In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
plt.rcParams['figure.figsize']=[19,8]
```

In [3]:
```python
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```python
import pandas as pd
startups_df=pd.read_csv("C:\\Users\\megha\\OneDrive\\Documents\\50_Startups.csv")
df=pd.DataFrame(startups_df)
df
```

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
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 [5]:
```python
startups_df.shape
```

Out[5]: (50, 5)

In [7]:
```python
startups_df.head()
```

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

In [8]:
```python
startups_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 [9]:
```python
startups_df.describe()
```

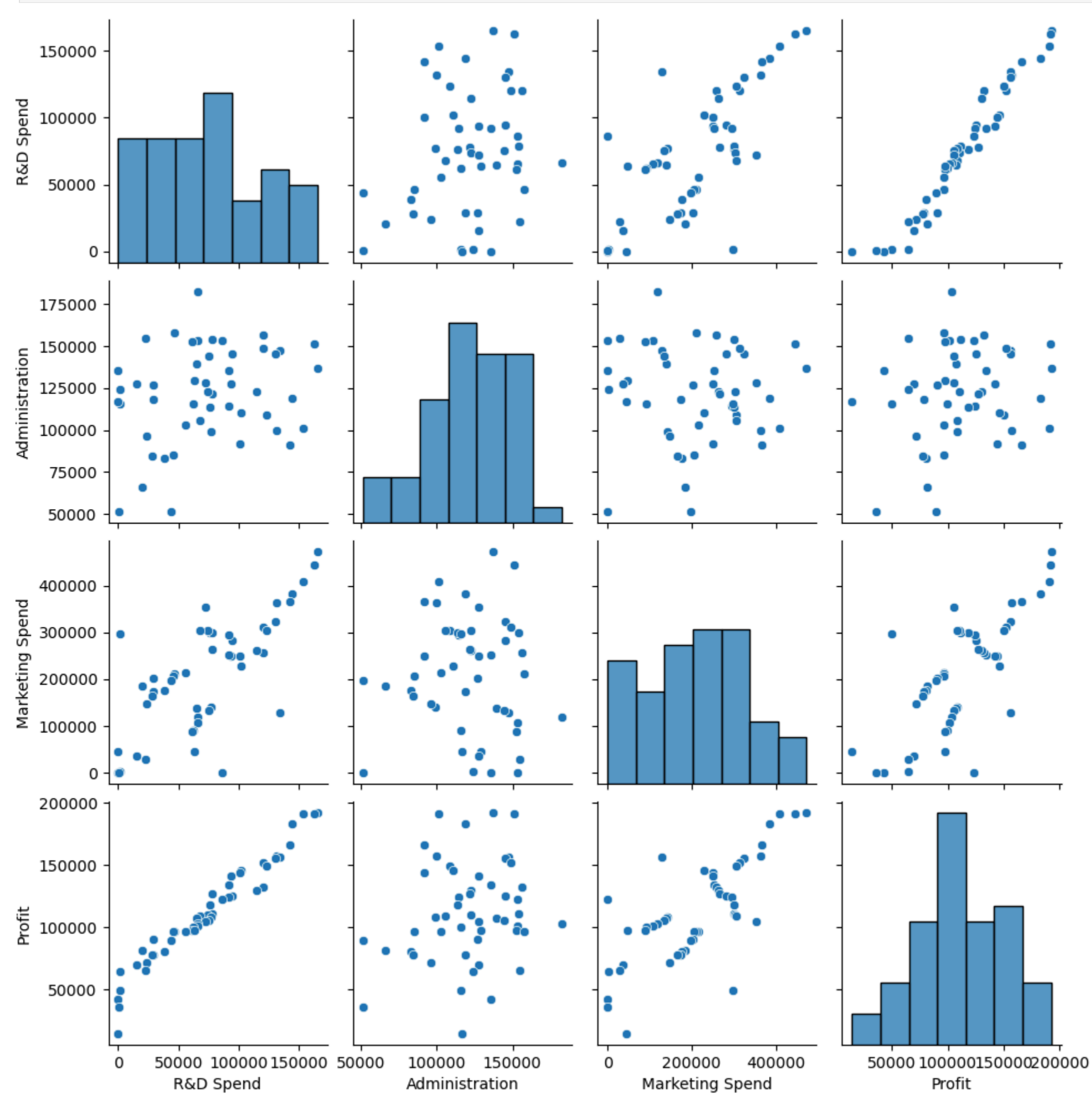
Out[9]:

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

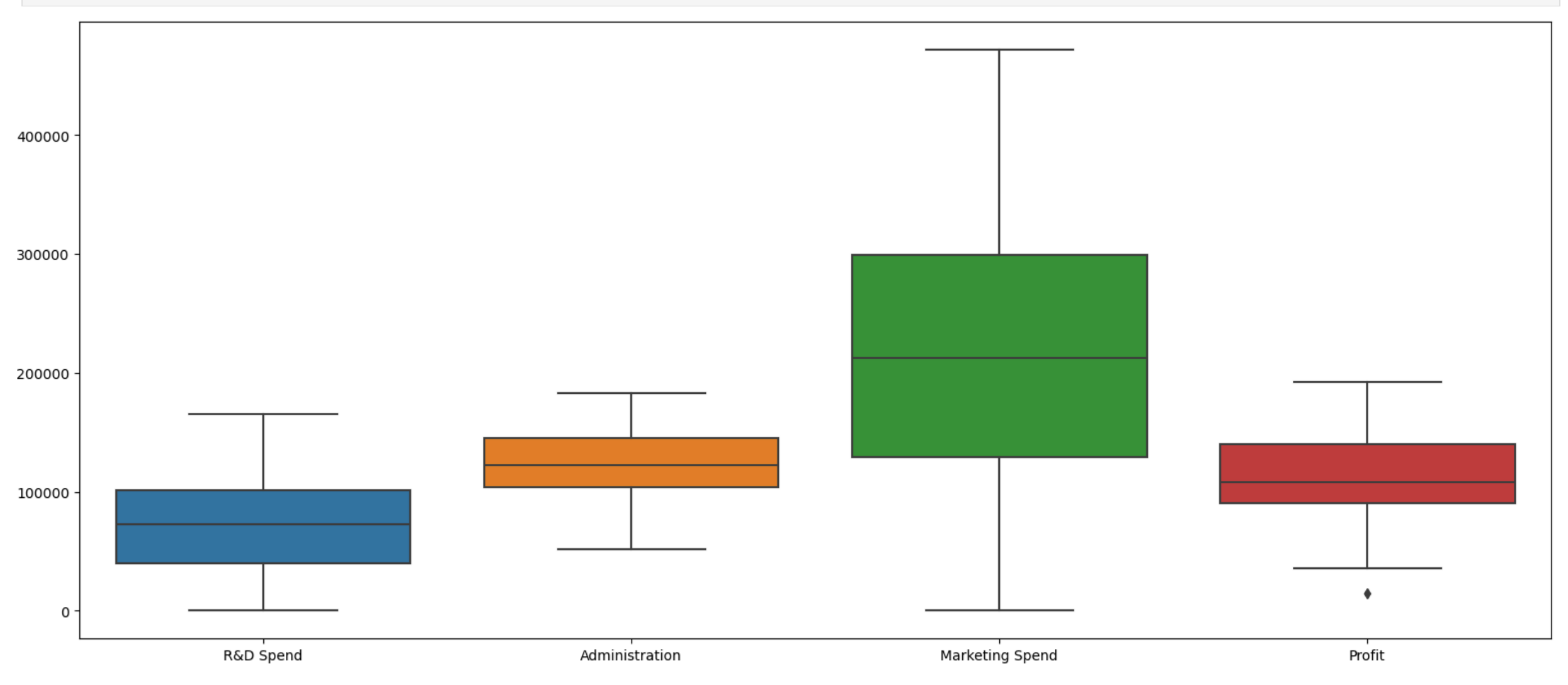
In [10]:
```python
startups_df.duplicated().sum()
```

Out[10]: 0

In [11]:
```python
sns.pairplot(data=startups_df)
plt.show()
```



In [12]:
```python
sns.boxplot(data=startups_df)
plt.show()
```



In [18]:
```python
startups_df = pd.get_dummies( data=startups_df, columns=['State'], drop_first=True, dtype=np.int64)
```

In [19]:
```python
startups_df.head()
```

Out[19]:

	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 [20]:
```python
from sklearn.preprocessing import StandardScaler
```

In [21]:
```python
scaler=StandardScaler()
```

In [22]:
```python
startups_df=pd.DataFrame(scaler.fit_transform(startups_df),columns=startups_df.columns)
```

In [23]:
```python
startups_df.head()
```

Out[23]:

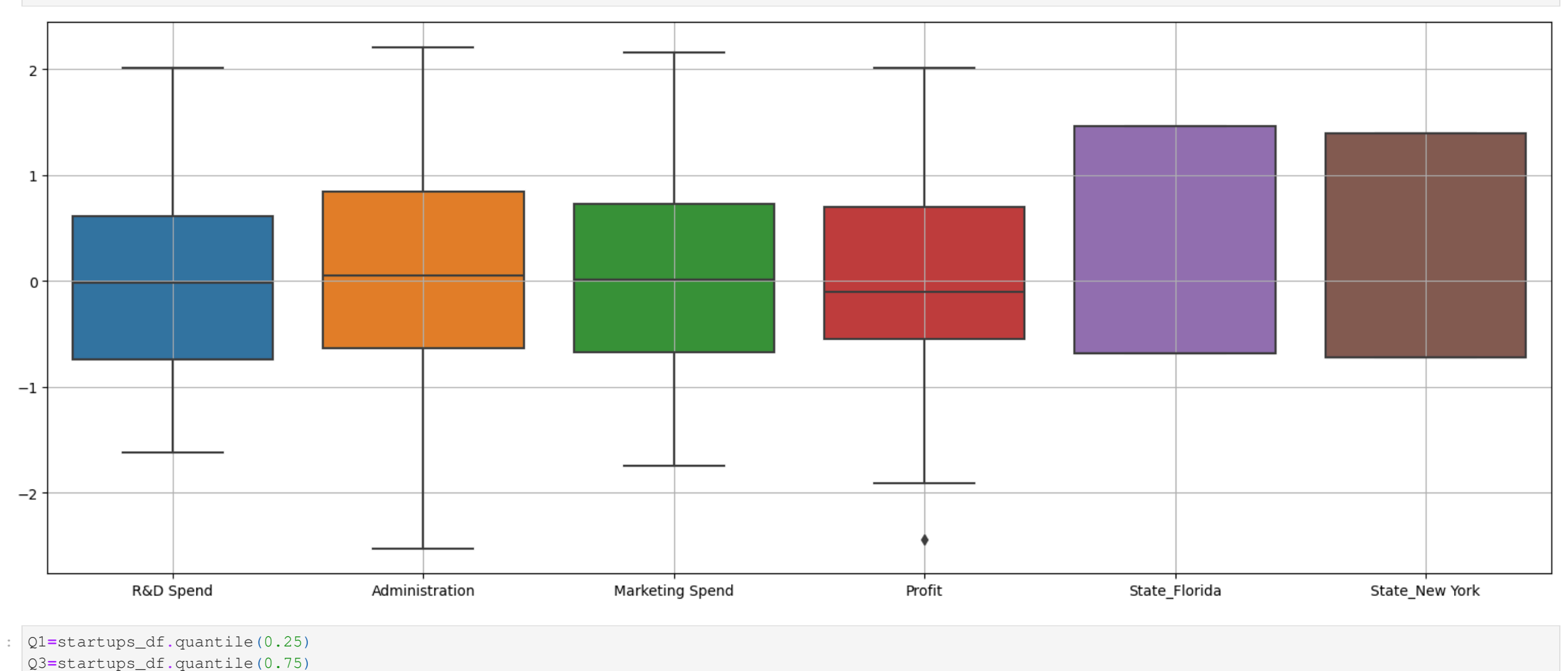
	R&D Spend	Administration	Marketing Spend	Profit	State_Florida	State_New York
0	2.016411	0.560753	2.153943	2.011203	-0.685994	1.393261
1	1.955860	1.082807	1.923600	1.999430	-0.685994	-0.717741
2	1.754364	-0.728257	1.626528	1.980842	1.457738	-0.717741
3	1.554784	-0.096365	1.422210	1.776627	-0.685994	1.393261
4	1.504937	-1.079919	1.281528	1.357740	1.457738	-0.717741

In [24]:
```python
startups_df.describe()
```

Out[24]:

	R&D Spend	Administration	Marketing Spend	Profit	State_Florida	State_New York
count	5.000000e+01	5.000000e+01	5.000000e+01	5.000000e+01	5.000000e+01	5.000000e+01
mean	-7.549517e-17	-2.564615e-16	-1.554312e-16	-5.151435e-16	-7.549517e-17	-1.043610e-16
std	1.010153e+00	1.010153e+00	1.010153e+00	1.010153e+00	1.010153e+00	1.010153e+00
min	-1.622362e+00	-2.525994e+00	-1.743127e+00	-2.439313e+00	-6.859943e-01	-7.177406e-01
25%	-7.434983e-01	-6.350458e-01	-6.750713e-01	-5.481991e-01	-6.859943e-01	-7.177406e-01
50%	-1.475621e-02	4.885871e-02	1.396931e-02	-1.011113e-01	-6.859943e-01	-7.177406e-01
75%	6.135700e-01	8.471792e-01	7.305723e-01	6.955535e-01	1.457738e+00	1.393261e+00
max	2.016411e+00	2.210141e+00	2.153943e+00	2.011203e+00	1.457738e+00	1.393261e+00

In [25]:
```python
plt.figure(figsize=(19,7))
sns.boxplot(startups_df)
plt.grid()
plt.show()
```



In [26]:
```python
Q1=startups_df.quantile(0.25)
Q3=startups_df.quantile(0.75)
IQR=Q3-Q1
print(IQR)
```

```
R&D Spend          1.357068
Administration     1.482225
Marketing Spend    1.405644
Profit             1.243753
State_Florida      2.143732
State_New York     2.111002
dtype: float64
```

In [30]:
```python
ul=Q3+1.5*IQR
ll=Q1-1.5*IQR
```

In [31]:
```python
startups_df=startups_df[~((startups_df<ll)|(startups_df>ul)).any (axis=1)]
```

In [32]:
```python
startups_df.shape
```

Out[32]: (49, 6)

In [33]:
```python
plt.figure(figsize=(19,7))
sns.boxplot(startups_df)
plt.grid()
plt.show()
```

In [36]:
```python
x=startups_df.drop('Profit',axis=1).values
y=startups_df.loc[:,'Profit'].values
```

In [41]:
```python
from sklearn.model_selection import train_test_split
```

In [42]:
```python
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=1)
```

In [43]:
```python
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[43]: ((39, 5), (10, 5), (39,), (10,))

In [44]:
```python
from sklearn.linear_model import LinearRegression
```

In [45]:
```python
linear_model=LinearRegression()
```

In [46]:
```python
linear_model.fit(x_train,y_train)
```

Out[46]:

▾ LinearRegression

LinearRegression()

In [47]:
```python
linear_model.coef_
```

Out[47]: array([0.87740372, -0.04218498, 0.06661508, -0.02635625, -0.02332294])

In [48]:
```python
linear_model.intercept_
```

Out[48]: 0.018473530654336434

In [49]:
```python
linear_model.score(x_train,y_train)
```

Out[49]: 0.9627805058018273

In [50]:
```python
y_predict=linear_model.predict(x_test)
```

In [51]:
```python
y_predict
```

Out[51]:
```
array([ 0.04120213, -0.52915719, -0.58713586, -1.41920057,  1.67515091,
        1.46703762, -0.91011834, -0.25801214, -1.51943683, -0.28694085])
```

In [52]:
```python
from sklearn.metrics import r2_score
```

In [53]:
```python
r2_score(y_test,y_predict)
```

Out[53]: 0.9511446681547796

In [54]:
```python
linear_model.score(x_test,y_test)
```

Out[54]: 0.9511446681547796

In [ ]: