In [1]:

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

In [2]:

```
df=pd.read_csv('datasets/IceCreamData.csv')
df.head()
```

Out[2]:

	Temperature	Revenue
0	24.566884	534.799028
1	26.005191	625.190122
2	27.790554	660.632289
3	20.595335	487.706960
4	11.503498	316.240194

In [3]:

```
df.corr()
```

Out[3]:

	Temperature	Revenue
Temperature	1.000000	0.989802
Revenue	0.989802	1.000000

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Temperature  500 non-null    float64
 1   Revenue      500 non-null    float64
dtypes: float64(2)
memory usage: 7.9 KB
```

In [5]:

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

Out[5]:

```
Temperature    0
Revenue        0
dtype: int64
```

In [6]:

```
x = df['Temperature'].values.reshape(-1,1)  #Independent variable
y = df['Revenue'].values.reshape(-1,1)   #dependent variable
```

In [7]:

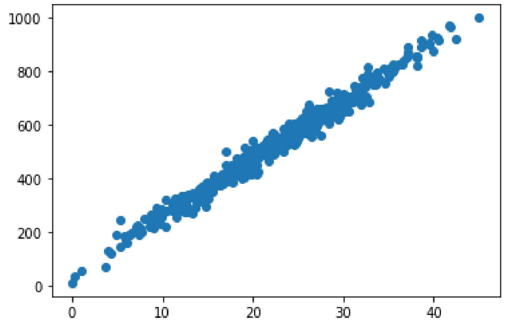
```
from matplotlib import pyplot as plt
```

In [8]:

```
plt.scatter(x,y)
```

Out[8]:

```
<matplotlib.collections.PathCollection at 0x22979b8fc70>
```



In [9]:

```
df.shape
```

Out[9]:

```
(500, 2)
```

In [10]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,random_state=0,test_size=0.1)
```

In [11]:

```
x_train.shape
```

Out[11]:

```
(450, 1)
```

```
In [12]: x_test.shape

Out[12]: (50, 1)
```

```
In [13]: x_test
```

```
        [27.72149999],
        [11.05909651],
        [12.57151377],
        [19.81463838],
        [23.98464085],
        [23.05621357],
        [12.27096675],
        [21.61064376],
        [24.5288527 ],
        [32.00436506],
        [18.43998163],
        [20.94791347],
        [24.34910395],
        [18.60275025],
        [40.47398918],
        [26.00519115],
        [23.15300185],
        [ 7.2613484 ],
        [30.22810362],
        [14.28719594]])
```

```
In [14]:  from sklearn.linear_model import LinearRegression
```

```
In [15]: lr = LinearRegression()
```

```
In [16]: lr.fit(x_train,y_train)

Out[16]: LinearRegression()
```

```
In [17]: y_pred = lr.predict(x_test)
```

In [18]:
```
prediction = pd.DataFrame(y_test,columns = ['Y'])
prediction['y_hat'] = y_pred
prediction['residuals'] = y_test - y_pred
prediction
```

Out[18]:

	Y	y_hat	residuals
0	704.281439	697.707072	6.574367
1	632.901914	652.739041	-19.837127
2	662.558990	664.134040	-1.575050
3	449.813300	450.147723	-0.334423
4	636.298374	664.877682	-28.579308
5	469.909033	441.006651	28.902383
6	587.221246	583.553776	3.667470
7	581.074005	623.271996	-42.197991
8	675.828916	666.888049	8.940867
9	493.710333	468.333683	25.376650
10	506.432135	546.354759	-39.922623
11	427.138369	443.047811	-15.909442
12	644.488633	622.399213	22.089420
13	350.629036	377.351271	-26.722234
14	366.247714	366.776707	-0.528993
15	965.493040	944.779684	20.713356
16	898.805423	892.959033	5.846390
17	648.453609	693.827041	-45.373432
18	586.138767	545.578718	40.560049
19	405.661446	420.245072	-14.583626
20	395.273750	390.775779	4.497971
21	572.537048	596.488947	-23.951899
22	288.158145	283.039720	5.118425
23	643.788331	654.913996	-11.125665
24	396.935648	380.689328	16.246321
25	412.082357	411.986607	0.095750
26	353.325633	370.762345	-17.436712
27	478.598509	509.804905	-31.206397
28	474.749392	479.300534	-4.551142
29	463.065614	456.304042	6.761573
30	654.894955	639.545333	15.349621
31	306.749930	281.457797	25.292133
32	319.349462	313.960895	5.388568
33	471.701557	469.621630	2.079927
34	559.135869	559.238431	-0.102561
35	552.819351	539.285761	13.533590
36	335.156856	307.501891	27.654965
37	537.664801	508.219271	29.445530
38	594.110352	570.933993	23.176359
39	675.807151	731.588934	-55.781783
40	463.480508	440.079120	23.401388
41	500.925065	493.976649	6.948416
42	572.672047	567.071044	5.601004
43	472.549343	443.577152	28.972190
44	918.391232	913.608158	4.783074
45	625.190122	602.661727	22.528395
46	506.493748	541.365821	-34.872073
47	223.435016	199.841051	23.593965
48	679.712058	693.415607	-13.703548
49	322.592741	350.832323	-28.239582

In [19]:
```
m = lr.coef_  #slop
```

In [20]:
```
b = lr.intercept_  #y_intercept
```

In [21]:
```
x_test
```

Out[21]:
```
array([[30.42779184],
       [28.33536277],
       [28.86558895],
       [18.90848865],
       [28.90019172],
       [18.48314099],
       [25.11606991],
       [26.96421749],
       [28.99373705],
       [19.75470829],
       [23.38514451],
       [18.57811922],
       [26.9236056 ],
       [15.52116187],
       [15.02911176],
       [41.92444647],
       [39.5131548 ],
       [30.24724825],
       [23.34903419],
       [17.51707397],
       [16.14582413],
       [25.71796257],
       [11.13270573],
       [28.43656665],
       [15.67648661],
       [17.13279538],
       [15.21456942],
       [21.68442569],
       [20.26501213],
       [19.19495126],
       [27.72143999],
       [11.05909651],
       [12.57151377],
       [19.81463838],
       [23.98464085],
       [23.05621357],
       [12.27096675],
       [21.61064376],
       [24.5288527 ],
       [32.00436506],
       [18.43998163],
       [20.94791347],
       [24.34910395],
       [18.60275025],
       [40.47398918],
       [26.00519115],
       [23.15300185],
       [ 7.2613484 ],
       [30.22810362],
       [14.28719594]])
```

In [22]:
```
y_1=m*30.42779184 + b
y_1
```

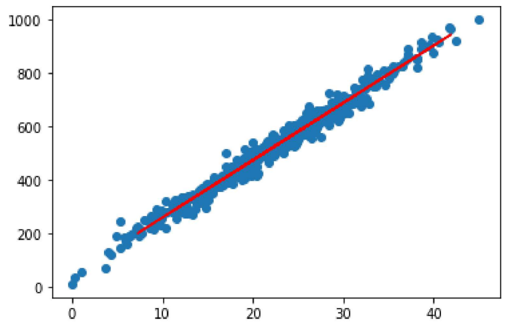
Out[22]:
```
array([[697.70707182]])
```

In [23]:
```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test,y_pred)
mse
```

Out[23]:
```
510.36278285590174
```

```
In [24]: plt.scatter(x,y)
         plt.plot(x_test,y_pred,color = 'red')
```

```
Out[24]: [<matplotlib.lines.Line2D at 0x2297c650f40>]
```



```
In [ ]:
```

```
In [ ]:
```