In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [107]:
```
a=pd.read_csv(r"C:\Users\user\Downloads\3_Fitness-1 (1).csv")
a
```

Out[107]:

	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 [108]:
```
a=a.head(50)
a
```

Out[108]:

	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 [109]:
```
a.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 [110]:
```
a.columns
```

Out[110]:
```
Index(['Row Labels', 'Sum of Jan', 'Sum of Feb', 'Sum of Mar',
       'Sum of Total Sales'],
      dtype='object')
```

In [111]:
```
a.head()
```

Out[111]:

	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

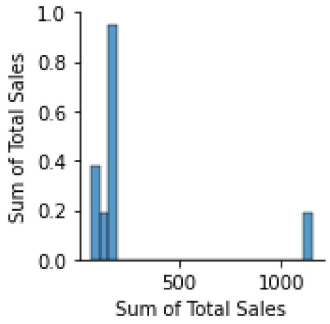
In [112]:
```
a.describe()
```

Out[112]:

	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 [149]:
```
sns.pairplot(a)
```

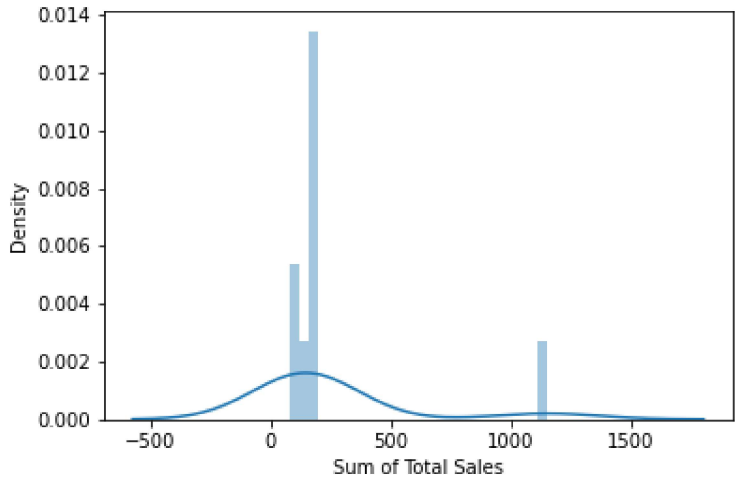
Out[149]:
```
<seaborn.axisgrid.PairGrid at 0x22eb365b3d0>
```



In [150]:
```
sns.distplot(a['Sum of Total Sales'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarnin
g: `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 flexibil
ity) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[150]:
```
<AxesSubplot:xlabel='Sum of Total Sales', ylabel='Density'>
```



In [153]:
```
x1=a[['Sum of Total Sales']]
```

In [154]:
```
sns.heatmap(x1.corr())
```

Out[154]: <AxesSubplot:>

In [155]:
```
x=a[['Sum of Total Sales']]
y=a['Sum of Total Sales']
```

In [156]:
```
from sklearn.model_selection import train_test_split

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

In [157]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[157]: LinearRegression()

In [158]:
```
print(lr.intercept_)
```

```
0.0
```

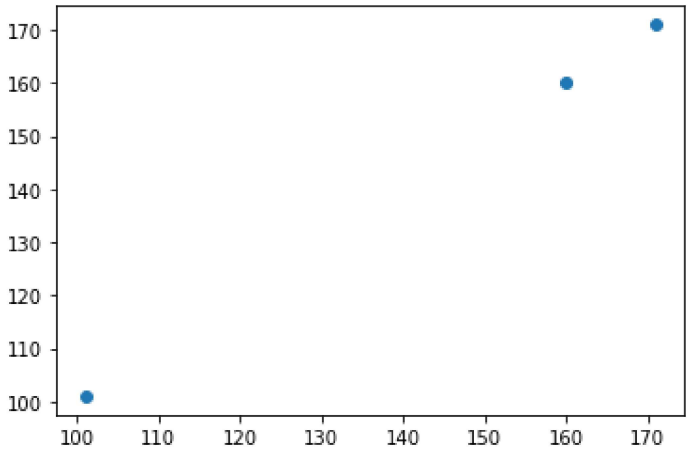
In [159]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[159]:

	Co-efficient
Sum of Total Sales	1.0

In [160]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[160]:
```
<matplotlib.collections.PathCollection at 0x22eb3ba37f0>
```



In [161]:
```
print(lr.score(x_test,y_test))
```

```
1.0
```

In [162]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [163]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[163]:
```
Ridge(alpha=10)
```

In [164]:
```
rr.score(x_test,y_test)
```

Out[164]:
```
0.9999999957742527
```

In [165]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[165]:
```
Lasso(alpha=10)
```

In [166]:
```
la.score(x_test,y_test)
```

Out[166]:
```
0.9999998478695237
```

In [167]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[167]:
```
ElasticNet()
```

In [168]:
```
print(en.coef_)
```

```
[0.99999295]
```

In [169]:
```
print(en.intercept_)
```

```
0.00219373361005637
```

In [170]:
```
print(en.predict(x_test))
```

```
[101.00148206 160.00106633 171.00098882]
```

In [171]:
```
en.score(x_test,y_test)
```

Out[171]:
```
0.9999999984787059
```

In [172]:
```
from sklearn import metrics
```

In [173]:
```
print("Mean Absolute Error",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error 0.0
```

In [174]:
```
print("Mean Squared Error",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error 0.0
```

In [175]:
```
print(" Root Mean Squared Error",np.sqrt(metrics.mean_squared_error(y_test,prediction))
```

```
 Root Mean Squared Error 0.0
```