Title: Profit Predictions 📊

Goal 🎯 : Find how to get the highest profit with least spending 💸 .

Before we start working with the data, let's import the main libraries we will use

In [40]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Now, lets load the data and check it

In [42]:
```
startups = pd.read_csv('50_Startups.csv')
df = startups.copy()
df
```

Out[42]:

	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

	R&D Spend	Administration	Marketing Spend	State	Profit
48	542.05	51743.15	0.00	New York	35673.41
49	0.00	116983.80	45173.06	California	14681.40

Lets check if there are null values we should deal with them or not

In [44]:
```
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 [45]:
```
df.isna().sum()
```

Out[45]:
```
R&D Spend          0
Administration     0
Marketing Spend    0
State              0
Profit             0
dtype: int64
```

Now I want to know some information about the dataset like the mean, std, and min & max values

In [47]:
```
df.describe().T
```

Out[47]:

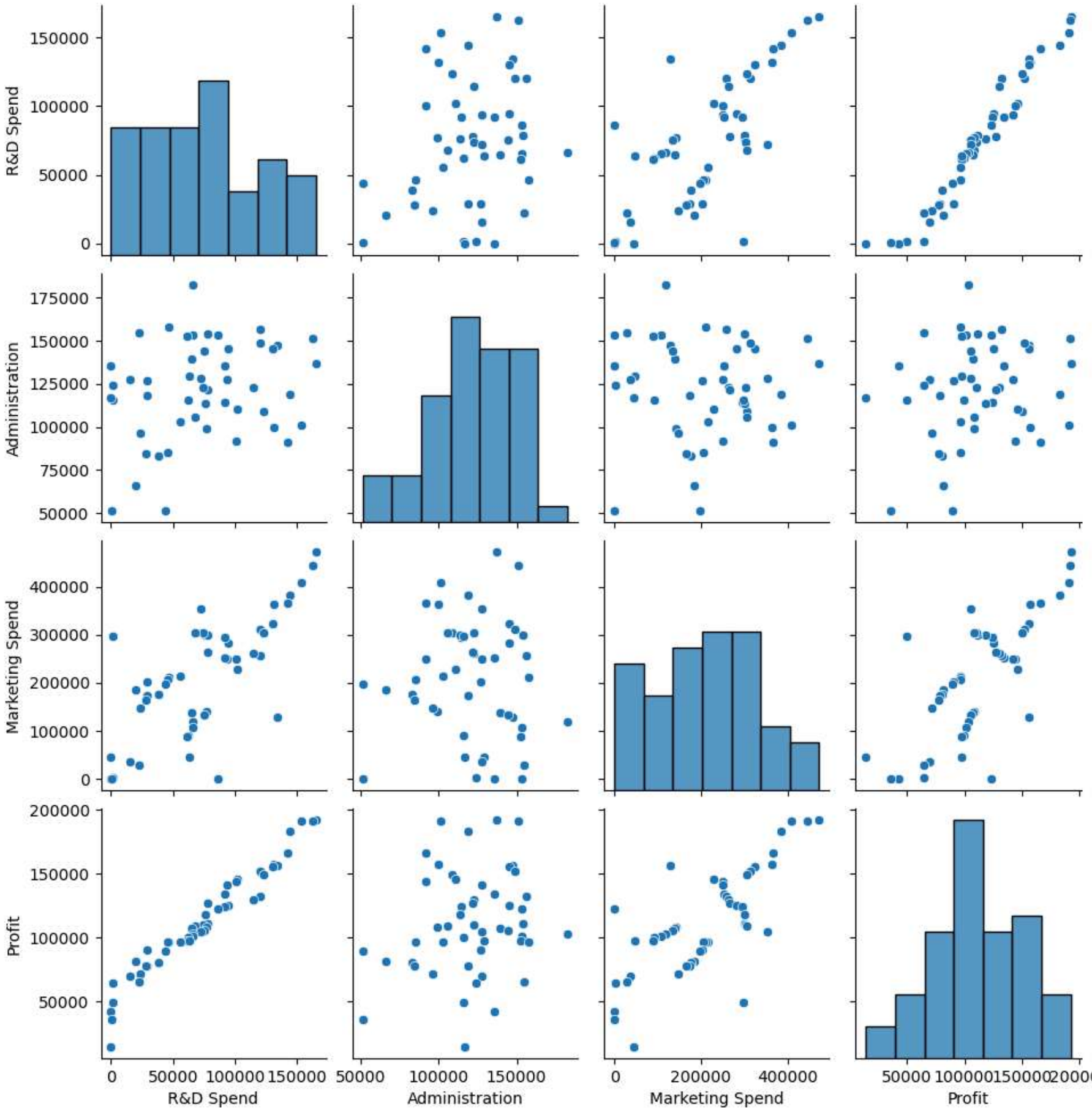
	count	mean	std	min	25%	50%	75%	max
R&D Spend	50.0	73721.6156	45902.256482	0.00	39936.3700	73051.080	101602.8000	165349.20
Administration	50.0	121344.6396	28017.802755	51283.14	103730.8750	122699.795	144842.1800	182645.56
Marketing Spend	50.0	211025.0978	122290.310726	0.00	129300.1325	212716.240	299469.0850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

After checking the information. I want to understand the coorelation between the features and target, so I can know which feature affects the most on the target

In [49]:
```
plt.figure(figsize=(7,3))
sns.heatmap(df.drop('State',axis=1).corr(),linewidths=0.5,annot=True)
plt.show()
```

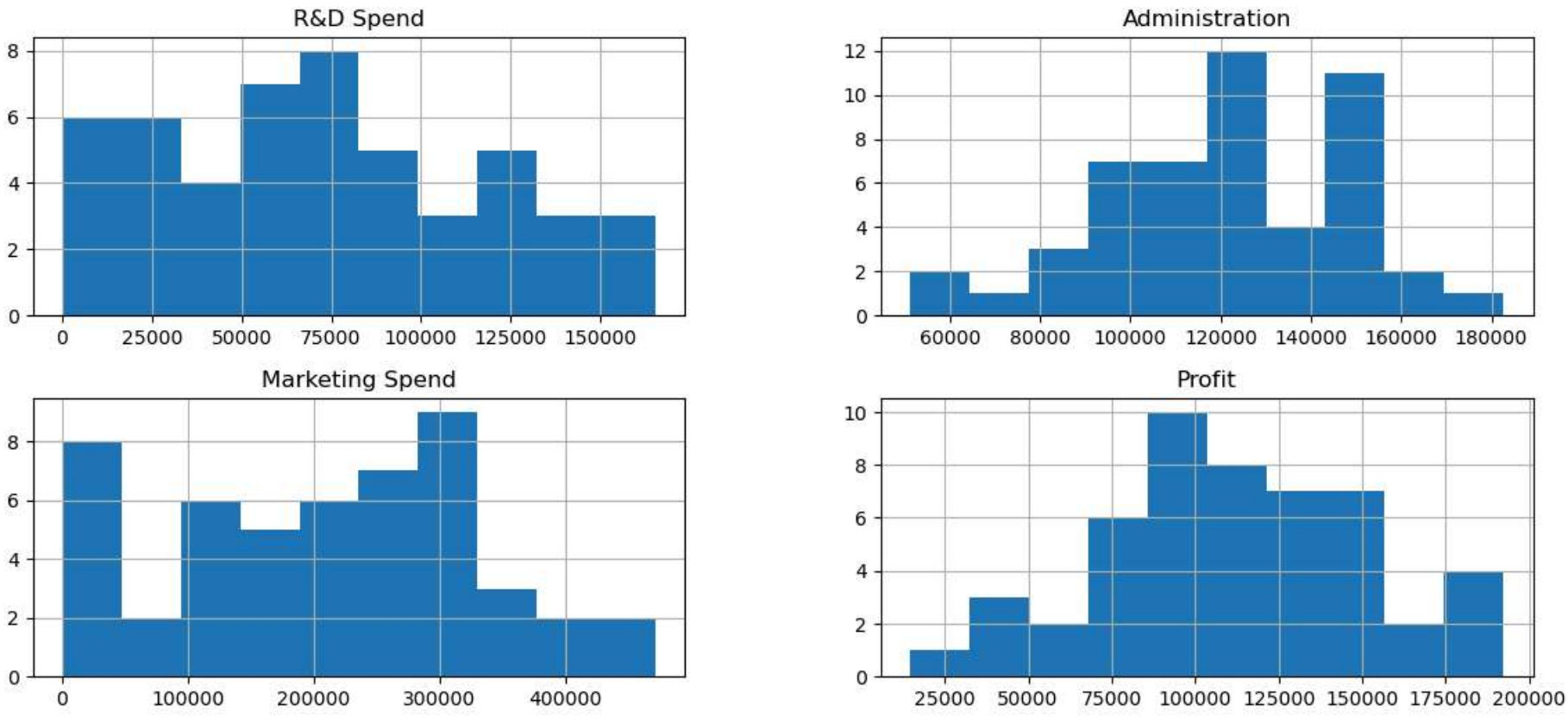
In [50]:
```
sns.pairplot(df)
```

Out[50]:
```
<seaborn.axisgrid.PairGrid at 0x2350796d670>
```



In [51]:
```
df.hist(figsize=(14,6))
```

Out[51]:
```
array([[<Axes: title={'center': 'R&D Spend'}>,
        <Axes: title={'center': 'Administration'}>],
       [<Axes: title={'center': 'Marketing Spend'}>,
        <Axes: title={'center': 'Profit'}>]], dtype=object)
```



In [52]:
```
top_5_profits = df.sort_values(by='Profit',ascending=False).head()
top_5_profits
```

Out[52]:

	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

From the visualalazations, I can understand that R&D Spends affects the most, then Marketing, and lastly the Administration.
Also the numerical ranges of each feature.
And what are the top 5 companies with highest profit

Lets start with preparing the data to use the Linear Regression.
I will start with changing the non-numerical values into numerical values so it can work with the model.
I used the dummies method, so I dont give a specific state an overhand over the others

In [55]:
```
df_state = pd.get_dummies(df['State'],dtype=int,prefix='State')
df_state
```

Out[55]:

	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

	State_California	State_Florida	State_New York
48	0	0	1
49	1	0	0

In [56]:
```
df = pd.concat([df,df_state.drop('State_Florida',axis=1)],axis=1)
df.drop('State',axis=1,inplace=True)
df
```

Out[56]:

	R&D Spend	Administration	Marketing Spend	Profit	State_California	State_New York
0	165349.20	136897.80	471784.10	192261.83	0	1
1	162597.70	151377.59	443898.53	191792.06	1	0
2	153441.51	101145.55	407934.54	191050.39	0	0
3	144372.41	118671.85	383199.62	182901.99	0	1
4	142107.34	91391.77	366168.42	166187.94	0	0
5	131876.90	99814.71	362861.36	156991.12	0	1
6	134615.46	147198.87	127716.82	156122.51	1	0
7	130298.13	145530.06	323876.68	155752.60	0	0
8	120542.52	148718.95	311613.29	152211.77	0	1
9	123334.88	108679.17	304981.62	149759.96	1	0
10	101913.08	110594.11	229160.95	146121.95	0	0
11	100671.96	91790.61	249744.55	144259.40	1	0
12	93863.75	127320.38	249839.44	141585.52	0	0
13	91992.39	135495.07	252664.93	134307.35	1	0
14	119943.24	156547.42	256512.92	132602.65	0	0
15	114523.61	122616.84	261776.23	129917.04	0	1
16	78013.11	121597.55	264346.06	126992.93	1	0
17	94657.16	145077.58	282574.31	125370.37	0	1
18	91749.16	114175.79	294919.57	124266.90	0	0
19	86419.70	153514.11	0.00	122776.86	0	1
20	76253.86	113867.30	298664.47	118474.03	1	0
21	78389.47	153773.43	299737.29	111313.02	0	1
22	73994.56	122782.75	303319.26	110352.25	0	0
23	67532.53	105751.03	304768.73	108733.99	0	0
24	77044.01	99281.34	140574.81	108552.04	0	1
25	64664.71	139553.16	137962.62	107404.34	1	0
26	75328.87	144135.98	134050.07	105733.54	0	0
27	72107.60	127864.55	353183.81	105008.31	0	1
28	66051.52	182645.56	118148.20	103282.38	0	0
29	65605.48	153032.06	107138.38	101004.64	0	1
30	61994.48	115641.28	91131.24	99937.59	0	0
31	61136.38	152701.92	88218.23	97483.56	0	1
32	63408.86	129219.61	46085.25	97427.84	1	0
33	55493.95	103057.49	214634.81	96778.92	0	0
34	46426.07	157693.92	210797.67	96712.80	1	0
35	46014.02	85047.44	205517.64	96479.51	0	1
36	28663.76	127056.21	201126.82	90708.19	0	0
37	44069.95	51283.14	197029.42	89949.14	1	0
38	20229.59	65947.93	185265.10	81229.06	0	1
39	38558.51	82982.09	174999.30	81005.76	1	0
40	28754.33	118546.05	172795.67	78239.91	1	0
41	27892.92	84710.77	164470.71	77798.83	0	0
42	23640.93	96189.63	148001.11	71498.49	1	0
43	15505.73	127382.30	35534.17	69758.98	0	1
44	22177.74	154806.14	28334.72	65200.33	1	0
45	1000.23	124153.04	1903.93	64926.08	0	1
46	1315.46	115816.21	297114.46	49490.75	0	0
47	0.00	135426.92	0.00	42559.73	1	0

	R&D Spend	Administration	Marketing Spend	Profit	State_California	State_New York
48	542.05	51743.15	0.00	35673.41	0	1
49	0.00	116983.80	45173.06	14681.40	1	0

Its time to split the data to train and test sets. So we can train the model using the train set and test the predicted values using the test set

In [58]:
```
X = df.drop('Profit',axis=1)
y = df['Profit']
```

In [59]:
```
X
```

Out[59]:

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

	R&D Spend	Administration	Marketing Spend	State_California	State_New York
48	542.05	51743.15	0.00	0	1
49	0.00	116983.80	45173.06	1	0

In [60]:
```
y
```

Out[60]:
```
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 [61]:
```
from sklearn.model_selection import train_test_split as tts
X_train, X_test, y_train, y_test = tts(X,y,test_size=0.3,random_state=35)
```

In [62]:
```
X_train
```

Out[62]:

	R&D Spend	Administration	Marketing Spend	State_California	State_New York
48	542.05	51743.15	0.00	0	1
34	46426.07	157693.92	210797.67	1	0
2	153441.51	101145.55	407934.54	0	0
20	76253.86	113867.30	298664.47	1	0
37	44069.95	51283.14	197029.42	1	0
10	101913.08	110594.11	229160.95	0	0
1	162597.70	151377.59	443898.53	1	0
36	28663.76	127056.21	201126.82	0	0
16	78013.11	121597.55	264346.06	1	0
6	134615.46	147198.87	127716.82	1	0
28	66051.52	182645.56	118148.20	0	0
5	131876.90	99814.71	362861.36	0	1
7	130298.13	145530.06	323876.68	0	0
24	77044.01	99281.34	140574.81	0	1
38	20229.59	65947.93	185265.10	0	1
13	91992.39	135495.07	252664.93	1	0
49	0.00	116983.80	45173.06	1	0
46	1315.46	115816.21	297114.46	0	0
19	86419.70	153514.11	0.00	0	1
3	144372.41	118671.85	383199.62	0	1
21	78389.47	153773.43	299737.29	0	1
12	93863.75	127320.38	249839.44	0	0
45	1000.23	124153.04	1903.93	0	1
27	72107.60	127864.55	353183.81	0	1
30	61994.48	115641.28	91131.24	0	0
8	120542.52	148718.95	311613.29	0	1
42	23640.93	96189.63	148001.11	1	0
11	100671.96	91790.61	249744.55	1	0
40	28754.33	118546.05	172795.67	1	0
0	165349.20	136897.80	471784.10	0	1
47	0.00	135426.92	0.00	1	0
33	55493.95	103057.49	214634.81	0	0
44	22177.74	154806.14	28334.72	1	0
15	114523.61	122616.84	261776.23	0	1
9	123334.88	108679.17	304981.62	1	0

In [63]:
```
X_test
```

Out[63]:

	R&D Spend	Administration	Marketing Spend	State_California	State_New York
39	38558.51	82982.09	174999.30	1	0
26	75328.87	144135.98	134050.07	0	0
22	73994.56	122782.75	303319.26	0	0
31	61136.38	152701.92	88218.23	0	1
29	65605.48	153032.06	107138.38	0	1
43	15505.73	127382.30	35534.17	0	1
41	27892.92	84710.77	164470.71	0	0
17	94657.16	145077.58	282574.31	0	1
25	64664.71	139553.16	137962.62	1	0
23	67532.53	105751.03	304768.73	0	0
35	46014.02	85047.44	205517.64	0	1
4	142107.34	91391.77	366168.42	0	0
18	91749.16	114175.79	294919.57	0	0
32	63408.86	129219.61	46085.25	1	0
14	119943.24	156547.42	256512.92	0	0

In [64]:
```
y_train
```

Out[64]:
```
48     35673.41
34     96712.80
2     191050.39
20    118474.03
37     89949.14
10    146121.95
1     191792.06
36     90708.19
16    126992.93
6     156122.51
28    103282.38
5     156991.12
7     155752.60
24    108552.04
38     81229.06
13    134307.35
49     14681.40
46     49490.75
19    122776.86
3     182901.99
21    111313.02
12    141585.52
45     64926.08
27    105008.31
30     99937.59
8     152211.77
42     71498.49
11    144259.40
40     78239.91
0     192261.83
47     42559.73
33     96778.92
44     65200.33
15    129917.04
9     149759.96
Name: Profit, dtype: float64
```

In [65]:
```
y_test
```

Out[65]:
```
39     81005.76
26    105733.54
22    110352.25
31     97483.56
29    101004.64
43     69758.98
41     77798.83
17    125370.37
25    107404.34
23    108733.99
35     96479.51
4     166187.94
18    124266.90
32     97427.84
14    132602.65
Name: Profit, dtype: float64
```

Now it is time to train the model on the train sets and predict the values

In [67]:
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
model = lr.fit(X_train,y_train)
```

In [68]:
```
y_prediction = model.predict(X_test)
y_prediction = y_prediction.round(2)
```

I want to check if the model did well and predicted with close results to the test sets.
I will compare the predicted values with the real values, and use evaluation metrics like RMSE and R^2.

What are RMSE and R^2?
RMSE: The average difference between the real and predicted values
R^2: Model score. It is between 0-1, the higher the better, and it can be negative but this means it is a bad model

In [70]:
```
model_check = pd.DataFrame({'y_test':y_test,
                            'y_predection':y_prediction})
model_check
```

Out[70]:

	y_test	y_predection
39	81005.76	82650.91
26	105733.54	114969.40
22	110352.25	118676.51
31	97483.56	96544.93
29	101004.64	100747.91
43	69758.98	57717.40
41	77798.83	77491.15
17	125370.37	129634.20
25	107404.34	102260.23
23	108733.99	113685.43
35	96479.51	88383.71
4	166187.94	177651.49
18	124266.90	133417.00
32	97427.84	99051.43
14	132602.65	155021.60

In [71]:
```
from sklearn.metrics import root_mean_squared_error, mean_absolute_error, r2_score
print(f'MAE: {mean_absolute_error(y_test,y_prediction).round(2)}')
print(f'RMAE: {np.sqrt(mean_absolute_error(y_test,y_prediction)).round(2)}')
print(f'RMSE: {root_mean_squared_error(y_test,y_prediction).round(2)}')
print(f'R^2: {round(r2_score(y_test,y_prediction),2)}')
```

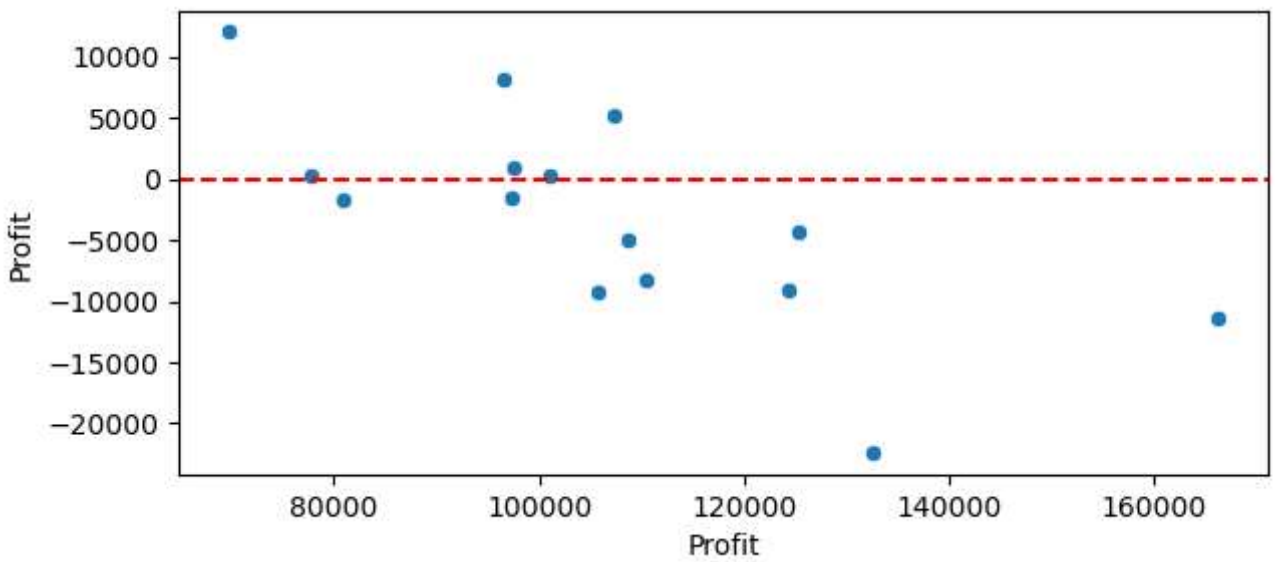
```
MAE: 6657.42
RMAE: 81.59
RMSE: 8787.97
R^2: 0.86
```

The evaluation metrics gave us a kinda good results, but there is of course better results we can achieve

I want to check if Linear Regression model is good for this dataset by using the residual plot. If the dots are in a specific pattern and they are random, it means that Linear Regression can be used.

In [74]:
```
test_residuals = y_test - y_prediction
plt.figure(figsize=(7,3))
sns.scatterplot(x=y_test,y=test_residuals)
plt.axhline(y=0,color='red',ls='--')
```

Out[74]:
```
<matplotlib.lines.Line2D at 0x23509333c20>
```



So yes, the dots are random which means that we can use the Linear Regression.

Lets see if the states affects on the target or not using backward elimination. If not, I will drop the states

In [77]:
```
df.drop('Profit',axis=1).head(10)
```

Out[77]:

	R&D Spend	Administration	Marketing Spend	State_California	State_New York
0	165349.20	136897.80	471784.10	0	1
1	162597.70	151377.59	443898.53	1	0
2	153441.51	101145.55	407934.54	0	0
3	144372.41	118671.85	383199.62	0	1
4	142107.34	91391.77	366168.42	0	0
5	131876.90	99814.71	362861.36	0	1
6	134615.46	147198.87	127716.82	1	0
7	130298.13	145530.06	323876.68	0	0
8	120542.52	148718.95	311613.29	0	1
9	123334.88	108679.17	304981.62	1	0

In [78]:
```
model.coef_
```

Out[78]:
```
array([ 8.32481595e-01, -2.06061407e-02,  2.58634933e-02, -4.02707371e+03,
       -5.24760110e+03])
```

In [79]:
```
import statsmodels.api as sm
stmodel = sm.OLS(y,X).fit()
stmodel.summary()
```

Out[79]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared (uncentered):**	0.988
Model:	OLS	**Adj. R-squared (uncentered):**	0.987
Method:	Least Squares	**F-statistic:**	765.9
Date:	Sat, 21 Jun 2025	**Prob (F-statistic):**	2.49e-42
Time:	10:50:44	**Log-Likelihood:**	-543.87
No. Observations:	50	**AIC:**	1098.
Df Residuals:	45	**BIC:**	1107.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
R&D Spend	0.7217	0.064	11.260	0.000	0.593	0.851
Administration	0.2847	0.038	7.465	0.000	0.208	0.362
Marketing Spend	0.0830	0.022	3.833	0.000	0.039	0.127
State_California	7720.4710	4539.299	1.701	0.096	-1422.146	1.69e+04
State_New York	7126.6916	4530.146	1.573	0.123	-1997.491	1.63e+04

Omnibus:	0.862	**Durbin-Watson:**	1.480
Prob(Omnibus):	0.650	**Jarque-Bera (JB):**	0.929
Skew:	-0.212	**Prob(JB):**	0.628
Kurtosis:	2.484	**Cond. No.**	8.02e+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, 8.02e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

In [80]:
```
df.drop(['State_New York','State_California'],axis=1,inplace=True)
df
```

Out[80]:

	R&D 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
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

	R&D Spend	Administration	Marketing Spend	Profit
48	542.05	51743.15	0.00	35673.41
49	0.00	116983.80	45173.06	14681.40

As we saw, the states gave us P-Value > 0.05 which means they affect the least on the target. So, I dropped them.

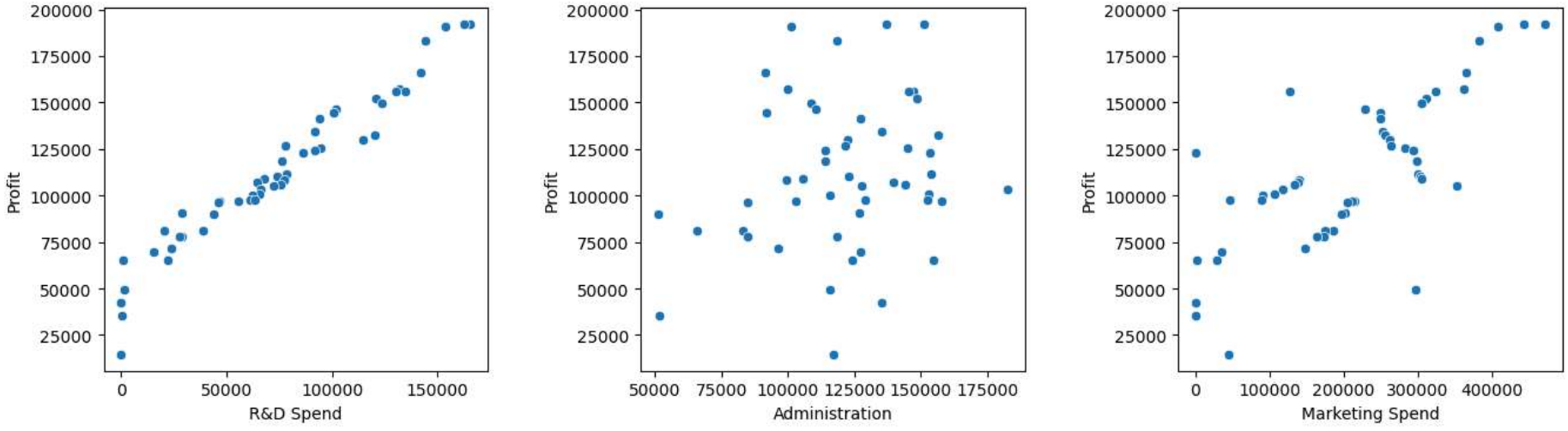
Let me do a quick visualazation for the features

In [83]:
```
plt.figure(figsize=(16,4))
plt.subplot(1,3,1)
sns.scatterplot(data=df,x='R&D Spend',y='Profit')

plt.subplot(1,3,2)
sns.scatterplot(data=df,x='Administration',y='Profit')

plt.subplot(1,3,3)
sns.scatterplot(data=df,x='Marketing Spend',y='Profit')

plt.subplots_adjust(wspace=0.4,hspace=0.4)
```



I noticed some zero values and outliers in R&D and Marketing spends. It is not realistic since it is hard to find a company that doesnt spend on them and gets a high profit, I will assume that they are the values are N/A and they replaced them with zeros. So, I will drop the zero values.

In [85]:
```
df = df[~((df["R&D Spend"] == 0) | (df["Marketing Spend"] == 0 ))]
df
```

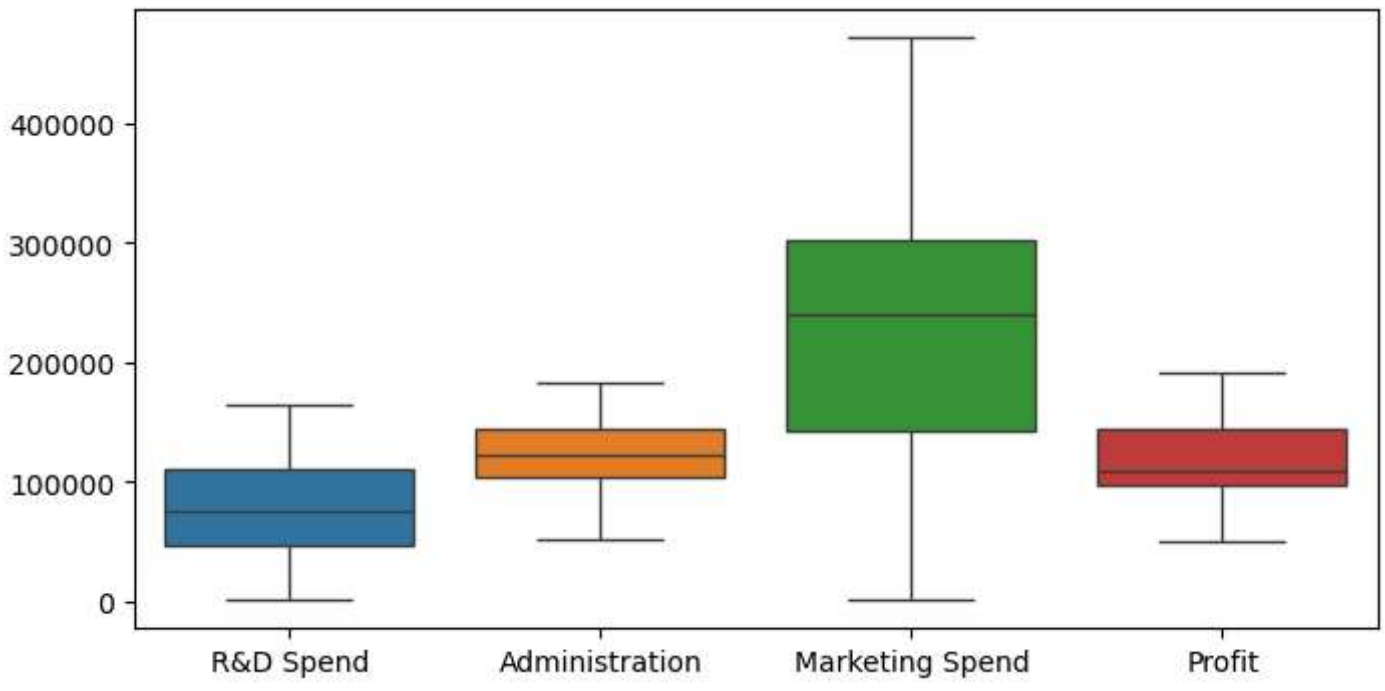
Out[85]:

	R&D 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
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
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

I want to check if there is any outliers

In [87]:
```
plt.figure(figsize=(8,4))
sns.boxplot(df)
```

Out[87]: <Axes: >



After re-cleaning the data, lets split it again the do the modeling process one more time to check how the model will perform now.

In [89]:
```
X = df.drop('Profit',axis=1)
y = df['Profit']
```

In [90]:
```
X_train, X_test, y_train, y_test = tts(X,y,test_size=0.2,random_state=35)
```

In [91]:
```
new_model = lr.fit(X_train,y_train)
```

In [92]:
```
y_pred = new_model.predict(X_test)
y_pred = y_pred.round(2)
```

In [93]:
```
model_check = pd.DataFrame({'y_test':y_test,
                            'y_predection':y_pred})
model_check
```

Out[93]:

	y_test	y_predection
29	101004.64	104513.35
26	105733.54	113168.38
24	108552.04	116766.52
40	78239.91	77484.69
30	99937.59	103129.60
45	64926.08	52282.42
14	132602.65	150351.14
44	65200.33	68280.21
21	111313.02	117698.64
4	166187.94	172935.00

In [94]:
```
print(f'MAE: {mean_absolute_error(y_test,y_pred).round(2)}')
print(f'RMAE: {np.sqrt(mean_absolute_error(y_test,y_pred)).round(2)}')
print(f'RMSE: {root_mean_squared_error(y_test,y_pred).round(2)}')
print(f'R^2: {round(r2_score(y_test,y_pred),2)}')
```

```
MAE: 6971.0
RMAE: 83.49
RMSE: 8464.52
R^2: 0.92
```

Yes, It gave us much better results with no overfitting or underfitting. Lets see the coefficient values to know the features relationships with each other.

In [96]:
```
new_model.coef_
```

Out[96]: array([0.80477776, -0.04723058, 0.01522351])

We can relize that spending on the R&D will give us more profit, while spending on Administration will decrease the profit.

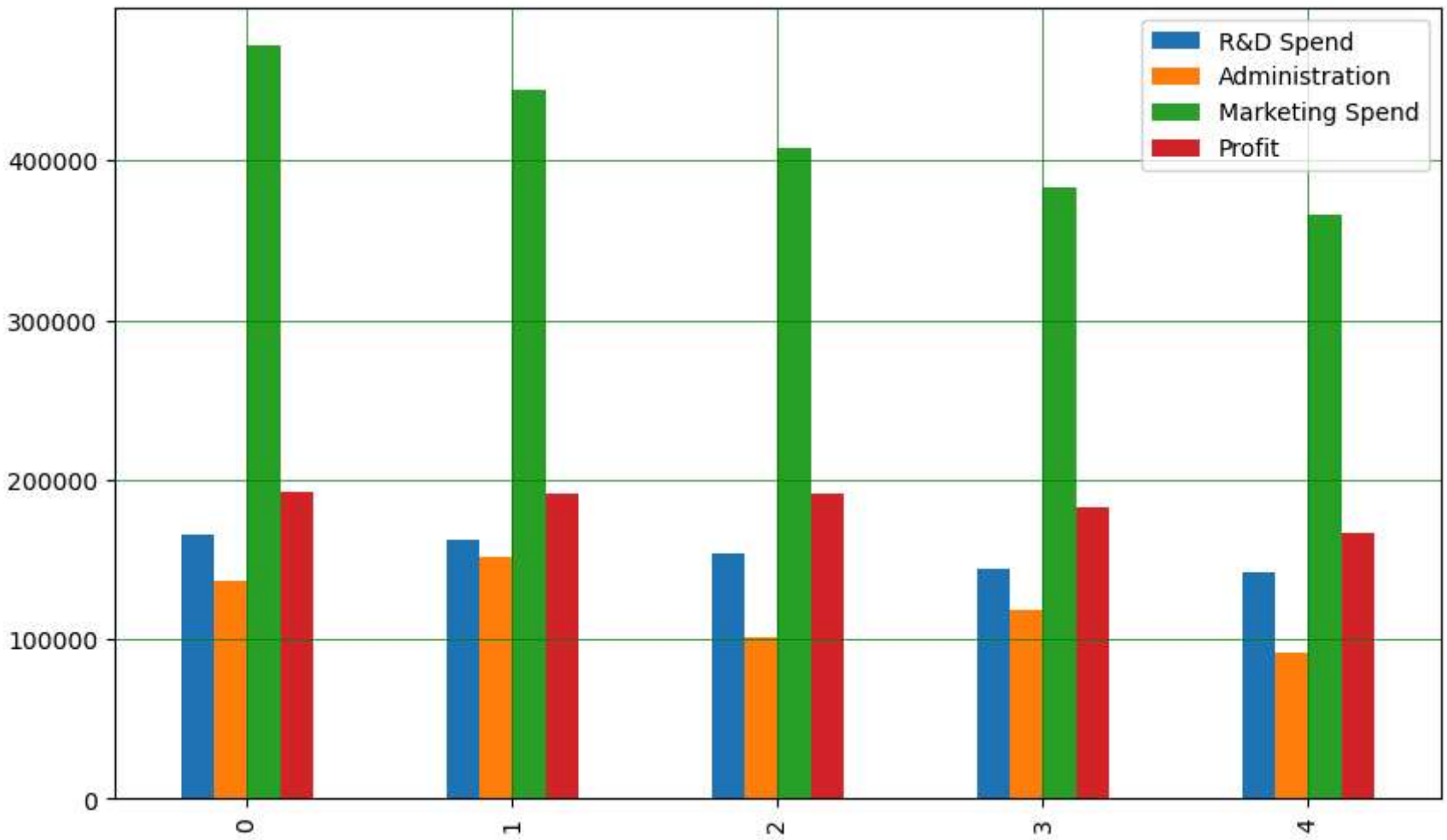
I want to visualize the top 5 companies with highest profit, so I can decide which company I want to buy between them that spends the least and get a high profit

In [98]:
```
df.head()
```

Out[98]:

	R&D 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

In [99]:
```
df.head().plot(kind='bar',figsize=(10,6))
plt.grid(which='major',linestyle='-',linewidth=0.5,color='green')
plt.show()
```



I will choose the company with index 2. Why?
It spends the least between the top 3 and gets a very close profit.

Lastly, lets use the model to how can I have the highest profit with the lowest spendings.

In [100…
```
final_model = lr.fit(X,y)
```

In [101…
```
final_model.coef_
```

Out[101…
```
array([ 0.77712216, -0.06028686,  0.0192574 ])
```

In [158…
```
campaign = [[165000,50000,420000]]
final_model.predict(campaign).round(2)
```

```
C:\Users\Lenovo\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[158…
```
array([192513.02])
```

Conclusion

We can understand that increasing the R&D spending and focus in it, increase Marketing spending a little bit, and decrease the spending of Administration spendings can lead to having a higher profit.

In [ ]:
```
```