

Matplotlib

line / plot

Plotting x and y points

The `plot()` function is used to draw points (markers) in a diagram.

By default, the `plot()` function draws a line from point to point.

The function takes parameters for specifying points in the diagram.

Parameter 1 is an array containing the points on the x-axis.

Parameter 2 is an array containing the points on the y-axis.

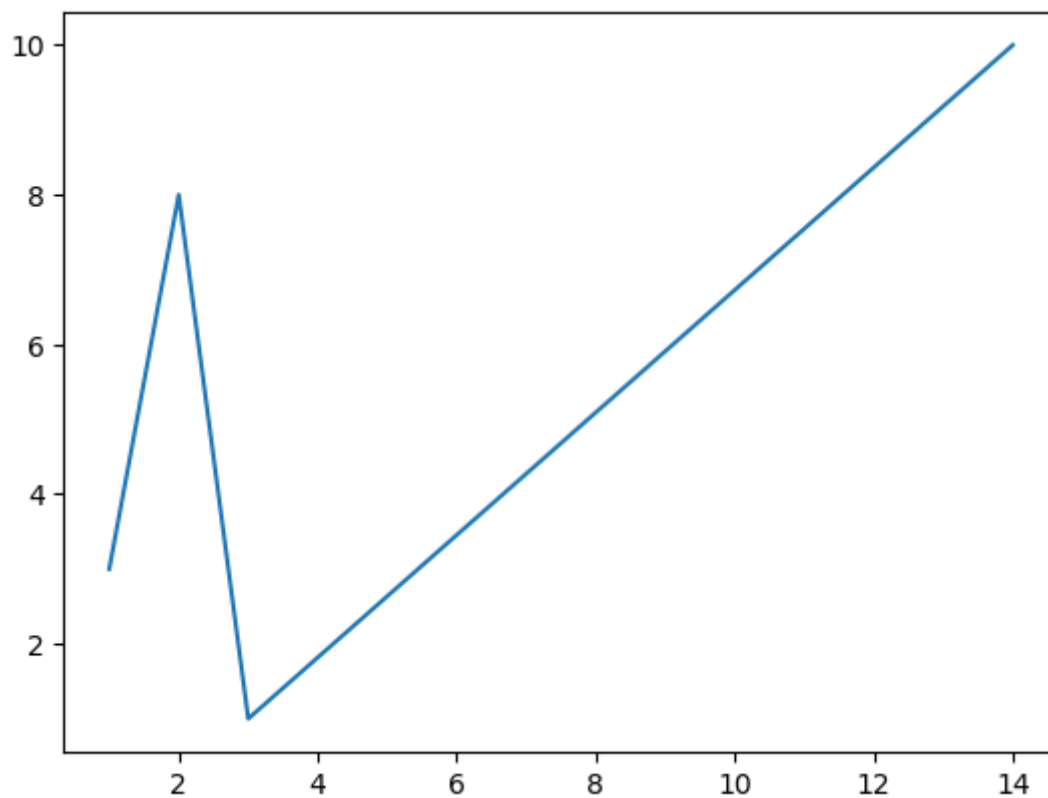
If we need to plot a line from (1, 3) to (8, 10), we have to pass two arrays [1, 8] and [3, 10] to the `plot` function.

In [1]:

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

x = np.array([1, 2, 3, 14])
y = np.array([3, 8, 1, 10])

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



Marker

Marker Description

'o' Circle

'*' Star

'.' Point

',' Pixel

'x' X

'X' X (filled)

'+' Plus

'P' Plus (filled)

's' Square

'D' Diamond

'd' Diamond (thin)

'p' Pentagon

'H' Hexagon

'h' Hexagon

'v' Triangle Down

'^' Triangle Up

'<' Triangle Left

'>' Triangle Right

'1' Tri Down

'2' Tri Up

'3' Tri Left

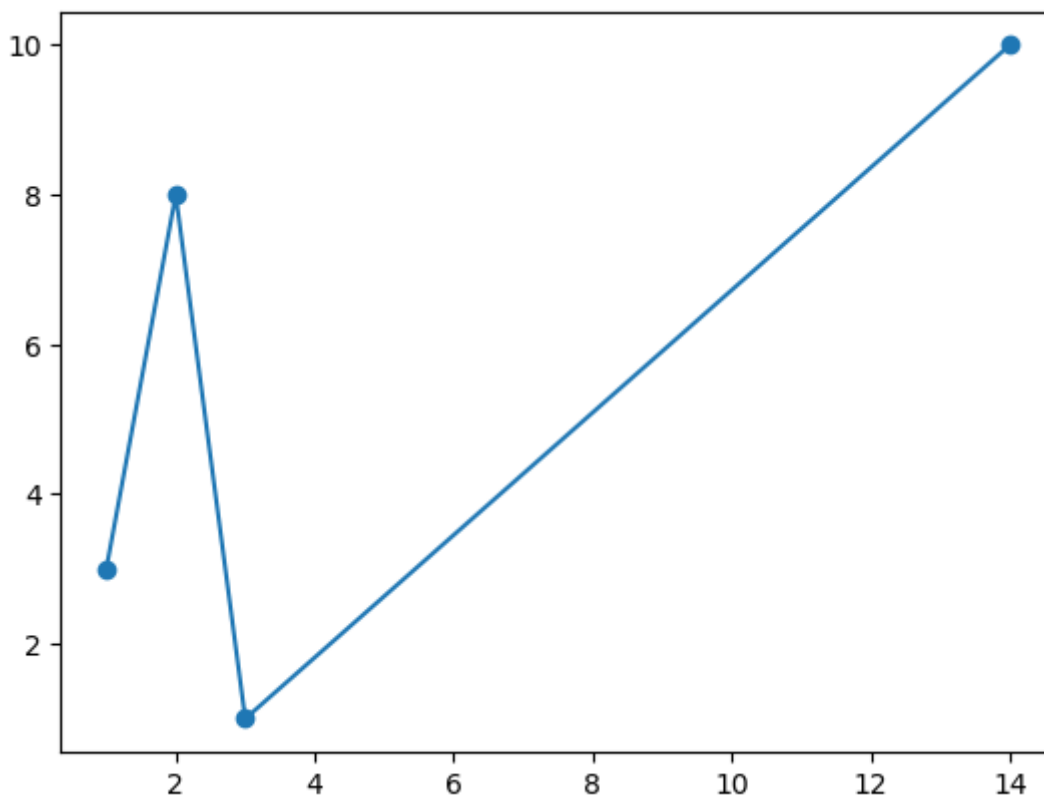
'4' Tri Right

'|' Vline

' ' Hline

In [4]:

```
plt.plot(x, y, marker = 'o')  
plt.show()
```



Line Reference

Line Syntax Description

'-' Solid line

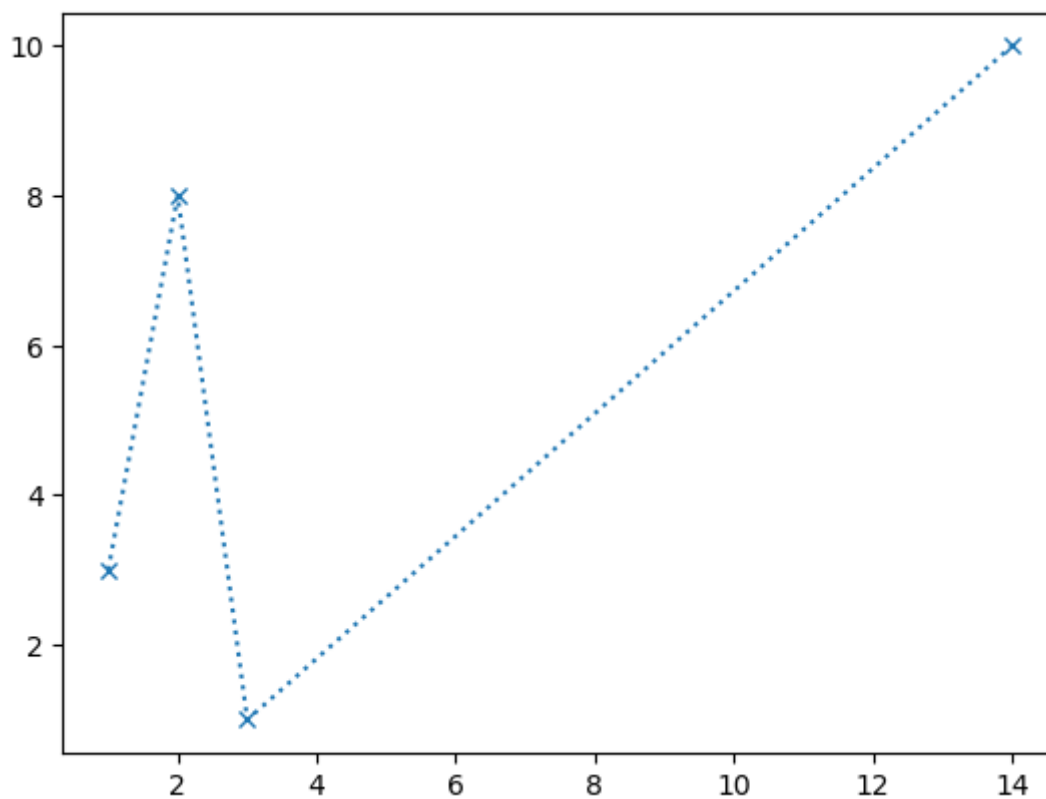
'.' Dotted line

'--' Dashed line

'-.' Dashed/dotted line

In [5]:

```
plt.plot(x, y, 'x:')  
plt.show()
```



Color Reference

Color Syntax Description

'r' Red

'g' Green

'b' Blue

'c' Cyan

'm' Magenta

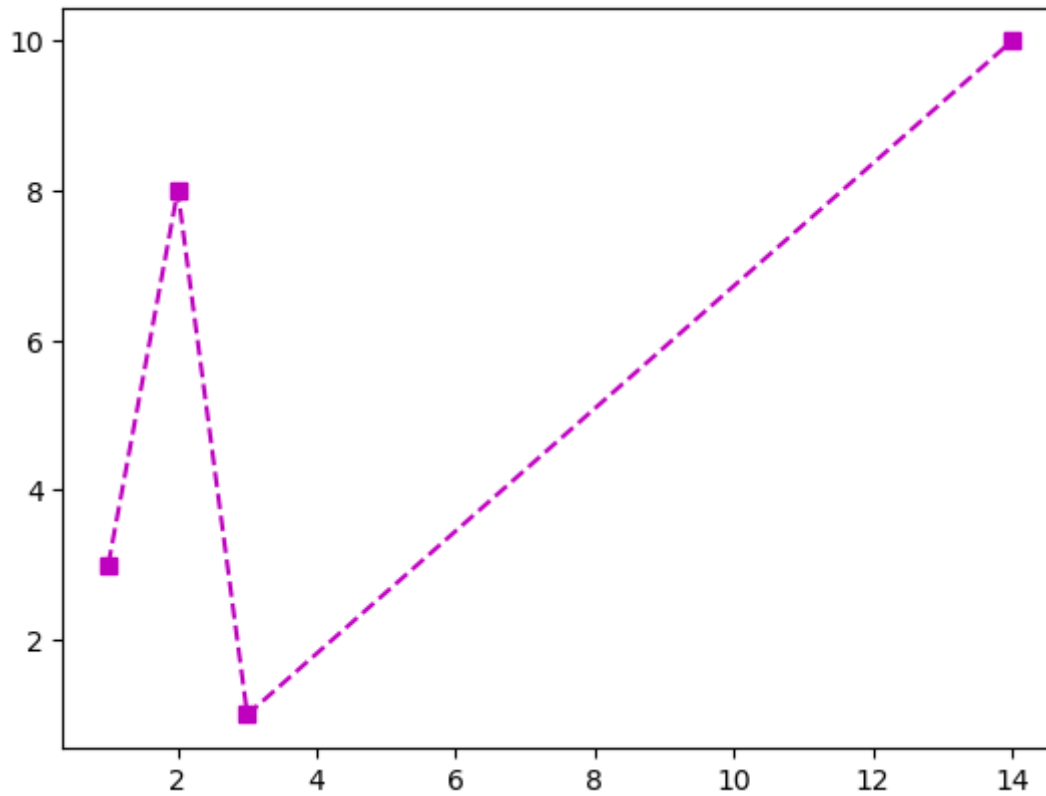
'y' Yellow

'k' Black

'w' White

In [6]:

```
plt.plot(x, y, 's--m')  
plt.show()
```



ms

size of the markers

mec

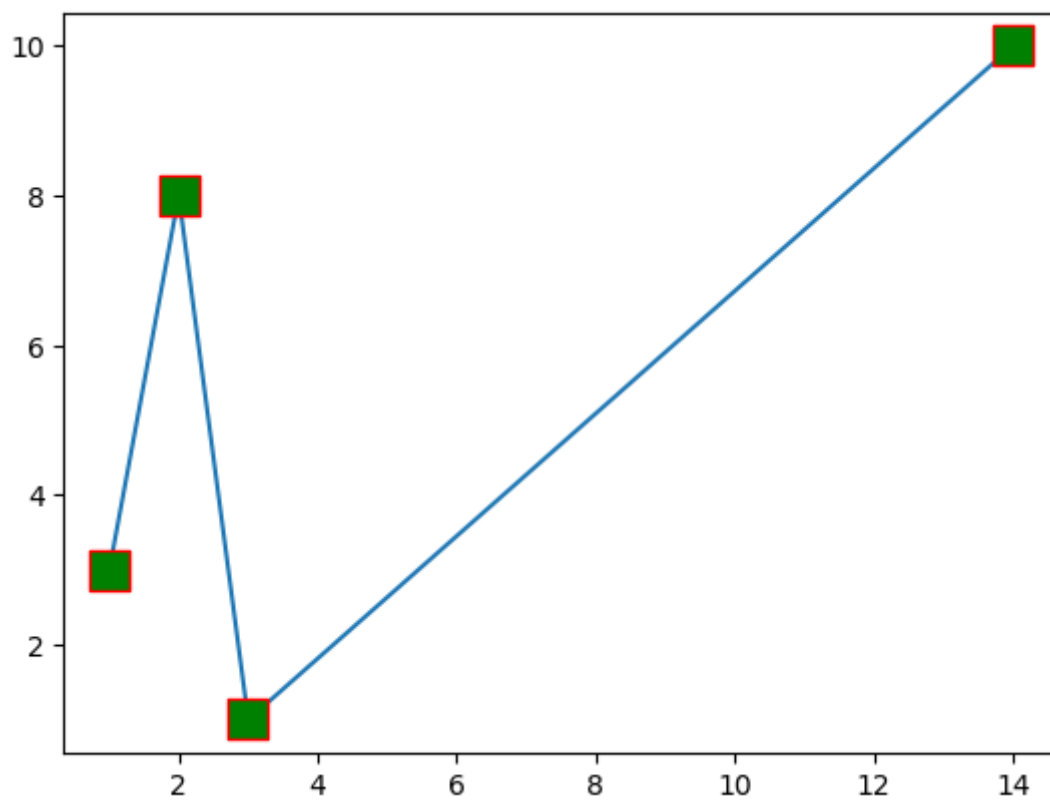
markeredgcolor

mfc

markerfacecolor

In [7]:

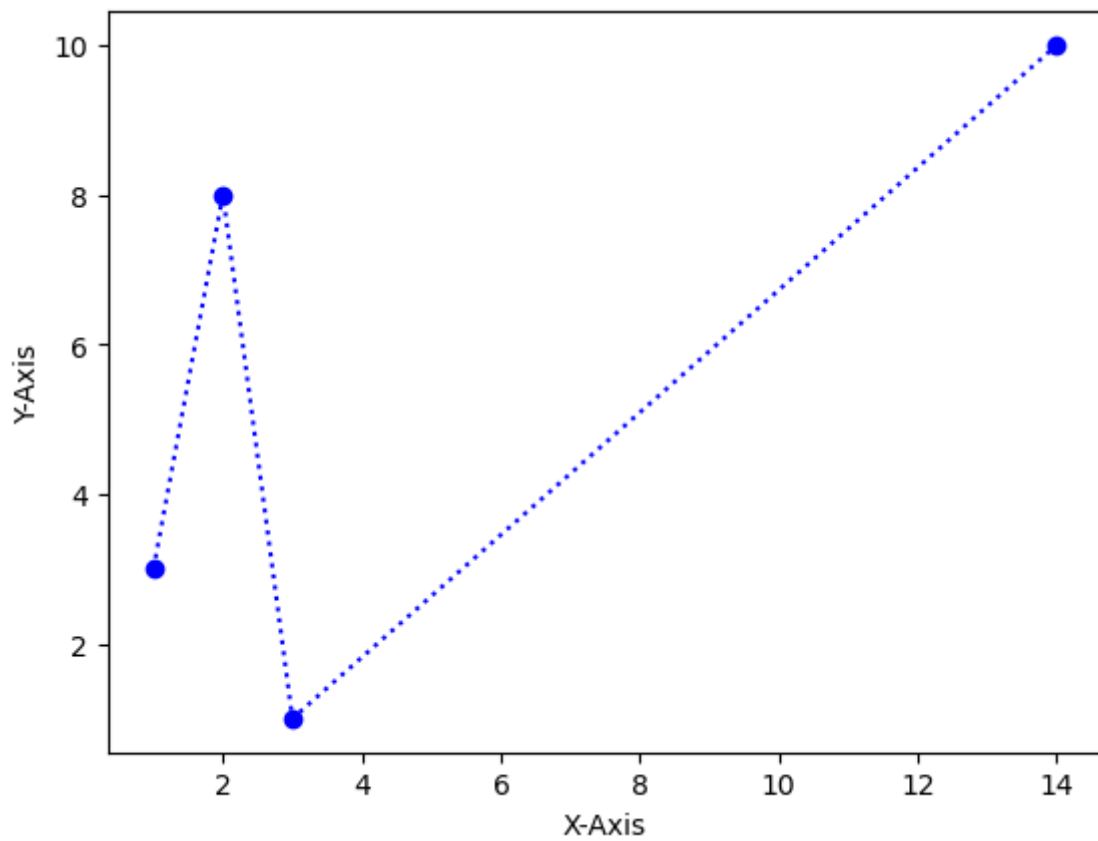
```
plt.plot(x,y, marker = 's', ms = 15, mec = 'r', mfc = 'g')  
plt.show()
```



Create Labels for a Plot

In [8]:

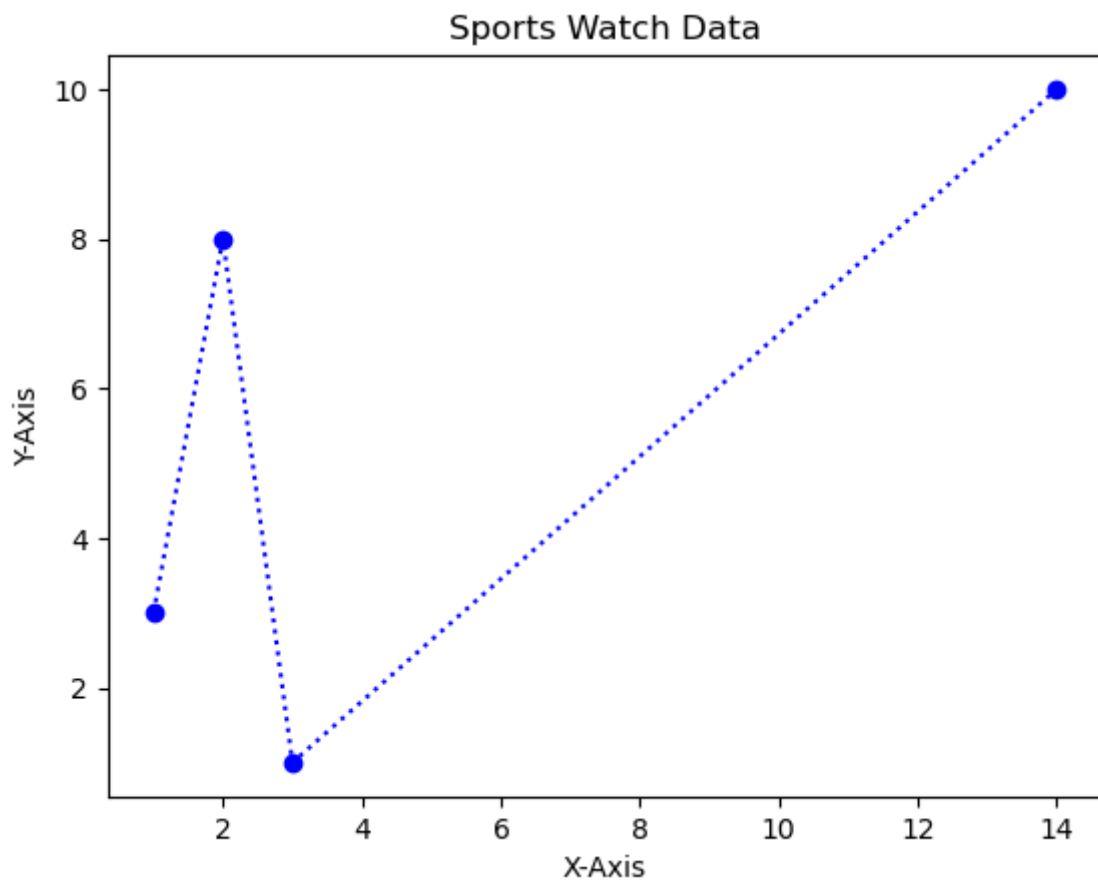
```
plt.plot(x,y,'o:b')  
  
plt.xlabel("X-Axis")  
plt.ylabel("Y-Axis")  
  
plt.show()
```



Create a Title for a Plot

In [9]:

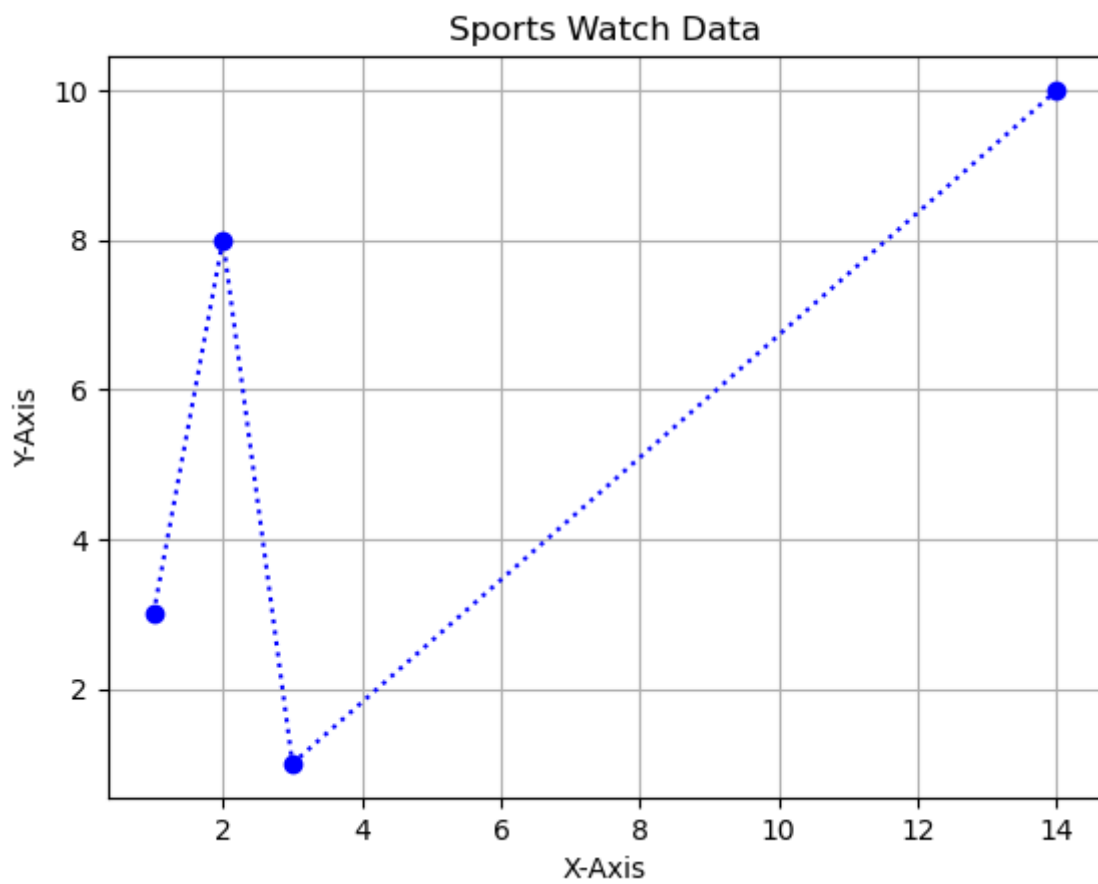
```
plt.plot(x,y,'o:b')  
  
plt.title("Sports Watch Data")  
plt.xlabel("X-Axis")  
plt.ylabel("Y-Axis")  
  
plt.show()
```



Add Grid Lines to a Plot

In [10]:

```
plt.plot(x,y, 'o:b')  
  
plt.title("Sports Watch Data")  
plt.xlabel("X-Axis")  
plt.ylabel("Y-Axis")  
  
plt.grid()  
  
plt.show()
```



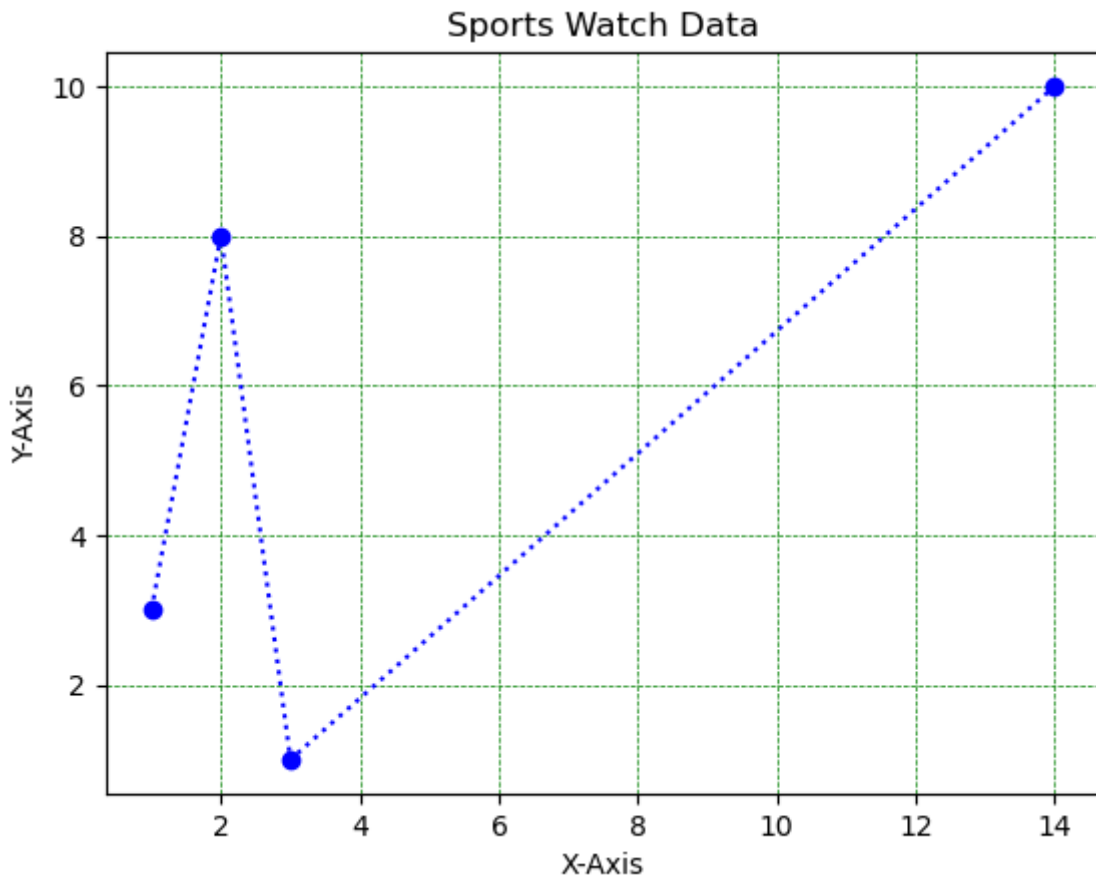
In [11]:

```
plt.plot(x,y,'o:b')

plt.title("Sports Watch Data")
plt.xlabel("X-Axis")
plt.ylabel("Y-Axis")

plt.grid(color = 'green', linestyle = '--', linewidth = 0.5)

plt.show()
```



Display Multiple Plots

plt.subplot(1, 2, 1)

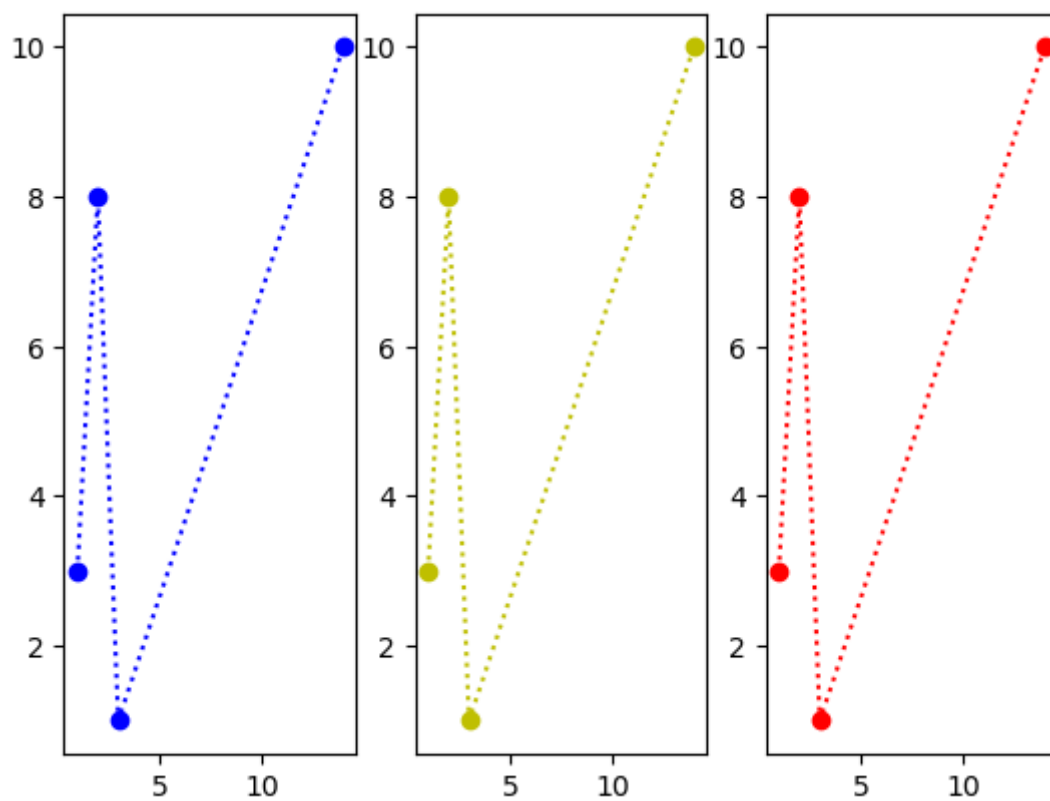
the figure has 1 row, 2 columns, and this plot is the first plot.

plt.subplot(1, 2, 2)

the figure has 1 row, 2 columns, and this plot is the second plot.

In [12]:

```
#plot 1:  
plt.subplot(1, 3, 1)  
plt.plot(x,y,'o:b')  
  
#plot 2:  
plt.subplot(1, 3, 3)  
plt.plot(x,y,'o:r')  
  
#plot 3:  
plt.subplot(1, 3, 2)  
plt.plot(x,y,'o:y')  
  
plt.show()
```



In [13]:

```
x = np.array([1, 2, 3, 14])
y = np.array([3, 8, 1, 10])
```

#plot 1:

```
plt.subplot(1, 3, 1)
plt.plot(x,y,'o:b')
```

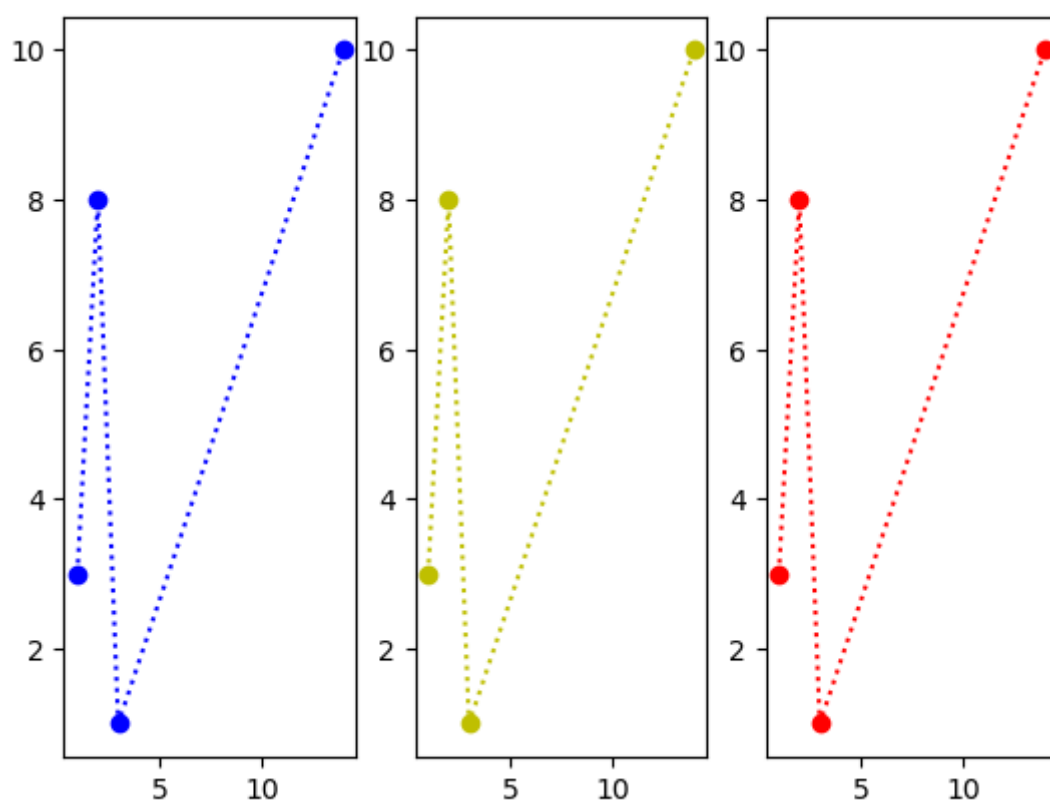
#plot 2:

```
plt.subplot(1, 3, 3)
plt.plot(x,y,'o:r')
```

#plot 3:

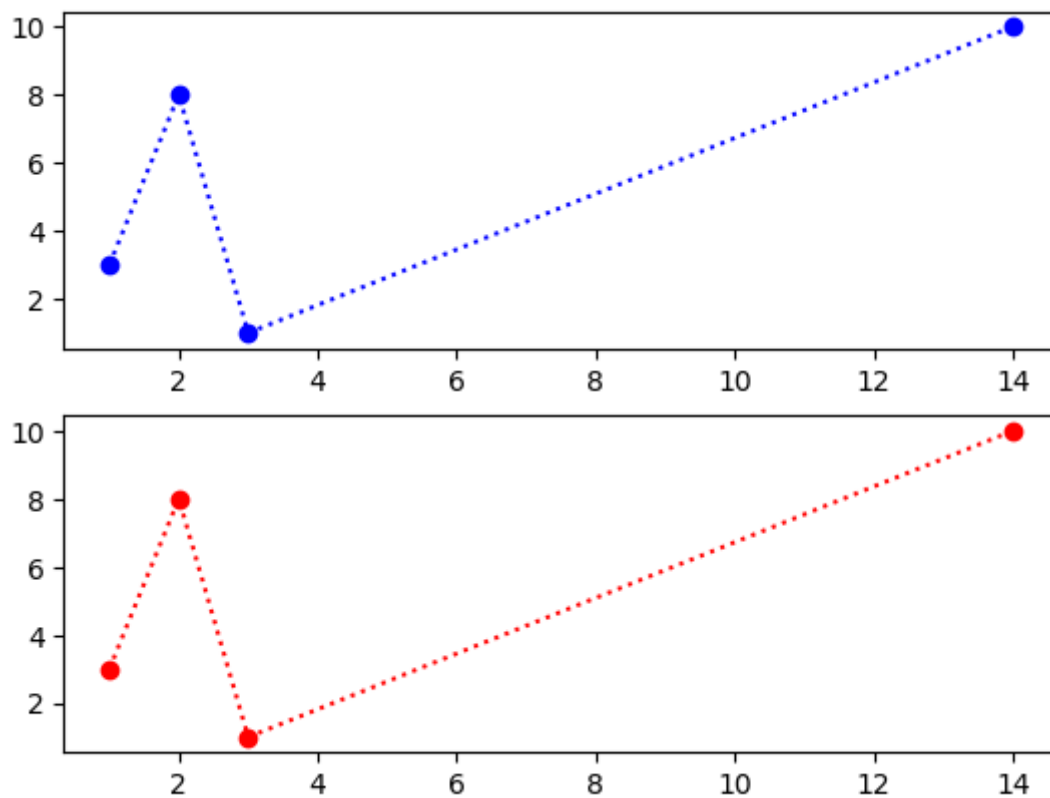
```
plt.subplot(1, 3, 2)
plt.plot(x,y,'o:y')
```

```
plt.show()
```



In [14]:

```
#plot 1:  
plt.subplot(2, 1, 1)  
plt.plot(x,y,'o:b')  
  
#plot 2:  
plt.subplot(2, 1, 2)  
plt.plot(x,y,'o:r')  
  
plt.show()
```



In [15]:

```
# plot 1
plt.subplot(2, 3, 1)
plt.plot(x,y,'s:b')

# plot 2
plt.subplot(2, 3, 2)
plt.plot(x,y,'s:r')

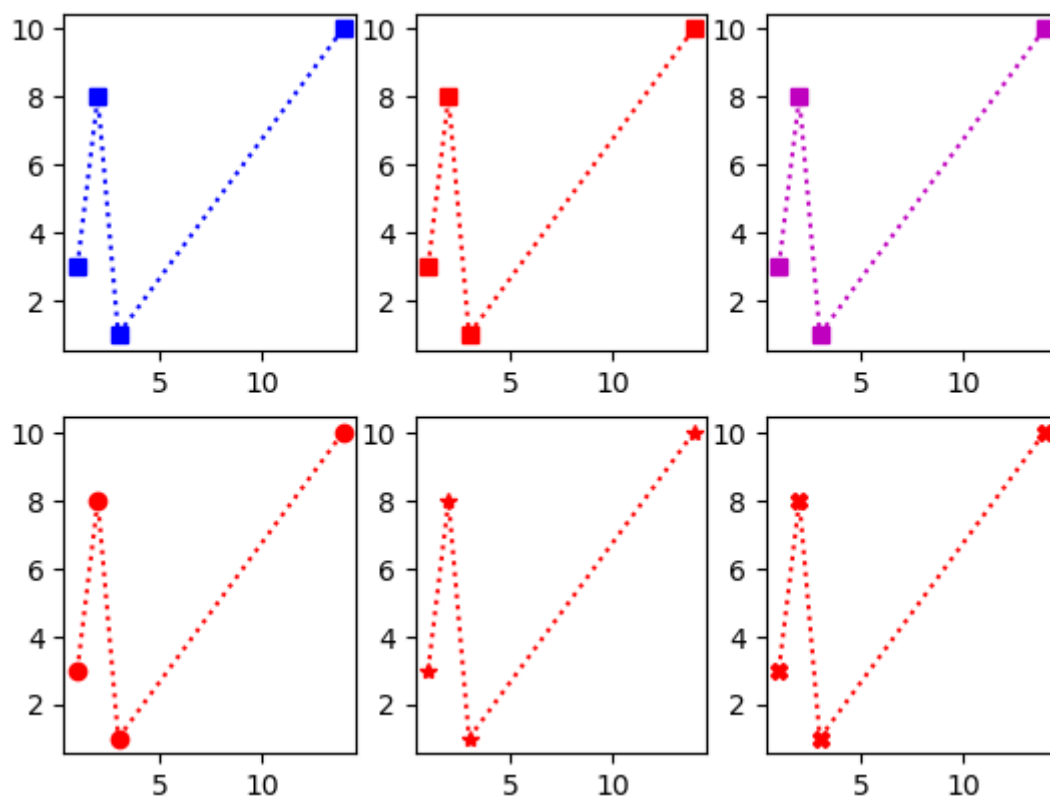
# plot 3
plt.subplot(2, 3, 3)
plt.plot(x,y,'s:m')

# plot 4
plt.subplot(2, 3, 4)
plt.plot(x,y,'o:r')

# plot 5
plt.subplot(2, 3, 5)
plt.plot(x,y,'*:r')

# plot 6
plt.subplot(2, 3, 6)
plt.plot(x,y,'X:r')

plt.show()
```



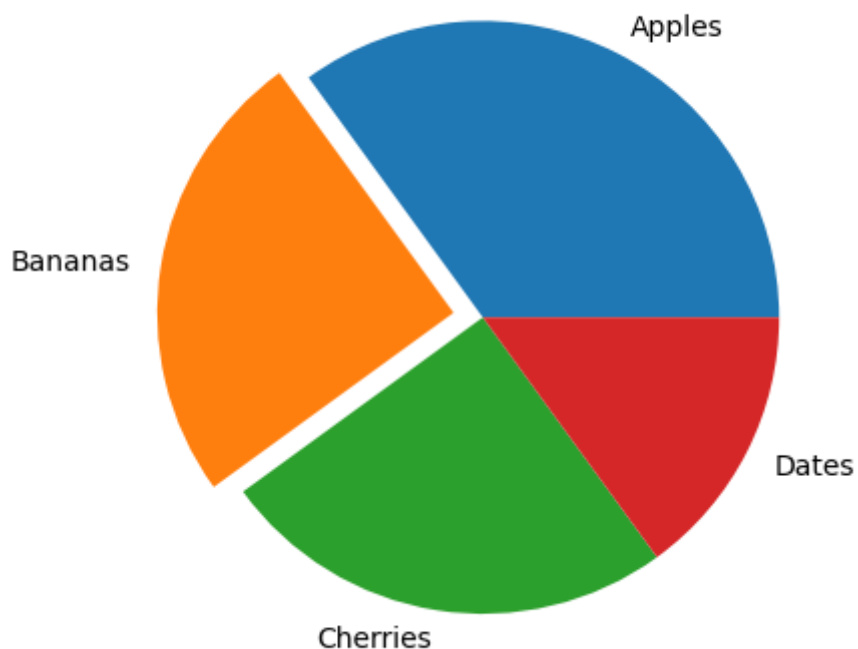
Pie Chart

In []:

```
y = np.array([35, 25, 25, 15])  
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]  
  
plt.pie(y, labels = mylabels)  
plt.show()
```

In [16]:

```
y = np.array([35, 25, 25, 15])  
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]  
myexplode = [0,0.1,0,0]  
plt.pie(y, labels = mylabels, explode = myexplode)  
plt.show()
```



In []: