PROBLEM 1

In [25]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.preprocessing import StandardScaler
import warnings
warnings.filterwarnings("ignore")
```

In [6]:

```
#reading the data
df=pd.read_csv('Consumo_cerveja.csv')
df
```

Out[6]:

	Data	Temperatura Media (C)	Temperatura Minima (C)	Temperatura Maxima (C)	Precipitacao (mm)	Final de Semana	Consumo de cerveja (litros)
0	2015-01-01	27,3	23,9	32,5	0	0.0	25.461
1	2015-01-02	27,02	24,5	33,5	0	0.0	28.972
2	2015-01-03	24,82	22,4	29,9	0	1.0	30.814
3	2015-01-04	23,98	21,5	28,6	1,2	1.0	29.799
4	2015-01-05	23,82	21	28,3	0	0.0	28.900
...	...	...	...	...	...	...	...
936	NaN	NaN	NaN	NaN	NaN	NaN	NaN
937	NaN	NaN	NaN	NaN	NaN	NaN	NaN
938	NaN	NaN	NaN	NaN	NaN	NaN	NaN
939	NaN	NaN	NaN	NaN	NaN	NaN	NaN
940	NaN	NaN	NaN	NaN	NaN	NaN	NaN

941 rows × 7 columns

In [7]:

```
df.describe()
```

Out[7]:

	Final de Semana	Consumo de cerveja (litros)
count	365.000000	365.000000
mean	0.284932	25.401367
std	0.452001	4.399143
min	0.000000	14.343000
25%	0.000000	22.008000

	Final de Semana	Consumo de cerveja (litros)
50%	0.000000	24.867000
75%	1.000000	28.631000
max	1.000000	37.937000

In [8]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 941 entries, 0 to 940
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Data                          365 non-null    object 
 1   Temperatura Media (C)         365 non-null    object 
 2   Temperatura Minima (C)        365 non-null    object 
 3   Temperatura Maxima (C)        365 non-null    object 
 4   Precipitacao (mm)             365 non-null    object 
 5   Final de Semana               365 non-null    float64
 6   Consumo de cerveja (litros)   365 non-null    float64
dtypes: float64(2), object(5)
memory usage: 51.6+ KB
```

In [9]:

```
df.shape
```

Out[9]:

```
(941, 7)
```

In [10]:

```
#removing nan values and duplicate values
df=df.dropna(axis=0).reset_index(drop=True)
df.drop_duplicates()
```

Out[10]:

	Data	Temperatura Media (C)	Temperatura Minima (C)	Temperatura Maxima (C)	Precipitacao (mm)	Final de Semana	Consumo de cerveja (litros)
0	2015-01-01	27,3	23,9	32,5	0	0.0	25.461
1	2015-01-02	27,02	24,5	33,5	0	0.0	28.972
2	2015-01-03	24,82	22,4	29,9	0	1.0	30.814
3	2015-01-04	23,98	21,5	28,6	1,2	1.0	29.799
4	2015-01-05	23,82	21	28,3	0	0.0	28.900
...	...	...	...	...	...	...	...
360	2015-12-27	24	21,1	28,2	13,6	1.0	32.307
361	2015-12-28	22,64	21,1	26,7	0	0.0	26.095
362	2015-12-29	21,68	20,3	24,1	10,3	0.0	22.309
363	2015-12-30	21,38	19,3	22,4	6,3	0.0	20.467
364	2015-12-31	24,76	20,2	29	0	0.0	22.446

365 rows × 7 columns

In [11]:

```
#converting the data into float
import re
for column in ['Temperatura Media (C)','Temperatura Minima (C)','Temperatura Maxima (C)'
,'Precipitacao (mm)']:
    df[column]=df[column].apply(lambda x: np.float(re.sub(r',','.',x)))
```

In [12]:

```
#splitting data to month year and day
df["Data"]=pd.to_datetime(df["Data"])
df["Year"]=df["Data"].apply(lambda x: x.year)
df["Month"]=df["Data"].apply(lambda x: x.month)
df["Day"]=df["Data"].apply(lambda x: x.day)
df=df.drop("Data",axis=1)
```

In [ ]:

```

```

In [13]:

```
df.head()
```

Out[13]:

	Temperatura Media (C)	Temperatura Minima (C)	Temperatura Maxima (C)	Precipitacao (mm)	Final de Semana	Consumo de cerveja (litros)	Year	Month	Day
0	27.30	23.9	32.5	0.0	0.0	25.461	2015	1	1
1	27.02	24.5	33.5	0.0	0.0	28.972	2015	1	2
2	24.82	22.4	29.9	0.0	1.0	30.814	2015	1	3
3	23.98	21.5	28.6	1.2	1.0	29.799	2015	1	4
4	23.82	21.0	28.3	0.0	0.0	28.900	2015	1	5

In [14]:

```
df.nunique()
```

Out[14]:

```
Temperatura Media (C)          277
Temperatura Minima (C)         110
Temperatura Maxima (C)         151
Precipitacao (mm)               93
Final de Semana                  2
Consumo de cerveja (litros)    359
Year                             1
Month                           12
Day                             31
dtype: int64
```

In [15]:

```
#handling outliers
df.describe(include='all')
```

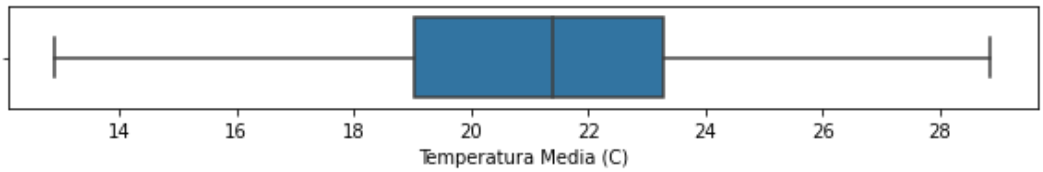
Out[15]:

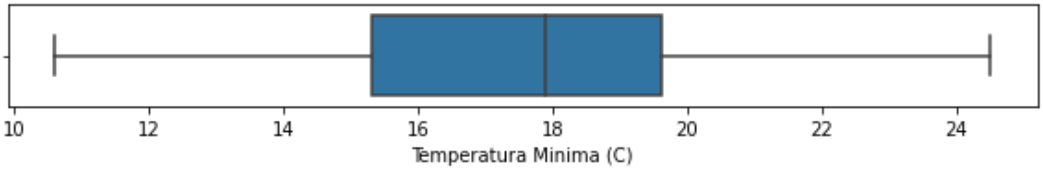
	Temperatura Media (C)	Temperatura Minima (C)	Temperatura Maxima (C)	Precipitacao (mm)	Final de Semana	Consumo de cerveja (litros)	Year	Month	Day
count	365.000000	365.000000	365.000000	365.000000	365.000000	365.000000	365.0	365.000000	365.000000
mean	21.226356	17.461370	26.611507	5.196712	0.284932	25.401367	2015.0	6.526027	15.720548
std	3.180108	2.826185	4.317366	12.417844	0.452001	4.399143	0.0	3.452584	8.808321

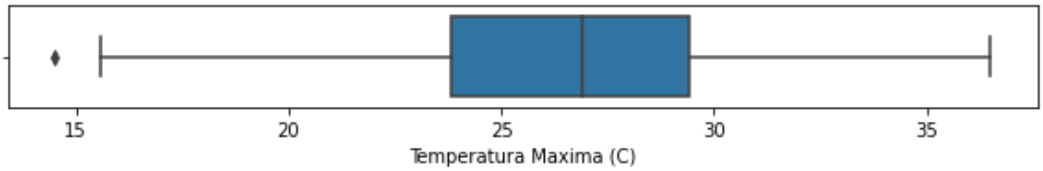
	Temperatura Media (C)	Temperatura Minima (C)	Temperatura Maxima (C)	Precipitacao (mm)	Final de Semana	Consumo de cerveja (litros)	Year	Month	Day
min	12.900000	10.600000	18.300000	0.000000	0.000000	14.343000	2015.0	1.000000	1.000000
25%	19.020000	15.300000	23.800000	0.000000	0.000000	22.008000	2015.0	4.000000	8.000000
50%	21.380000	17.900000	26.900000	0.000000	0.000000	24.867000	2015.0	7.000000	16.000000
75%	23.280000	19.600000	29.400000	3.200000	1.000000	28.631000	2015.0	10.000000	23.000000
max	28.860000	24.500000	36.500000	94.800000	1.000000	37.937000	2015.0	12.000000	31.000000

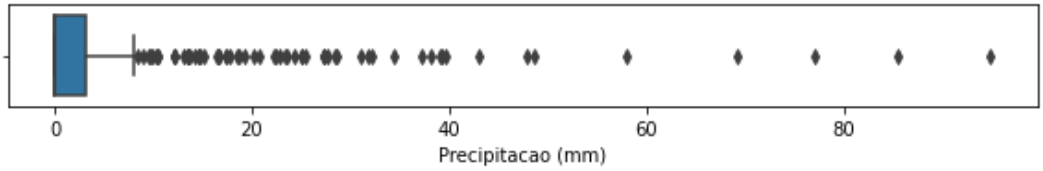
In [16]:

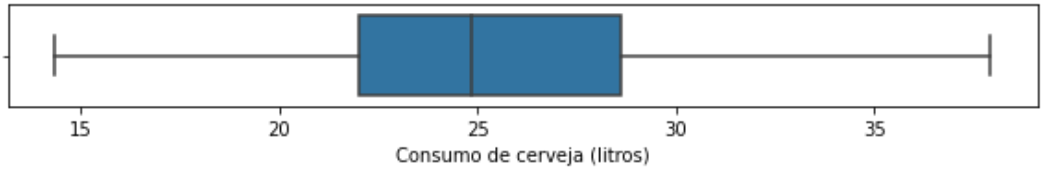
```
#outliers
for column in df.columns[0:-1]:
    plt.figure(figsize=(10,1))
    sns.boxplot(x=(column),data=df)
```

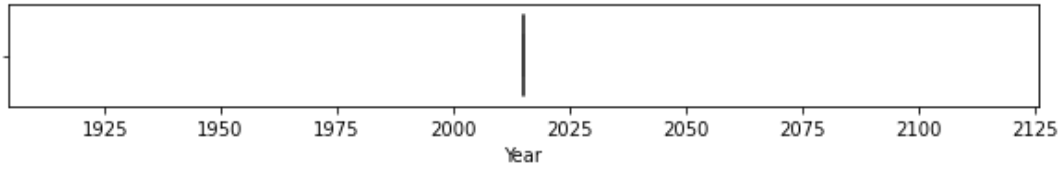


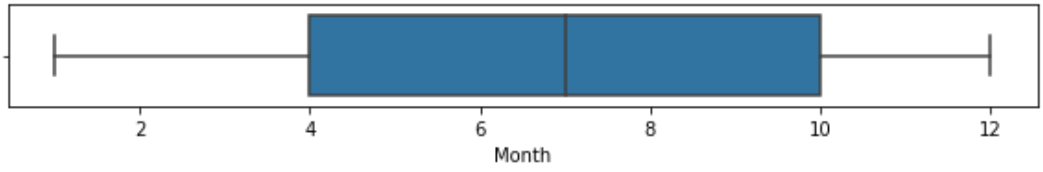










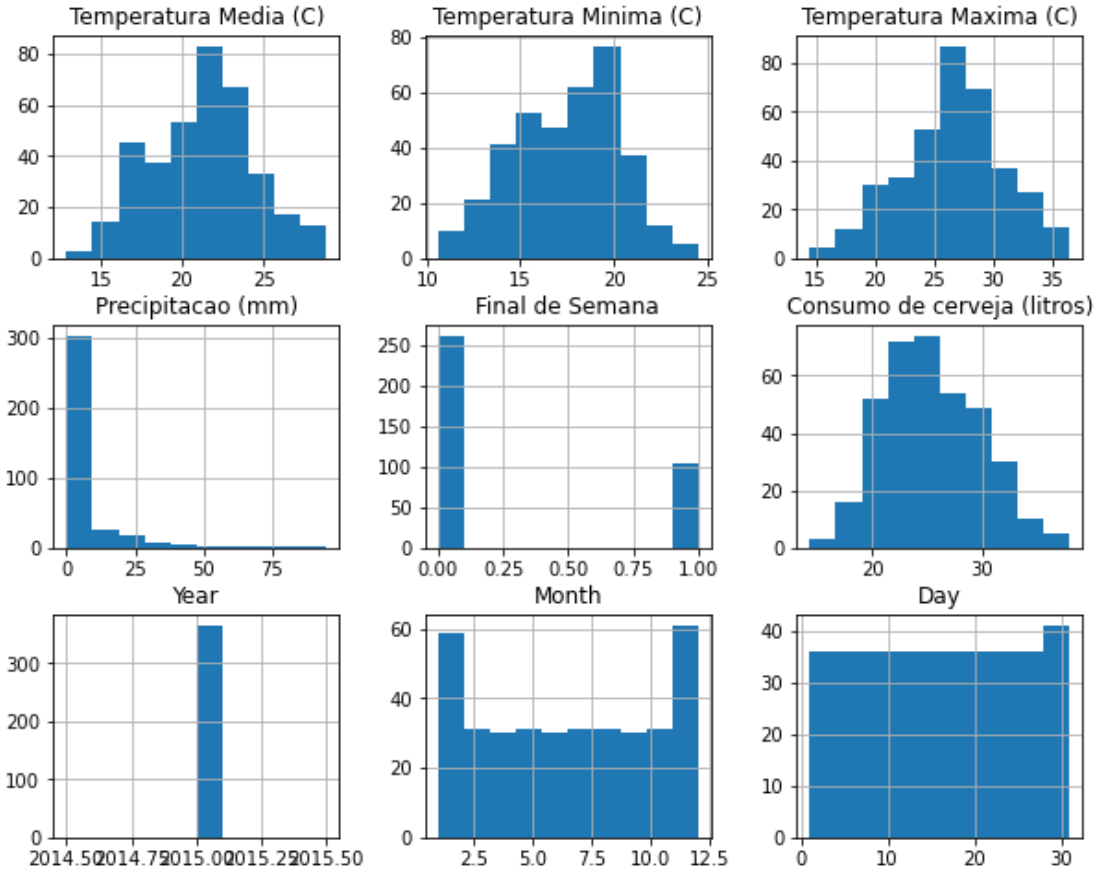


In [17]:

```
#plotting and analyzing corelation
```

In [18]:

```
corelation=df.corr(method='pearson')
corelation
plot=df.hist(figsize=(10,8))
```



In [19]:

```
#data splitting and training and testing
#splitting
X=df.drop("Consumo de cerveja (litros)",axis=1).copy()
Y=df["Consumo de cerveja (litros)"].copy()
```

In [20]:

```
X
```

Out[20]:

	Temperatura Media (C)	Temperatura Minima (C)	Temperatura Maxima (C)	Precipitacao (mm)	Final de Semana	Year	Month	Day
0	27.30	23.9	32.5	0.0	0.0	2015	1	1
1	27.02	24.5	33.5	0.0	0.0	2015	1	2
2	24.82	22.4	29.9	0.0	1.0	2015	1	3
3	23.98	21.5	28.6	1.2	1.0	2015	1	4
4	23.82	21.0	28.3	0.0	0.0	2015	1	5
...	...	...	...	...	...	...	...	...
360	24.00	21.1	28.2	13.6	1.0	2015	12	27
361	22.64	21.1	26.7	0.0	0.0	2015	12	28
362	21.68	20.3	24.1	10.3	0.0	2015	12	29
363	21.38	19.3	22.4	6.3	0.0	2015	12	30
364	24.76	20.2	29.0	0.0	0.0	2015	12	31

365 rows × 8 columns

In [21]:

```
Y
```

Out[21]:

```
0      25.461
1      28.972
2      30.814
3      29.799
4      28.900
        ...
360    32.307
361    26.095
362    22.309
363    20.467
364    22.446
Name: Consumo de cerveja (litros), Length: 365, dtype: float64
```

In [22]:

```
X_train,X_test,y_train,y_test=train_test_split(X,Y,test_size=0.33,random_state=24)
```

In [23]:

```
#feature scaling
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
```

In [26]:

```
#LR
regressor=LinearRegression()
regressor.fit(X_train,y_train)
print(regressor.intercept_)
print(regressor.coef_)
```

```
25.55133606557377
[-3.10342010e-01  1.81660610e-01  2.79429927e+00 -6.69461945e-01
  2.44604248e+00 -8.88178420e-16  2.64216219e-01 -8.87725270e-02]
```

In [ ]:

```

```

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
df=pd.read_csv('california_housing_train.csv')
df
```

Out[2]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	media
0	-114.31	34.19	15.0	5612.0	1283.0	1015.0	472.0	1.4936	
1	-114.47	34.40	19.0	7650.0	1901.0	1129.0	463.0	1.8200	
2	-114.56	33.69	17.0	720.0	174.0	333.0	117.0	1.6509	
3	-114.57	33.64	14.0	1501.0	337.0	515.0	226.0	3.1917	
4	-114.57	33.57	20.0	1454.0	326.0	624.0	262.0	1.9250	
...	...	...	...	...	...	...	...	...	
16995	-124.26	40.58	52.0	2217.0	394.0	907.0	369.0	2.3571	
16996	-124.27	40.69	36.0	2349.0	528.0	1194.0	465.0	2.5179	
16997	-124.30	41.84	17.0	2677.0	531.0	1244.0	456.0	3.0313	
16998	-124.30	41.80	19.0	2672.0	552.0	1298.0	478.0	1.9797	
16999	-124.35	40.54	52.0	1820.0	300.0	806.0	270.0	3.0147	

17000 rows × 9 columns

In [4]:

```
df.shape
```

Out[4]:

```
(17000, 9)
```

In [7]:

```
df.isnull()
```

Out[7]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	media
0	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	
16995	False	False	False	False	False	False	False	False	

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	media
16996	False	False	False	False	False	False	False	False	
16997	False	False	False	False	False	False	False	False	
16998	False	False	False	False	False	False	False	False	
16999	False	False	False	False	False	False	False	False	

17000 rows × 9 columns

In [ ]:

```
#data splitting and training and testing
#splitting
X=df.drop("Consumo de cerveja (litros)",axis=1).copy()
Y=df["Consumo de cerveja (litros)"].copy()
```

PROGRAM 3

In [24]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
```

In [17]:

```
df=pd.read_csv('insurance.csv')
df
```

Out[17]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [18]:

```
df.shape
```

Out[18]:

```
(1338, 7)
```

In [19]:

```
df.isnull().sum()
```

Out[19]:

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [20]:

```
y=df['charges']
x=df.drop('charges',axis=1)

x1=pd.get_dummies(x,drop_first=True,columns=['sex','smoker','region'])
x1
```

Out[20]:

	age	bmi	children	sex_male	smoker_yes	region_northwest	region_southeast	region_southwest
0	19	27.900	0	0	1	0	0	1
1	18	33.770	1	1	0	0	1	0
2	28	33.000	3	1	0	0	1	0
3	33	22.705	0	1	0	1	0	0
4	32	28.880	0	1	0	1	0	0
...	...	...	...	...	...	...	...	...
1333	50	30.970	3	1	0	1	0	0
1334	18	31.920	0	0	0	0	0	0
1335	18	36.850	0	0	0	0	1	0
1336	21	25.800	0	0	0	0	0	1
1337	61	29.070	0	0	1	1	0	0

1338 rows × 8 columns

In [41]:

```
X_train,X_test,y_train,y_test=train_test_split(x1,y,test_size=0.33,random_state=24)
```

In [42]:

```
#feature scaling
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
```

In [43]:

```
#LR
regressor=LinearRegression()
regressor.fit(X_train,y_train)
print(regressor.intercept_)
print(regressor.coef_)
```

```
13408.230184920762
[3787.4913775  2253.98199903  553.08164946 -138.41198423 9433.53013528
  -49.25705854 -432.02542667 -358.37822796]
```

In [ ]:

```

```

In [ ]:

```

```

PROGRAM4

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.preprocessing import StandardScaler
import warnings
warnings.filterwarnings("ignore")
```

In [3]:

```
df=pd.read_csv("50_Startups.csv")
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

	R&D Spend	Administration	Marketing Spend	State	Profit
27	[illegible]	[illegible]	[illegible]	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]:

```
df.isnull().sum()
```

Out[5]:

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

In [6]:

```
df.shape
```

Out[6]:

```
(50, 5)
```

In [25]:

```
x=df['State'].replace(['California','Florida','New York'],[0,1,2],inplace=True)
df
```

Out[25]:

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	2	192261.83
1	162597.70	151377.59	443898.53	0	191792.06
2	153441.51	101145.55	407934.54	1	191050.39

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

	R&D Spend	Administration	Marketing Spend	State	Profit
48	542.05	51743.15	0.00	2	35673.41
49	0.00	116983.80	45173.06	0	14681.40

In [29]:

```
y=df['State']
```

In [30]:

```
X_train,X_test,y_train,y_test=train_test_split(df,y,test_size=0.22,random_state=1)
```

In [31]:

```
#feature scaling
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
```

In [32]:

```
#LR
regressor=LinearRegression()
regressor.fit(X_train,y_train)
print(regressor.intercept_)
print(regressor.coef_)
```

```
0.8717948717948718
[-2.94662527e-16  3.46944695e-16  3.74700271e-16  7.90309488e-01
  5.55111512e-17]
```

In [ ]:

In [ ]:

In [ ]: