In [1]:
```
import pandas as pd
city=pd.read_csv("cities.csv")
city
```

Out[1]:

	LatD	"LatM"	"LatS"	"NS"	"LonD"	"LonM"	"LonS"	"EW"	"City"	"State"
0	41	5	59	"N"	80	39	0	"W"	"Youngstown"	OH
1	42	52	48	"N"	97	23	23	"W"	"Yankton"	SD
2	46	35	59	"N"	120	30	36	"W"	"Yakima"	WA
3	42	16	12	"N"	71	48	0	"W"	"Worcester"	MA
4	43	37	48	"N"	89	46	11	"W"	"Wisconsin Dells"	WI
...	...	...	...	...	...	...	...	...	...	...
123	39	31	12	"N"	119	48	35	"W"	"Reno"	NV
124	50	25	11	"N"	104	39	0	"W"	"Regina"	SA
125	40	10	48	"N"	122	14	23	"W"	"Red Bluff"	CA
126	40	19	48	"N"	75	55	48	"W"	"Reading"	PA
127	41	9	35	"N"	81	14	23	"W"	"Ravenna"	OH

128 rows × 10 columns

In [3]:
```
city.head(3)
```

Out[3]:

	LatD	"LatM"	"LatS"	"NS"	"LonD"	"LonM"	"LonS"	"EW"	"City"	"State"
0	41	5	59	"N"	80	39	0	"W"	"Youngstown"	OH
1	42	52	48	"N"	97	23	23	"W"	"Yankton"	SD
2	46	35	59	"N"	120	30	36	"W"	"Yakima"	WA

In [4]:
```
city.tail(3)
```

Out[4]:

	LatD	"LatM"	"LatS"	"NS"	"LonD"	"LonM"	"LonS"	"EW"	"City"	"State"
125	40	10	48	"N"	122	14	23	"W"	"Red Bluff"	CA
126	40	19	48	"N"	75	55	48	"W"	"Reading"	PA
127	41	9	35	"N"	81	14	23	"W"	"Ravenna"	OH

In [5]:
```
city.shape
```

Out[5]:
```
(128, 10)
```

In [6]:
```
city.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 128 entries, 0 to 127
Data columns (total 10 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   LatD      128 non-null    int64
 1    "LatM"   128 non-null    int64
 2    "LatS"   128 non-null    int64
 3    "NS"     128 non-null    object
 4    "LonD"   128 non-null    int64
 5    "LonM"   128 non-null    int64
 6    "LonS"   128 non-null    int64
 7    "EW"     128 non-null    object
 8    "City"   128 non-null    object
 9    "State"  128 non-null    object
dtypes: int64(6), object(4)
memory usage: 10.1+ KB
```

In [8]:
```
city.isna().sum()
```

Out[8]:
```
LatD        0
 "LatM"     0
 "LatS"     0
 "NS"       0
 "LonD"     0
 "LonM"     0
 "LonS"     0
 "EW"       0
 "City"     0
 "State"    0
dtype: int64
```

In [9]:
```
city.LatD.isna().sum()
```

Out[9]:
```
0
```

In [10]:
```
city.describe()
```

Out[10]:

	LatD	"LatM"	"LatS"	"LonD"	"LonM"	"LonS"
count	128.000000	128.000000	128.000000	128.000000	128.000000	128.000000
mean	38.820312	30.765625	27.492188	93.250000	27.742188	26.960938
std	5.200596	16.426158	18.977814	15.466499	16.927937	18.727807
min	26.000000	1.000000	0.000000	71.000000	0.000000	0.000000
25%	35.000000	16.000000	11.000000	80.000000	14.000000	11.000000
50%	39.000000	31.000000	24.000000	89.500000	26.500000	23.500000
75%	42.250000	45.000000	47.000000	103.250000	40.250000	47.000000
max	50.000000	59.000000	59.000000	123.000000	58.000000	59.000000

In [11]:
```
city.nunique()
```

Out[11]:
```
LatD         25
 "LatM"      51
 "LatS"      10
 "NS"         1
 "LonD"      44
 "LonM"      53
 "LonS"      10
 "EW"         1
 "City"     120
 "State"     47
dtype: int64
```

```
import numpy
```

```
array_data=numpy.array([10,20,30,50,78])

#display mean
print(numpy.mean(array_data))
#display average
print(numpy.average(array_data))
#display median
print(numpy.median(array_data))
#display variance
print(numpy.var(array_data))
#display standard deviation
print(numpy.std(array_data))
#display maximum
print(numpy.max(array_data))
#display minimum
print(numpy.min(array_data))
#50th percentile
print(numpy.percentile(array_data, 50))
#25th percentile
print(numpy.percentile(array_data, 25))
#75th percentile
print(numpy.percentile(array_data, 75))
```

```
37.6
37.6
30.0
583.04
24.146221236458512
78
10
30.0
20.0
50.0
```

```
import numpy as np
arr1 = np.array([[1, 2], [5, 6]])
arr2 = np.array([[7, 8], [3,4]])
a = np.concatenate((arr1, arr2))
print(a)
```

```
[[1 2]
 [5 6]
 [7 8]
 [3 4]]
```

```
arr1 = np.array([[1, 2], [5, 6]])
arr2 = np.array([[7, 8], [3,4]])
a = np.append(arr1,arr2, axis=1)
print(a)
```

```
[[1 2 7 8]
 [5 6 3 4]]
```

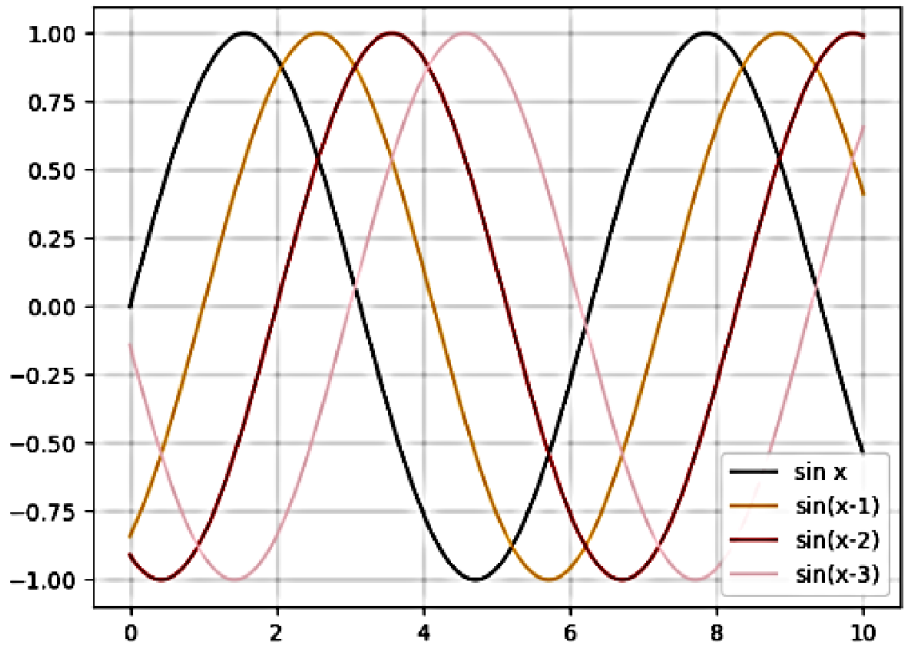
```
a = np.array([1, 2, 3])
b = np.array([5,6,7])
a = np.vstack((a,b))
print(a)
```

```
[[1 2 3]
 [5 6 7]]
```

```
a = np.array([[1, 2],[3,4]])
b = np.array([[5,6],[7,8]])
a = np.dstack((a,b))
print(a)
```

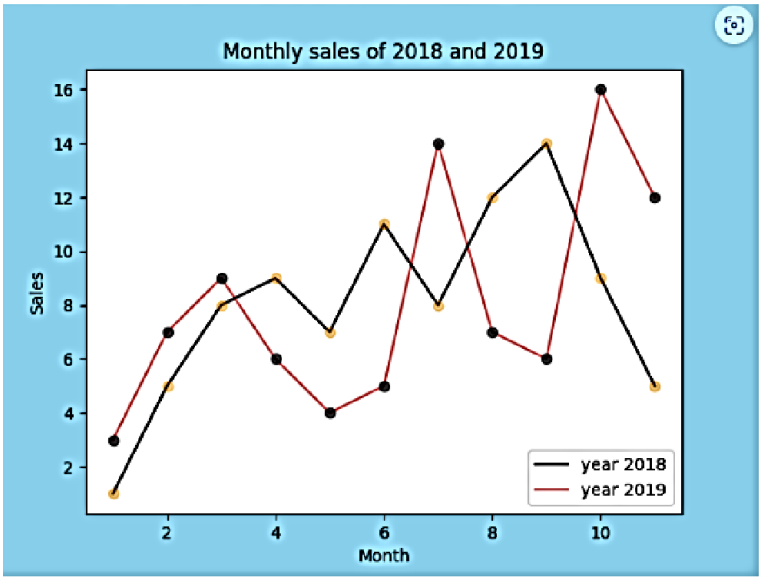
```
[[[1 5]
  [2 6]]

 [[3 7]
  [4 8]]]
```

```
import matplotlib.pyplot as plt
import numpy as np
fig = plt.figure()
ax = plt.axes()
ax.grid(color='black', linestyle='-', linewidth=2,alpha=0.1)
#creating numeric sequence with evenly spaced
x = np.linspace(0, 10, 1000)
plt.plot(x, np.sin(x - 0), color='black',alpha=0.8,label='sin x')
plt.plot(x, np.sin(x - 1), color='orange',label='sin(x-1)')
plt.plot(x, np.sin(x - 2),color='r' ,label='sin(x-2)')
plt.plot(x, np.sin(x - 3), color='pink',label='sin(x-3)')

plt.legend(loc='best')

plt.show()
```

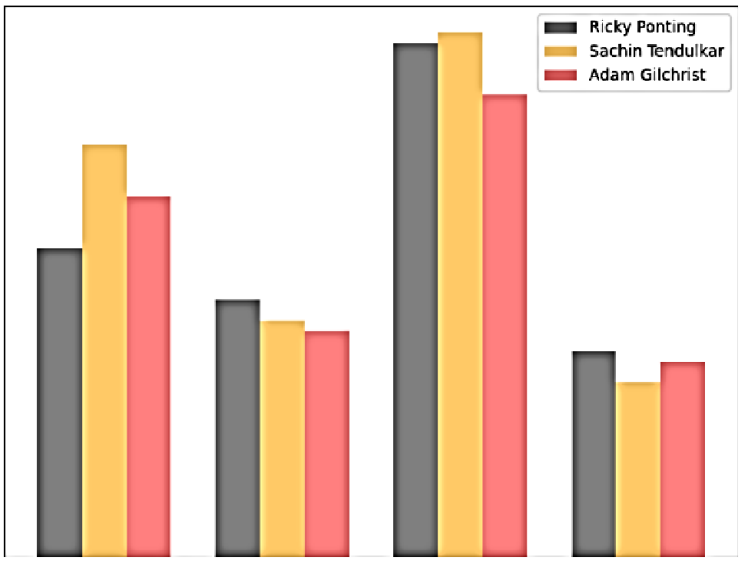



```
<Figure size 640x480 with 0 Axes>
```

```
#data-points or markers
sales1 = [1, 5, 8, 9, 7, 11, 8, 12, 14, 9, 5]
sales2 = [3, 7, 9, 6, 4, 5, 14, 7, 6, 16, 12]
#plotting line plots
line_chart1 = plt.plot(range(1,12), sales1,color='black')
line_chart2 = plt.plot(range(1,12), sales2,color='r',alpha=0.6)
#creating  points
plt.plot(range(1,12), sales2,'C0o', alpha=0.5,color='black')
plt.plot(range(1,12), sales1,'C0o', alpha=0.5,color='orange')
#creating title,Y-Label,X-Label & legend
plt.title('Monthly sales of 2018 and 2019')
plt.ylabel('Sales')
plt.xlabel('Month')
plt.legend(['year 2018', 'year 2019'], loc=4)
#saving figure to my device in .png format
plt.savefig('plot15.png', dpi=300, bbox_inches='tight')
plt.show()
```

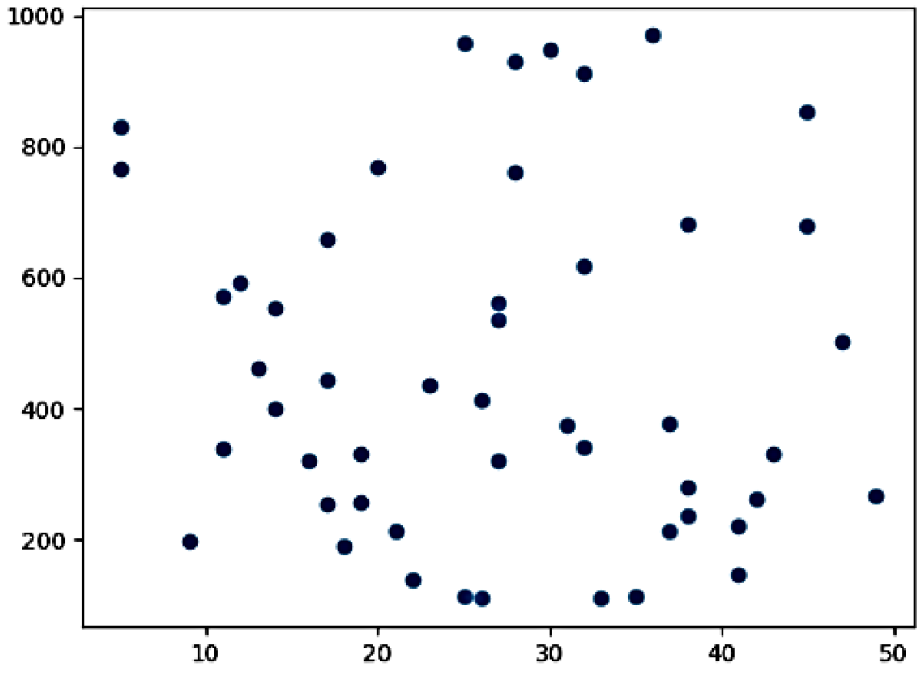


```
data = [[30, 25, 50, 20],
[40, 23, 51, 17],
[35, 22, 45, 19]]
label=['Match-1','Match-2','Match-3','Match-4']
X = np.arange(len(label))
y=[0.25,1.25,2.25,3.25]
#creating figure,axex,title&labels
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
plt.xlabel('Match')
plt.ylabel('Score')
plt.title('Comparing scores of three players in different cricket matches')
plt.xticks(y,label)
#creating bar plots
ax.bar(X + 0.00, data[0], color = 'black', width = 0.25,alpha=0.5,label='Ricky Ponting')
ax.bar(X + 0.25, data[1], color = 'orange', width = 0.25,alpha=0.6,label='Sachin Tendulkar')
ax.bar(X + 0.50, data[2], color = 'red', width = 0.25,alpha=0.5,label='Adam Gilchrist')
plt.legend()
plt.show()
```




```
<Figure size 640x480 with 0 Axes>
```

```
x = np.random.randint(5, 50, 50)
y = np.random.randint(100, 1000, 50)

plt.scatter(x, y)
plt.show()
```




```
<Figure size 640x480 with 0 Axes>
```