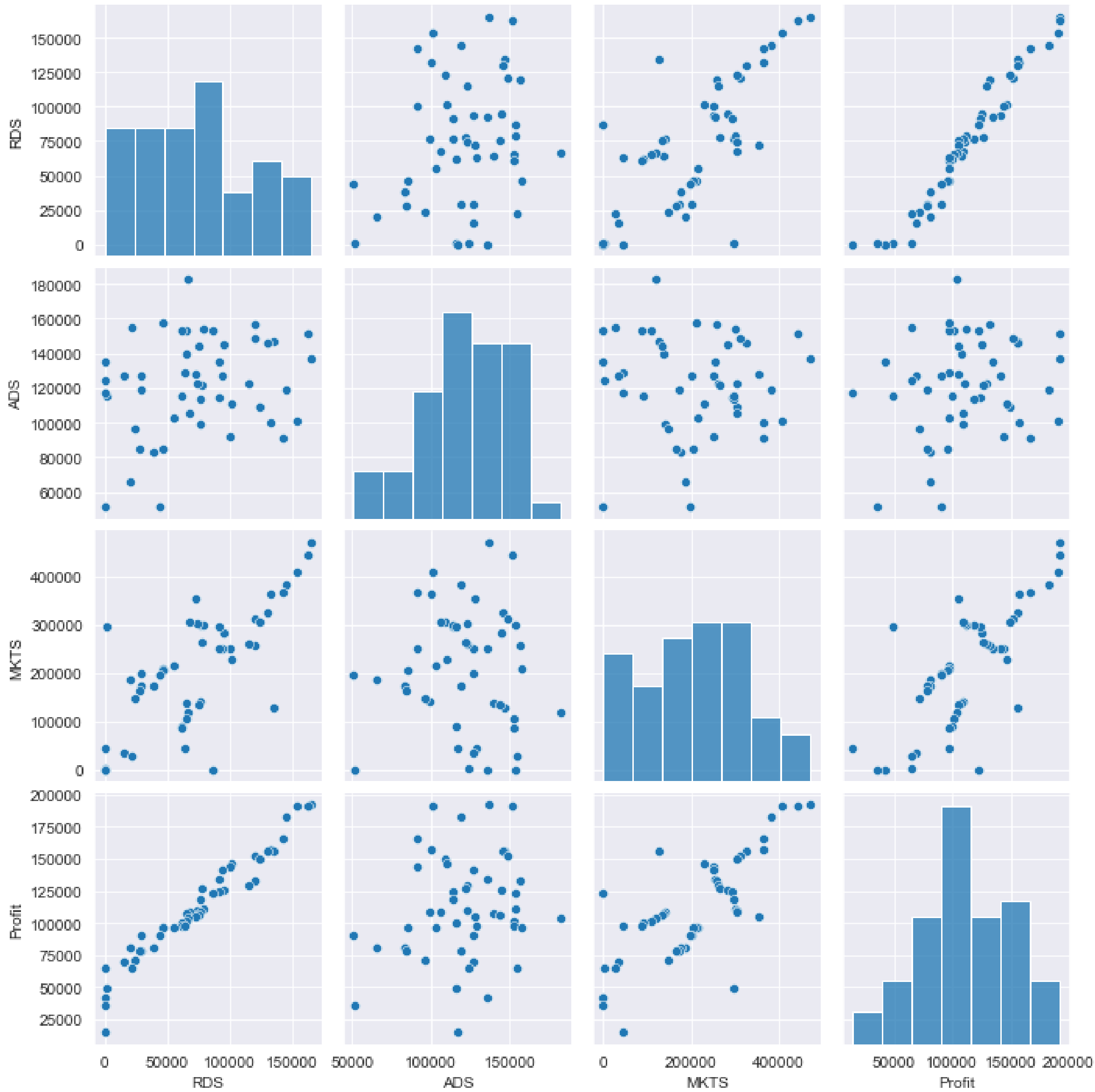
```
dt.head()
```

Out[161]:

	RDS	ADS	MKTS	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

In [163]:

```
### FORMAT THE PLOT BACKGROUND AND SCATTERPLOTS FOR ALL VARIABLES
sns.set_style(style='darkgrid')
sns.pairplot(dt)
plt.show()
```



In [164]:

```
## Log Function
```

In [165]:

```
X_inputs = dt.copy()
X_inputs.head()
```

Out[165]:

	RDS	ADS	MKTS	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

In [166]:

```
X_inputs['log_RDS']  = np.log(X_inputs['RDS'])
X_inputs['log_ADS']  = np.log(X_inputs['ADS'])
X_inputs['log_MKTS'] = np.log(X_inputs['MKTS'])
X_inputs
```

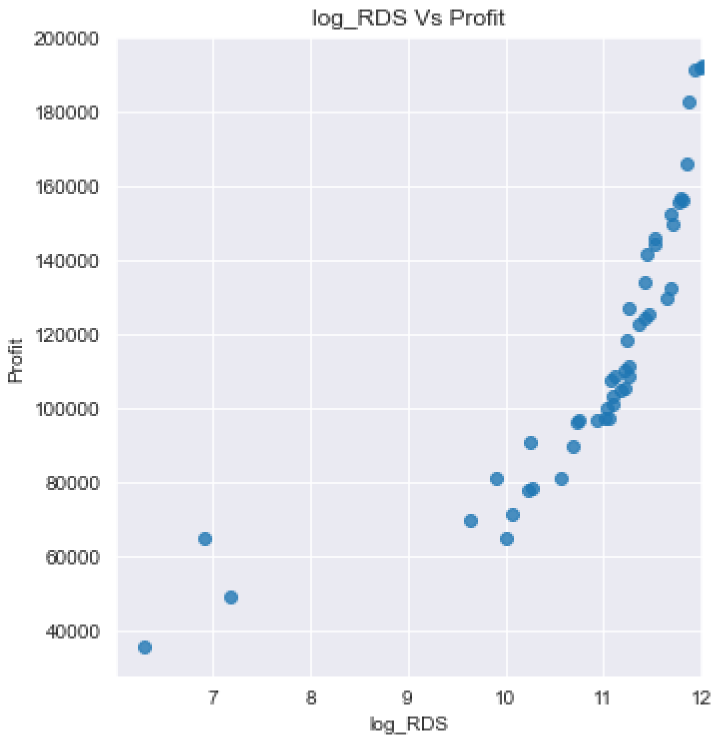
Out[166]:

	RDS	ADS	MKTS	State	Profit	log_RDS	log_ADS	log_MKTS
0	165349.20	136897.80	471784.10	New York	192261.83	12.015815	11.826990	13.064277
1	162597.70	151377.59	443898.53	California	191792.06	11.999034	11.927533	13.003351
2	153441.51	101145.55	407934.54	Florida	191050.39	11.941075	11.524316	12.918862
3	144372.41	118671.85	383199.62	New York	182901.99	11.880151	11.684117	12.856311
4	142107.34	91391.77	366168.42	Florida	166187.94	11.864338	11.422911	12.810849
5	131876.90	99814.71	362861.36	New York	156991.12	11.789624	11.511071	12.801776
6	134615.46	147198.87	127716.82	California	156122.51	11.810178	11.899540	11.757571
7	130298.13	145530.06	323876.68	Florida	155752.60	11.777580	11.888138	12.688118
8	120542.52	148718.95	311613.29	New York	152211.77	11.699758	11.909814	12.649518
9	123334.88	108679.17	304981.62	California	149759.96	11.722659	11.596155	12.628007
10	101913.08	110594.11	229160.95	Florida	146121.95	11.531876	11.613622	12.342180
11	100671.96	91790.61	249744.55	California	144259.40	11.519623	11.427265	12.428194
12	93863.75	127320.38	249839.44	Florida	141585.52	11.449600	11.754462	12.428574
13	91992.39	135495.07	252664.93	California	134307.35	11.429461	11.816691	12.439820
14	119943.24	156547.42	256512.92	Florida	132602.65	11.694774	11.961114	12.454934
15	114523.61	122616.84	261776.23	New York	129917.04	11.648536	11.716820	12.475245
16	78013.11	121597.55	264346.06	California	126992.93	11.264632	11.708472	12.485014
17	94657.16	145077.58	282574.31	New York	125370.37	11.458017	11.885024	12.551697
18	91749.16	114175.79	294919.57	Florida	124266.90	11.426814	11.645495	12.594458
19	86419.70	153514.11	0.00	New York	122776.86	11.366971	11.941548	-inf
20	76253.86	113867.30	298664.47	California	118474.03	11.241823	11.642789	12.607076
21	78389.47	153773.43	299737.29	New York	111313.02	11.269445	11.943236	12.610662
22	73994.56	122782.75	303319.26	Florida	110352.25	11.211747	11.718172	12.622541
23	67532.53	105751.03	304768.73	Florida	108733.99	11.120365	11.568843	12.627309
24	77044.01	99281.34	140574.81	New York	108552.04	11.252132	11.505713	11.853495
25	64664.71	139553.16	137962.62	California	107404.34	11.076971	11.846201	11.834738
26	75328.87	144135.98	134050.07	Florida	105733.54	11.229619	11.878512	11.805969
27	72107.60	127864.55	353183.81	New York	105008.31	11.185915	11.758727	12.774744
28	66051.52	182645.56	118148.20	Florida	103282.38	11.098190	12.115303	11.679695
29	65605.48	153032.06	107138.38	New York	101004.64	11.091415	11.938403	11.581877
30	61994.48	115641.28	91131.24	Florida	99937.59	11.034801	11.658248	11.420056
31	61136.38	152701.92	88218.23	New York	97483.56	11.020862	11.936243	11.387569
32	63408.86	129219.61	46085.25	California	97427.84	11.057359	11.769269	10.738248

	RDS	ADS	MKTS	State	Profit	log_RDS	log_ADS	log_MKTS
33	55493.95	103057.49	214634.81	Florida	96778.92	10.924029	11.543042	12.276693
34	46426.07	157693.92	210797.67	California	96712.80	10.745616	11.968411	12.258654
35	46014.02	85047.44	205517.64	New York	96479.51	10.736701	11.350964	12.233287
36	28663.76	127056.21	201126.82	Florida	90708.19	10.263389	11.752385	12.211691
37	44069.95	51283.14	197029.42	California	89949.14	10.693533	10.845117	12.191108
38	20229.59	65947.93	185265.10	New York	81229.06	9.914902	11.096621	12.129543
39	38558.51	82982.09	174999.30	California	81005.76	10.559932	11.326380	12.072537
40	28754.33	118546.05	172795.67	California	78239.91	10.266544	11.683057	12.059865
41	27892.92	84710.77	164470.71	Florida	77798.83	10.236128	11.346998	12.010488
42	23640.93	96189.63	148001.11	California	71498.49	10.070735	11.474077	11.904975
43	15505.73	127382.30	35534.17	New York	69758.98	9.648965	11.754948	10.478250
44	22177.74	154806.14	28334.72	California	65200.33	10.006844	11.949929	10.251843
45	1000.23	124153.04	1903.93	New York	64926.08	6.907985	11.729270	7.551675
46	1315.46	115816.21	297114.46	Florida	49490.75	7.181942	11.659760	12.601873
47	0.00	135426.92	0.00	California	42559.73	-inf	11.816187	-inf
48	542.05	51743.15	0.00	New York	35673.41	6.295358	10.854047	-inf
49	0.00	116983.80	45173.06	California	14681.40	-inf	11.669791	10.718256

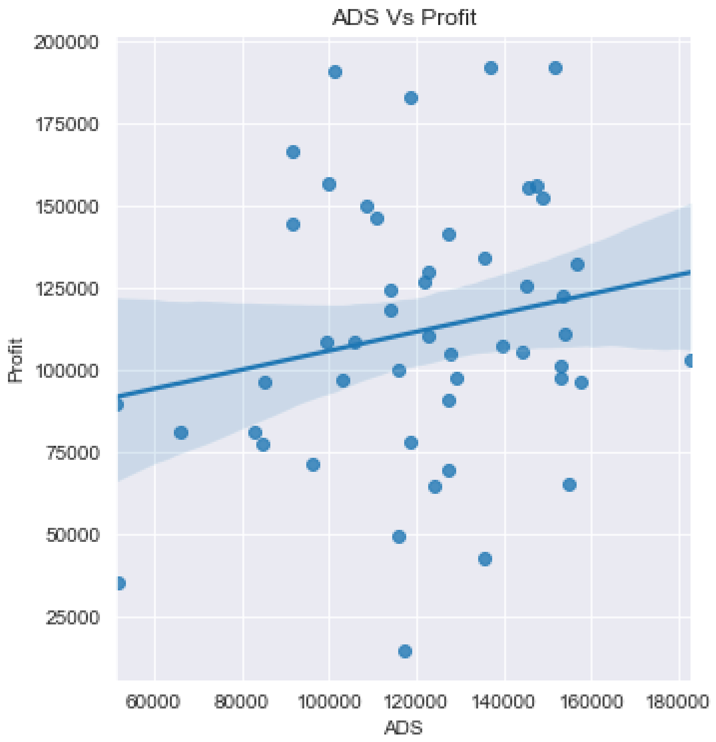
In [167]:

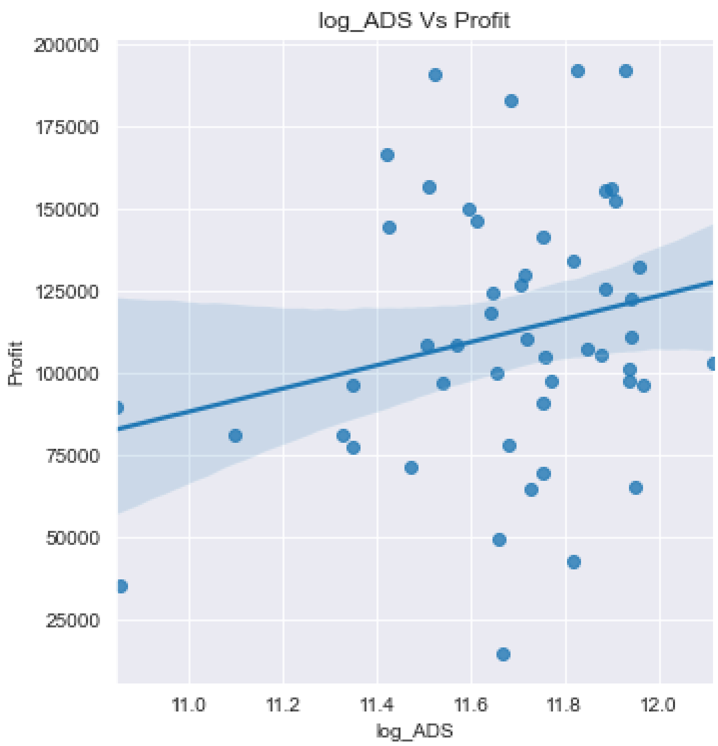
```
sns.lmplot(x='RDS',y='Profit',data=X_inputs)
plt.title('RDS Vs Profit')

sns.lmplot(x='log_RDS',y='Profit',data=X_inputs)
plt.title('log_RDS Vs Profit')
plt.show()
```



In [168]:

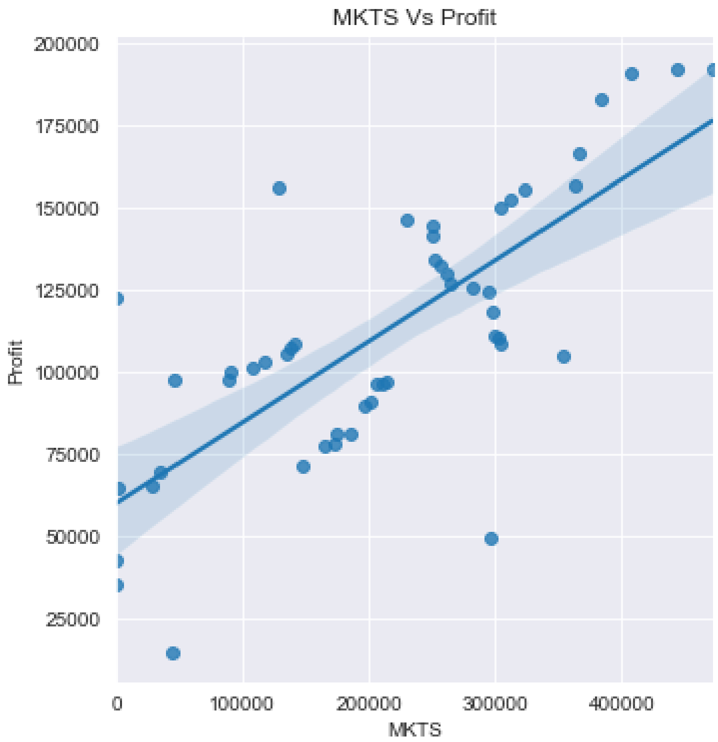
```
sns.lmplot(x='ADS',y='Profit',data=X_inputs)
plt.title('ADS Vs Profit')

sns.lmplot(x='log_ADS',y='Profit',data=X_inputs)
plt.title('log_ADS Vs Profit')
plt.show()
```

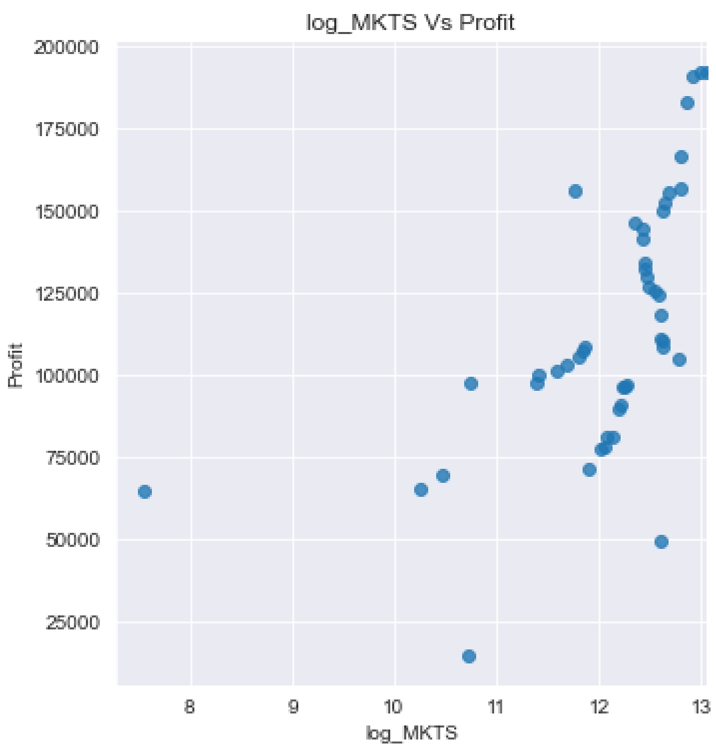




In [169]:

```
sns.lmplot(x='MKTS',y='Profit',data=X_inputs)
plt.title('MKTS Vs Profit')

sns.lmplot(x='log_MKTS',y='Profit',data=X_inputs)
plt.title('log_MKTS Vs Profit')
plt.show()
```





In [170]:

```
## Model Building
```

In [172]:

```
model = smf.ols('Profit~RDS+ADS+MKTS',data=dt).fit()
```

In [174]:

```
## Model Testing
```

In [173]:

```
model.params
```

Out[173]:

```
Intercept    50122.192990
RDS              0.805715
ADS             -0.026816
MKTS             0.027228
dtype: float64
```

In [175]:

```
#FINDING PVALUES AND TVALUES
print(model.tvalues, '\n', model.pvalues)
```

```
Intercept     7.626218
RDS          17.846374
ADS          -0.525507
MKTS          1.655077
dtype: float64
 Intercept    1.057379e-09
RDS          2.634968e-22
ADS          6.017551e-01
MKTS         1.047168e-01
dtype: float64
```

In [176]:

```
#R SQUARED VALUE
model.rsquared, model.rsquared_adj
```

Out[176]:

```
(0.9507459940683246, 0.9475337762901719)
```

Simple linear regression model

In [178]:

```
slr_1 = smf.ols('Profit~ADS' ,data=dt).fit()
slr_1.tvalues,slr_1.pvalues
#ADS HAS MORE SIGNIFICANT PVALUE
slr_1.summary()
```

Out[178]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.040
Model:	OLS	**Adj. R-squared:**	0.020
Method:	Least Squares	**F-statistic:**	2.015
Date:	Wed, 23 Feb 2022	**Prob (F-statistic):**	0.162
Time:	14:12:10	**Log-Likelihood:**	-599.63
No. Observations:	50	**AIC:**	1203.
Df Residuals:	48	**BIC:**	1207.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	7.697e+04	2.53e+04	3.040	0.004	2.61e+04	1.28e+05
ADS	0.2887	0.203	1.419	0.162	-0.120	0.698

Omnibus:	0.126	**Durbin-Watson:**	0.099
Prob(Omnibus):	0.939	**Jarque-Bera (JB):**	0.110
Skew:	0.093	**Prob(JB):**	0.947
Kurtosis:	2.866	**Cond. No.**	5.59e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.59e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

In [179]:

```
slr_2 = smf.ols('Profit~MKTS' ,data=dt).fit()
slr_2.tvalues,slr_1.pvalues #MKTC HAS MORE SIGNIFICANT PVALUE
slr_2.summary()
```

Out[179]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.559
Model:	OLS	**Adj. R-squared:**	0.550
Method:	Least Squares	**F-statistic:**	60.88
Date:	Wed, 23 Feb 2022	**Prob (F-statistic):**	4.38e-10
Time:	14:12:29	**Log-Likelihood:**	-580.18
No. Observations:	50	**AIC:**	1164.
Df Residuals:	48	**BIC:**	1168.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	6e+04	7684.530	7.808	0.000	4.46e+04	7.55e+04
MKTS	0.2465	0.032	7.803	0.000	0.183	0.310

Omnibus:	4.420	**Durbin-Watson:**	1.178
Prob(Omnibus):	0.110	**Jarque-Bera (JB):**	3.882
Skew:	-0.336	**Prob(JB):**	0.144
Kurtosis:	4.188	**Cond. No.**	4.89e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.89e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

In [181]:

```
slr_3= smf.ols('Profit~ADS+MKTS' ,data=dt).fit()
slr_3.tvalues,slr_1.pvalues #VARIABLES HAVE SIGNIFICANT PVALUES
slr_3.summary()
```

Out[181]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.610
Model:	OLS	**Adj. R-squared:**	0.593
Method:	Least Squares	**F-statistic:**	36.71
Date:	Wed, 23 Feb 2022	**Prob (F-statistic):**	2.50e-10
Time:	14:12:42	**Log-Likelihood:**	-577.13
No. Observations:	50	**AIC:**	1160.
Df Residuals:	47	**BIC:**	1166.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	2.022e+04	1.77e+04	1.143	0.259	-1.54e+04	5.58e+04
ADS	0.3237	0.131	2.468	0.017	0.060	0.588
MKTS	0.2488	0.030	8.281	0.000	0.188	0.309

Omnibus:	6.584	**Durbin-Watson:**	1.279
Prob(Omnibus):	0.037	**Jarque-Bera (JB):**	6.524
Skew:	-0.512	**Prob(JB):**	0.0383
Kurtosis:	4.443	**Cond. No.**	1.30e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.3e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

In [182]:

```
# Model validation techniques
```

In [183]:

```
## Two Techniques: 1. Collinearity Check
```

In [184]:

```
# 1) Collinearity Problem Check
# Calculate VIF = 1/(1-Rsquare) for all independent variables

rsq_r=smf.ols("RDS~ADS+MKTS",data=dt).fit().rsquared
vif_r=1/(1-rsq_r)

rsq_a=smf.ols("ADS~RDS+MKTS",data=dt).fit().rsquared
vif_a=1/(1-rsq_a)

rsq_m=smf.ols("MKTS~RDS+ADS",data=dt).fit().rsquared
vif_m=1/(1-rsq_m)

# Putting the values in Dataframe format
d1={'Variables':['RDS','ADS','MKTS'],'Vif':[vif_r,vif_a,vif_m]}
Vif_df=pd.DataFrame(d1)
Vif_df
```

Out[184]:

	Variables	Vif
0	RDS	2.468903
1	ADS	1.175091
2	MKTS	2.326773

In [185]:

```
# NONE VARIABLE HAS VID>20 , NO COLLINEARITY, SO CONSIDER ALL VARIABLE IN REGRESSION EQUATI
```

In [188]:

```
## 2. Residual test
###Q-Q plot
```

In [189]:

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

In [190]:

```
list(np.where(model.resid<-20000)) #OUTLIER DETECTION FROM ABOVE Q-Q PLOT OF RESIDUALS.
```

Out[190]:

```
[array([49], dtype=int64)]
```

In [191]:

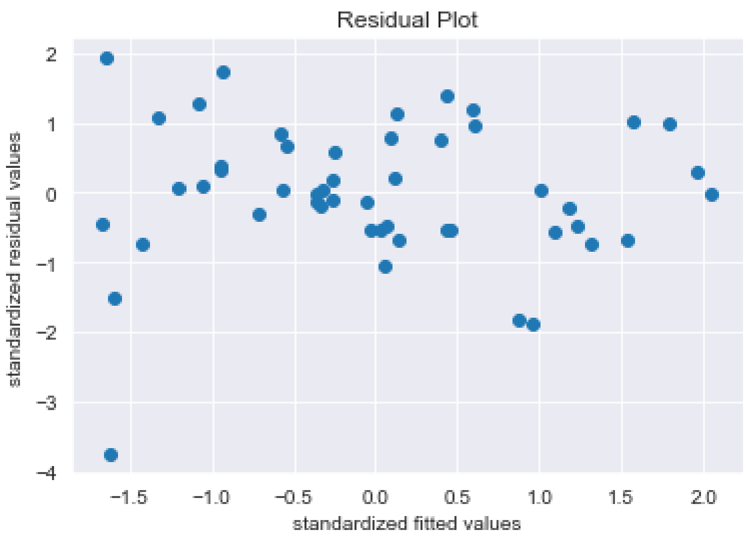
```
# Homoscedasticity or Heteroscedasticity
```

In [192]:

```
def standard_values( vals ):
    return (vals - vals.mean())/vals.std()
```

In [193]:

```
 plt.scatter(standard_values(model.fittedvalues),standard_values(model.resid))
plt.title('Residual Plot')
plt.xlabel('standardized fitted values')
plt.ylabel('standardized residual values')
plt.show()
```

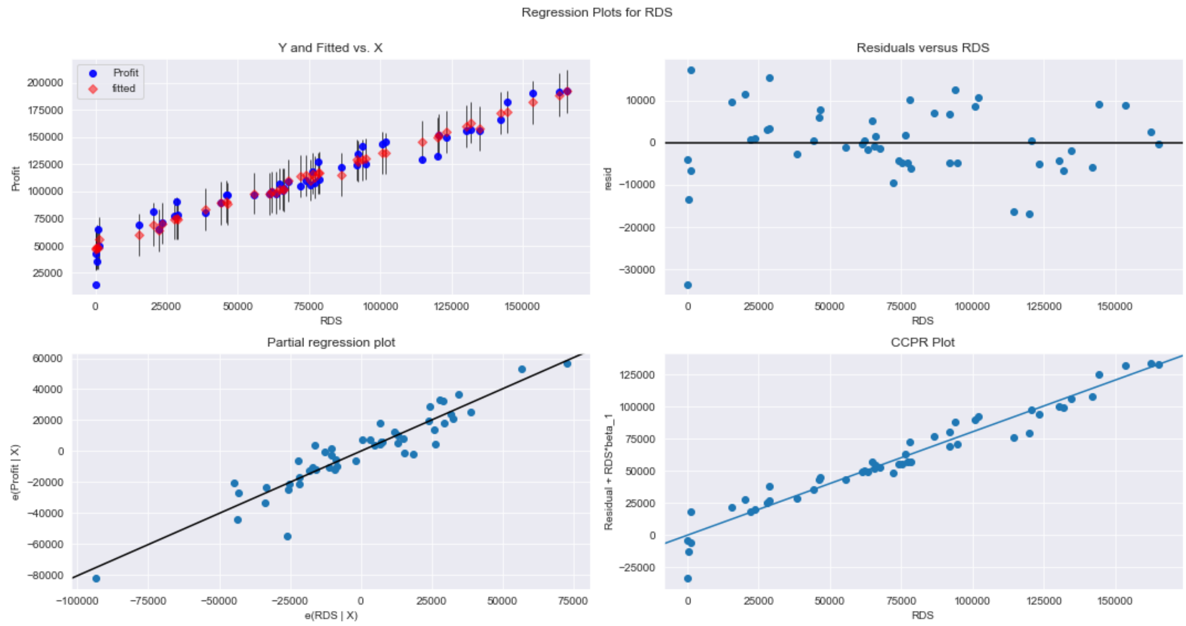


In [194]:

```
## Residuals Vs Regressor
```

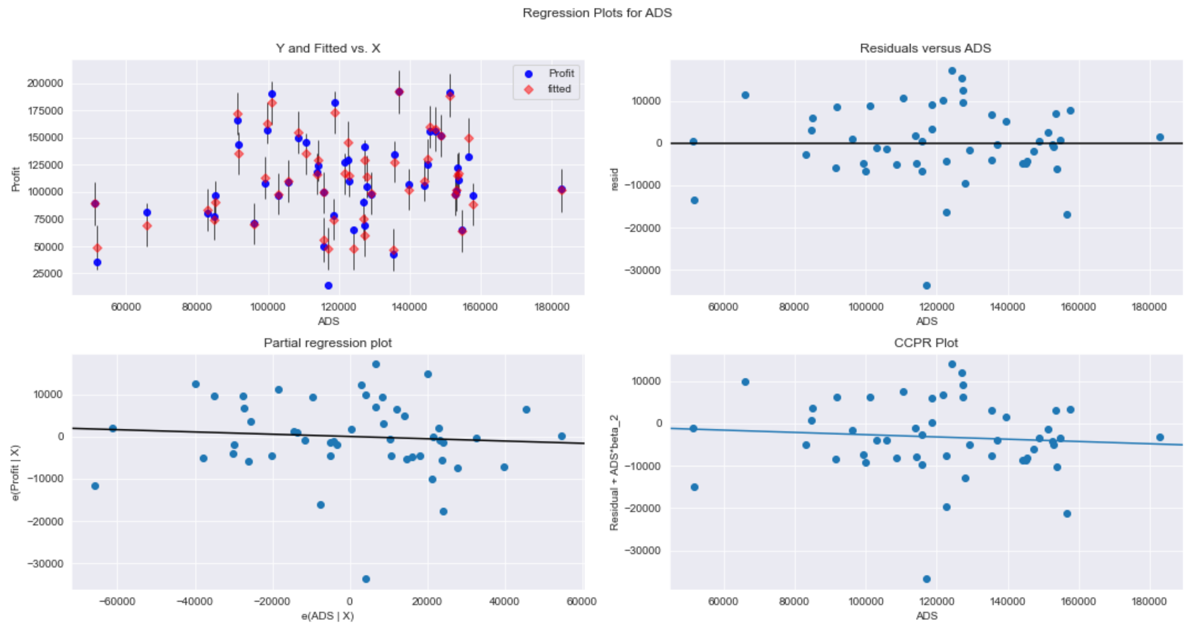
In [195]:

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



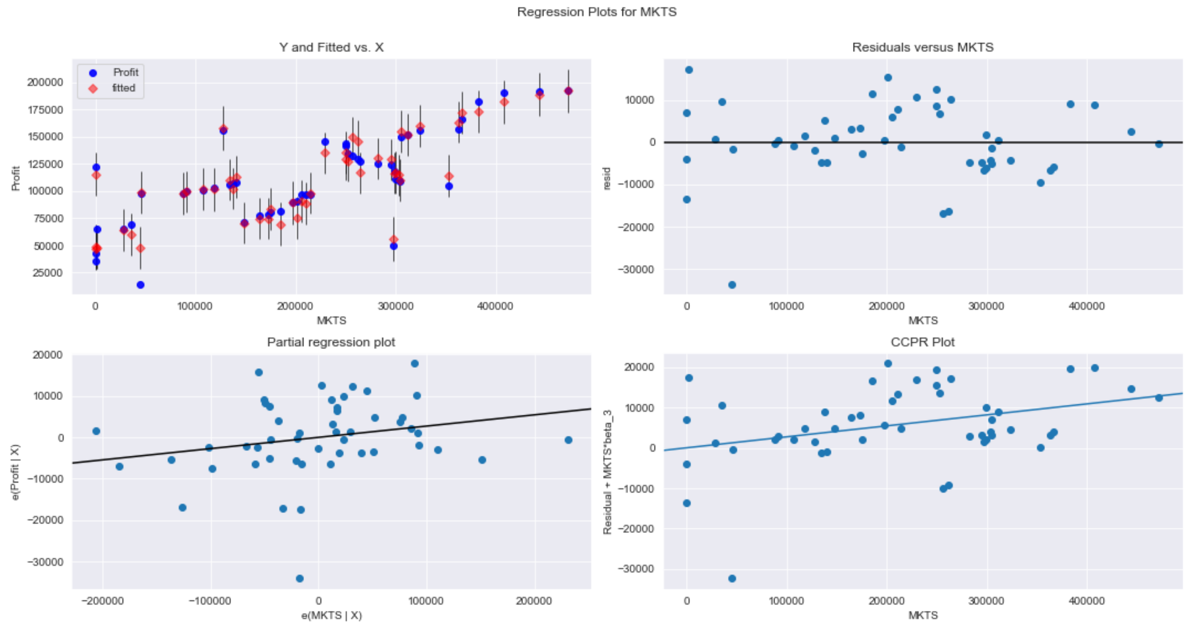
In [196]:

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



In [197]:

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



In [198]:

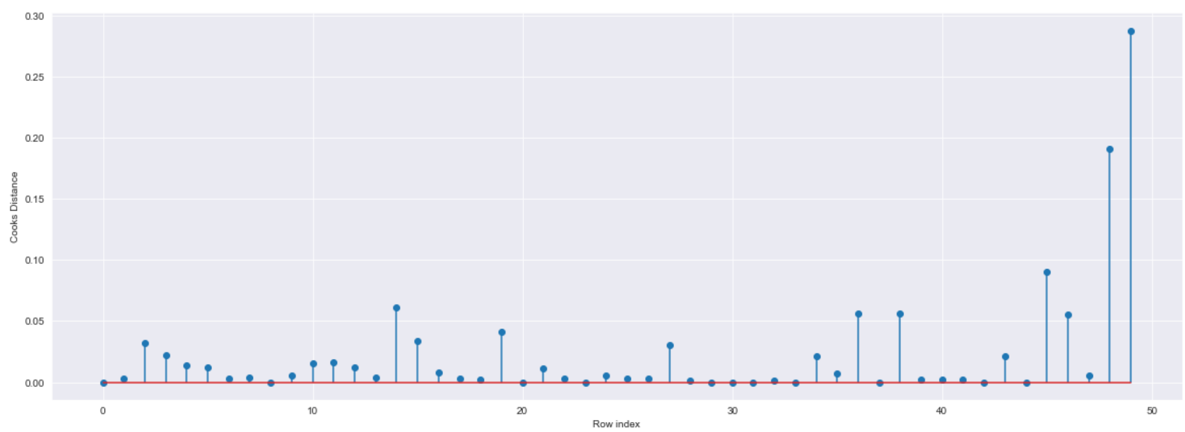
```
# Checking outliers or influencers
```

In [199]:

```
model_influence = model.get_influence()
(c, _) = model_influence.cooks_distance
```

In [201]:

```
#Plot the influencers values using stem plot
fig = plt.subplots(figsize=(20, 7))
plt.stem(np.arange(len(dt)), np.round(c, 3))
plt.xlabel('Row index')
plt.ylabel('Cooks Distance')
plt.show()
```



In [202]:

```
#index and value of influencer where c is more than .5
(np.argmax(c),np.max(c))
```

Out[202]:

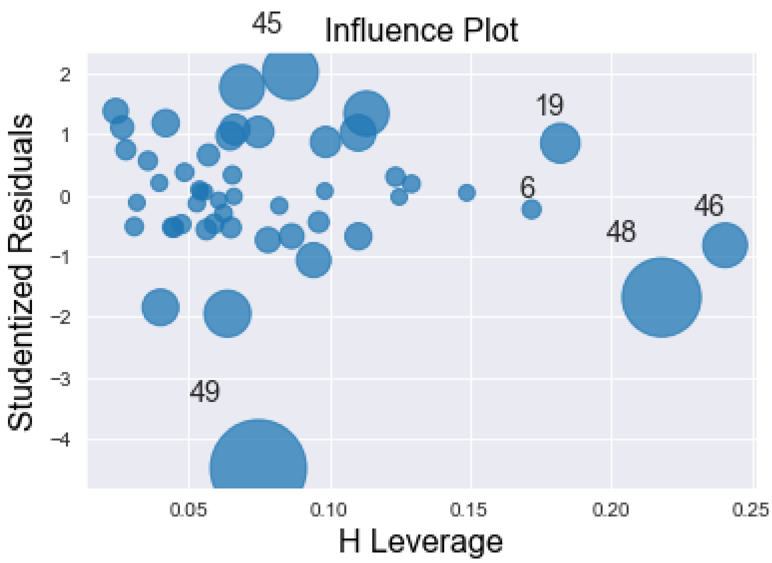
```
(49, 0.28808229275432634)
```

In [203]:

```
## 2. Leverage value
```

In [205]:

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



In [206]:

```
dt.shape
```

Out[206]:

```
(50, 5)
```

In [207]:

```
k = dt.shape[1]
n = dt.shape[0]
leverage_cutoff = 3*((k + 1)/n)
leverage_cutoff
```

Out[207]:

```
0.36
```

In [208]:

```
dt[dt.index.isin([49])]
```

Out[208]:

	RDS	ADS	MKTS	State	Profit
49	0.0	116983.8	45173.06	California	14681.4

Improving the model

In [209]:

```
dt1 = dt.copy()
dt1
```

Out[209]:

	RDS	ADS	MKTS	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

	RDS	ADS	MKTS	State	Profit
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
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 [210]:

```
dt2=dt1.drop(dt1.index[[49]],axis=0).reset_index(drop=True)
dt2
```

Out[210]:

	RDS	ADS	MKTS	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

	RDS	ADS	MKTS	State	Profit
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
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

In [211]:

```
fnl_data = smf.ols('Profit~ADS+RDS+MKTS',data=dt2).fit()
fnl_data.summary()
```

Out[211]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.961
Model:	OLS	**Adj. R-squared:**	0.959
Method:	Least Squares	**F-statistic:**	372.8
Date:	Wed, 23 Feb 2022	**Prob (F-statistic):**	8.85e-32
Time:	14:20:15	**Log-Likelihood:**	-506.28
No. Observations:	49	**AIC:**	1021.
Df Residuals:	45	**BIC:**	1028.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	5.238e+04	5542.657	9.451	0.000	4.12e+04	6.35e+04
ADS	-0.0222	0.043	-0.518	0.607	-0.109	0.064
RDS	0.7830	0.038	20.470	0.000	0.706	0.860
MKTS	0.0252	0.014	1.825	0.075	-0.003	0.053

Omnibus:	0.082	**Durbin-Watson:**	1.598
Prob(Omnibus):	0.960	**Jarque-Bera (JB):**	0.232
Skew:	-0.082	**Prob(JB):**	0.890
Kurtosis:	2.706	**Cond. No.**	1.41e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.41e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

In [213]:

```
# R squared values
```

In [212]:

```
fnl_data.rsquared ,fnl_data.rsquared_adj
```

Out[212]:

```
(0.9613162435129847, 0.9587373264138503)
```

In [214]:

```
# Predicting for new data
```

In [215]:

```
new_data=pd.DataFrame({'RDS':50000,"ADS":90000,"MKTS":180000},index=[0])
new_data
```

Out[215]:

	RDS	ADS	MKTS
0	50000	90000	180000

In [216]:

```
## For manual Prediction
```

In [217]:

```
fnl_data.predict(new_data)
```

Out[217]:

```
0    94076.462322
dtype: float64
```

In [219]:

```
pred_y=fnl_data.predict(dt2)
pred_y
```

Out[219]:

```
0     190716.676999
1     187537.122227
2     180575.526396
3     172461.144642
4     170863.486721
5     162582.583177
6     157741.338633
7     159347.735318
8     151328.826941
9     154236.846778
10    135507.792682
11    135472.855621
12    129355.599449
13    127780.129139
14    149295.404796
15    145937.941975
16    117437.627921
17    130408.626295
18    129129.234457
19    116641.003121
20    117097.731866
21    117911.019038
22    115248.217796
23    110603.139045
24    114051.073877
25    103398.054385
26    111547.638935
27    114916.165026
28    103027.229434
29    103057.621761
30    100656.410227
31     99088.213693
32    100325.741335
33     98962.303136
34     90552.307809
35     91709.288672
36     77080.554255
37     90722.503244
38     71433.021956
39     85147.375646
40     76625.510303
41     76492.145175
42     72492.394974
43     62592.049718
44     67025.731107
45     50457.297206
46     58338.443625
47     49375.776655
48     51658.096812
dtype: float64
```

In [223]:

```
## Table containing R^2 value
```

In [222]:

```
d2={'Prep_Models':['Model','Fnl_Model'],'Rsquared':[model.rsquared,fnl_data.rsquared]}
table=pd.DataFrame(d2)
table
```

Out[222]:

	Prep_Models	Rsquared
0	Model	0.950746
1	Fnl_Model	0.961316

In [ ]:

```

```