In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
df=pd.read_csv("50_Startups.csv")
df.head()
```

Out[2]:

	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 [3]:
```
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 [4]:
```
df.isnull().sum()
```

Out[4]:
```
R&D Spend          0
Administration     0
Marketing Spend    0
State              0
Profit             0
dtype: int64
```

In [5]:
```
df.drop(["State"],axis=1,inplace=True)
```

In [6]:
```
df.describe()
```

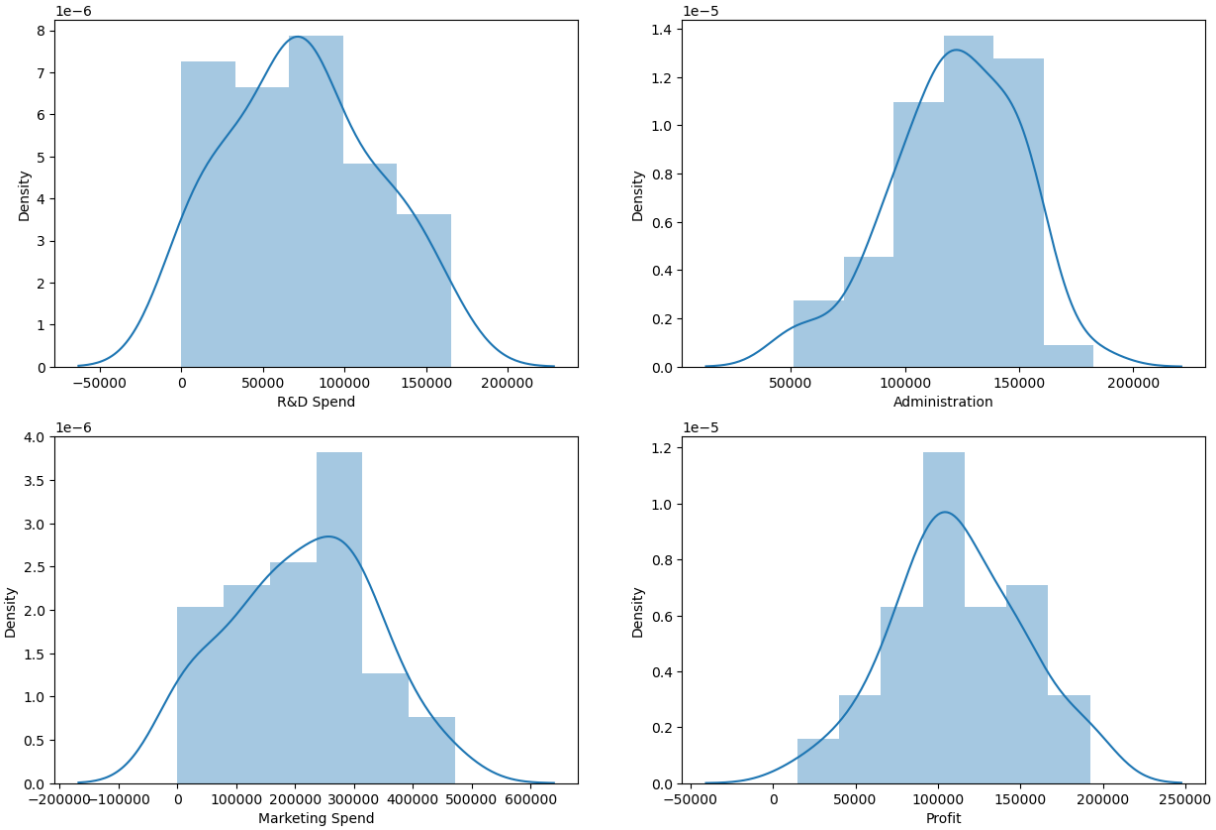
Out[6]:

	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

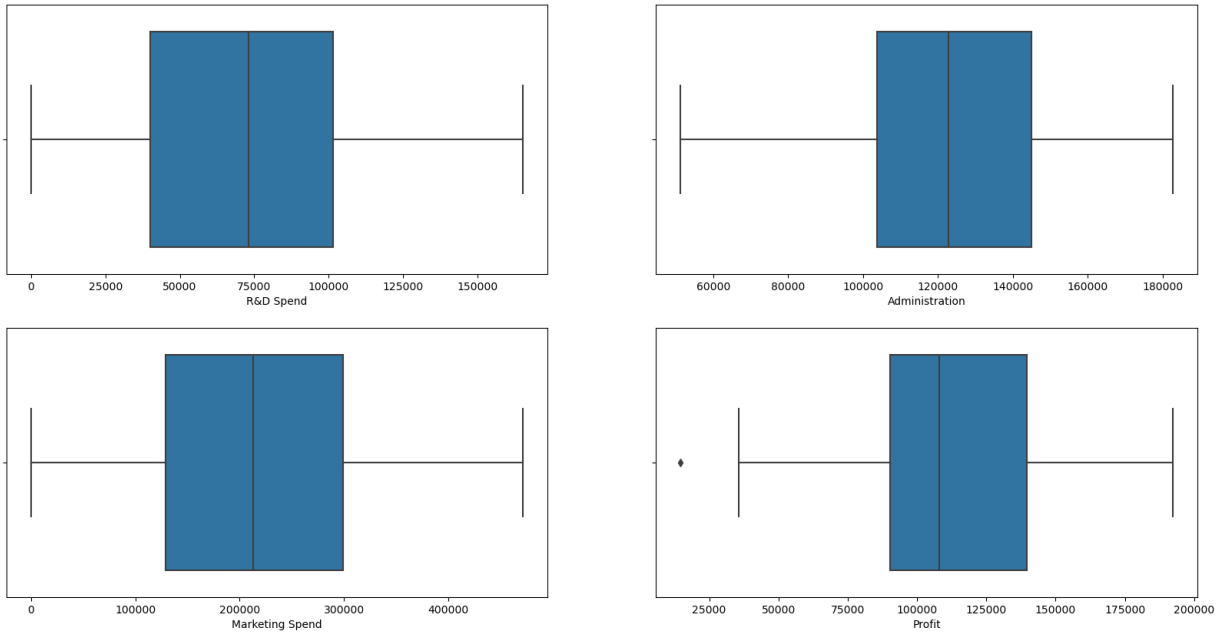
In [7]:
```
num_col=df.select_dtypes(include=["int","float"]).columns
num_col
```

Out[7]:
```
Index(['R&D Spend', 'Administration', 'Marketing Spend', 'Profit'], dtype='obje
ct')
```

In [8]:
```
plt.figure(figsize=(15,10))
count=1
for i in num_col:
    plt.subplot(2,2,count)
    sns.distplot(df[i])
    count+=1
plt.show()
```



In [9]:
```
plt.figure(figsize=(20,10))
count=1
for i in num_col:
    plt.subplot(2,2,count)
    sns.boxplot(df[i])
    count+=1
plt.show()
```



In [10]:
```
df[df["Profit"]<20000]
```

Out[10]:

	R&D Spend	Administration	Marketing Spend	Profit
49	0.0	116983.8	45173.06	14681.4

In [11]:
```
df.drop(index=49,inplace=True)
```

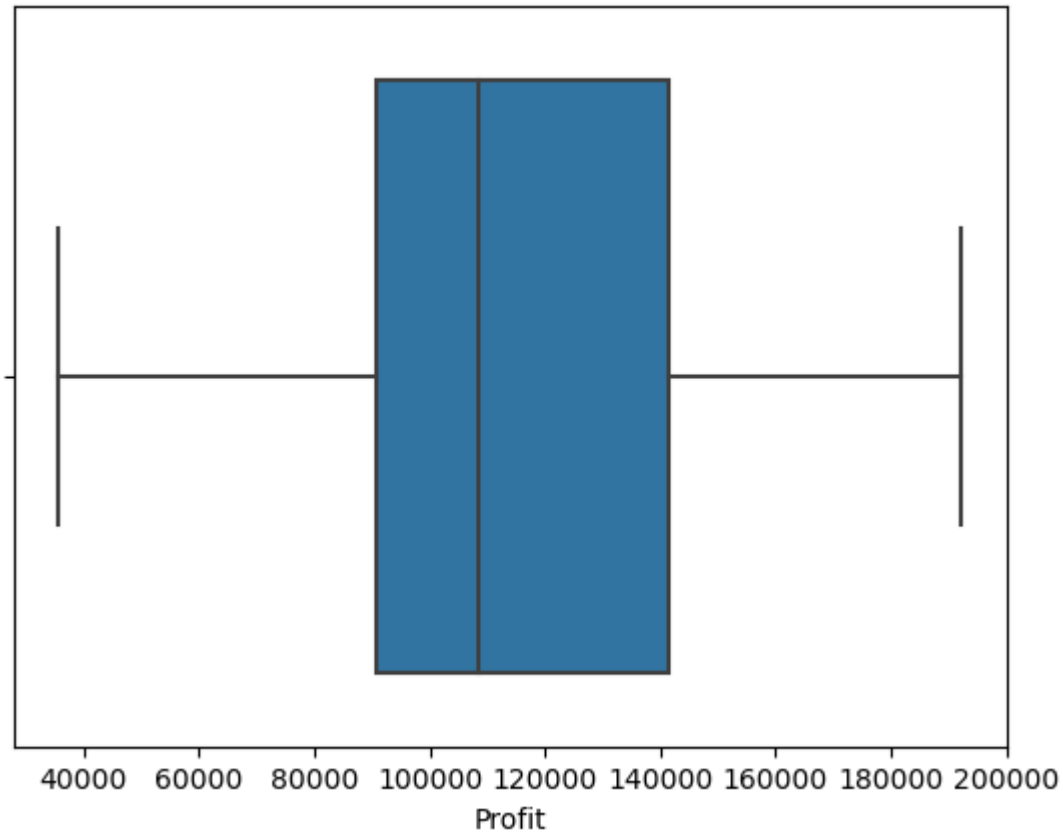
In [12]:
```
df.describe()
```

Out[12]:

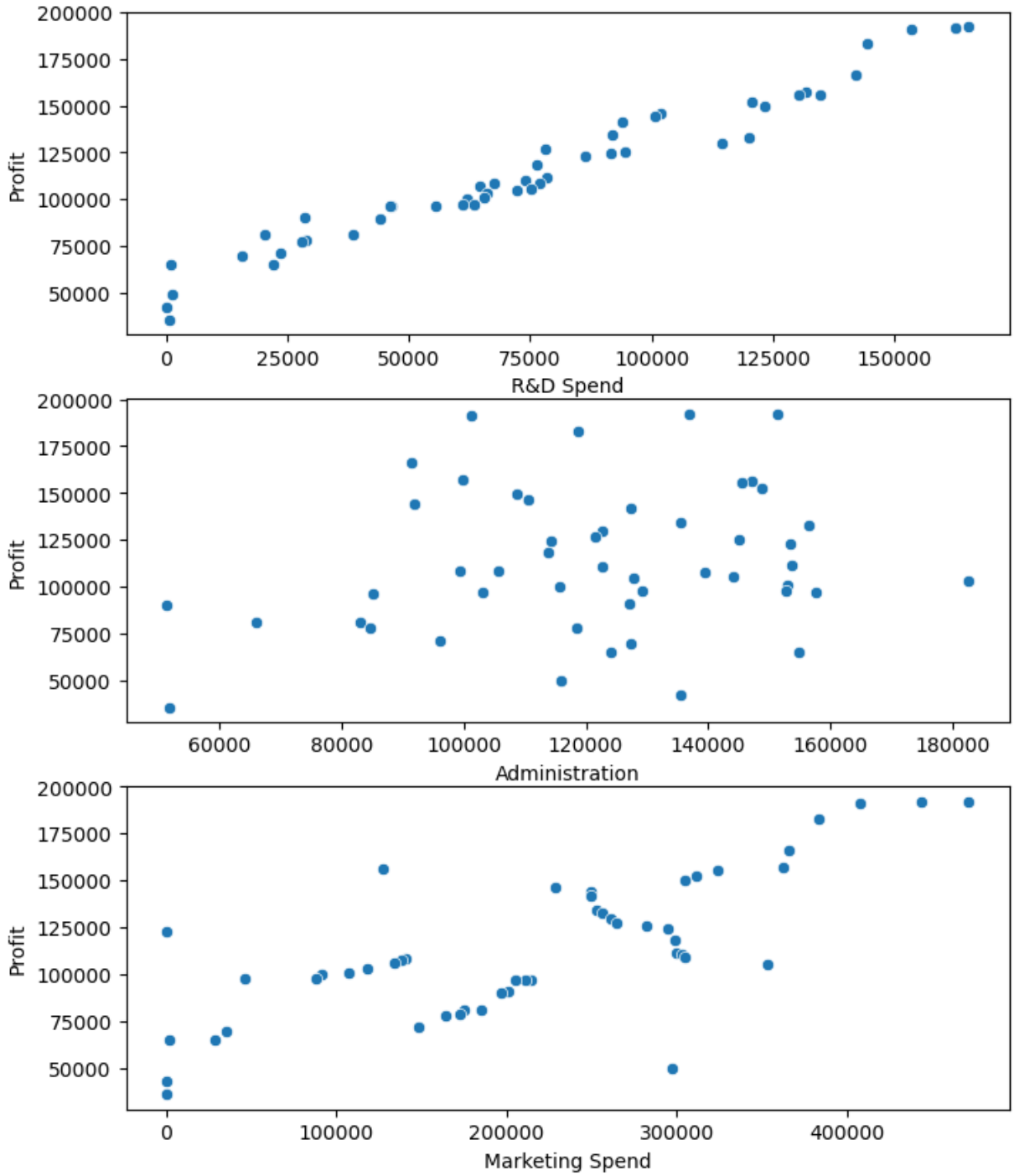
	R&D Spend	Administration	Marketing Spend	Profit
count	49.000000	49.000000	49.000000	49.000000
mean	75226.138367	121433.636327	214409.833265	113998.991020
std	45115.141560	28301.008988	121168.170072	38171.246893
min	0.000000	51283.140000	0.000000	35673.410000
25%	44069.950000	103057.490000	134050.070000	90708.190000
50%	73994.560000	122782.750000	214634.810000	108552.040000
75%	101913.080000	145077.580000	299737.290000	141585.520000
max	165349.200000	182645.560000	471784.100000	192261.830000

In [13]:
```
sns.boxplot(df["Profit"])
```

Out[13]:
```
<AxesSubplot:xlabel='Profit'>
```



In [14]:
```
plt.figure(figsize=(8,10))
count=1
for i in num_col:
    if i!="Profit":
        plt.subplot(3,1,count)
        sns.scatterplot(df[i],df["Profit"])
        count+=1
plt.show()
```



In [15]:
```
x=df[["R&D Spend","Marketing Spend"]]
```

In [16]:
```
y=df["Profit"]
```

In [17]:
```
x=df.iloc[:,:-1]
y=df.iloc[:,-1]
```

In [18]:
```
x
```

Out[18]:

	R&D Spend	Administration	Marketing Spend
0	165349.20	136897.80	471784.10
1	162597.70	151377.59	443898.53
2	153441.51	101145.55	407934.54
3	144372.41	118671.85	383199.62
4	142107.34	91391.77	366168.42
5	131876.90	99814.71	362861.36
6	134615.46	147198.87	127716.82
7	130298.13	145530.06	323876.68
8	120542.52	148718.95	311613.29
9	123334.88	108679.17	304981.62
10	101913.08	110594.11	229160.95
11	100671.96	91790.61	249744.55
12	93863.75	127320.38	249839.44
13	91992.39	135495.07	252664.93
14	119943.24	156547.42	256512.92
15	114523.61	122616.84	261776.23
16	78013.11	121597.55	264346.06
17	94657.16	145077.58	282574.31
18	91749.16	114175.79	294919.57
19	86419.70	153514.11	0.00
20	76253.86	113867.30	298664.47
21	78389.47	153773.43	299737.29
22	73994.56	122782.75	303319.26
23	67532.53	105751.03	304768.73
24	77044.01	99281.34	140574.81
25	64664.71	139553.16	137962.62
26	75328.87	144135.98	134050.07
27	72107.60	127864.55	353183.81
28	66051.52	182645.56	118148.20
29	65605.48	153032.06	107138.38
30	61994.48	115641.28	91131.24
31	61136.38	152701.92	88218.23
32	63408.86	129219.61	46085.25
33	55493.95	103057.49	214634.81

	R&D Spend	Administration	Marketing Spend
34	46426.07	157693.92	210797.67
35	46014.02	85047.44	205517.64
36	28663.76	127056.21	201126.82
37	44069.95	51283.14	197029.42
38	20229.59	65947.93	185265.10
39	38558.51	82982.09	174999.30
40	28754.33	118546.05	172795.67
41	27892.92	84710.77	164470.71
42	23640.93	96189.63	148001.11
43	15505.73	127382.30	35534.17
44	22177.74	154806.14	28334.72
45	1000.23	124153.04	1903.93
46	1315.46	115816.21	297114.46
47	0.00	135426.92	0.00
48	542.05	51743.15	0.00

In [19]:
```
y
```

Out[19]:
```
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
Name: Profit, dtype: float64
```

In [20]:
```
from sklearn.model_selection import train_test_split
```

In [21]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=12
```

In [22]:
```
x_train.shape
```

Out[22]: (39, 3)

In [23]:
```
y_train.shape
```

Out[23]: (39,)

In [24]:
```
x_test.shape
```

Out[24]: (10, 3)

In [25]:
```
y_test.shape
```

Out[25]: (10,)

In [26]:
```
from sklearn.linear_model import LinearRegression
```

In [27]:
```
mlr=LinearRegression()
```

In [28]:
```
mlr.fit(x_train,y_train)
```

Out[28]: LinearRegression()

In [29]:
```
y_pred_train=mlr.predict(x_train)
y_pred_test=mlr.predict(x_test)
```

In [30]:
```
from sklearn.metrics import r2_score,mean_squared_error
```

In [31]:
```
def model_performance(y_actual,y_pred):
    r2=r2_score(y_actual,y_pred)
    RMSE=np.sqrt(mean_squared_error(y_actual,y_pred))
    print("R2 Score:{}|RMSE:{}".format(round(r2,2),round(RMSE,2)))
```

In [32]:
```
print("Train Performance")
model_performance(y_train,y_pred_train)
```

```
Train Performance
R2 Score:0.96|RMSE:7169.27
```

In [33]:
```
print("Test Performance")
model_performance(y_test,y_pred_test)
```

```
Test Performance
R2 Score:0.95|RMSE:9276.04
```

In [ ]:
```

```