In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
df=pd.read_csv("3_Fitness-1.csv")
df
```

Out[4]:

	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 [5]:
```
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
```

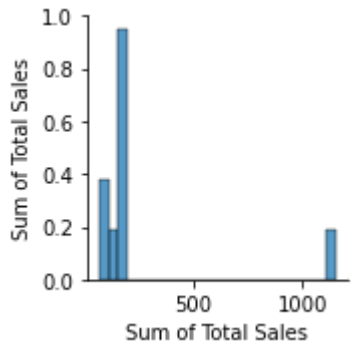
In [6]:
```
df.describe()
```

Out[6]:

	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

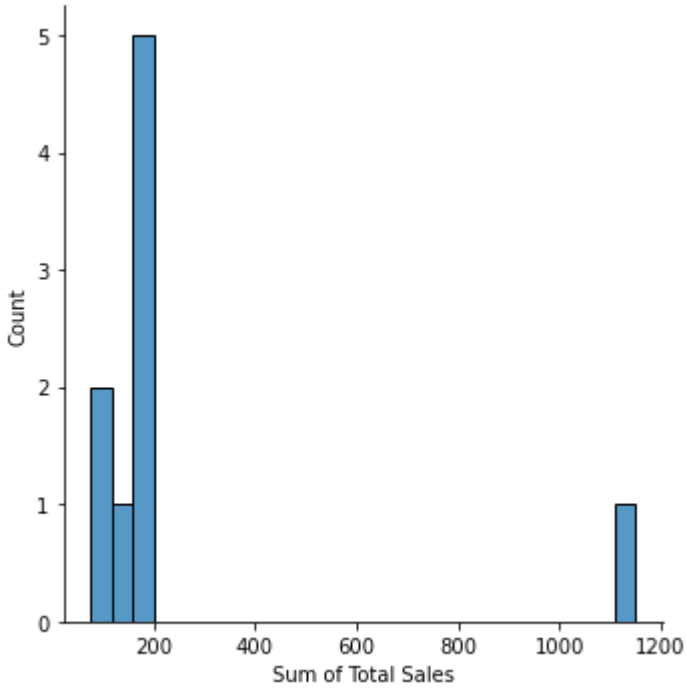
In [7]:
```
sns.pairplot(df)
```

Out[7]:
```
<seaborn.axisgrid.PairGrid at 0x23f386d79d0>
```



In [8]:
```
sns.displot(df['Sum of Total Sales'])
```

Out[8]:
```
<seaborn.axisgrid.FacetGrid at 0x23f3888db80>
```



In [9]:
```
df1=df.drop(['Row Labels'],axis=1)
df1
```

Out[9]:

	Sum of Jan	Sum of Feb	Sum of Mar	Sum of Total Sales
0	5.62%	7.73%	6.16%	75
1	4.21%	17.27%	19.21%	160
2	9.83%	11.60%	5.17%	101
3	2.81%	21.91%	7.88%	127
4	25.28%	10.57%	11.82%	179
5	8.15%	16.24%	18.47%	167
6	18.54%	8.76%	17.49%	171
7	25.56%	5.93%	13.79%	170
8	100.00%	100.00%	100.00%	1150

In [10]:
```
sns.heatmap(df1.corr())
```

Out[10]: <AxesSubplot:>

In [11]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [12]:
```
y=df['Sum of Total Sales']
x=df1.drop(['Sum of Jan','Sum of Feb','Sum of Mar'],axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
print(x_train)
```

```
   Sum of Total Sales
7                 170
4                 179
8                1150
2                 101
0                  75
5                 167
```

In [13]:
```
model=LinearRegression()
model.fit(x_train,y_train)
model.intercept_
```

Out[13]: 1.7053025658242404e-13

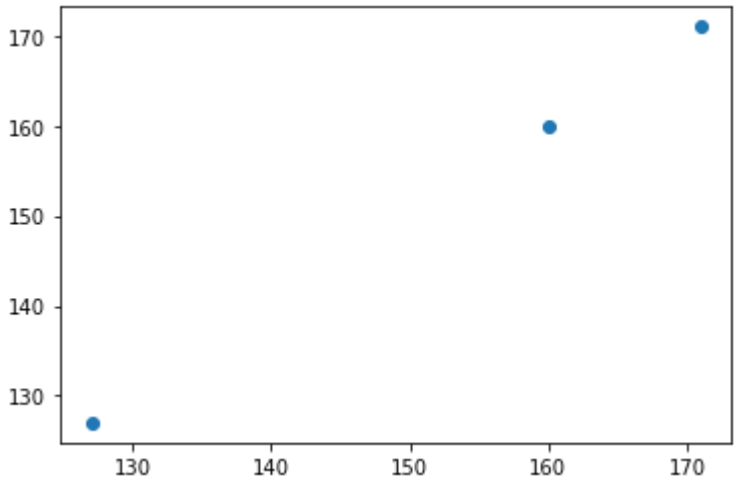
In [14]:
```
coeff=pd.DataFrame(model.coef_,x.columns,columns=["Coefficient"])
coeff
```

Out[14]:

	Coefficient
Sum of Total Sales	1.0

In [15]:
```
prediction=model.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[15]: <matplotlib.collections.PathCollection at 0x23f3a8f0fa0>



In [16]:
```
model.score(x_test,y_test)
```

Out[16]: 1.0

In [17]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [18]:
```
rr = Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[18]: Ridge(alpha=10)

In [19]:
```
rr.score(x_test,y_test)
```

Out[19]: 0.9999999906879236

In [20]:
```
la = Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[20]: Lasso(alpha=10)

In [21]:
```
la.score(x_test,y_test)
```

Out[21]: 0.9999996647574675

In [22]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
print(en.coef_)
print(en.intercept_)
print(en.predict(x_test))
print(en.score(x_test,y_test))
from sklearn import metrics
print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,prediction))
print("Mean Squared Error:",metrics.mean_squared_error(y_test,prediction))
print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test,pred
```

```
[0.99999304]
0.002137721736403364
[160.0010236  171.000947   127.00125339]
0.999999996647598
Mean Absolute Error: 9.000207986294602e-14
Mean Squared Error: 8.145251800042021e-27
Root Mean Squared Error: 9.025104874760193e-14
```

In [ ]: