Python CheetSheet

Naga Ganesh B J
bjnaga@gmail.com

A Quick Guide to NumPy

1 Introduction

Created by John D. Hunter, its a free, open source, low level plotting library for python written in C, Objective C and JavaScript for platform compatibility.

1.1 Installation

```
pip install matplotlib
```

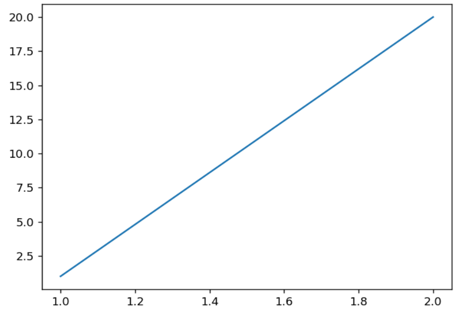
2 Pyplot Plotting

Listing 1: Insert code directly in your document

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

x_points = np.array([1,2])
y_points = np.array([1,20])
# prints the version of Matplotlib installed
print(matplotlib.__version__)

plt.plot(x_points,y_points)
plt.show()
```



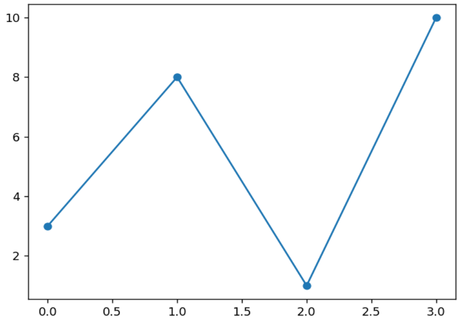
2.1 Markers

You can use the keyword argument marker to emphasize each point with a specified marker:

Listing 2: Insert code directly in your document

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

ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, marker = 'o')
plt.show()
```



Marker Reference

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

2.2 fmt

You can use also use the shortcut string notation parameter to specify the marker. This parameter is also called **fmt**, and is written with this syntax:

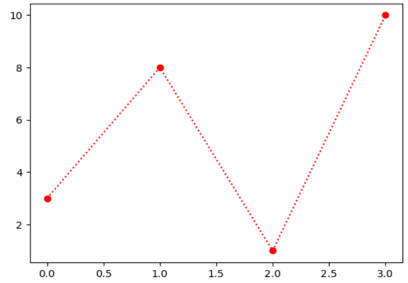
```
marker|line|color
```

Listing 3: Insert code directly in your document

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

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, 'o:r')
plt.show()
```



LineSyntax	Description
'-'	Solid line
':'	Dotted line
'--'	Dashed line
'-.'	Dashed/dotted line

Note: If you leave out the line value in the fmt parameter, no line will be plotted.

Color Reference

ColorSyntax	Description
'r'	Red
'g'	Green
'b'	Blue
'c'	Cyan
'm'	Magenta
'y'	Yellow
'k'	Black
'w'	White

2.3 Marker Size

ms to set the size of the markers.

Listing 4: Insert code directly in your document

```
plt.plot(ypoints, marker = 'o', ms = 20)
```

2.4 Marker Color

markeredgecolor = mec, markerfacecolor = mfc, markersize= ms . You can also use Hexadecimal color values.

Listing 5: Insert code directly in your document

```
plt.plot(ypoints, marker = 'o', ms = 20,
         mec = '#4CAF50', mfc = '#4CAF50')
```

Or any of the 140 supported color names.

Listing 6: Insert code directly in your document

```
plt.plot(ypoints, marker = 'o',ms = 20,
         mec = 'hotpink', mfc = 'hotpink')
```

2.5 Line Style

linestyle = ls,

Style	Or
'solid'	(default) '-'
'dotted'	':'
'dashed'	'--'
'dashdot'	'-.'
'None'	'' or ' '

2.6 Line Color

Listing 7: Insert code directly in your document

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

ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, color = 'r')
# Hex code can be used here too.
plt.plot(ypoints, c = '#4CAF50')

plt.show()
```

2.7 Line Width

keyword argument **linewidth** or the shorter **lw** Plot with a 20.5pt wide line:

Listing 8: Insert code directly in your document

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

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, linewidth = '20.5')
plt.show()
```

2.8 Multiline Plot

Listing 9: Insert code directly in your document

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

y1 = np.array([3, 8, 1, 10])
y2 = np.array([6, 2, 7, 11])
# x-axis is given default value 0,1,2 3
plt.plot(y1)
plt.plot(y2)

plt.show()
```

Listing 10: Insert code directly in your document

```
# This and the above example produce the
# same result
import matplotlib.pyplot as plt
import numpy as np

x1 = np.array([0, 1, 2, 3])
y1 = np.array([3, 8, 1, 10])
x2 = np.array([0, 1, 2, 3])
y2 = np.array([6, 2, 7, 11])
# Here x, y come in pain
plt.plot(x1, y1, x2, y2)
plt.show()
```

3 Label and Title

Listing 11: Insert code directly in your document

```
plt.plot(x, y)
plt.title(”Sports␣Watch␣Data”)
plt.xlabel(”Average␣Pulse”)
plt.ylabel(”Calorie␣Burnage”)
plt.show()
```

3.1 Font for Title and Label

Listing 12: Insert code directly in your document

```
font1 = {'family':'serif','color':'blue',
         'size':20}
font2 = {'family':'serif','color':'darkred',
         'size':15}

plt.title(”Sports␣Watch␣Data”, fontdict = font1)
plt.xlabel(”Average␣Pulse”, fontdict = font2)
plt.ylabel(”Calorie␣Burnage”, fontdict = font2)
```

3.2 Title Position

Listing 13: Insert code directly in your document

```
plt.title(”Sports␣Watch␣Data”, loc = 'left')
```

4 Adding Grid Lines

Listing 14: Insert code directly in your document

```
# Prints both x, y grid lines in graph
plt.grid()
# Prints x only grid lines perpendicular
# to x−axis
plt.grid(axis = 'x')
```

4.1 Set Line Properties for grid

Listing 15: Insert code directly in your document

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

5 Subplot

draw multiple graphs in single figure so that we can save in single picture for future reference.

Listing 16: Insert code directly in your document

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

#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

#(no. of rows, no. of col,current plot index)
plt.subplot(1, 2, 1)
plt.title(”SALES”) # title of plot 1
plt.plot(x,y)

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(1, 2, 2)
plt.plot(x,y)
plt.title(”INCOME”) # Title of plot 2
plt.show()
```

6 Scatter

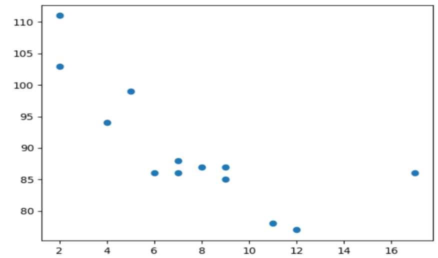
scatter() function to draw a scatter plot. It plots one dot for each observation.It needs two arrays of the same length, one for the values of the x-axis, and one for values on the y-axis.

Listing 17: Insert code directly in your document

```
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()
```



6.1 Color

You can set your own color for each scatter plot with the color or the c argument, **Note:** The two plots are plotted with two different colors, by default blue and orange, you will learn how to change colors later in this chapter.

Listing 18: Insert code directly in your document

```
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()
```

6.2 Color Each Dot

Note: You cannot use the color argument for this, only the c argument.You can even set a specific color for each dot by using an array of colors as value for the c argument:

Listing 19: Insert code directly in your document

```
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()
```

6.3 ColorMap

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

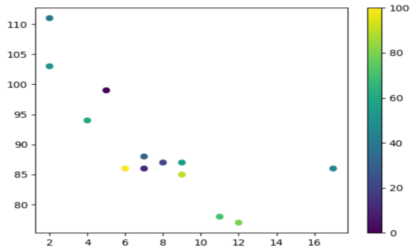
Listing 20: Insert code directly in your document

```
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()
```

You can include the colormap in the drawing by including the **plt.colorbar()** statement:



6.3.1 ColorMap

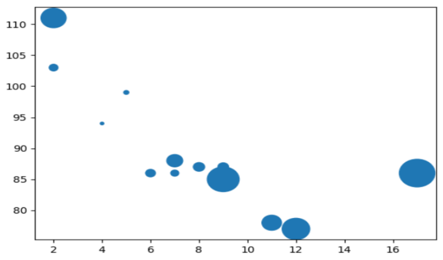
Below are the list of color maps available.
Accent, Accent_r, Blues , Blues_r, BrBG , BrBG_r, BuGn , BuGn_r, BuPu , BuPu_r, CMRmap , CMRmap_r, Dark2 , Dark2_r, GnBu , GnBu_r, Greens , Greens_r, Greys , Greys_r, OrRd , OrRd_r, Oranges , Oranges_r, PRGn , PRGn_r, Paired , Paired_r, Pastel1 , Pastel1_r, 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 , W,istia_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

6.3.2 Size

Listing 21: Insert code directly in your document

```
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()
```



6.3.3 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:

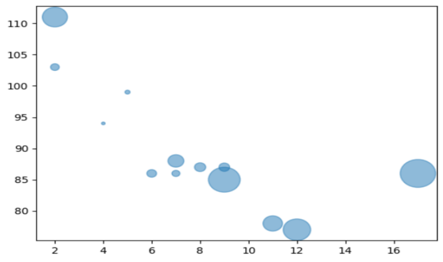
Listing 22: Insert code directly in your document

```
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()
```



6.3.4 Combine Color Size and Alpha

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

Listing 23: Insert code directly in your document

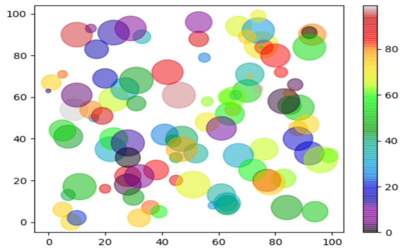
```
import matplotlib.pyplot as plt
import numpy as np

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()
```



7 Bar Chart

Listing 24: Insert code directly in your document

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

x = np.array(["A","B","C","D"])
y = np.array([3,8,1,10])

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

7.1 Horizontal Bar chart

Listing 25: Insert code directly in your document

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

x = np.array(["A","B","C","D"])
y = np.array([3,4,5,6])

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

7.2 Bar Color

Listing 26: Insert code directly in your document

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

x = np.array(["A","B","C","D"])
y = np.array([3,4,5,6])

plt.bar(x,y,color = "red")
// plt.bar(x,y,color="#4CAF50")
plt.show()
```

7.3 Bar Width and Height

Default bar width is 0.8 which is equal to 80

Listing 27: Insert code directly in your document

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

x = np.array(["A","B","C","D"])
y = np.array([3,4,5,6])

plt.bar(x,y,width=0.1)
# plt.barh(x,y, height=0.1)
# height parameter can only be used with plt.barh()
plt.show()
```

8 Histogram

Histogram is graph showing frequency distribution. Observations between interval. We use hist().

Listing 28: Insert code directly in your document

```
import matplotlib.pyplot as plt
import numpy as np
# generate normal distributed data set with
# 250 values and concentrate at 170 and 10 is
# the standard distribution.
x = np.random.normal(170,10,250)
print(x)

plt.hist(x)
plt.show()
```

9 Pie Charts

Listing 29: Insert code directly in your document

```
import matplotlib.pyplot as plt
import numpy as np
y = np.array([35,25,25,15])
plt.pie(y)
plt.show()
```

Each piece out of 5 is called wedge, by default first wedge starts from x-axis.

9.1 Label the wedges

Listing 30: Insert code directly in your document

```
import myplotlib.pyplot as plt
import numpy as np

y = np.array([35,25,25,15])
mylabes = ["Apples", "Bananas", "Cherries",
 "Dates"]

plt.pie(y,labels = mylabel)
plt.show()
```

9.2 Start Angle

Default angle is 0 degree from x-axis.

Listing 31: Insert code directly in your document

```

import myplotlib.pyplot as plt
import numpy as np

y = np.array([35,25,25,15])
mylabes = ["Apples", "Bananas", "Cherries",
 "Dates"]

plt.pie(y,labels = mylabel,startangle=90)
plt.show()
```

9.3 Explode

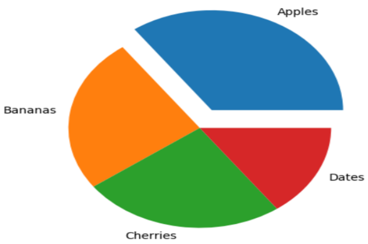
If we want one of the sections to stand out. It must be an list with one value each for wedge.

Listing 32: Insert code directly in your document

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

y = np.array([35,25,25,15])
mylabes = ["Apples", "Bananas", "Cherries",
         "Dates"]
myexplode = [0.2,0,0,0]

plt.pie(y, labels = mylabel, startangle = 90,
        explode = myexplode, shadow = True)
# Here Shadow parameter will create a shadow
# for the pie chart
plt.show()
```



9.4 Color

Listing 33: Insert code directly in your document

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

y =  np.array([35,25,25,15])
mylabels = ["Apple", "Bananas", "Cherries", "Dates"]
mycolors = ["black", "hotpink", "b", "#4CAF50"]

plt.pie(y, labels = mylabels, colors =
mycolors)
plt.show()
```

You can use one of the shortcuts: ’r’ : red, ’g’: Green, ’b’: Blue, ’c’: Cyan, ’m’: Magneta, ’y’: Yellow, ’k’: Black, ’w’: White

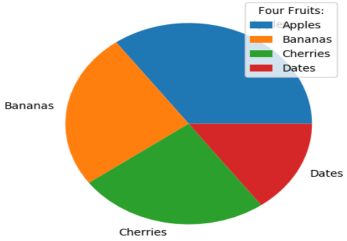
9.5 Legend with header

Listing 34: Insert code directly in your document

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

y =  np.array([35,25,25,15])
mylabels = ["Apple", "Bananas", "Cherries", "Dates"]
mycolors = ["black", "hotpink", "b", "#4CAF50"]

plt.pie(y, labels = mylabels, colors =
mycolors)
plt.legend(title="Four␣Fruits:")
plt.show()
```


Four Fruits:
Apples
Bananas
Cherries
Dates
Bananas
Dates
Cherries