

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

df=pd.read_csv('/content/Ice Cream.csv')
df
```

	Temperature	Revenue	
0	24.6	535	
1	26.1	626	
2	27.8	661	
3	20.6	488	
4	11.6	317	
...	
495	22.3	525	
496	32.9	756	
497	12.6	307	
498	22.4	567	
499	29.0	656	

500 rows × 2 columns

Next steps:

Generate code with df

☒ View recommended plots

```
df.head()
```

	Temperature	Revenue	
0	24.6	535	
1	26.1	626	
2	27.8	661	
3	20.6	488	
4	11.6	317	

Next steps:

Generate code with df

☒ View recommended plots

```
df.tail()
```

	Temperature	Revenue	
495	22.3	525	
496	32.9	756	
497	12.6	307	
498	22.4	567	
499	29.0	656	

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

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

```
df.dtypes
```

```
Temperature    float64
Revenue        int64
dtype: object
```

```
x=df.iloc[0,:-1]
x
```

	Temperature	
0	24.6	
1	26.1	
2	27.8	
3	20.6	
4	11.6	
...	...	
495	22.3	
496	32.9	
497	12.6	
498	22.4	
499	29.0	

500 rows × 1 columns

Next steps:

[Generate code with x](#)[View recommended plots](#)

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

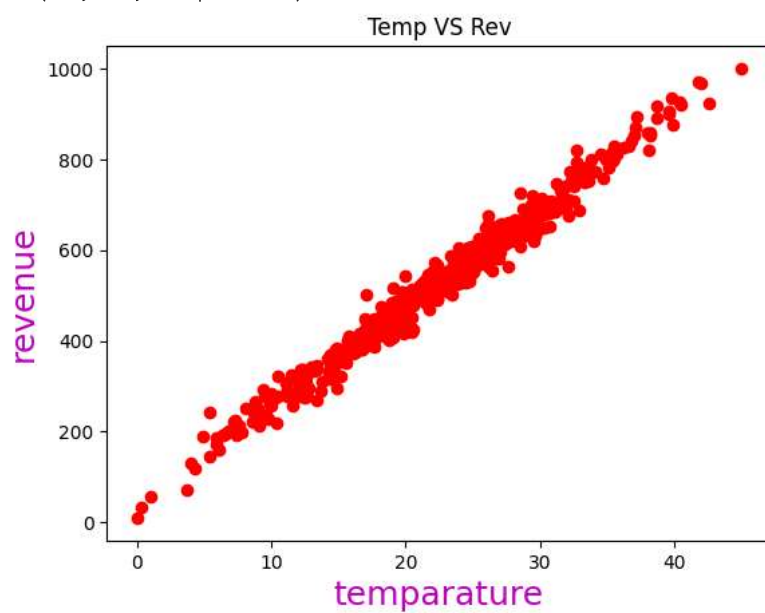
```
y
```

```
0      535
1      626
2      661
3      488
4      317
...
495    525
496    756
497    307
498    567
499    656
```




```
Name: Revenue, Length: 500, dtype: int64
```

```
plt.scatter(x,y,color='r')
plt.xlabel("temparature",fontsize=18,color='m')
plt.ylabel("revenue",fontsize=18,color='m')
plt.title('Temp VS Rev')
```

```
Text(0.5, 1.0, 'Temp VS Rev')
```






```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
x_train
```

Temperature		
5	14.4	 
116	26.0	
45	15.7	
16	26.0	
462	12.2	
...	...	
106	23.8	
270	24.7	
348	38.2	
435	16.0	
102	14.3	

350 rows × 1 columns

Next steps: [Generate code with x_train](#) [View recommended plots](#)

x_test

Temperature		
361	29.8	 
73	27.4	
374	24.0	
155	21.2	
104	26.6	
...	...	
266	18.6	
23	16.3	
222	25.9	
261	23.6	
426	26.2	

150 rows × 1 columns

Next steps: [Generate code with x_test](#) [View recommended plots](#)

y_train

5	368
116	573
45	397
16	649
462	298
...	...
106	554
270	604
348	857
435	382
102	323

Name: Revenue, Length: 350, dtype: int64

y_test

361	660
73	624
374	560
155	498
104	628
...	...
266	421
23	375
222	597
261	571
426	618

Name: Revenue, Length: 150, dtype: int64

```

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

```

```

array([[683.42932418, 632.0173136, 559.18363196, 499.20295296,
        614.87997675, 304.2657462, 284.98624224, 314.97658174,
        505.62945428, 685.57149128, 724.13049921, 400.66326603,
        430.65360553, 672.71848864, 51.49002755, 199.29955795,
        522.76679114, 563.46796617, 584.88963725, 683.42932418,
        557.04146485, 784.11117821, 655.58115178, 569.89446749,
        912.64120465, 666.29198732, 314.97658174, 632.0173136,
        732.69916764, 698.42449393, 434.93793974, 471.35478056,
        711.27749657, 801.24851507, 640.58598203, 492.77645164,
        745.55217029, 317.11874885, 261.42240406, 130.75021052,
        582.74747014, 229.28989745, 548.47279642, 392.0945976,
        437.08010685, 593.45830567, 572.0366346, 124.3237092,
        475.63911478, 531.33545957, 623.44864517, 503.48728717,
        484.20778321, 499.20295296, 550.61496353, 632.0173136,
        574.17880171, 679.14498996, 171.45138556, 662.0076531,
        584.88963725, 704.85099525, 535.61979378, 629.8751465,
        677.00282285, 411.37410156, 509.91378849, 494.91861874,
        732.69916764, 312.83441463, 319.26091595, 454.21744371,
        796.96418086, 424.22710421, 520.62462403, 689.8558255,
        805.53284929, 447.79094238, 494.91861874, 492.77645164,
        546.33062932, 520.62462403, 664.14982021, 548.47279642,
        589.17397146, 698.42449393, 237.85856588, 627.73297939,
        584.88963725, 769.11600846, 428.51143842, 522.76679114,
        610.59564253, 353.53558967, 563.46796617, 580.60530303,
        640.58598203, 662.0076531, 413.51626867, 441.36444106,
        434.93793974, 233.57423166, 512.0559556, 441.36444106,
        385.66809628, 758.40517293, 621.30647807, 137.17671184,
        672.71848864, 462.78611213, 657.72331889, 805.53284929,
        578.46313592, 452.0752766, 419.94276999, 548.47279642,
        576.32096882, 687.71365839, 674.86065575, 447.79094238,
        578.46313592, 503.48728717, 370.67292653, 469.21261346,
        477.78128189, 623.44864517, 636.30164782, 269.99107249,
        387.81026338, 944.77371125, 372.81509363, 724.13049921,
        246.42723431, 655.58115178, 833.38102168, 829.09668747,
        494.91861874, 557.04146485, 957.6267139, 252.85373563,
        471.35478056, 559.18363196, 443.50660817, 310.69224752,
        719.846165, 443.50660817, 394.2367647, 599.884807,
        550.61496353, 606.31130832])

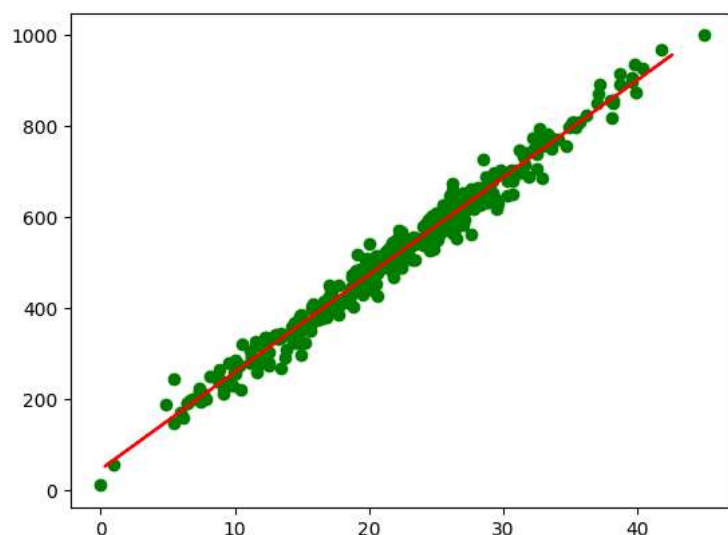
```

```

plt.scatter(x_train,y_train,color='g')
plt.plot(x_test,y_pred,color='r')

```

[<matplotlib.lines.Line2D at 0x7a3cb82b28f0>]



```
print(lr.predict([[18.5]]))
```

[441.36444106]

/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegre: warnings.warn()

```

data = {'Actualvalue':y_test,
        'predictedvalue':y_pred,
        'difference':y_test-y_pred}
df1=pd.DataFrame(data)
df1

```

	Actualvalue	predictedvalue	difference
361	660	683.429324	-23.429324
73	624	632.017314	-8.017314
374	560	559.183632	0.816368
155	498	499.202953	-1.202953
104	628	614.879977	13.120023
...
266	421	443.506608	-22.506608
23	375	394.236765	-19.236765
222	597	599.884807	-2.884807
261	571	550.614964	20.385036
426	618	606.311308	11.688692

150 rows × 3 columns

Next steps:

[Generate code with df1](#)[View recommended plots](#)

```
print("Slope is",lr.coef_)
```

```
Slope is [21.42167107]
```

```
print("Constant is",lr.intercept_)
```

```
Constant is 45.06352623099866
```

```
from sklearn.metrics import mean_absolute_percentage_error,r2_score
print(" Percentage Error is",mean_absolute_percentage_error(y_test,y_pred))
print('\n')
print("r2 score is: ",r2_score(y_test,y_pred))
```

```
Percentage Error is 0.04669937595245471
```

```
r2 score is: 0.9799112633959458
```