# SUMESH R -20104169

In [1]:

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

In [2]:

```
df = pd.read_csv("3_Fitness-1.csv")
# .dropna(axis="columns")
df
```

Out[2]:

| | Row Labels | Sum of Jan | Sum of Feb | Sum of Mar | Sum of Total Sales |
|---|---|---|---|---|---|
| **0** | A | 5.62% | 7.73% | 6.16% | 75 |
| **1** | B | 4.21% | 17.27% | 19.21% | 160 |
| **2** | C | 9.83% | 11.60% | 5.17% | 101 |
| **3** | D | 2.81% | 21.91% | 7.88% | 127 |
| **4** | E | 25.28% | 10.57% | 11.82% | 179 |
| **5** | F | 8.15% | 16.24% | 18.47% | 167 |
| **6** | G | 18.54% | 8.76% | 17.49% | 171 |
| **7** | H | 25.56% | 5.93% | 13.79% | 170 |
| **8** | Grand Total | 100.00% | 100.00% | 100.00% | 1150 |

In [3]:

```
df.head()
```

Out[3]:

| | Row Labels | Sum of Jan | Sum of Feb | Sum of Mar | Sum of Total Sales |
|---|---|---|---|---|---|
| **0** | A | 5.62% | 7.73% | 6.16% | 75 |
| **1** | B | 4.21% | 17.27% | 19.21% | 160 |
| **2** | C | 9.83% | 11.60% | 5.17% | 101 |
| **3** | D | 2.81% | 21.91% | 7.88% | 127 |
| **4** | E | 25.28% | 10.57% | 11.82% | 179 |

# Data cleaning and pre processing

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Row Labels          9 non-null      object
 1   Sum of Jan          9 non-null      object
 2   Sum of Feb          9 non-null      object
 3   Sum of Mar          9 non-null      object
 4   Sum of Total Sales  9 non-null      int64 
dtypes: int64(1), object(4)
memory usage: 488.0+ bytes
```

In [5]:

```
df.describe()
```

Out[5]:

| | Sum of Total Sales |
|---|---|
| **count** | 9.000000 |
| **mean** | 255.555556 |
| **std** | 337.332963 |
| **min** | 75.000000 |
| **25%** | 127.000000 |
| **50%** | 167.000000 |
| **75%** | 171.000000 |
| **max** | 1150.000000 |

In [6]:

```
df.columns
```

Out[6]:

```
Index(['Row Labels', 'Sum of Jan', 'Sum of Feb', 'Sum of Mar',
       'Sum of Total Sales'],
      dtype='object')
```

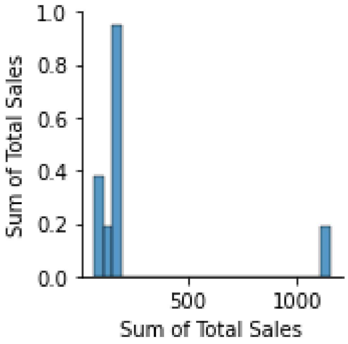
# EDA and VISUALIZATION

In [7]:

```
sns.pairplot(df)
```

Out[7]:

```
<seaborn.axisgrid.PairGrid at 0x274644ac760>
```



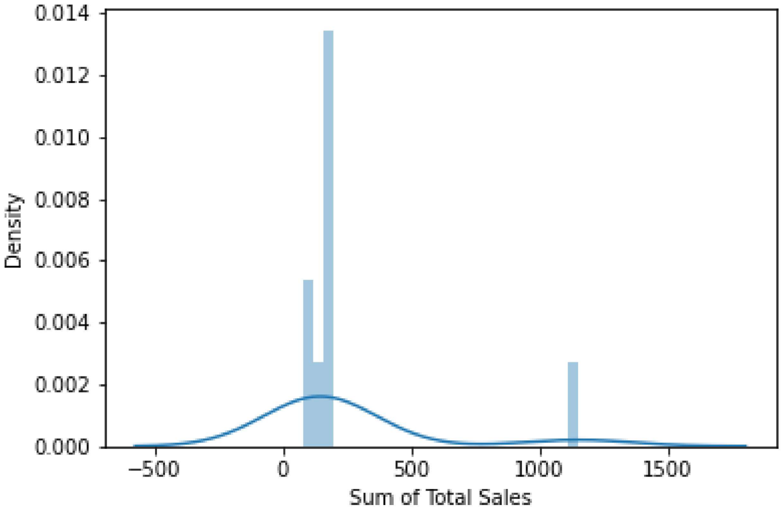
In [8]:

```
sns.distplot(df["Sum of Total Sales"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[8]:

```
<AxesSubplot:xlabel='Sum of Total Sales', ylabel='Density'>
```



In [9]:

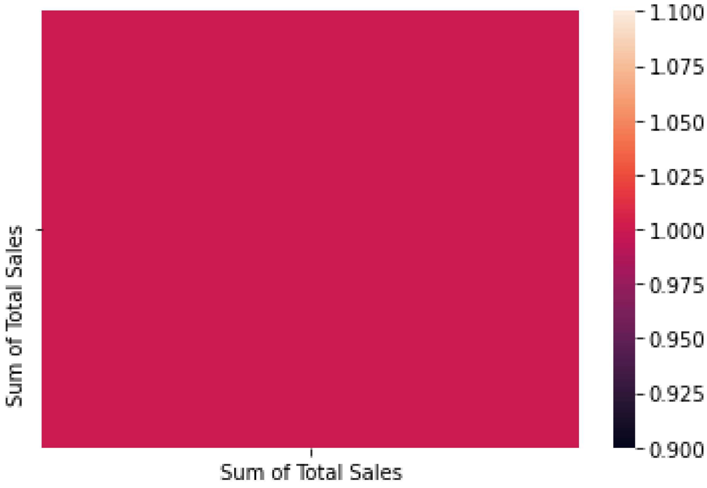
```python
df1 = df[['Row Labels', 'Sum of Jan', 'Sum of Feb', 'Sum of Mar',
       'Sum of Total Sales']]
```

In [10]:

```python
sns.heatmap(df1.corr())
```

Out[10]:

```
<AxesSubplot:>
```



In [11]:

```python
x = df1[['Sum of Total Sales','Sum of Total Sales']]
y = df1['Sum of Total Sales']
```

## split the data into training and test data

In [12]:

```python
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3)
```

In [13]:

```
lr = LinearRegression()
lr.fit(x_train, y_train)
```

Out[13]:

```
LinearRegression()
```

In [14]:

```
lr.intercept_
```

Out[14]:

```
1.1368683772161603e-13
```

In [15]:

```
coeff = pd.DataFrame(lr.coef_, x.columns, columns =['Co-efficient'])
coeff
```

Out[15]:

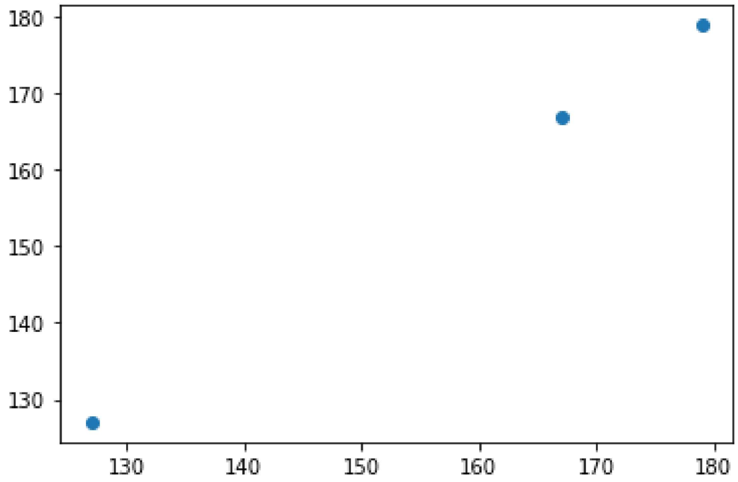
| | Co-efficient |
|---|---|
| **Sum of Total Sales** | 0.5 |
| **Sum of Total Sales** | 0.5 |

In [16]:

```
prediction = lr.predict(x_test)
plt.scatter(y_test, prediction)
```

Out[16]:

```
<matplotlib.collections.PathCollection at 0x27464f061c0>
```



In [17]:

```
lr.score(x_test,y_test)
```

Out[17]:

```
1.0
```