SUMESH R - 20104169

Basic Analysis using NumPy and Pandas

Import Libraries

In [1]:
```
import pandas as pd
```

In [2]:
```
import numpy as np
```

Import Dataset

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

In [4]:
```
display(data)
```

	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

To display top 10 rows

In [5]:
```
data.head(10)
```

Out[5]:

	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

	Row Labels	Sum of Jan	Sum of Feb	Sum of Mar	Sum of Total Sales
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

to display last 5 rows

In [6]:
```
data.tail()
```

Out[6]:

	Row Labels	Sum of Jan	Sum of Feb	Sum of Mar	Sum of Total Sales
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

statistical summary

In [7]:
```
data.describe()
```

Out[7]:

	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

To print number of elements

In [8]:
```
data.size
```

Out[8]: 45

to print number of row and cols

In [9]:
```
data.shape
```

Out[9]: (9, 5)

to find missing values

In [10]:
```
data.isna()
```

Out[10]:

	Row Labels	Sum of Jan	Sum of Feb	Sum of Mar	Sum of Total Sales
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	False	False
6	False	False	False	False	False
7	False	False	False	False	False
8	False	False	False	False	False

fill null values with a constant

In [11]:
```
data.fillna(5)
```

Out[11]:

	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

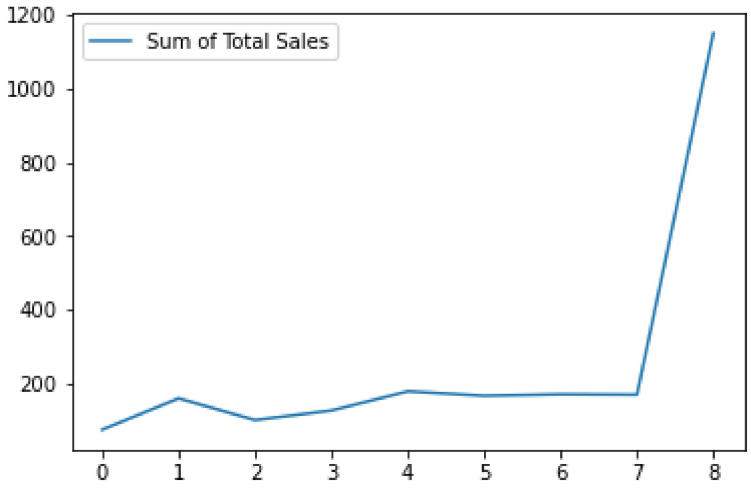
to select a particular columns

In [12]:
```
df=pd.DataFrame(data[['Sum of Jan','Sum of Total Sales']])
import matplotlib.pyplot as plt
```

line plot

In [13]:
```
df.plot.line()
```

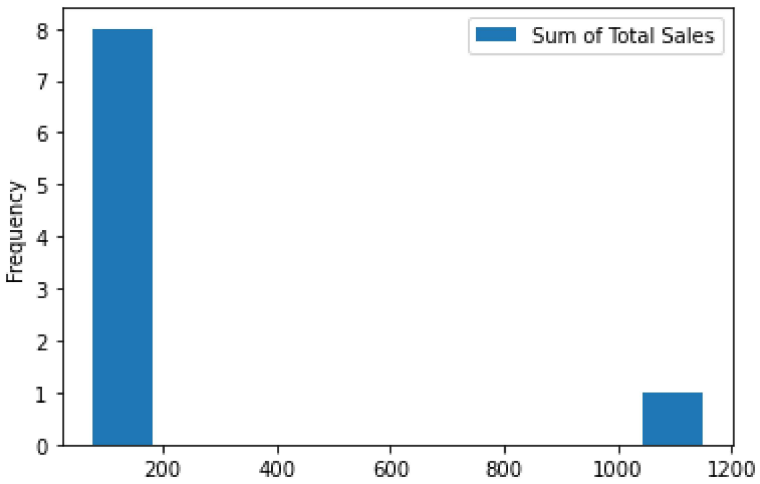
Out[13]: <AxesSubplot:>



histogram

In [14]:
```
df.plot.hist()
```

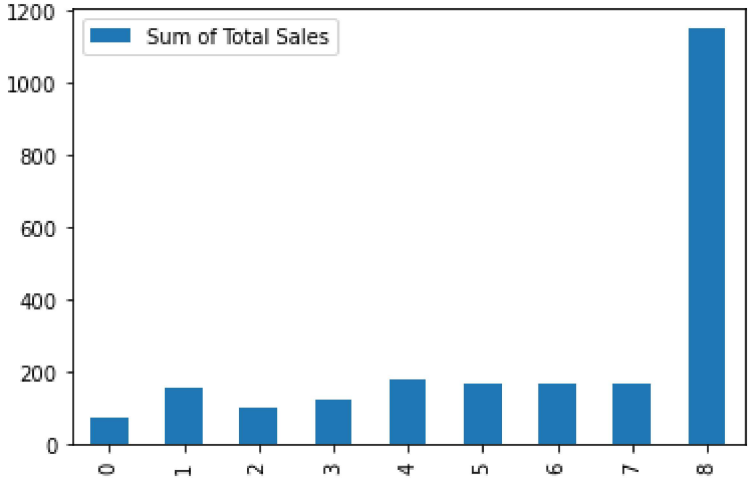
Out[14]: <AxesSubplot:ylabel='Frequency'>



bar chart

In [15]:
```
df.plot.bar()
```

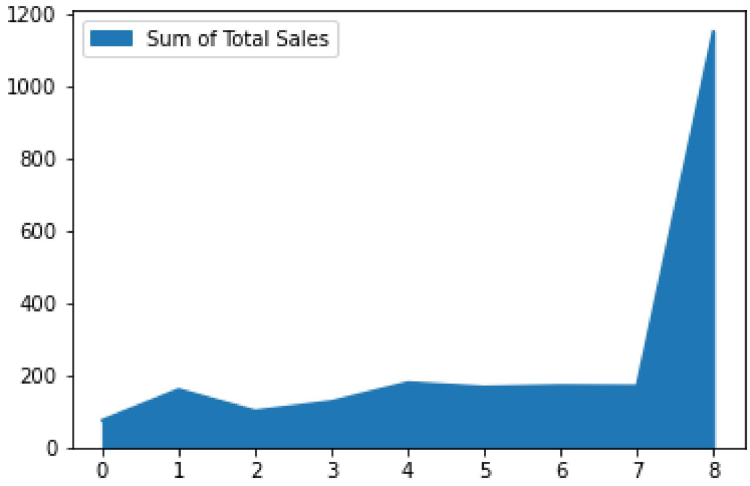
Out[15]: <AxesSubplot:>



area plot

In [16]:
```
df.plot.area()
```

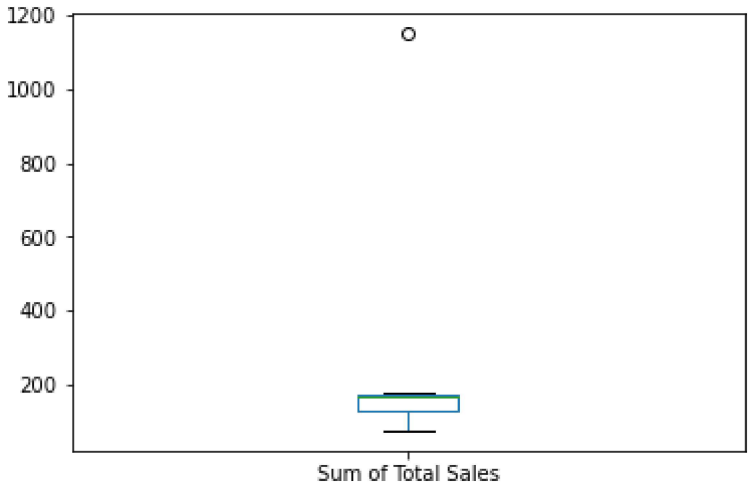
Out[16]: <AxesSubplot:>



box plot

In [17]:
```
df.plot.box()
```

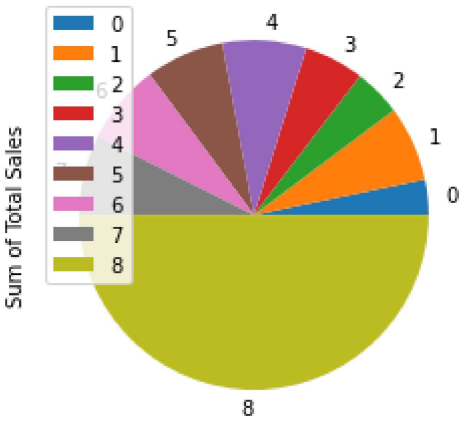
Out[17]: <AxesSubplot:>



pie plot

In [20]:
```
df.plot.pie(y="Sum of Total Sales")
```

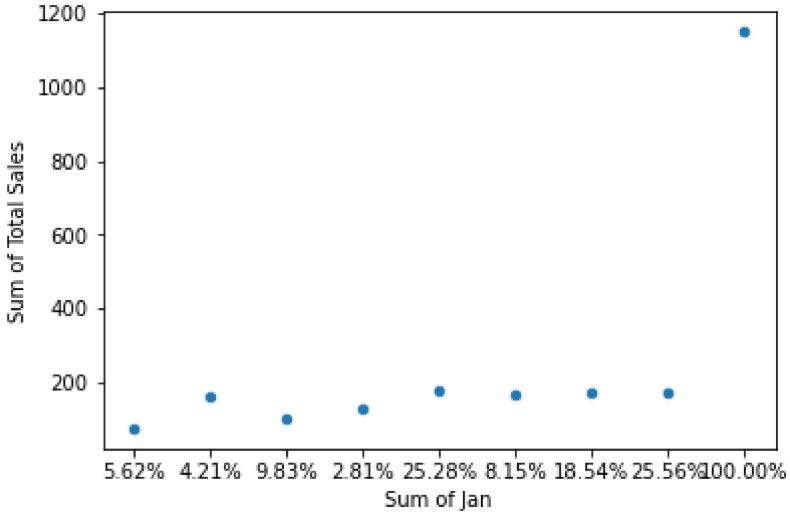
Out[20]: <AxesSubplot:ylabel='Sum of Total Sales'>



scatter plot

In [19]:
```
df.plot.scatter(x="Sum of Jan",y="Sum of Total Sales")
```

Out[19]: <AxesSubplot:xlabel='Sum of Jan', ylabel='Sum of Total Sales'>


1200
1000
800
600
400
200
Sum of Total Sales
5.62%
4.21%
9.83%
2.81%
25.28%
8.15%
18.54%
25.56%
100.00%
Sum of Jan