In [1]:

```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:

```python
startups = pd.read_csv('50_Startups.csv')
df = startups.copy()
df
```

Out[3]:

	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
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 [4]:

```python
df.head()
```

Out[4]:

	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 [5]:

```python
df.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 [7]:

```python
df.shape
```

Out[7]:

```
(50, 5)
```

In [8]:

```python
df.isna().sum()
```

Out[8]:

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

In [9]:

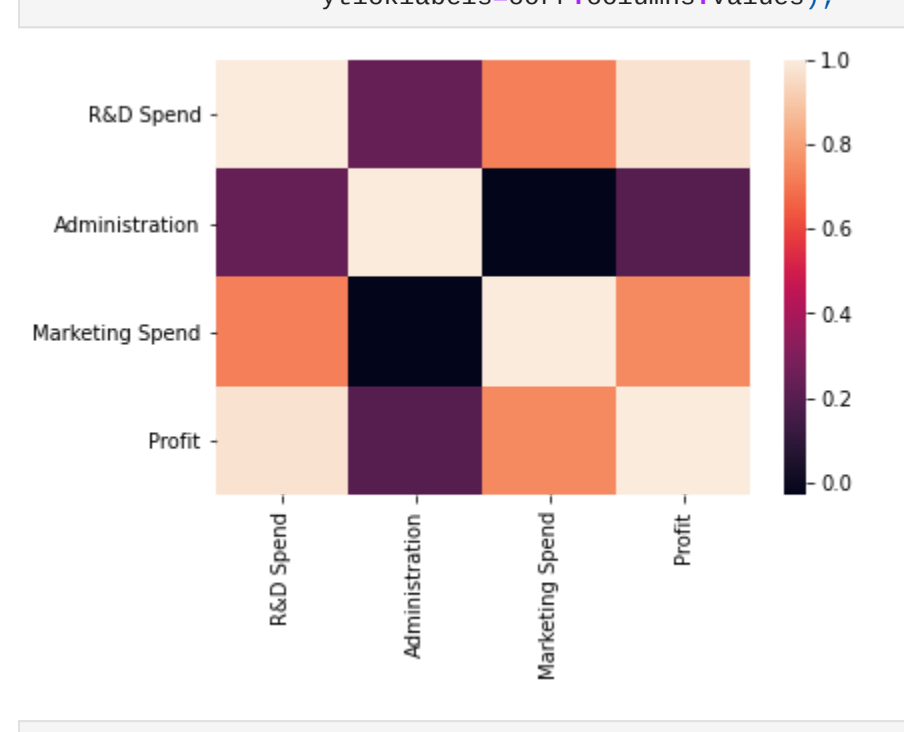
```python
corr = df.corr()
corr
```

Out[9]:

	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 [10]:

```python
sns.heatmap(corr, xticklabels=corr.columns.values,
                  yticklabels=corr.columns.values);
```

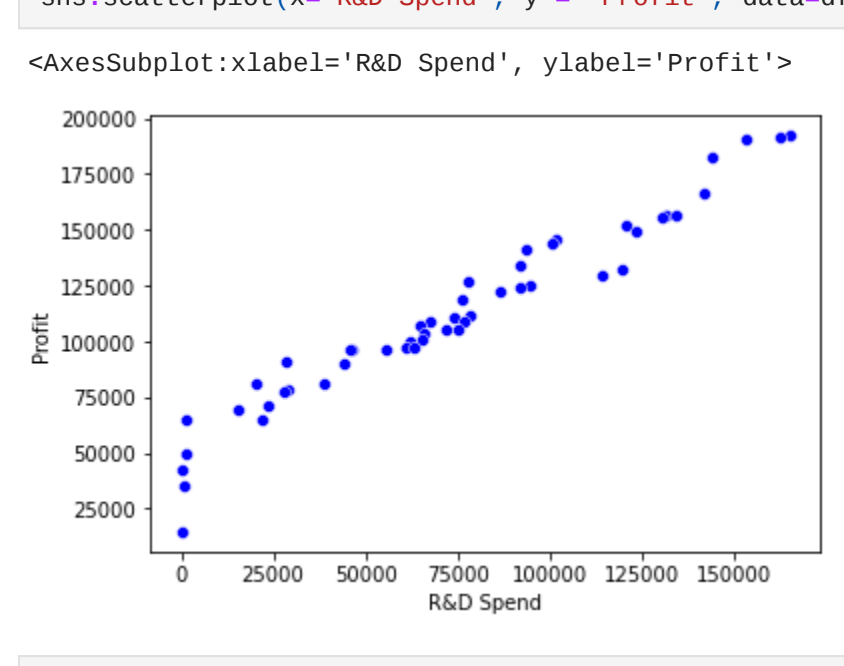


In [12]:

```python
sns.scatterplot(x="R&D Spend", y = "Profit", data=df, color="blue")
```

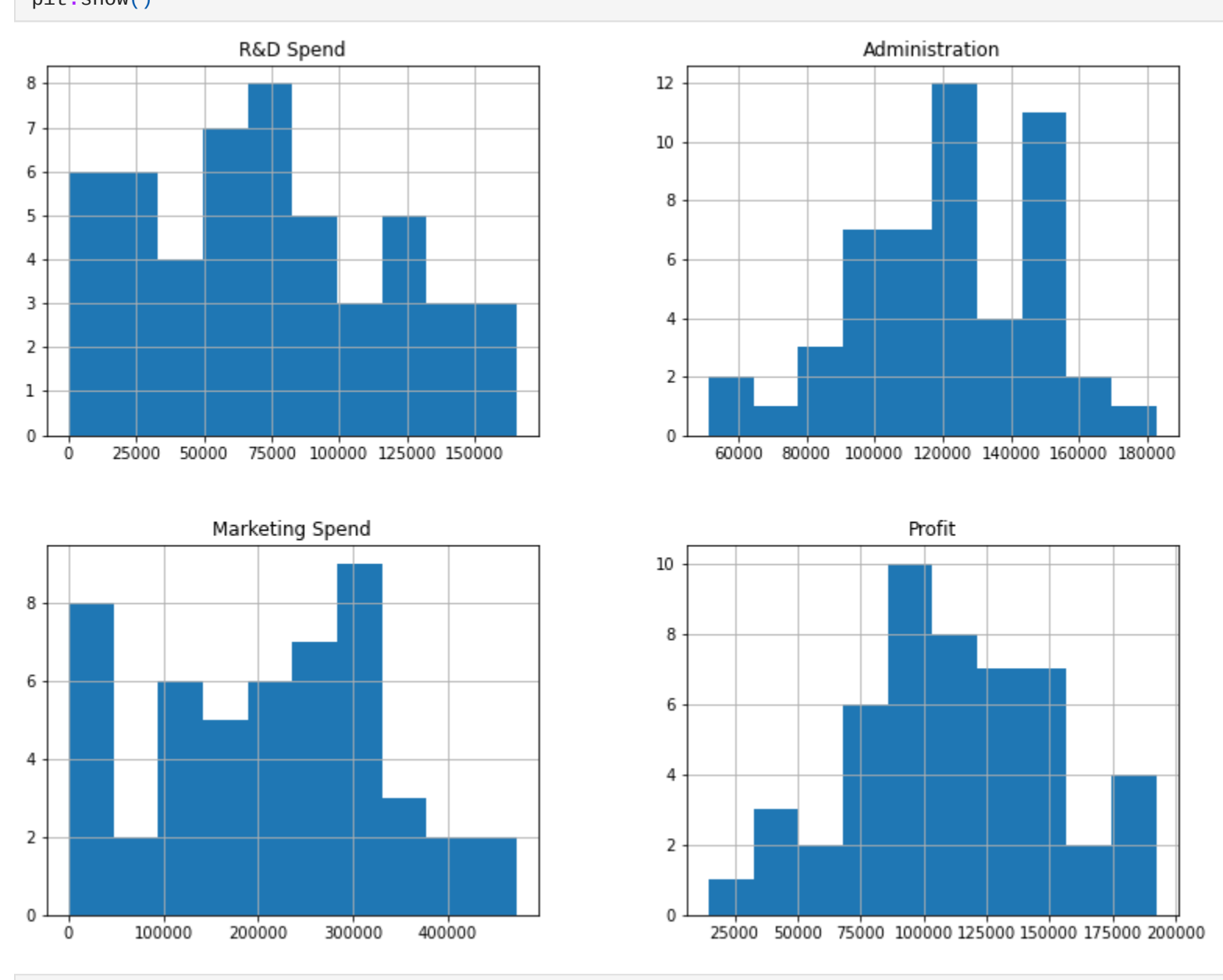
Out[12]:

```
<AxesSubplot:xlabel='R&D Spend', ylabel='Profit'>
```



In [13]:

```python
df.hist(figsize = (13,10))
plt.show()
```



In [14]:

```python
df.describe().T
```

Out[14]:

	count	mean	std	min	25%	50%	75%	max
R&D Spend	50.0	73721.6156	45902.256482	0.00	39936.3700	73051.080	101602.8000	165349.20
Administration	50.0	121344.6396	28017.802755	51283.14	103730.8750	122699.795	144842.1800	182645.56
Marketing Spend	50.0	211025.0978	122290.310726	0.00	129300.1325	212716.240	299469.0850	471784.10
Profit	50.0	112012.6392	40306.180338	14681.40	90138.9025	107978.190	139765.9775	192261.83

In [15]:

```python
df_State = pd.get_dummies(df["State"])
```

In [18]:

```python
dfDummies = pd.get_dummies(df["State"], prefix="State")
```

In [19]:

```python
dfDummies
```

Out[19]:

	State_California	State_Florida	State_New York
0	0	0	1
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0
5	0	0	1
6	1	0	0
7	0	1	0
8	0	0	1
9	1	0	0
10	0	1	0
11	1	0	0
12	0	1	0
13	1	0	0
14	0	1	0
15	0	0	1
16	1	0	0
17	0	0	1
18	0	1	0
19	0	0	1
20	1	0	0
21	0	0	1
22	0	1	0
23	0	1	0
24	0	0	1
25	1	0	0
26	0	1	0
27	0	0	1
28	0	1	0
29	0	0	1
30	0	1	0
31	0	0	1
32	1	0	0
33	0	1	0
34	1	0	0
35	0	0	1
36	0	1	0
37	1	0	0
38	0	0	1
39	1	0	0
40	1	0	0
41	0	1	0
42	1	0	0
43	0	0	1
44	1	0	0
45	0	0	1
46	0	1	0
47	1	0	0
48	0	0	1
49	1	0	0

In [21]:

```python
df = pd.concat([df, dfDummies], axis=1)
df = df.drop(["State_California"], axis = 1)
df = df.drop(["State"], axis = 1)
df.head()
```

Out[21]:

	R&D Spend	Administration	Marketing Spend	Profit	State_Florida	State_New York
0	165349.20	136897.80	471784.10	192261.83	0	1
1	162597.70	151377.59	443898.53	191792.06	0	0
2	153441.51	101145.55	407934.54	191050.39	1	0
3	144372.41	118671.85	383199.62	182901.99	0	1
4	142107.34	91391.77	366168.42	166187.94	1	0

In [22]:

```python
x = df.drop("Profit", axis = 1)
y = df["Profit"]
```

In [23]:

```python
y
```

Out[23]:

```
0     192261.83
1     191792.06
2     191050.39
3     182901.99
4     166187.94
5     156991.12
6     156122.51
7     155752.60
8     152211.77
9     149759.96
10    146121.95
11    144259.40
12    141585.52
13    134307.35
14    132602.65
15    129917.04
16    126992.93
17    125370.37
18    124266.90
19    122776.86
20    118474.03
21    111313.02
22    110352.25
23    108733.99
24    108552.04
25    107404.34
26    105733.54
27    105008.31
28    103282.38
29    101004.64
30     99937.59
31     97483.56
32     97427.84
33     96778.92
34     96712.80
35     96479.51
36     90708.19
37     89949.14
38     81229.06
39     81005.76
40     78239.91
41     77798.83
42     71498.49
43     69758.98
44     65200.33
45     64926.08
46     49490.75
47     42559.73
48     35673.41
49     14681.40
Name: Profit, dtype: float64
```

In [24]:

```python
x
```

Out[24]:

	R&D Spend	Administration	Marketing Spend	State_Florida	State_New York
0	165349.20	136897.80	471784.10	0	1
1	162597.70	151377.59	443898.53	0	0
2	153441.51	101145.55	407934.54	1	0
3	144372.41	118671.85	383199.62	0	1
4	142107.34	91391.77	366168.42	1	0
5	131876.90	99814.71	362861.36	0	1
6	134615.46	147198.87	127716.82	0	0
7	130298.13	145530.06	323876.68	1	0
8	120542.52	148718.95	311613.29	0	1
9	123334.88	108679.17	304981.62	0	0
10	101913.08	110594.11	229160.95	1	0
11	100671.96	91790.61	249744.55	0	0
12	93863.75	127320.38	249839.44	1	0
13	91992.39	135495.07	252664.93	0	0
14	119943.24	156547.42	256512.92	1	0
15	114523.61	122616.84	261776.23	0	1
16	78013.11	121597.55	264346.06	0	0
17	94657.16	145077.58	282574.31	0	1
18	91749.16	114175.79	294919.57	1	0
19	86419.70	153514.11	0.00	0	1
20	76253.86	113867.30	298664.47	0	0
21	78389.47	153773.43	299737.29	0	1
22	73994.56	122782.75	303319.26	1	0
23	67532.53	105751.03	304768.73	1	0
24	77044.01	99281.34	140574.81	0	1
25	64664.71	139553.16	137962.62	0	0
26	75328.87	144135.98	134050.07	1	0
27	72107.60	127864.55	353183.81	0	1
28	66051.52	182645.56	118148.20	1	0
29	65605.48	153032.06	107138.38	0	1
30	61994.48	115641.28	91131.24	1	0
31	61136.38	152701.92	88218.23	0	1
32	63408.86	129219.61	46085.25	0	0
33	55493.95	103057.49	214634.81	1	0
34	46426.07	157693.92	210797.67	0	0
35	46014.02	85047.44	205517.64	0	1
36	28663.76	127056.21	201126.82	1	0
37	44069.95	51283.14	197029.42	0	0
38	20229.59	65947.93	185265.10	0	1
39	38558.51	82982.09	174999.30	0	0
40	28754.33	118546.05	172795.67	0	0
41	27892.92	84710.77	164470.71	1	0
42	23640.93	96189.63	148001.11	0	0
43	15505.73	127382.30	35534.17	0	1
44	22177.74	154806.14	28334.72	0	0
45	1000.23	124153.04	1903.93	0	1
46	1315.46	115816.21	297114.46	1	0
47	0.00	135426.92	0.00	0	0
48	542.05	51743.15	0.00	0	1
49	0.00	116983.80	45173.06	0	0

In [26]:

```python
from sklearn.model_selection import train_test_split
```

In [27]:

```python
x_train, x_test, y_train, y_test = train_test_split(x , y, test_size = 0.20, random_state = 42)
```

In [28]:

```python
x_train
```

Out[28]:

	R&D Spend	Administration	Marketing Spend	State_Florida	State_New York
12	93863.75	127320.38	249839.44	1	0
4	142107.34	91391.77	366168.42	1	0
37	44069.95	51283.14	197029.42	0	0
8	120542.52	148718.95	311613.29	0	1
3	144372.41	118671.85	383199.62	0	1
6	134615.46	147198.87	127716.82	0	0
41	27892.92	84710.77	164470.71	1	0
46	1315.46	115816.21	297114.46	1	0
47	0.00	135426.92	0.00	0	0
15	114523.61	122616.84	261776.23	0	1
9	123334.88	108679.17	304981.62	0	0
16	78013.11	121597.55	264346.06	0	0
24	77044.01	99281.34	140574.81	0	1
34	46426.07	157693.92	210797.67	0	0
31	61136.38	152701.92	88218.23	0	1
0	165349.20	136897.80	471784.10	0	1
44	22177.74	154806.14	28334.72	0	0
27	72107.60	127864.55	353183.81	0	1
33	55493.95	103057.49	214634.81	1	0
5	131876.90	99814.71	362861.36	0	1
29	65605.48	153032.06	107138.38	0	1
11	100671.96	91790.61	249744.55	0	0
36	28663.76	127056.21	201126.82	1	0
1	162597.70	151377.59	443898.53	0	0
21	78389.47	153773.43	299737.29	0	1
2	153441.51	101145.55	407934.54	1	0
43	15505.73	127382.30	35534.17	0	1
35	46014.02	85047.44	205517.64	0	1
23	67532.53	105751.03	304768.73	1	0
40	28754.33	118546.05	172795.67	0	0
10	101913.08	110594.11	229160.95	1	0
22	73994.56	122782.75	303319.26	1	0
18	91749.16	114175.79	294919.57	1	0
49	0.00	116983.80	45173.06	0	0
20	76253.86	113867.30	298664.47	0	0
7	130298.13	145530.06	323876.68	1	0
42	23640.93	96189.63	148001.11	0	0
14	119943.24	156547.42	256512.92	1	0
28	66051.52	182645.56	118148.20	1	0
38	20229.59	65947.93	185265.10	0	1

In [29]:

```python
x_test
```

Out[29]:

	R&D Spend	Administration	Marketing Spend	State_Florida	State_New York
13	91992.39	135495.07	252664.93	0	0
39	38558.51	82982.09	174999.30	0	0
30	61994.48	115641.28	91131.24	1	0
45	1000.23	124153.04	1903.93	0	1
17	94657.16	145077.58	282574.31	0	1
48	542.05	51743.15	0.00	0	1
26	75328.87	144135.98	134050.07	1	0
25	64664.71	139553.16	137962.62	0	0
32	63408.86	129219.61	46085.25	0	0
19	86419.70	153514.11	0.00	0	1

In [38]:

```python
y_train
```

Out[38]:

```
12    141585.52
4     166187.94
37     89949.14
8     152211.77
3     182901.99
6     156122.51
41     77798.83
46     49490.75
47     42559.73
15    129917.04
9     149759.96
16    126992.93
24    108552.04
34     96712.80
31     97483.56
0     192261.83
44     65200.33
27    105008.31
33     96778.92
5     156991.12
29    101004.64
11    144259.40
36     90708.19
1     191792.06
21    111313.02
2     191050.39
43     69758.98
35     96479.51
23    108733.99
40     78239.91
10    146121.95
22    110352.25
18    124266.90
49     14681.40
20    118474.03
7     155752.60
42     71498.49
14    132602.65
28    103282.38
38     81229.06
Name: Profit, dtype: float64
```

In [39]:

```python
y_test
```

Out[39]:

```
13    134307.35
39     81005.76
30     99937.59
45     64926.08
17    125370.37
48     35673.41
26    105733.54
25    107404.34
32     97427.84
19    122776.86
Name: Profit, dtype: float64
```

In [42]:

```python
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
```

In [43]:

```python
model = lm.fit(x_train, y_train)
```

In [45]:

```python
y_pred = model.predict(x_test)
y_pred
```

Out[45]:

```
array([126362.87908255,  84608.45383634,  99677.49425147,  46357.46068582,
       128750.48288504,  50912.4174188 , 109741.35032702, 100643.24281647,
        97599.27574594, 113097.42524432])
```

In [47]:

```python
df =pd.DataFrame({"Main values" : y_test, "Predict values" : y_pred, "Difference" :abs(y_pred-y_test)})
df
```

Out[47]:

	Main values	Predict values	Difference
13	134307.35	126362.879083	7944.470917
39	81005.76	84608.453836	3602.693836
30	99937.59	99677.494251	260.095749
45	64926.08	46357.460686	18568.619314
17	125370.37	128750.482885	3380.112885
48	35673.41	50912.417419	15239.007419
26	105733.54	109741.350327	4007.810327
25	107404.34	100643.242816	6761.097184
32	97427.84	97599.275746	171.435746
19	122776.86	113097.425244	9679.434756

In [49]:

```python
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
```

In [50]:

```python
mean_absolute_error(y_test, model.predict(x_test))
```

Out[50]:

```
6961.477813252382
```

In [52]:

```python
mean_squared_error(y_test, model.predict(x_test))
```

Out[52]:

```
82010363.04430102
```

In [53]:

```python
np.sqrt(mean_squared_error([y_test], [y_pred]))
```

Out[53]:

```
9055.957323458466
```

In [55]:

```python
model.score(x_train, y_train)
```

Out[55]:

```
0.9537019995248526
```

In [56]:

```python
import statsmodels.api as sm
```

In [57]:

```python
sts_model = sm.OLS(y,x).fit()
```

In [58]:

```python
sts_model.summary()
```

Out[58]:

OLS Regression Results

Dep. Variable:	Profit	R-squared (uncentered):	0.988
Model:	OLS	Adj. R-squared (uncentered):	0.986
Method:	Least Squares	F-statistic:	727.1
Date:	Tue, 25 May 2021	Prob (F-statistic):	7.87e-42
Time:	17:22:40	Log-Likelihood:	-545.15
No. Observations:	50	AIC:	1100.
Df Residuals:	45	BIC:	1110.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
R&D Spend	0.7182	0.066	10.916	0.000	0.586	0.851
Administration	0.3113	0.035	8.885	0.000	0.241	0.382
Marketing Spend	0.0786	0.023	3.429	0.001	0.032	0.125
State_Florida	3464.4536	4905.406	0.706	0.484	-6415.541	1.33e+04
State_New York	5067.8937	4668.238	1.086	0.283	-4334.419	1.45e+04

Omnibus:	1.355	Durbin-Watson:	1.288
Prob(Omnibus):	0.508	Jarque-Bera (JB):	1.241
Skew:	-0.237	Prob(JB):	0.538
Kurtosis:	2.391	Cond. No.	8.28e+05

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 8.28e+05. This might indicate that there are
strong multicollinearity or other numerical problems.