Import neccessery libraries

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from numpy.polynomial.polynomial import polyfit
from sklearn.linear_model import LinearRegression
import seaborn as sns
import statsmodels.stats.tests.test_influence
from sklearn.feature_selection import RFE
from statsmodels.stats.outliers_influence import variance_inflation_factor
import math
```

Problem

Prepare a prediction model for profit of 50startups data Do transformations for getting better predictions of profit and make a table containing R^2 value for each prepared model

Import data

In [3]:
```
import os
```

In [4]:
```
os.getcwd()
```

Out[4]:
```
'C:\\Users\\Akarsh\\assignment-5'
```

In [5]:
```
os.chdir('C:\\Users\\Akarsh\\Desktop\\assignments\\multiple linear regressi
```

In [6]:
```
os.getcwd()
```

Out[6]:
```
'C:\\Users\\Akarsh\\Desktop\\assignments\\multiple linear regression'
```

In [7]:
```
startup_data = pd.read_csv('50_Startups.csv')
startup_data
```

Out[7]:

	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

	R&D Spend	Administration	Marketing Spend	State	Profit
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

	R&D Spend	Administration	Marketing Spend	State	Profit
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

Data understanding

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

Out[8]:
```
(50, 5)
```

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

Out[9]:
```
R&D Spend          0
Administration     0
Marketing Spend    0
State              0
Profit             0
dtype: int64
```

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

Out[10]:
```
R&D Spend          float64
Administration     float64
Marketing Spend    float64
State               object
Profit             float64
dtype: object
```

In [11]:
```
startup_data.describe()
```

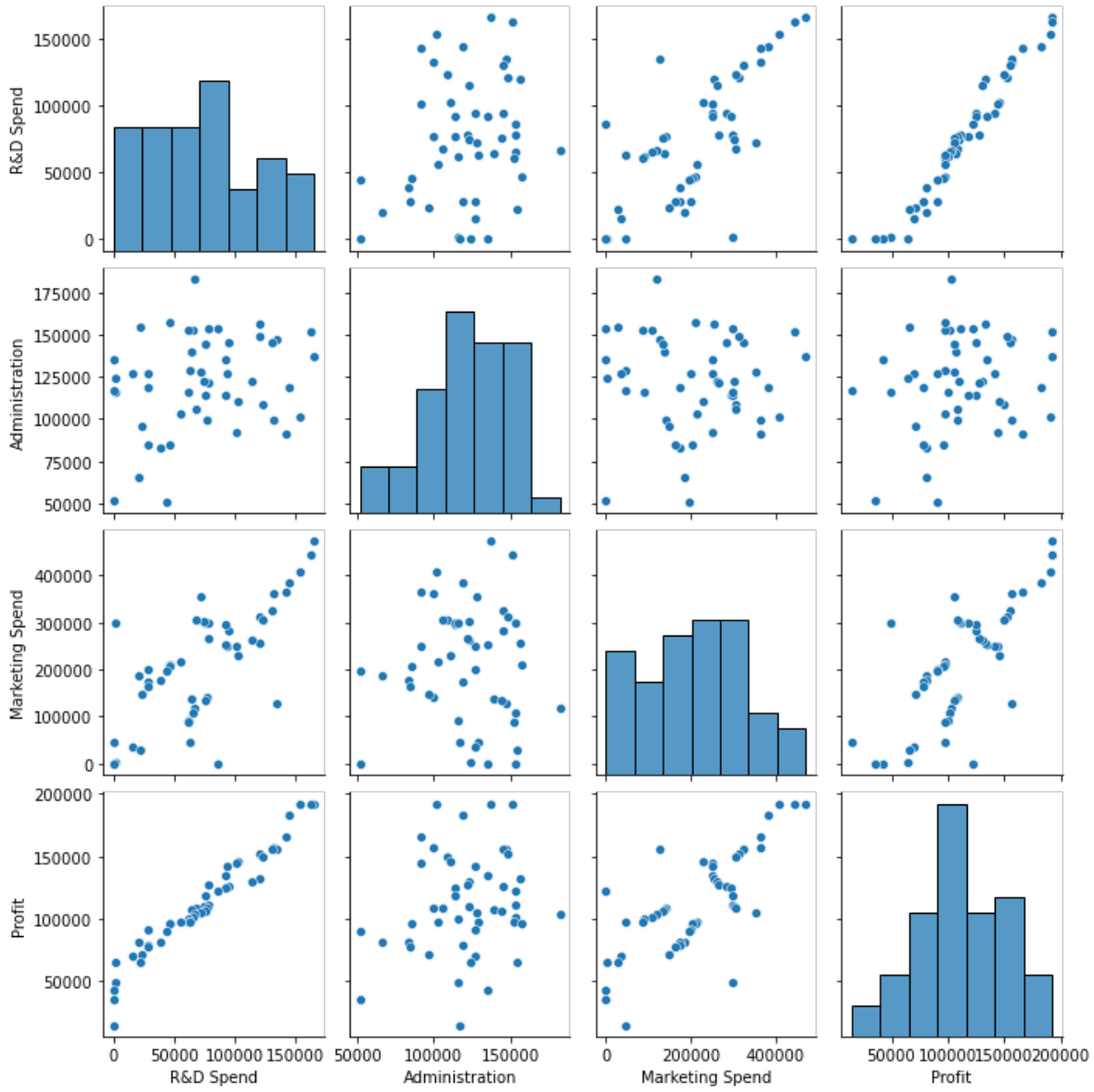
Out[11]:

	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

graphical representation of startup_data

In [12]:

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



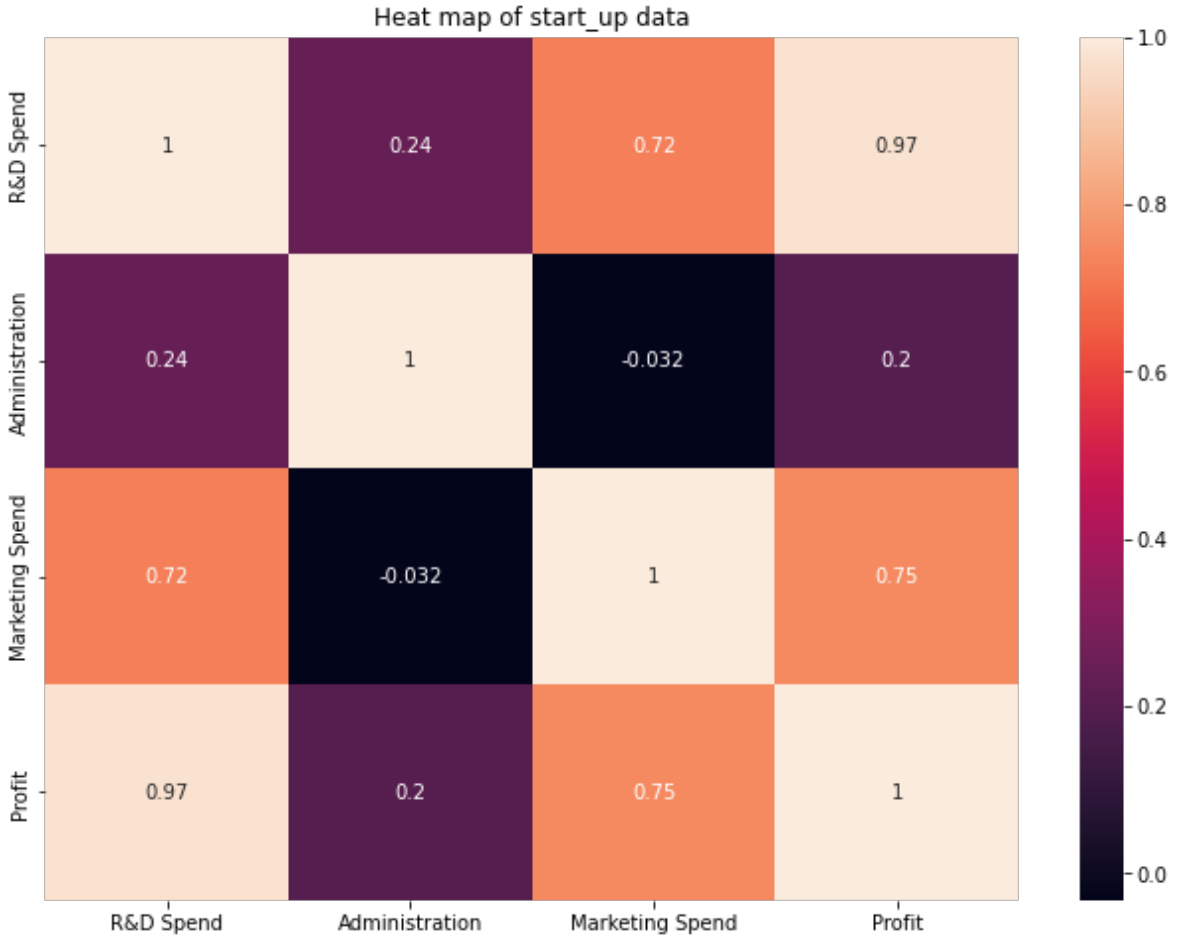
Correlation matrix

In [13]:

```
corrMatrix = startup_data.corr()
```

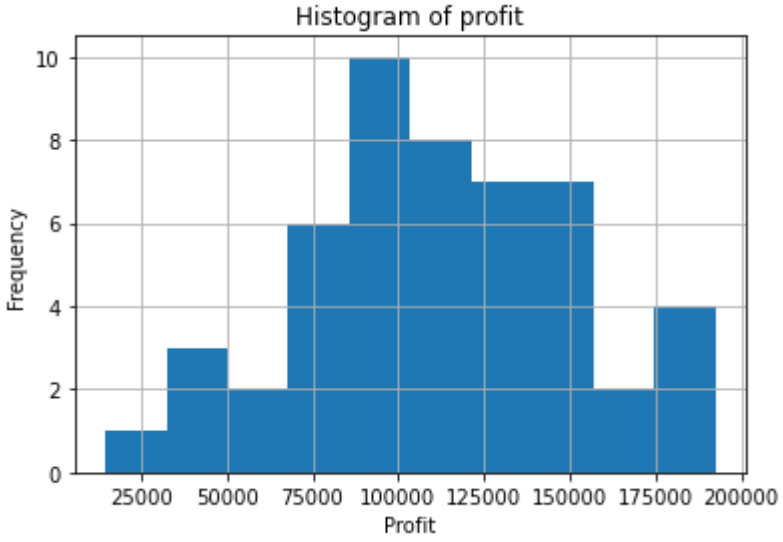
In [14]:

```
plt.figure(figsize=(11,8))
plt.title('Heat map of start_up data')
sns.heatmap(corrMatrix, annot=True)
plt.show()
```



In [17]:
```
startup_data.Profit.hist()
plt.title('Histogram of profit')
plt.xlabel('Profit')
plt.ylabel('Frequency')
```

Out[17]:
```
Text(0, 0.5, 'Frequency')
```



Regression model

In [19]:
```
startup_data_2 = pd.get_dummies(startup_data, columns=['State'])
```

In [20]:
```
X = startup_data_2[['R&D Spend','Administration', 'Marketing Spend', 'State
Y = startup_data_2[['Profit']]
```

In [21]:
```
model = sm.OLS(Y, X).fit()
predictions = model.predict(X)
model.summary()
```

Out[21]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.951
Model:	OLS	**Adj. R-squared:**	0.945
Method:	Least Squares	**F-statistic:**	169.9
Date:	Wed, 16 Feb 2022	**Prob (F-statistic):**	1.34e-27
Time:	10:40:12	**Log-Likelihood:**	-525.38
No. Observations:	50	**AIC:**	1063.
Df Residuals:	44	**BIC:**	1074.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
R&D Spend	0.8060	0.046	17.369	0.000	0.712	0.900
Administration	-0.0270	0.052	-0.517	0.608	-0.132	0.078
Marketing Spend	0.0270	0.017	1.574	0.123	-0.008	0.062
State_California	5.013e+04	6884.820	7.281	0.000	3.62e+04	6.4e+04
State_Florida	5.032e+04	7251.767	6.940	0.000	3.57e+04	6.49e+04
State_New York	5.008e+04	6952.587	7.204	0.000	3.61e+04	6.41e+04

Omnibus:	14.782	**Durbin-Watson:**	1.283
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	21.266
Skew:	-0.948	**Prob(JB):**	2.41e-05
Kurtosis:	5.572	**Cond. No.**	2.45e+06

Notes:

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

[2] The condition number is large, 2.45e+06. This might indicate that there are strong multicollinearity or other numerical problems.

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

Multicollinearity

Finding Cook's Distance

In [23]:
```
influence = model.get_influence()
```

In [24]:
```
summ_data = infl.summary_frame()
```

In [25]:
```
summ_data.sort_values('cooks_d', ascending=False)
```

Out[25]:

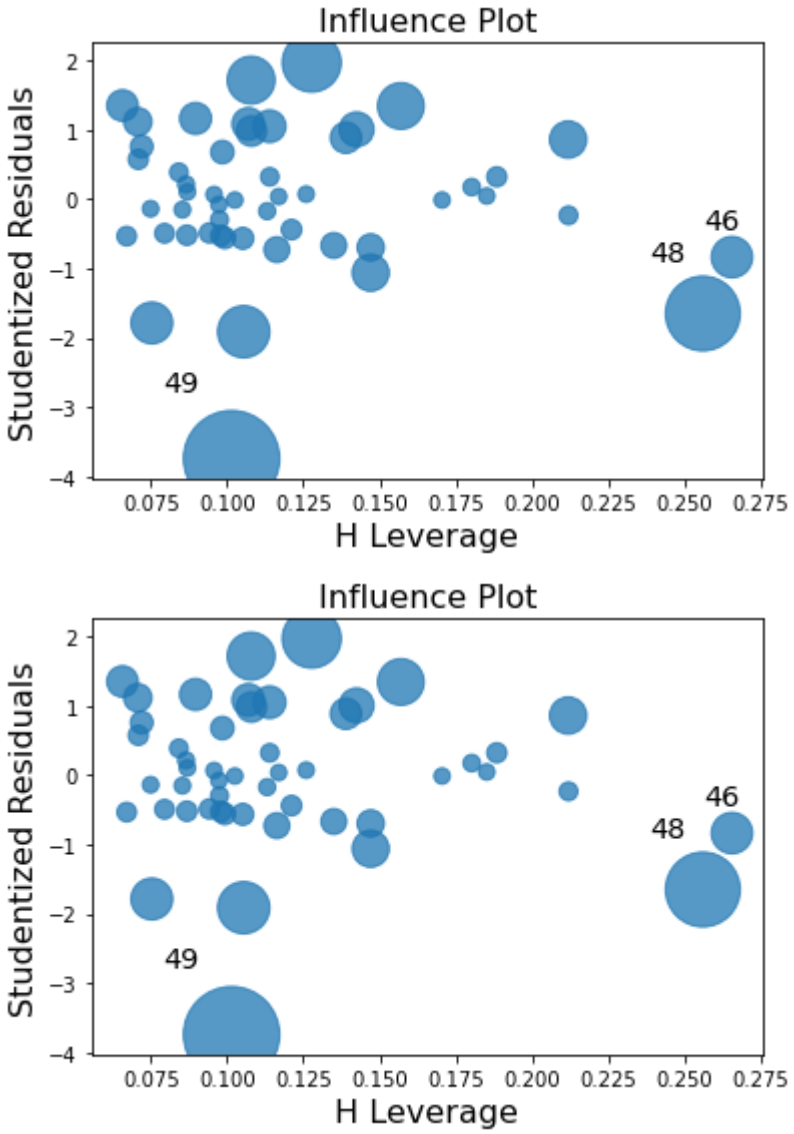
	dfb_R&D Spend	dfb_Administration	dfb_Marketing Spend	dfb_State_California	dfb_State_Florida	dfb_Stat
49	0.578956	-0.114232	0.080954	-0.566028	-0.246221	-0.
48	-0.112734	0.701599	0.418630	-0.783828	-0.801849	-0.
45	-0.212843	0.091394	-0.189969	0.095382	0.140857	0.
14	-0.221204	-0.257240	0.142195	0.267421	0.086725	0.
36	-0.379353	0.189523	0.218405	-0.107545	0.053174	-0.
38	-0.189819	-0.313449	0.109261	0.320201	0.309091	0.
15	-0.208289	0.066627	0.071114	-0.002577	0.007587	-0.
46	0.434369	-0.142646	-0.364064	0.106828	0.034265	0.
19	0.252210	0.039342	-0.342025	0.009492	0.035168	0.
27	0.271462	-0.146112	-0.339679	0.169919	0.186504	0.
2	0.197811	-0.174765	-0.013702	0.080109	0.147293	0.
3	0.110000	-0.049701	0.073783	-0.035947	-0.056370	0.
43	-0.090858	0.058906	-0.085052	0.024294	0.044997	0.
10	0.186811	-0.137400	-0.159583	0.116978	0.216711	0.
12	0.069420	0.010085	-0.048411	-0.017490	0.098185	-0.
34	-0.196836	0.242310	0.173390	-0.138400	-0.212207	-0.
11	0.152695	-0.197500	-0.063320	0.226735	0.131358	0.
16	-0.055403	0.029857	0.116725	0.037230	-0.065807	-0.
4	-0.153851	0.163418	0.047737	-0.104342	-0.152925	-0.
5	-0.081405	0.105665	-0.033055	-0.046292	-0.031296	-0.
21	0.134008	-0.163215	-0.156986	0.162458	0.165701	0.
35	-0.049889	-0.102221	0.034443	0.099727	0.095288	0.
13	0.005568	0.054245	0.040526	-0.007538	-0.074145	-0.
9	-0.087676	0.064819	0.009105	-0.073333	-0.016881	-0.
26	-0.047078	-0.040934	0.086363	0.018763	-0.038037	0.

	dfb_R&D Spend	dfb_Administration	dfb_Marketing Spend	dfb_State_California	dfb_State_Florida	dfb_Stat
24	-0.077455	0.089257	0.087987	-0.087996	-0.089711	-0.
47	0.046774	-0.046937	0.031389	-0.029301	-0.001347	0.
17	0.035255	-0.075381	-0.061166	0.081487	0.083661	0.
7	-0.045130	-0.041183	0.001538	0.056695	0.017003	0.
25	0.012139	0.046367	-0.028528	0.012690	-0.030258	-0.
1	0.020841	0.046268	0.058280	-0.049608	-0.085419	-0.
22	0.057093	-0.025665	-0.065599	0.028443	-0.010225	0.
18	-0.003027	0.019777	-0.017913	-0.008655	-0.050050	-0.
6	-0.095187	0.004676	0.082074	-0.021299	-0.002912	0.
40	-0.063137	0.020142	0.044268	0.021070	-0.010930	-0.
41	-0.022659	-0.051009	-0.006525	0.059567	0.087625	0.
39	0.011576	0.050925	-0.005414	-0.074335	-0.047479	-0.
28	-0.004892	0.054512	-0.014043	-0.041421	-0.021855	-0.
20	-0.018937	0.001269	0.034093	0.009204	-0.011262	-0.
32	-0.029851	0.005201	0.042128	-0.025767	-0.013770	-0.
23	0.017345	0.004924	-0.019954	-0.003001	-0.014564	-0.
33	0.005720	0.010764	-0.000214	-0.012256	-0.024085	-0.
42	-0.010791	-0.008202	0.004902	0.018523	0.009901	0.
44	-0.005257	0.013394	-0.004732	-0.001883	-0.006265	-0.
29	-0.001210	-0.011523	0.007045	0.007881	0.006612	0.
8	0.000992	0.008446	0.005091	-0.010483	-0.010897	-0.
37	0.001322	-0.016087	-0.000850	0.018397	0.013923	0.
30	0.004911	-0.003183	-0.008237	0.004519	0.008048	0.

In [26]:

```
infl.plot_influence()
```

Out[26]:



Index 48 and 49 has highest Cook's Distance

Finding Variance Inflation Factor (VIF)

In [29]:
```
vif = pd.DataFrame()
```

In [38]:
```
vif["VIF Factor"] = [variance_inflation_factor(X.values, i)for i in range(X
vif["features"] = X.columns
```

In [39]:
```
vif.round(1)
```

Out[39]:

	VIF Factor	features
0	2.5	R&D Spend
1	1.2	Administration
2	2.4	Marketing Spend
3	9.0	State_California

	VIF Factor	features
4	9.4	State_Florida

In [40]:
```
vif.round(2)
```

Out[40]:

	VIF Factor	features
0	2.50	R&D Spend
1	1.18	Administration
2	2.42	Marketing Spend
3	9.04	State_California
4	9.44	State_Florida
5	9.22	State_New York

Administration has the lowest variance inflation factor We would need to discard this variables to improve model and try to solve multicolinearity.

In [42]:
```
# Removed administration from the dataframe and Test the model
new_X = startup_data_2[['R&D Spend', 'Marketing Spend', 'State_California',
```

In [44]:
```
new_model = sm.OLS(Y, new_X).fit()
new_predictions = new_model.predict(new_X)
new_model.summary()
```

Out[44]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.950
Model:	OLS	**Adj. R-squared:**	0.946
Method:	Least Squares	**F-statistic:**	215.8
Date:	Wed, 16 Feb 2022	**Prob (F-statistic):**	9.72e-29
Time:	11:01:13	**Log-Likelihood:**	-525.53
No. Observations:	50	**AIC:**	1061.
Df Residuals:	45	**BIC:**	1071.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
R&D Spend	0.7967	0.042	18.771	0.000	0.711	0.882
Marketing Spend	0.0298	0.016	1.842	0.072	-0.003	0.062
State_California	4.696e+04	3119.471	15.053	0.000	4.07e+04	5.32e+04
State_Florida	4.71e+04	3670.129	12.833	0.000	3.97e+04	5.45e+04
State_New York	4.694e+04	3342.591	14.043	0.000	4.02e+04	5.37e+04

Omnibus:	14.640	**Durbin-Watson:**	1.257
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	21.037
Skew:	-0.938	**Prob(JB):**	2.70e-05
Kurtosis:	5.565	**Cond. No.**	9.45e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.45e+05. This might indicate that there are strong multicollinearity or other numerical problems.

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

In [48]:
```
# Removed Index with highest Cook's distance to remove the hightest influe
new_df = startup_data_2.drop(startup_data_2.index[[49,48]])
```

In [49]:
```
new_X = new_df[['R&D Spend', 'Marketing Spend', 'State_California', 'State_
new_Y = new_df[['Profit']]
```

In [50]:
```
final_model = sm.OLS(new_Y, new_X).fit()
predictions = final_model.predict(new_X)
final_model.summary()
```

Out[50]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.961
Model:	OLS	**Adj. R-squared:**	0.958
Method:	Least Squares	**F-statistic:**	265.9
Date:	Wed, 16 Feb 2022	**Prob (F-statistic):**	1.02e-29
Time:	11:06:14	**Log-Likelihood:**	-494.30
No. Observations:	48	**AIC:**	998.6
Df Residuals:	43	**BIC:**	1008.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
R&D Spend	0.7692	0.035	22.072	0.000	0.699	0.840
Marketing Spend	0.0251	0.013	1.908	0.063	-0.001	0.052
State_California	5.183e+04	2710.866	19.120	0.000	4.64e+04	5.73e+04
State_Florida	5.046e+04	3078.590	16.391	0.000	4.43e+04	5.67e+04

State_New York	5.09e+04	2936.767	17.333	0.000	4.5e+04	5.68e+04

Omnibus:	0.133	**Durbin-Watson:**	1.645
Prob(Omnibus):	0.936	**Jarque-Bera (JB):**	0.304
Skew:	0.097	**Prob(JB):**	0.859
Kurtosis:	2.661	**Cond. No.**	1.02e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.02e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

Now even through the Model has accpetable p Value and R sq value, we can still improve the R squre value.

Sq Root Transformation of X

In [51]:
```
X_sqrt = np.sqrt(new_df[['R&D Spend', 'Marketing Spend', 'State_California
```

In [59]:
```
model_3 = sm.OLS(new_Y, X_sqrt).fit()
predictions_3 = model_3.predict(X_sqrt)
model_3.summary()
```

Out[59]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.887
Model:	OLS	**Adj. R-squared:**	0.877
Method:	Least Squares	**F-statistic:**	84.44
Date:	Wed, 16 Feb 2022	**Prob (F-statistic):**	8.67e-20
Time:	11:12:52	**Log-Likelihood:**	-519.91
No. Observations:	48	**AIC:**	1050.
Df Residuals:	43	**BIC:**	1059.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
R&D Spend	340.5455	25.777	13.211	0.000	288.560	392.531
Marketing Spend	20.0497	15.481	1.295	0.202	-11.170	51.270
State_California	1.836e+04	6267.224	2.930	0.005	5724.219	3.1e+04
State_Florida	1.692e+04	7013.669	2.413	0.020	2779.320	3.11e+04
State_New York	1.908e+04	6591.247	2.894	0.006	5782.772	3.24e+04

Omnibus:	7.588	**Durbin-Watson:**	0.777
Prob(Omnibus):	0.023	**Jarque-Bera (JB):**	7.161
Skew:	0.941	**Prob(JB):**	0.0279
Kurtosis:	3.197	**Cond. No.**	3.04e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.04e+03. This might indicate that there are strong multicollinearity or other numerical problems.

- Square Root Transformation of Y

In [53]:
```
Y_sqrt = np.sqrt(new_df['Profit'])
```

In [57]:
```
model_4 = sm.OLS(Y_sqrt, new_X).fit()
predictions_4 = model_4.predict(new_X)
model_4.summary()
```

Out[57]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.954
Model:	OLS	**Adj. R-squared:**	0.950
Method:	Least Squares	**F-statistic:**	223.3
Date:	Wed, 16 Feb 2022	**Prob (F-statistic):**	3.68e-28
Time:	11:12:20	**Log-Likelihood:**	-185.87
No. Observations:	48	**AIC:**	381.7
Df Residuals:	43	**BIC:**	391.1
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
R&D Spend	0.0012	5.64e-05	20.622	0.000	0.001	0.001
Marketing Spend	2.473e-05	2.13e-05	1.159	0.253	-1.83e-05	6.78e-05
State_California	241.0032	4.390	54.894	0.000	232.149	249.857
State_Florida	240.7325	4.986	48.283	0.000	230.678	250.787
State_New York	240.9886	4.756	50.669	0.000	231.397	250.580

Omnibus:	4.530	**Durbin-Watson:**	1.406
Prob(Omnibus):	0.104	**Jarque-Bera (JB):**	3.371
Skew:	-0.532	**Prob(JB):**	0.185

Kurtosis:	3.745	**Cond. No.**	1.02e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.02e+06. This might indicate that there are

Square Root Transformation of X & Y

In [58]:
```
model_5 = sm.OLS(Y_sqrt, X_sqrt).fit()
predictions_5 = model_5.predict(X_sqrt)
model_5.summary()
```

Out[58]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.929
Model:	OLS	**Adj. R-squared:**	0.923
Method:	Least Squares	**F-statistic:**	141.7
Date:	Wed, 16 Feb 2022	**Prob (F-statistic):**	3.64e-24
Time:	11:12:27	**Log-Likelihood:**	-196.16
No. Observations:	48	**AIC:**	402.3
Df Residuals:	43	**BIC:**	411.7
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
R&D Spend	0.5271	0.030	17.371	0.000	0.466	0.588
Marketing Spend	0.0231	0.018	1.270	0.211	-0.014	0.060
State_California	187.8689	7.377	25.465	0.000	172.991	202.747
State_Florida	187.0162	8.256	22.652	0.000	170.366	203.666
State_New York	189.8076	7.759	24.463	0.000	174.160	205.455

Omnibus:	7.976	**Durbin-Watson:**	1.243
Prob(Omnibus):	0.019	**Jarque-Bera (JB):**	7.007
Skew:	0.870	**Prob(JB):**	0.0301
Kurtosis:	3.692	**Cond. No.**	3.04e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.04e+03. This might indicate that there are strong multicollinearity or other numerical problems.

We will use Model Square Root Transformation of X & Y as it has the best R square value 1 - p-value < 0.01

Thus the model is accepted

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

3 - Adj. R-sqared = 1 Thus the model explains 100% of the variance in dependent

variable

In [ ]: