PRACTICAL - 5

AIM: Write a program to implement Linear Regression.

In [2]:
```
import pandas as pd
df = pd.read_csv("sum_two_numbers.csv")
```

In [4]:
```
df.head()
```

Out[4]:

	x	y	z
0	4.0	8.0	12.0
1	5.0	2.0	7.0
2	9.0	44.0	53.0
3	123.0	12.0	135.0
4	10.0	1.0	11.0

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 59 entries, 0 to 58
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   x       59 non-null     float64
 1   y       59 non-null     float64
 2   z       59 non-null     float64
dtypes: float64(3)
memory usage: 1.5 KB
```

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

Out[6]:

	x	y	z
count	59.000000	59.000000	59.000000
mean	33.490508	36.119153	69.609831
std	22.777758	22.344767	40.820837
min	4.000000	1.000000	6.850000
25%	13.780000	17.250000	41.020000
50%	34.730000	38.440000	73.800000
75%	43.750000	48.905000	94.050000
max	125.300000	123.050000	248.350000

In [7]:
```
X = df[['x','y']]
```

In [12]:
```
X.head()
```

Out[12]:

	x	y
0	4.0	8.0
1	5.0	2.0
2	9.0	44.0
3	123.0	12.0
4	10.0	1.0

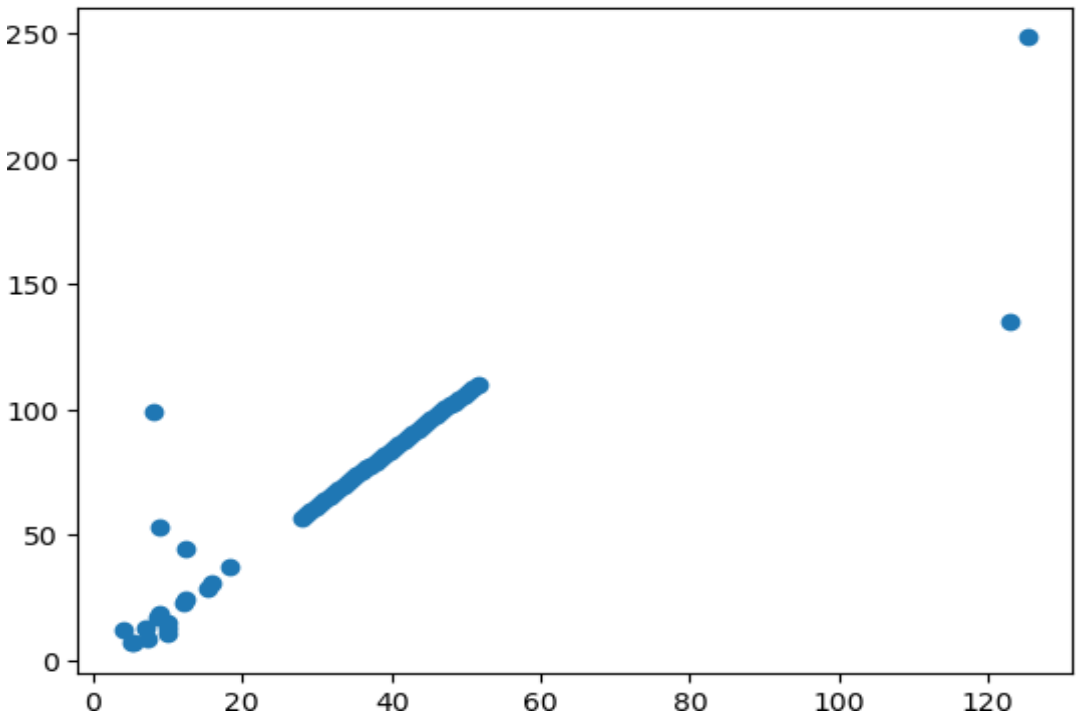
In [9]:
```
y = df['z']
```

In [11]:
```
y.head()
```

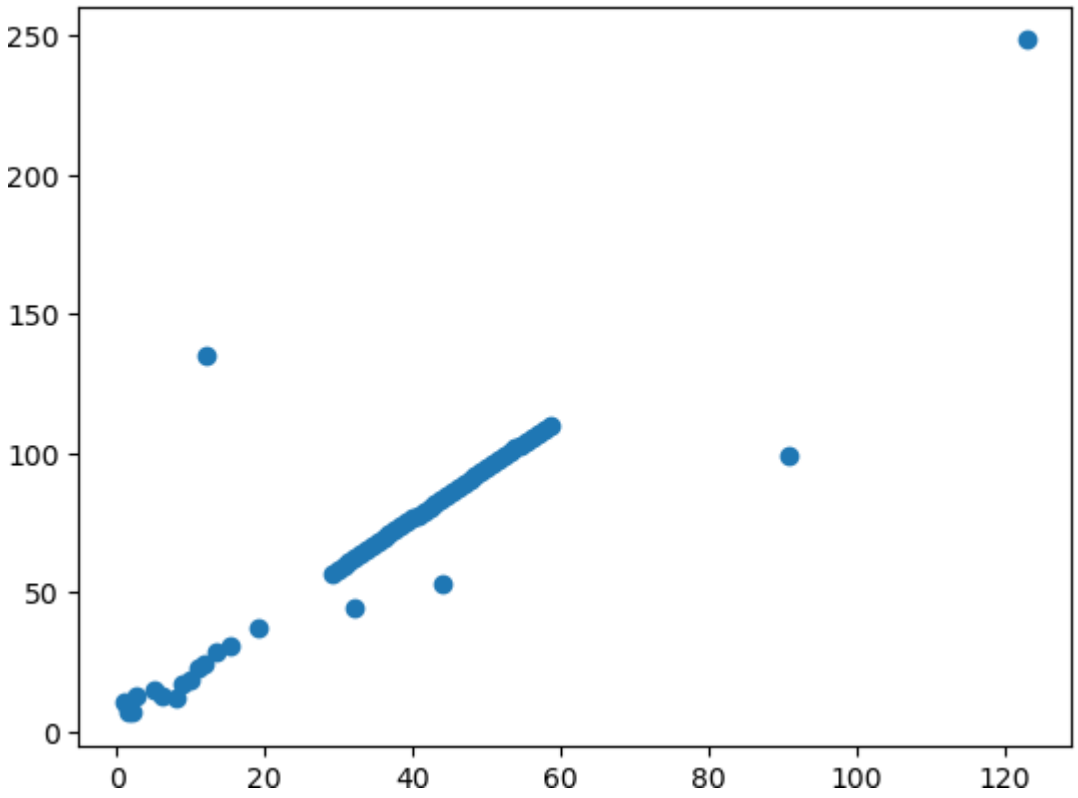
Out[11]:
```
0     12.0
1      7.0
2     53.0
3    135.0
4     11.0
Name: z, dtype: float64
```

In [27]:
```
import matplotlib.pyplot as plt

plt.scatter(df['x'],df['z'])
plt.show()
```



In [28]:
```
plt.scatter(df['y'],df['z'])
plt.show()
```



In [13]:
```
from sklearn.linear_model import LinearRegression
```

In [14]:
```
lr = LinearRegression()
```

In [15]:
```
lr.fit(X,y)
```

Out[15]:
```
LinearRegression
LinearRegression()
```

In [17]:
```
lr.predict([[1,1]])[0]
```

```
C:\Users\GCET\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[17]:
```
2.0003506478363855
```

In [18]:
```
from sklearn.model_selection import train_test_split
```

In [19]:
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.30,random_state
```

In [21]:
```
y_pred = lr.predict(X_test)
```

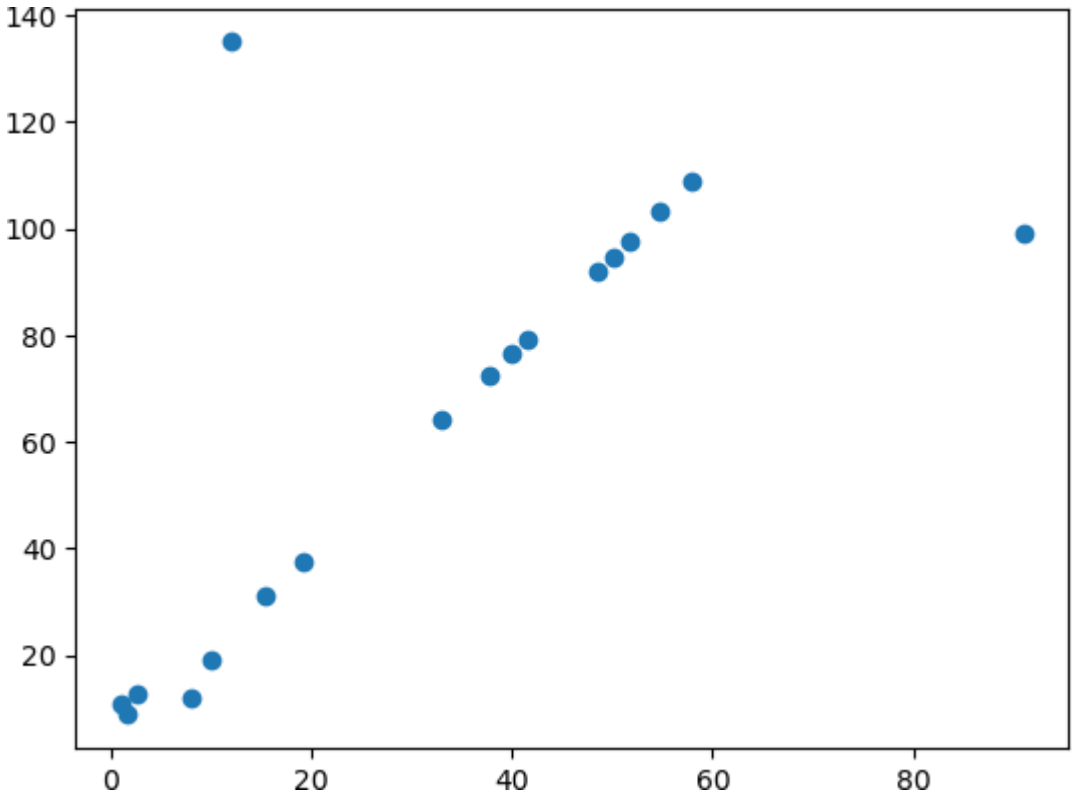
In [22]:
```
accuracy = pd.DataFrame({'Actual' : y_test , 'Predicted' : y_pred})
accuracy
```

Out[22]:

	Actual	Predicted
0	12.00	12.000319
5	19.00	19.000305
34	76.59	76.590150
13	31.20	31.200275
45	91.96	91.960107
53	103.13	103.130076
57	108.72	108.720061
25	64.02	64.020185
47	94.75	94.750100
12	12.60	12.600333
49	97.54	97.550092
3	135.00	135.000160
36	79.39	79.380142
31	72.40	72.400162
8	8.80	8.800341
17	37.50	37.500256
6	99.00	98.999978
4	11.00	11.000340

In [31]:
```
plt.scatter(X_test['x'],y_pred)
plt.show()
```

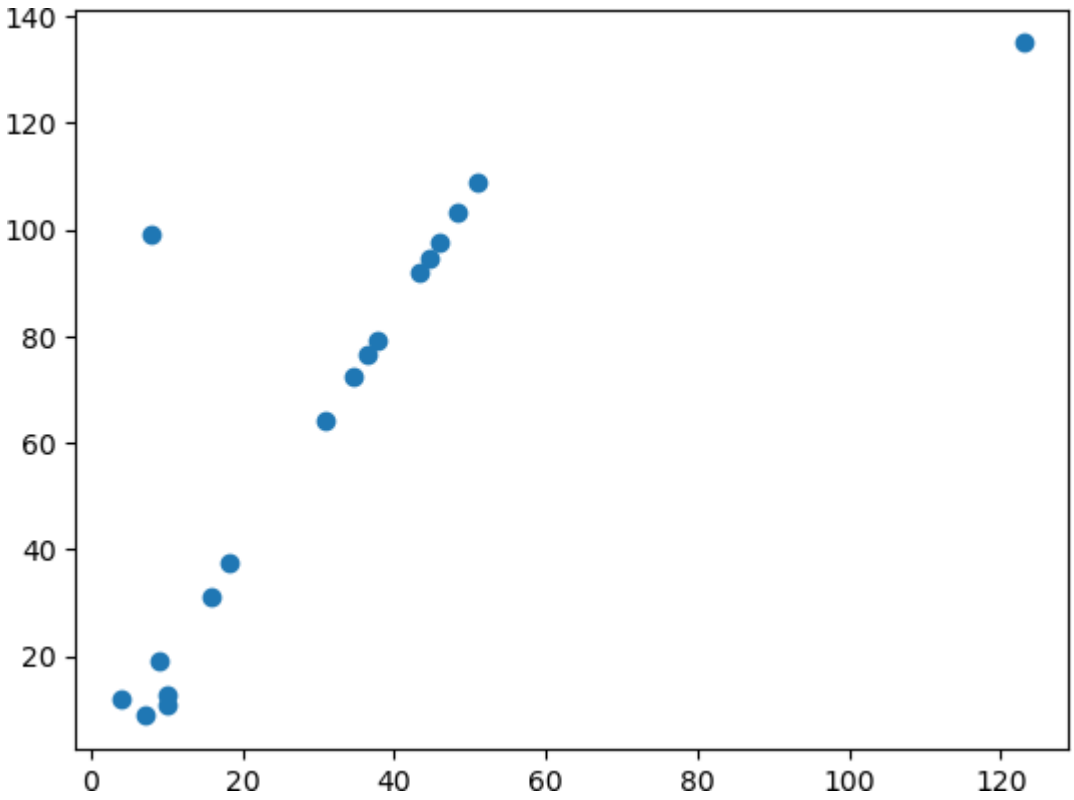
In [32]:
```
plt.scatter(X_test['y'],y_pred)
plt.show()
```



In [33]:
```
lr.score(X_train,y_train)
```

Out[33]: 0.9999999897014692

In [34]:
```
lr.score(X_test,y_test)
```



Out[34]:
0.9999999
928934155

In [51]:
```
import joblib
joblib.dump(lr,"Sum_Predictor")
```

Out[51]:
['Sum_Pr
edictor'
]