```
import numpy as np
import pandas as pd
```

```
#Load a built datasets from the git web
df=pd.read_csv('https://raw.githubusercontent.com/arib168/data/main/50_Startups.csv')
df
```

18	91749.16	114175.79	294919.57	Florida	1242
19	86419.70	153514.11	0.00	New York	1227
20	76253.86	113867.30	298664.47	California	1184
21	78389.47	153773.43	299737.29	New York	1113
22	73994.56	122782.75	303319.26	Florida	1103
23	67532.53	105751.03	304768.73	Florida	1087
24	77044.01	99281.34	140574.81	New York	1085
25	64664.71	139553.16	137962.62	California	1074
26	75328.87	144135.98	134050.07	Florida	1057
27	72107.60	127864.55	353183.81	New York	1050
28	66051.52	182645.56	118148.20	Florida	1032
29	65605.48	153032.06	107138.38	New York	1010
30	61994.48	115641.28	91131.24	Florida	999
31	61136.38	152701.92	88218.23	New York	974
32	63408.86	129219.61	46085.25	California	974
33	55493.95	103057.49	214634.81	Florida	967
34	46426.07	157693.92	210797.67	California	967
35	46014.02	85047.44	205517.64	New York	964
36	28663.76	127056.21	201126.82	Florida	907
37	44069.95	51283.14	197029.42	California	899
38	20229.59	65947.93	185265.10	New York	812

```
df.head()
```

	R&D Spend	Administration	Marketing Spend	State	Profi
0	165349.20	136897.80	471784.10	New York	192261.8
1	162597.70	151377.59	443898.53	California	191792.0
2	153441.51	101145.55	407934.54	Florida	191050.3
3	144372.41	118671.85	383199.62	New York	182901.9
47	0.00	135426.92	0.00	California	425

```
df.tail()
```

	R&D Spend	Administration	Marketing Spend	State	Profit
45	1000.23	124153.04	1903.93	New York	64926.08

```
df.columns
```

```
Index(['R&D Spend', 'Administration', 'Marketing Spend', 'State', 'Profit'],
dtype='object')
```

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

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

```
df.dtypes
```

```
R&D Spend          float64
Administration     float64
Marketing Spend    float64
State               object
Profit             float64
dtype: object
```

```
z=df['State'].value_counts()
z
```

```
New York      17
California    17
Florida       16
Name: State, dtype: int64
```

```
import matplotlib.pyplot as plt
y= [17,17,16]
labels_x=['New York','California','Florida']
plt.pie(y,labels=labels_x)
```

```
([<matplotlib.patches.Wedge at 0x7f3652102b50>
```

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

	R&D Spend	Administration	Marketing Spend	State
0	165349.20	136897.80	471784.10	New York
1	162597.70	151377.59	443898.53	California
2	153441.51	101145.55	407934.54	Florida
3	144372.41	118671.85	383199.62	New York
4	142107.34	91391.77	366168.42	Florida
5	131876.90	99814.71	362861.36	New York
6	134615.46	147198.87	127716.82	California
7	130298.13	145530.06	323876.68	Florida
8	120542.52	148718.95	311613.29	New York
9	123334.88	108679.17	304981.62	California
10	101913.08	110594.11	229160.95	Florida
11	100671.96	91790.61	249744.55	California
12	93863.75	127320.38	249839.44	Florida
13	91992.39	135495.07	252664.93	California
14	119943.24	156547.42	256512.92	Florida
15	114523.61	122616.84	261776.23	New York
16	78013.11	121597.55	264346.06	California

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

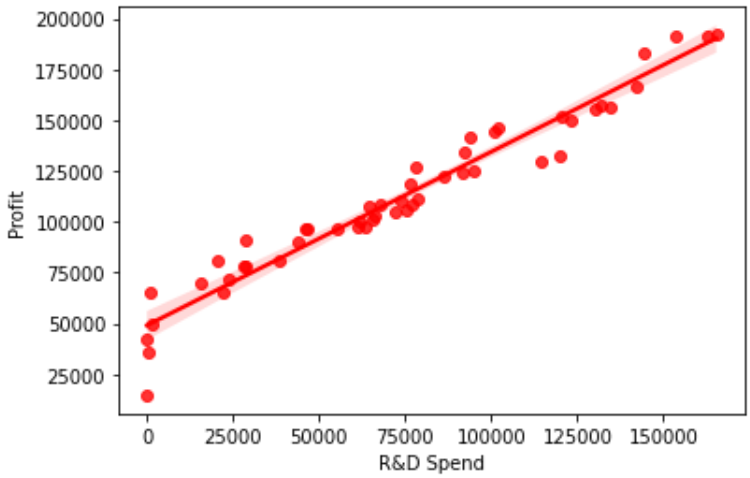
```
0     192261.83
1     191792.06
2     191050.39
3     182901.99
4     166187.94
5     156991.12
6     156122.51
7     155752.60
8     152211.77
9     149759.96
10    146121.95
11    144259.40
12    141585.52
13    134307.35
14    132602.65
15    129917.04
16    126992.93
17    125370.37
18    124266.90
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
```

```
48      542.05          51743.15                0.00   New York
```

```
import seaborn as sns
sns.regplot(x=df['R&D Spend'],y=y,color='red')
```

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

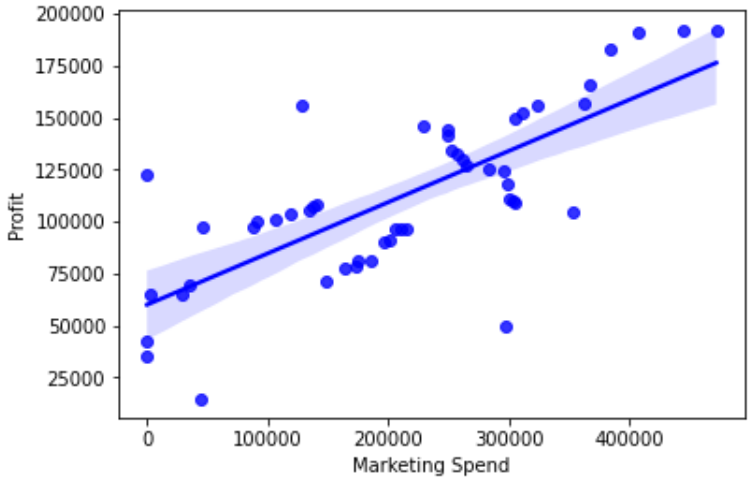



```
sns.regplot(x=df['Administration'],y=y,color='indigo')
```

```
<AxesSubplot:xlabel='Administration', ylabel='Profit'>
```

```
sns.regplot(x=df['Marketing Spend'],y=y,color='b')
```

```
<AxesSubplot:xlabel='Marketing Spend', ylabel='Profit'>
```




```
from sklearn.compose import make_column_transformer
from sklearn.preprocessing import OneHotEncoder
col_trans=make_column_transformer((OneHotEncoder(handle_unknown='ignore'),['State']),remainder='p
x=col_trans.fit_transform(x)
x
```

```
array([[0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 1.6534920e+05,
        1.3689780e+05, 4.7178410e+05],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 1.6259770e+05,
        1.5137759e+05, 4.4389853e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 1.5344151e+05,
        1.0114555e+05, 4.0793454e+05],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 1.4437241e+05,
        1.1867185e+05, 3.8319962e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 1.4210734e+05,
        9.1391770e+04, 3.6616842e+05],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 1.3187690e+05,
        9.9814710e+04, 3.6286136e+05],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 1.3461546e+05,
        1.4719887e+05, 1.2771682e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 1.3029813e+05,
        1.4553006e+05, 3.2387668e+05],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 1.2054252e+05,
        1.4871895e+05, 3.1161329e+05],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 1.2333488e+05,
        1.0867917e+05, 3.0498162e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 1.0191308e+05,
        1.1059411e+05, 2.2916095e+05],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 1.0067196e+05,
        9.1790610e+04, 2.4974455e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 9.3863750e+04,
        1.2732038e+05, 2.4983944e+05],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 9.1992390e+04,
        1.3549507e+05, 2.5266493e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 1.1994324e+05,
        1.5654742e+05, 2.5651292e+05],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 1.1452361e+05,
        1.2261684e+05, 2.6177623e+05],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 7.8013110e+04,
        1.2159755e+05, 2.6434606e+05],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 9.4657160e+04,
```

```
       1.4507758e+05, 2.8257431e+05],
      [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 9.1749160e+04,
       1.1417579e+05, 2.9491957e+05],
      [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 8.6419700e+04,
       1.5351411e+05, 0.0000000e+00],
      [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 7.6253860e+04,
       1.1386730e+05, 2.9866447e+05],
      [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 7.8389470e+04,
       1.5377343e+05, 2.9973729e+05],
      [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 7.3994560e+04,
       1.2278275e+05, 3.0331926e+05],
      [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 6.7532530e+04,
       1.0575103e+05, 3.0476873e+05],
      [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 7.7044010e+04,
       9.9281340e+04, 1.4057481e+05],
      [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 6.4664710e+04,
       1.3955316e+05, 1.3796262e+05],
      [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 7.5328870e+04,
       1.4413598e+05, 1.3405007e+05],
      [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 7.2107600e+04,
       1.2786455e+05, 3.5318381e+05],
      [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 6.6051520e+04,
       1.8264556e+05, 1.1814820e+05],
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.30,random_state=42)
x_train
```

```
array([[1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 1.3461546e+05,
        1.4719887e+05, 1.2771682e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 2.7892920e+04,
        8.4710770e+04, 1.6447071e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 1.3154600e+03,
        1.1581621e+05, 2.9711446e+05],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 0.0000000e+00,
        1.3542692e+05, 0.0000000e+00],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 1.1452361e+05,
        1.2261684e+05, 2.6177623e+05],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 1.2333488e+05,
        1.0867917e+05, 3.0498162e+05],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 7.8013110e+04,
        1.2159755e+05, 2.6434606e+05],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 7.7044010e+04,
        9.9281340e+04, 1.4057481e+05],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 4.6426070e+04,
        1.5769392e+05, 2.1079767e+05],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 6.1136380e+04,
        1.5270192e+05, 8.8218230e+04],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 1.6534920e+05,
        1.3689780e+05, 4.7178410e+05],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 2.2177740e+04,
        1.5480614e+05, 2.8334720e+04],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 7.2107600e+04,
        1.2786455e+05, 3.5318381e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 5.5493950e+04,
        1.0305749e+05, 2.1463481e+05],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 1.3187690e+05,
        9.9814710e+04, 3.6286136e+05],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 6.5605480e+04,
        1.5303206e+05, 1.0713838e+05],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 1.0067196e+05,
        9.1790610e+04, 2.4974455e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 2.8663760e+04,
        1.2705621e+05, 2.0112682e+05],
```

```
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 1.6259770e+05,
        1.5137759e+05, 4.4389853e+05],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 7.8389470e+04,
        1.5377343e+05, 2.9973729e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 1.5344151e+05,
        1.0114555e+05, 4.0793454e+05],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 1.5505730e+04,
        1.2738230e+05, 3.5534170e+04],
       [0.0000000e+00, 0.0000000e+00, 1.0000000e+00, 4.6014020e+04,
        8.5047440e+04, 2.0551764e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 6.7532530e+04,
        1.0575103e+05, 3.0476873e+05],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 2.8754330e+04,
        1.1854605e+05, 1.7279567e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 1.0191308e+05,
        1.1059411e+05, 2.2916095e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 7.3994560e+04,
        1.2278275e+05, 3.0331926e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, 9.1749160e+04,
        1.1417579e+05, 2.9491957e+05],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 0.0000000e+00,
        1.1698380e+05, 4.5173060e+04],
```

```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
y_pred
```

```
array([126187.39411505,  85788.82259512,  99777.02815177,  45706.12238326,
       127062.20722772,  51891.83884457, 109114.62977494, 100600.61123701,
        97953.99874714, 111730.57706807, 128818.49200668, 174195.35772633,
        93736.28538439, 148381.04097161, 172313.8713939 ])
```

```
y_test
```

```
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
12    141585.52
4     166187.94
37     89949.14
8     152211.77
3     182901.99
Name: Profit, dtype: float64
```

```
print('Slope is',model.coef_)
print('Constant value is',model.intercept_)
```

```
Slope is [ 2.59028652e+02  7.17099427e+02 -9.76128080e+02  8.04937292e-01
 -9.12577104e-02  2.80672826e-02]
Constant value is 57153.61206241345
```

```
from sklearn.metrics import mean_absolute_error
print('MAE is',mean_absolute_error(y_test,y_pred))
```

```
MAE is 7395.4335315232565
```

```
from sklearn.metrics import mean_absolute_percentage_error
print('"ERROR percentage is',mean_absolute_percentage_error(y_test,y_pred))
```

```
"ERROR percentage is 0.08929865344171896
```

```
from sklearn.metrics import r2_score
print('R2 score is',r2_score(y_test,y_pred))
```

```
R2 score is 0.9397108063355675
```

```
from sklearn.metrics import mean_squared_error
X=mean_squared_error(y_test,y_pred)
X
```

```
84826955.03534974
```

```
np.sqrt(X)
```

```
9210.154995186007
```

```
x.shape
```

```
(50, 6)
```