

In [1]:

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

In [20]:

```
df=pd.read_csv(r'C:\Users\user\Downloads\3_Fitness-1.csv')
df
```

Out[20]:

	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 [21]:

```
df.head(10)
```

Out[21]:

	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 [22]:

```
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 [23]:

```
df.describe()
```

Out[23]:

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 [24]:

```
df.columns
```

Out[24]:

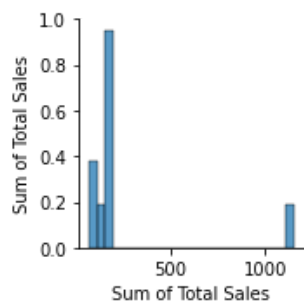
```
Index(['Row Labels', 'Sum of Jan', 'Sum of Feb', 'Sum of Mar',  
      'Sum of Total Sales'],  
      dtype='object')
```

In [25]:

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

Out[25]:

<seaborn.axisgrid.PairGrid at 0x1ed690d7dc0>



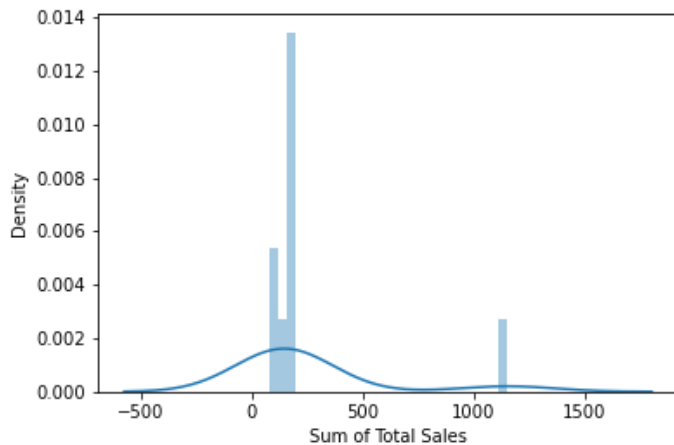
In [27]:

```
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 adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
 warnings.warn(msg, FutureWarning)

Out[27]:

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

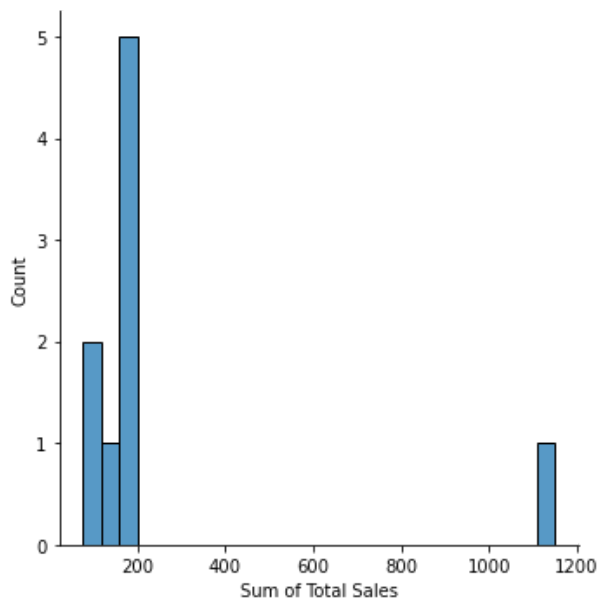


In [28]:

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

Out[28]:

```
<seaborn.axisgrid.FacetGrid at 0x1ed69015e20>
```



In [29]:

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

In [30]:

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

Out[30]:

<AxesSubplot:>



In [31]:

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

In [32]:

```
from sklearn.model_selection import train_test_split
```

In [33]:

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

In [34]:

```
from sklearn.linear_model import LinearRegression
```

```
lr=LinearRegression()
lr.fit(x_train,y_train)#ValueError: Input contains NaN, infinity or a value too large for dtype('float64')
```

Out[34]:

```
LinearRegression()
```

In [35]:

```
print(lr.intercept_)
```

```
[5.68434189e-14]
```

In [36]:

```
coef= pd.DataFrame(lr.coef_)
coef
```

Out[36]:

```
0
0 1.0
```

In [37]:

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

```
1.0
```

In [38]:

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

Out[38]:

<matplotlib.collections.PathCollection at 0x1ed6adb3af0>

