# **Matplotlib Scatter**

## **Creating Scatter Plots**

With Pyplot, you can use the scatter() function to draw a scatter plot.

The scatter() function plots one dot for each observation. It needs two arrays of the same length, one for the values of the x-axis, and one for values on the y-axis:

### Example

A simple scatter plot:

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])  
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])  
  
plt.scatter(x, y)  
plt.show()



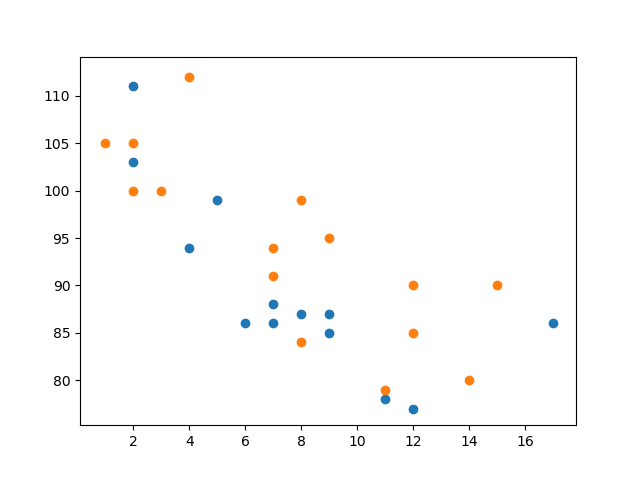
## **Compare Plots**

In the example above, there seems to be a relationship between speed and age, but what if we plot the observations from another day as well? Will the scatter plot tell us something else?

### Example

Draw two plots on the same figure:

import matplotlib.pyplot as plt  
import numpy as np  
  
#day one, the age and speed of 13 cars:  
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])  
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])  
plt.scatter(x, y)  
  
#day two, the age and speed of 15 cars:  
x = np.array([2,2,8,1,15,8,12,9,7,3,11,4,7,14,12])  
y = np.array([100,105,84,105,90,99,90,95,94,100,79,112,91,80,85])  
plt.scatter(x, y)  
  
plt.show()



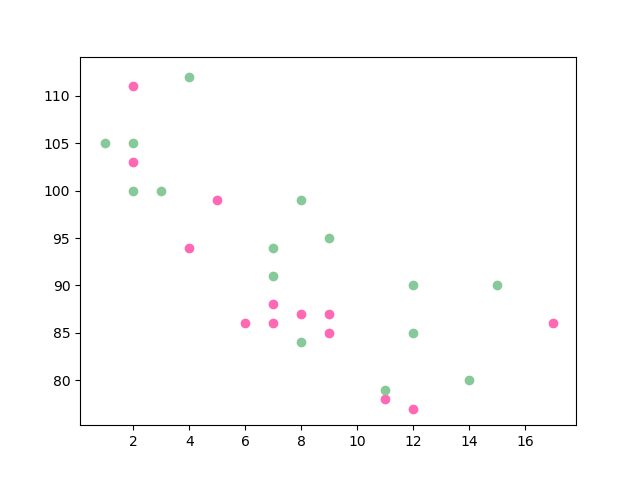
## **Colors**

You can set your own color for each scatter plot with the color or the c argument:

### Example

Set your own color of the markers:

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])  
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])  
plt.scatter(x, y, color = 'hotpink')  
  
x = np.array([2,2,8,1,15,8,12,9,7,3,11,4,7,14,12])  
y = np.array([100,105,84,105,90,99,90,95,94,100,79,112,91,80,85])  
plt.scatter(x, y, color = '#88c999')  
  
plt.show()



## **Color Each Dot**

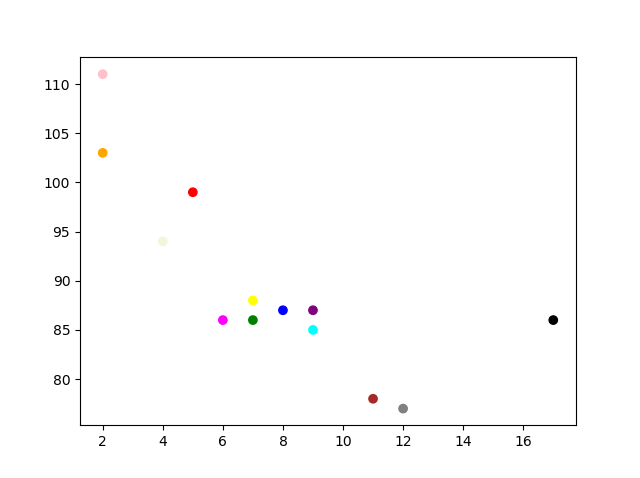
You can even set a specific color for each dot by using an array of colors as value for the c argument:

**Note:** You cannot use the color argument for this, only the c argument.

### Example

Set your own color of the markers:

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])  
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])  
colors = np.array(["red","green","blue","yellow","pink","black","orange","purple","beige","brown","gray","cyan","magenta"])  
  
plt.scatter(x, y, c=colors)  
  
plt.show()



## **ColorMap**

The Matplotlib module has a number of available colormaps.

A colormap is like a list of colors, where each color has a value that ranges from 0 to 100.

Here is an example of a colormap:



### How to Use the ColorMap

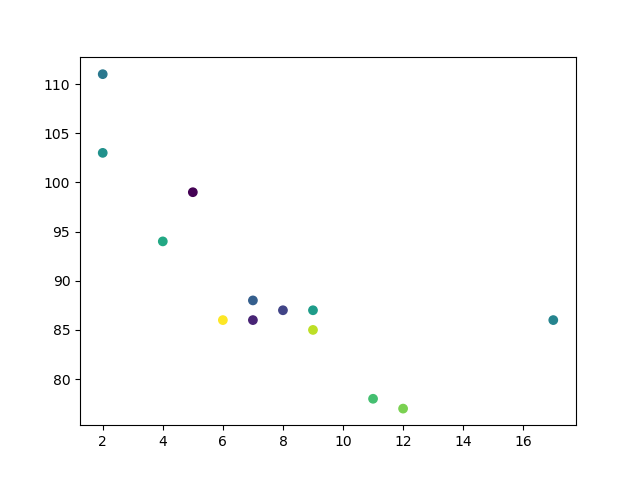
You can specify the colormap with the keyword argument cmap with the value of the colormap, in this case 'viridis' which is one of the built-in colormaps available in Matplotlib.

In addition you have to create an array with values (from 0 to 100), one value for each of the point in the scatter plot:

### Example

Create a color array, and specify a colormap in the scatter plot:

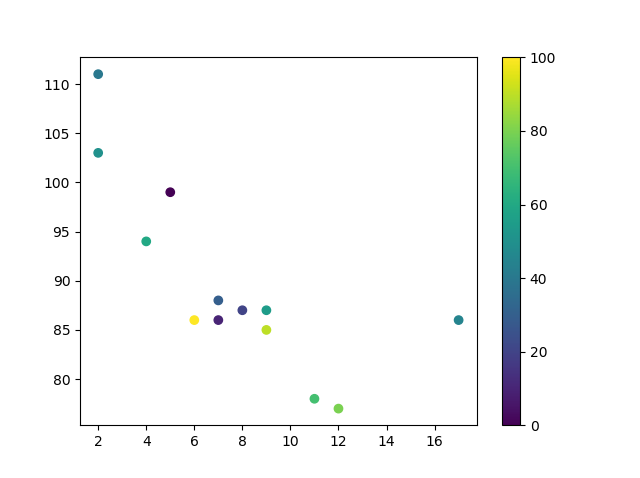
import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])  
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])  
colors = np.array([0, 10, 20, 30, 40, 45, 50, 55, 60, 70, 80, 90, 100])  
  
plt.scatter(x, y, c=colors, cmap='viridis')  
  
plt.show()



### Example

Include the actual colormap:

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])  
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])  
colors = np.array([0, 10, 20, 30, 40, 45, 50, 55, 60, 70, 80, 90, 100])  
  
plt.scatter(x, y, c=colors, cmap='viridis')  
  
plt.colorbar()  
  
plt.show()



### Available ColorMaps

You can choose any of the built-in colormaps:

|  |  |  |
| --- | --- | --- |
| **Name** |  | **Reverse** |
| Accent |  | Accent\_r |
| Blues |  | Blues\_r |
| BrBG |  | BrBG\_r |
| BuGn |  | BuGn\_r |
| BuPu |  | BuPu\_r |
| CMRmap |  | CMRmap\_r |
| Dark2 |  | Dark2\_r |
| GnBu |  | GnBu\_r |
| Greens |  | Greens\_r |
| Greys |  | Greys\_r |
| OrRd |  | OrRd\_r |
| Oranges |  | Oranges\_r |
| PRGn |  | PRGn\_r |
| Paired |  | Paired\_r |
| Pastel1 |  | Pastel1\_r |
| Pastel2 |  | Pastel2\_r |
| PiYG |  | PiYG\_r |
| PuBu |  | PuBu\_r |
| PuBuGn |  | PuBuGn\_r |
| PuOr |  | PuOr\_r |
| PuRd |  | PuRd\_r |
| Purples |  | Purples\_r |
| RdBu |  | RdBu\_r |
| RdGy |  | RdGy\_r |
| RdPu |  | RdPu\_r |
| RdYlBu |  | RdYlBu\_r |
| RdYlGn |  | RdYlGn\_r |
| Reds |  | Reds\_r |
| Set1 |  | Set1\_r |
| Set2 |  | Set2\_r |
| Set3 |  | Set3\_r |
| Spectral |  | Spectral\_r |
| Wistia |  | Wistia\_r |
| YlGn |  | YlGn\_r |
| YlGnBu |  | YlGnBu\_r |
| YlOrBr |  | YlOrBr\_r |
| YlOrRd |  | YlOrRd\_r |
| afmhot |  | afmhot\_r |
| autumn |  | autumn\_r |
| binary |  | binary\_r |
| bone |  | bone\_r |
| brg |  | brg\_r |
| bwr |  | bwr\_r |
| cividis |  | cividis\_r |
| cool |  | cool\_r |
| coolwarm |  | coolwarm\_r |
| copper |  | copper\_r |
| cubehelix |  | cubehelix\_r |
| flag |  | flag\_r |
| gist\_earth |  | gist\_earth\_r |
| gist\_gray |  | gist\_gray\_r |
| gist\_heat |  | gist\_heat\_r |
| gist\_ncar |  | gist\_ncar\_r |
| gist\_rainbow |  | gist\_rainbow\_r |
| gist\_stern |  | gist\_stern\_r |
| gist\_yarg |  | gist\_yarg\_r |
| gnuplot |  | gnuplot\_r |
| gnuplot2 |  | gnuplot2\_r |
| gray |  | gray\_r |
| hot |  | hot\_r |
| hsv |  | hsv\_r |
| inferno |  | inferno\_r |
| jet |  | jet\_r |
| magma |  | magma\_r |
| nipy\_spectral |  | nipy\_spectral\_r |
| ocean |  | ocean\_r |
| pink |  | pink\_r |
| plasma |  | plasma\_r |
| prism |  | prism\_r |
| rainbow |  | rainbow\_r |
| seismic |  | seismic\_r |
| spring |  | spring\_r |
| summer |  | summer\_r |
| tab10 |  | tab10\_r |
| tab20 |  | tab20\_r |
| tab20b |  | tab20b\_r |
| tab20c |  | tab20c\_r |
| terrain |  | terrain\_r |
| twilight |  | twilight\_r |
| twilight\_shifted |  | twilight\_shifted\_r |
| viridis |  | viridis\_r |
| winter |  | winter\_r |

## **Size**

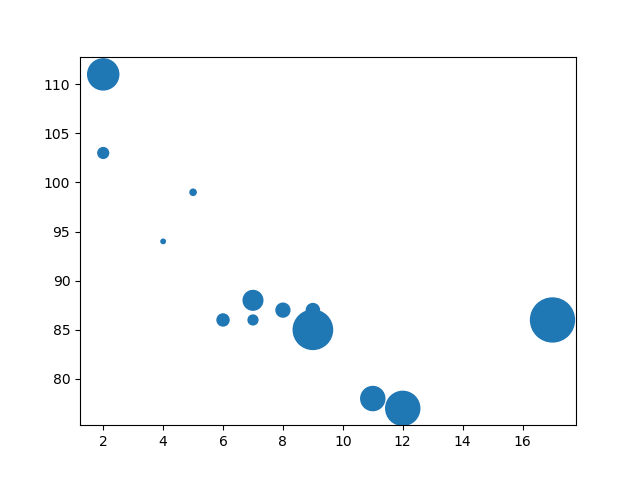
You can change the size of the dots with the s argument.

Just like colors, make sure the array for sizes has the same length as the arrays for the x- and y-axis:

### Example

Set your own size for the markers:

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])  
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])  
sizes = np.array([20,50,100,200,500,1000,60,90,10,300,600,800,75])  
  
plt.scatter(x, y, s=sizes)  
  
plt.show()



## **Alpha**

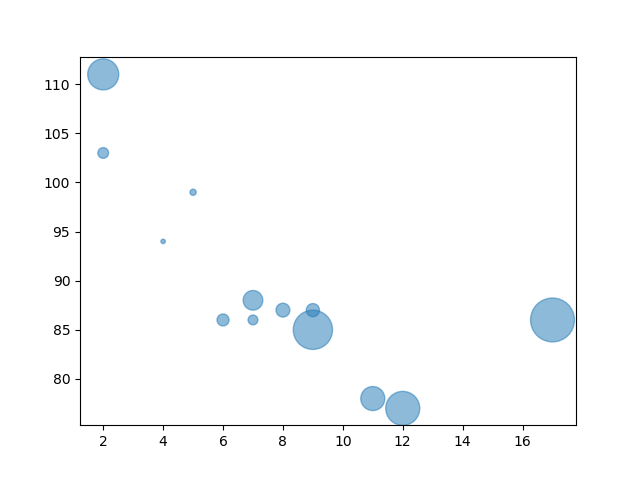
You can adjust the transparency of the dots with the alpha argument.

Just like colors, make sure the array for sizes has the same length as the arrays for the x- and y-axis:

### Example

Set your own size for the markers:

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])  
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])  
sizes = np.array([20,50,100,200,500,1000,60,90,10,300,600,800,75])  
  
plt.scatter(x, y, s=sizes, alpha=0.5)  
  
plt.show()



## **Combine Color Size and Alpha**

You can combine a colormap with different sizes on the dots. This is best visualized if the dots are transparent:

### Example

Create random arrays with 100 values for x-points, y-points, colors and sizes:

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.random.randint(100, size=(100))  
y = np.random.randint(100, size=(100))  
colors = np.random.randint(100, size=(100))  
sizes = 10 \* np.random.randint(100, size=(100))  
  
plt.scatter(x, y, c=colors, s=sizes, alpha=0.5, cmap='nipy\_spectral')  
  
plt.colorbar()  
  
plt.show()