Queen Mary
University of London

QHP4701
Introduction to Data Science Programming

Visualisation with Matplotlib

Lecturer: Nikesh Bajaj, PhD
School of Physical and Chemical Sciences
http://nikeshbajaj.in

Lecture Outline

Visualisation: Introduction to Matplotlib

- Basic Plots: Line Plot: Labels, title, colour, Legend
- Scatter, Bar, Stem
- Matrix, Heatmap, Colormap, Image
- Figure Size and Multiple plots: Subplots
- Statistics: Histogram, Pie-Chart
- Decorating plots with labels, title, colour, markers, texts
- More on Figures and Plots

Ref: Python Data Science Handbook, 2nd Edition, Chapter 4

Link: https://jakevdp.github.io/PythonDataScienceHandbook/

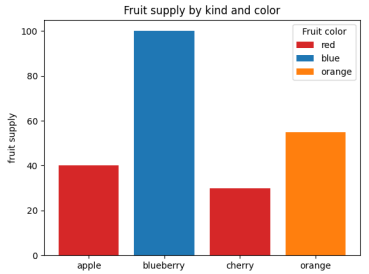
Matplotlib: Visualisation Library

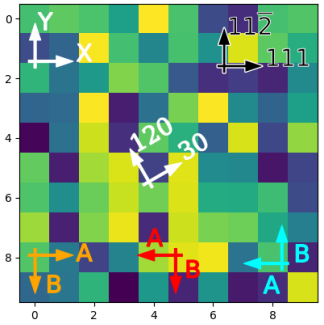
- Matplotlib is a python library to plot and visualise data. It is widely used library that supports many of the data-types to visualise them.
- Among others, Anaconda comes with matplotlib.
- To use matplotlib, first we need to import it as

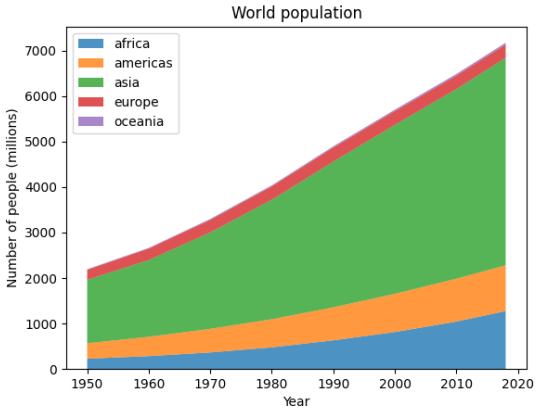
```
import matplotlib.pyplot as plt
```

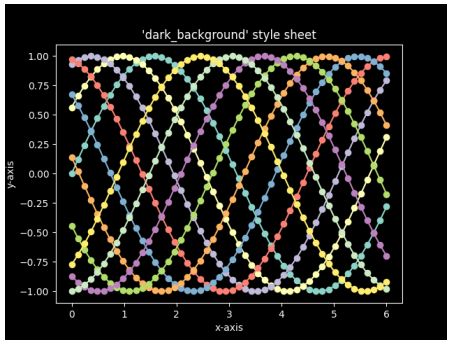
Matplotlib: Examples

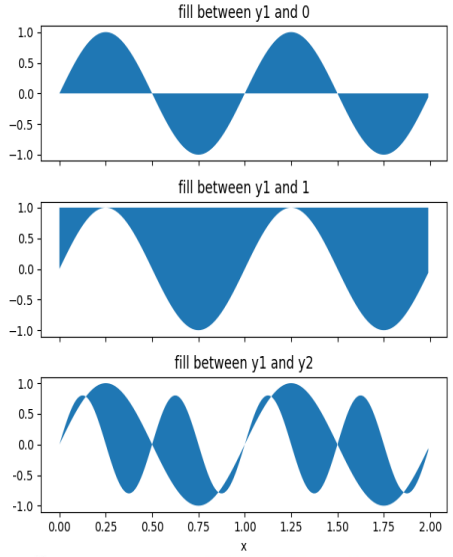
- Matplotlib allows you to visualise data in many different ways. Here are a few examples

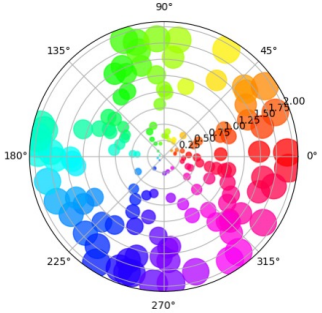


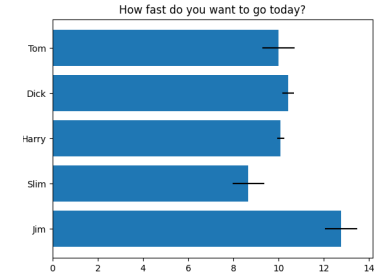


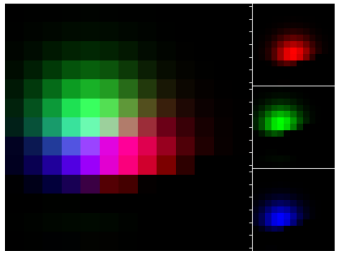


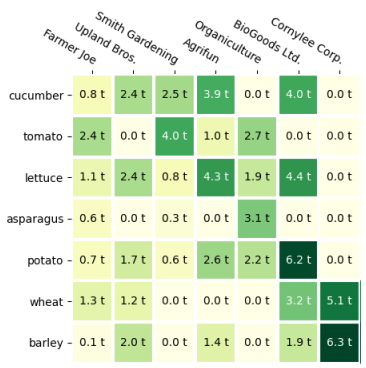


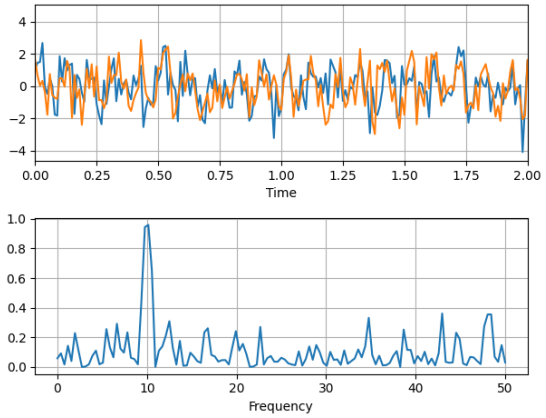


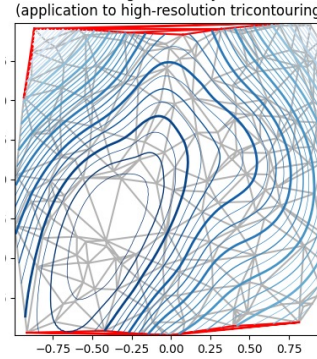


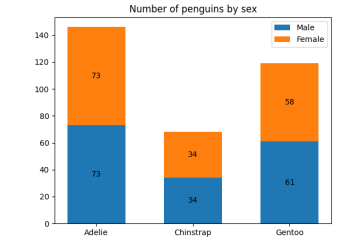


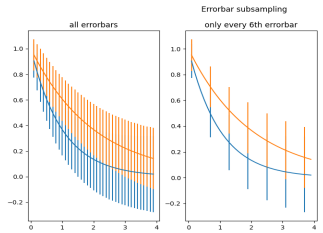


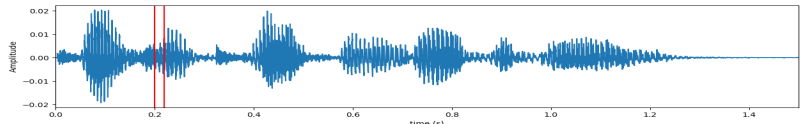


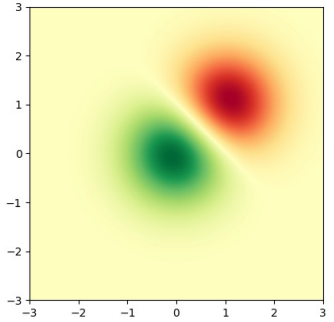










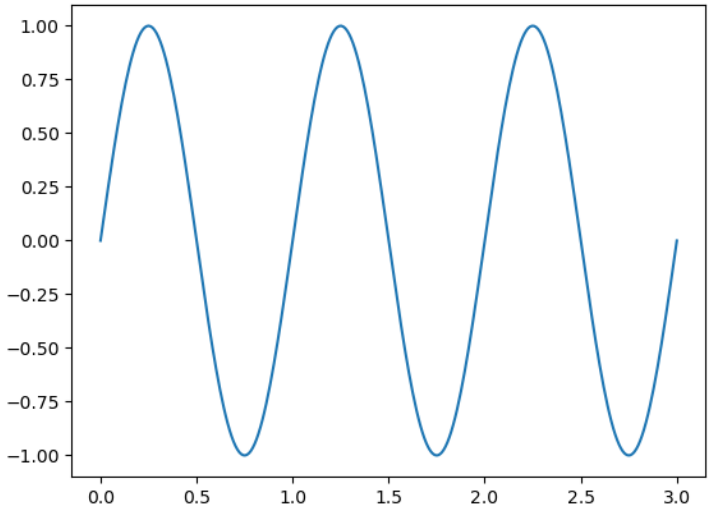


Line plot

- Simple line plot is a plot connecting the points represented by given variable.

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

Example: 3 sec sine wave of 1 Hz

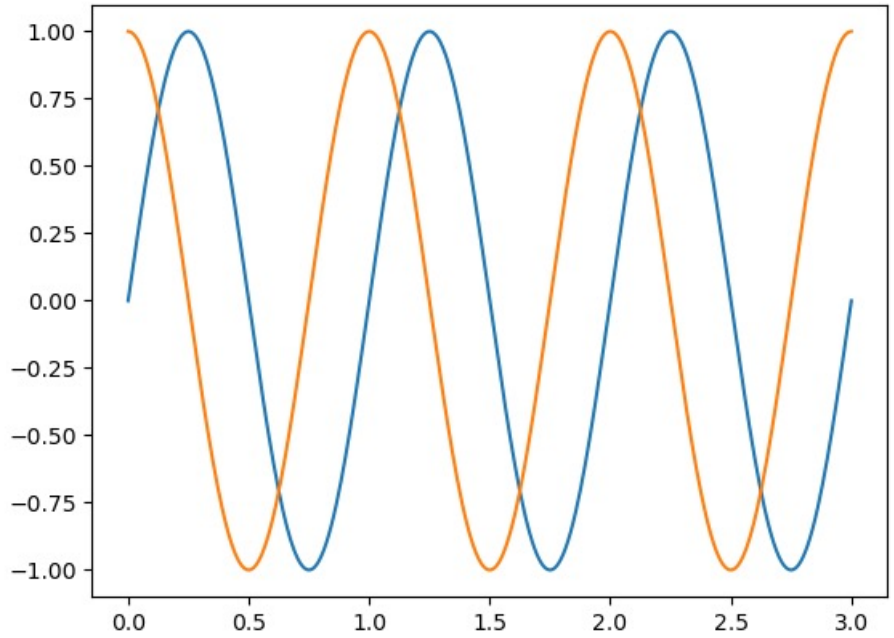



```
import matplotlib.pyplot as plt
import numpy as np


t = np.linspace(0, 3, 300)
f = 1
x = np.sin(2*np.pi*f*t)
plt.figure()
plt.plot(t,x)
plt.show()
```

Line plot

- Multiple line plots.

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


t = np.linspace(0, 3, 300)
f = 1
x = np.sin(2*np.pi*f*t)
y = np.cos(2*np.pi*f*t)
plt.figure()
plt.plot(t,x)
plt.plot(t,y)
plt.show()
```

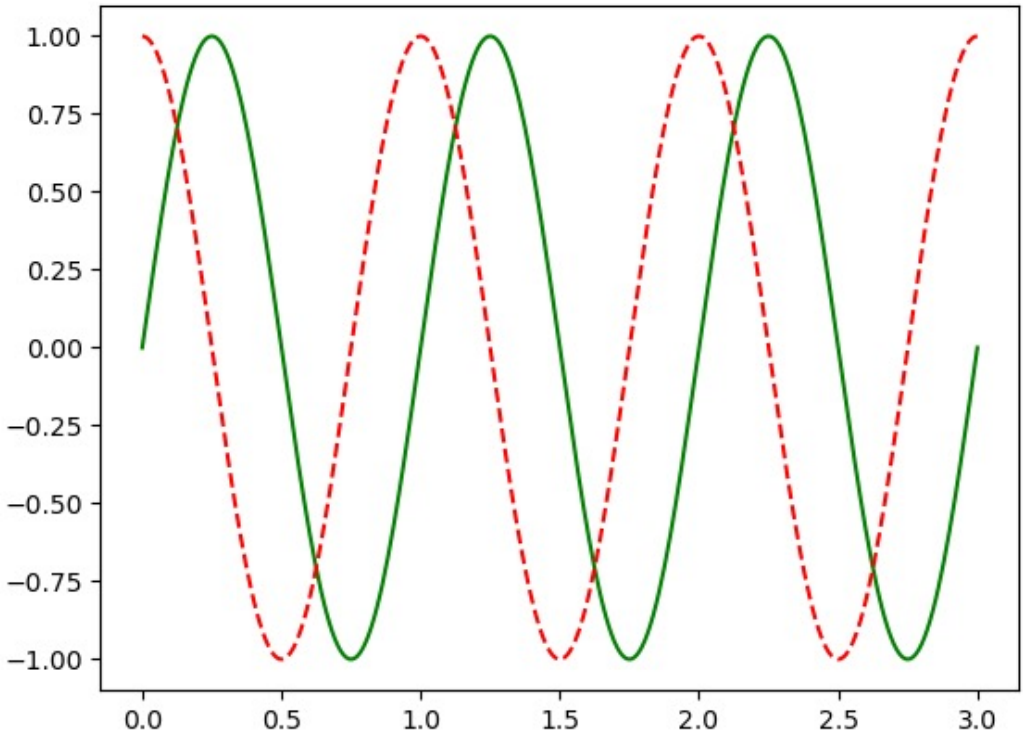


Line plot

- Line Colour: color = ‘g’
- Line Type: ls = ‘--’, ‘-’, ‘:’, ‘-.’

- 'b' as blue
- 'g' as green
- 'r' as red
- 'c' as cyan
- 'm' as magenta
- 'y' as yellow
- 'k' as black
- 'w' as white

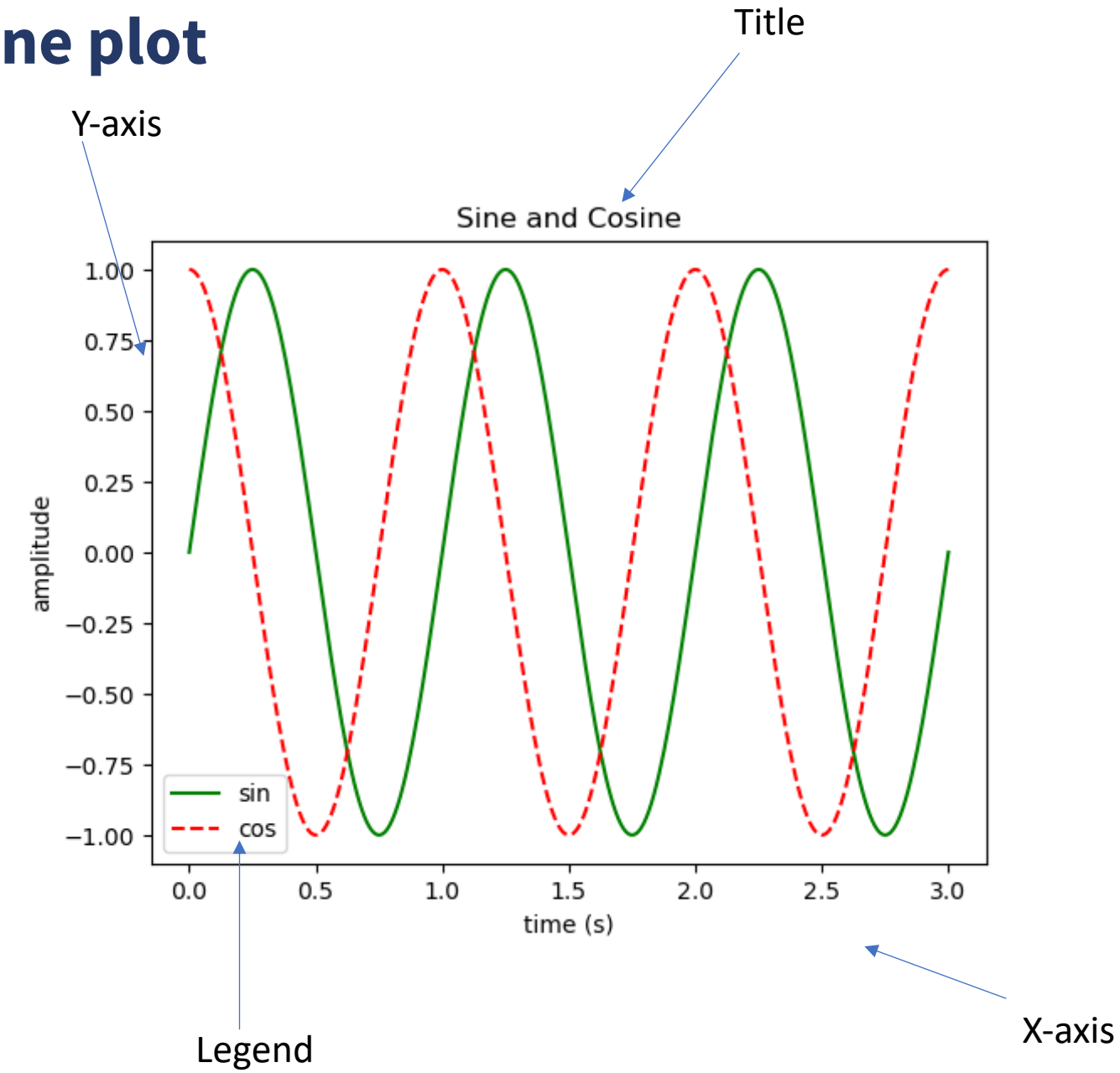
```
t = np.linspace(0, 3, 300)
f = 1
x = np.sin(2*np.pi*f*t)
y = np.cos(2*np.pi*f*t)
plt.figure()
plt.plot(t,x, color='g')
plt.plot(t,y, color='r', ls='--')
plt.show()
```



Line plot

- Axis Labels, Title, Legend

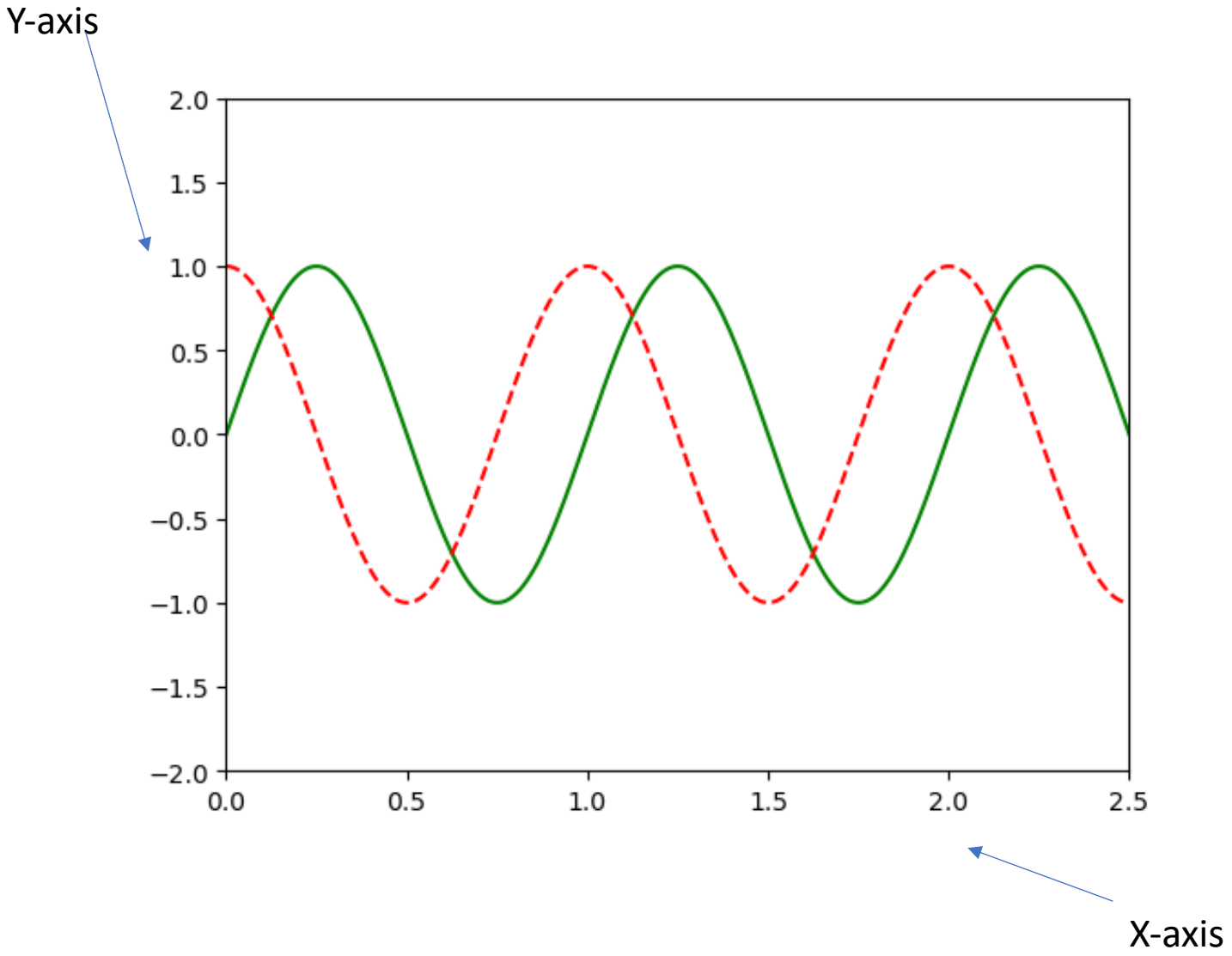
```
plt.figure()
plt.plot(t,x, color='g',label='sin')
plt.plot(t,y, color='r',ls='--',label='cos')
plt.xlabel('time (s)')
plt.ylabel('amplitude')
plt.title('Sine and Cosine')
plt.legend()
plt.show()
```



Line plot

- Limit X, Y axes

```
t = np.linspace(0, 3, 300)
f = 1
x = np.sin(2*np.pi*f*t)
y = np.cos(2*np.pi*f*t)
plt.plot(t,x, color='g')
plt.plot(t,y,color='r', ls='--')
plt.xlim([0,2.5])
plt.ylim([-2,2])
plt.show()
```



Lecture Outline

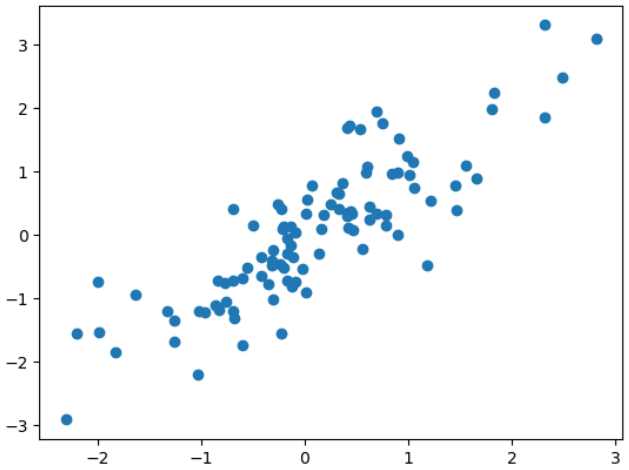
Visualisation: Introduction to Matplotlib

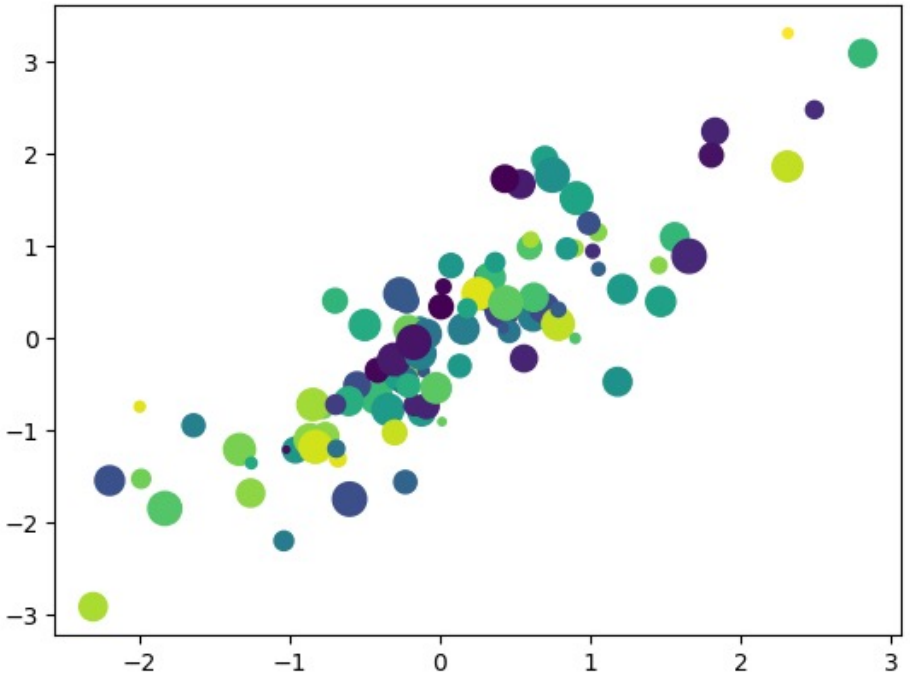
- Basic Plots: Line Plot: Labels, title, colour, Legend
- Scatter, Bar, Stem
- Matrix, Heatmap, Colormap, Image
- Figure Size and Multiple plots: Subplots
- Statistics: Histogram, Pie-Chart
- Decorating plots with labels, title, colour, markers, texts
- More on Figures and Plots

Scatter plot

- Plot X-Y Scatter

```
x = np.random.randn(100)
y = x + 0.5*np.random.randn(100)
color = np.random.rand(100)
size = np.random.rand(100)*200

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

plt.figure()
plt.scatter(x,y, c=color, s=size)
plt.show()
```



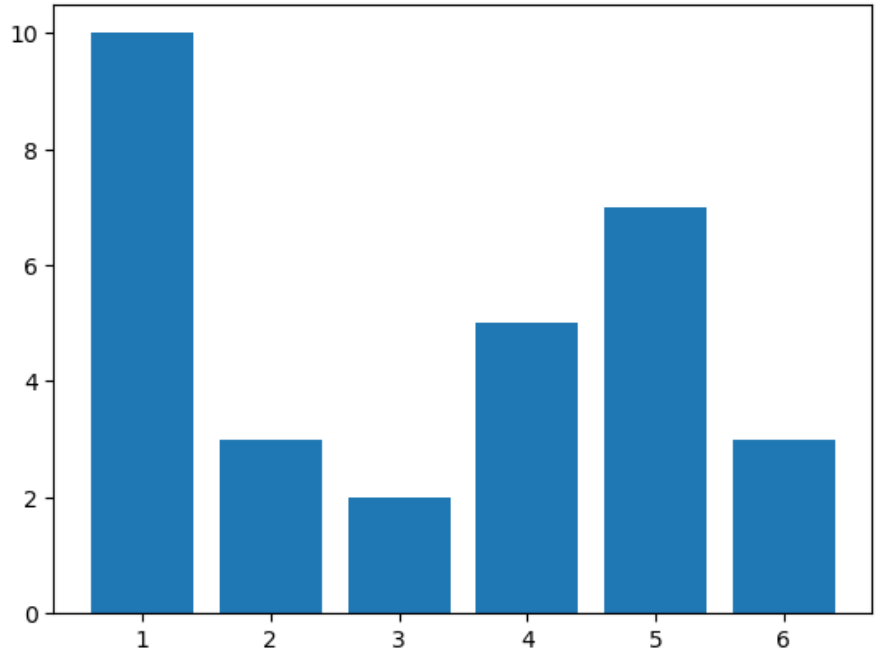


Bar plot

- Bar Plot

```
y = [10,3,2,5,7,3]
x = [1,2,3,4,5,6]
```

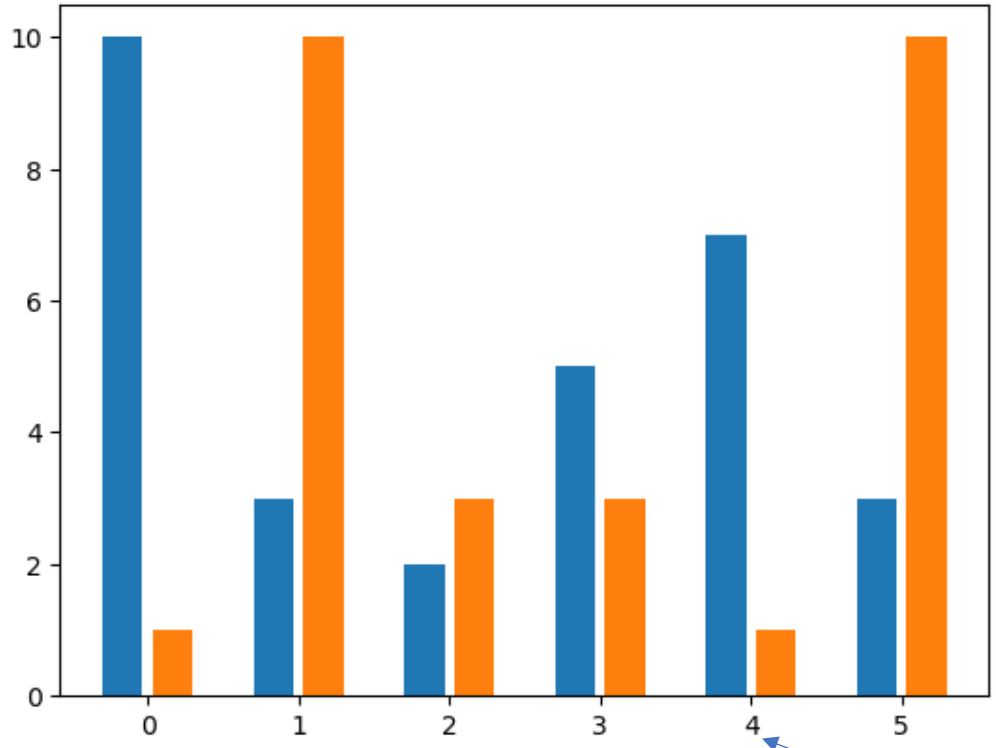
```
plt.figure()
plt.bar(x,y)
plt.show()
```



Bar plot

- Bar Plot

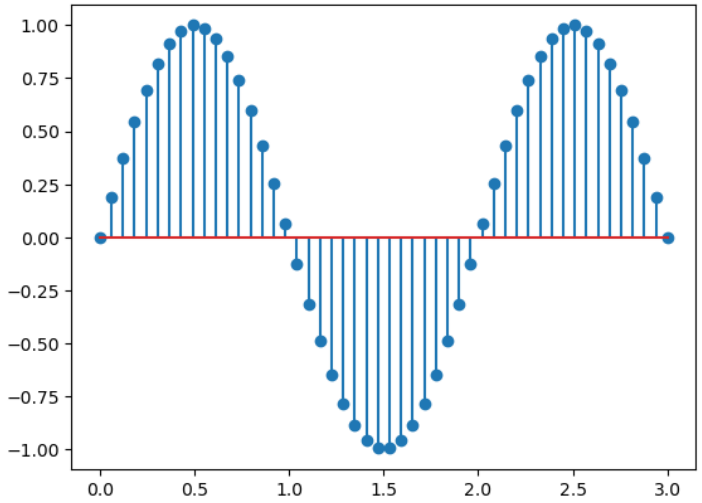
```
x1 = [1,4,7,10,13,16]
y1 = [10,3,2,5,7,3]


x2 = [2,5,8,11,14,17]
y2 = [1,10,3,3,1,10]


x  = np.array(x1)+0.5
x_label = np.arange(len(x))


plt.figure()
plt.bar(x1,y1)
plt.bar(x2,y2)
plt.xticks(x, x_label)
plt.show()
```

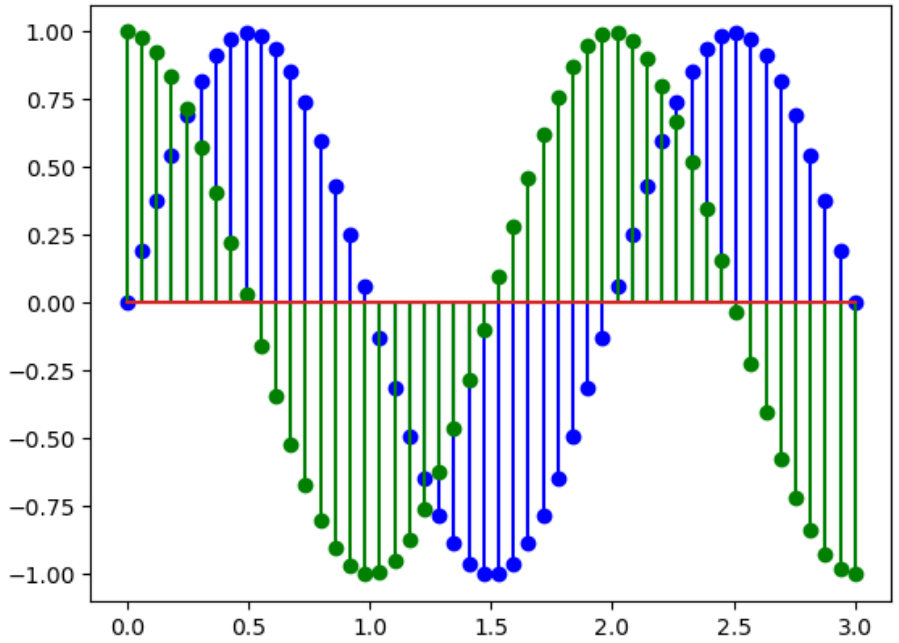


Stem plot

- Stem Plot

```
t = np.linspace(0, 3, 50)
f = 0.5
x = np.sin(2*np.pi*f*t)
y = np.cos(2*np.pi*f*t)

plt.figure()
plt.stem(t,x)
plt.show()

plt.figure()
plt.stem(t,x,linefmt ='b')
plt.stem(t,y,linefmt ='g')
plt.show()
```





Lecture Outline

Visualisation: Introduction to Matplotlib

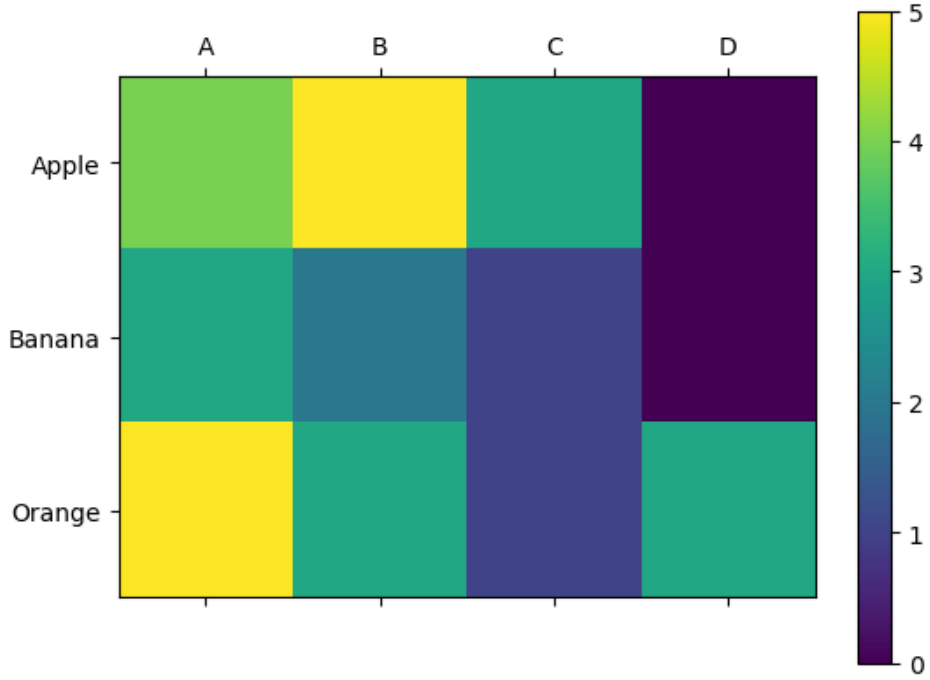
- Basic Plots: Line Plot: Labels, title, colour, Legend
- Scatter, Bar, Stem
- Matrix, Heatmap, Colormap, Image
- Figure Size and Multiple plots: Subplots
- Statistics: Histogram, Pie-Chart
- Decorating plots with labels, title, colour, markers, texts
- More on Figures and Plots

Matrix plot/ heatmap

- matshow

```
x = np.array([[4,5,3,0],[3,2,1,0],[5,3,1,3]])
```

```
plt.matshow(x)
plt.xticks(range(4), ['A','B','C','D'])
plt.yticks(range(3), ['Apple','Banana','Orange'])
#plt.xticks(x, x_label)
plt.colorbar()
plt.show()
```

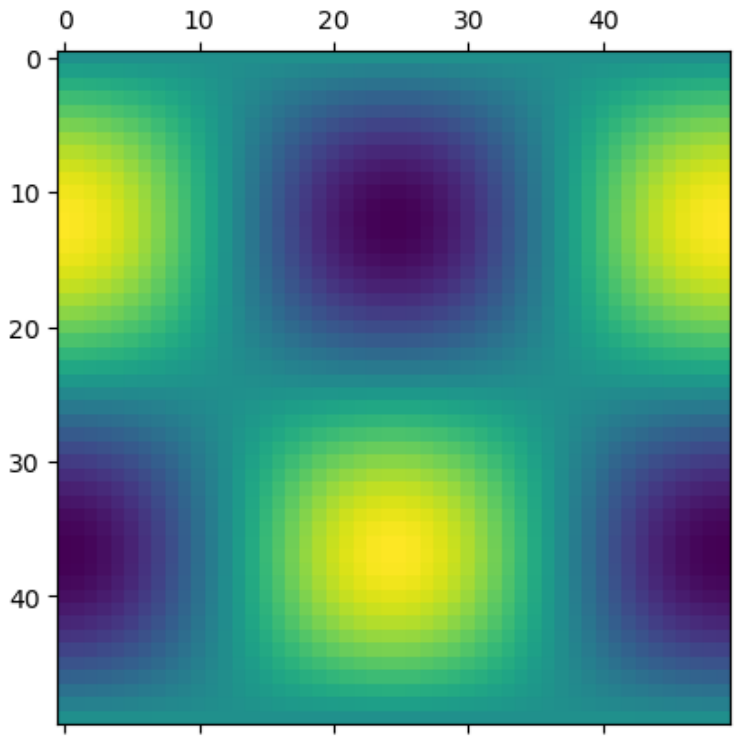


Heatmap, colormap

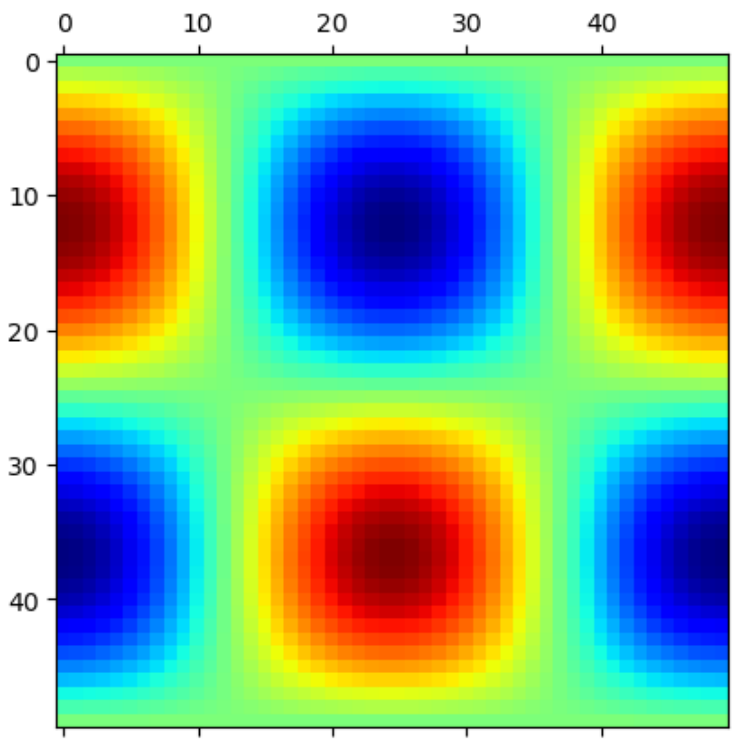
- colormap

```
t = np.linspace(0, 2, 50)
f = 0.5
x = np.sin(2*np.pi*f*t)
y = np.cos(2*np.pi*f*t)
z = x[:,None]@y[None,:]
```

```
plt.matshow(z)
plt.show()
```




```
plt.matshow(z, cmap='jet')
plt.show()
```

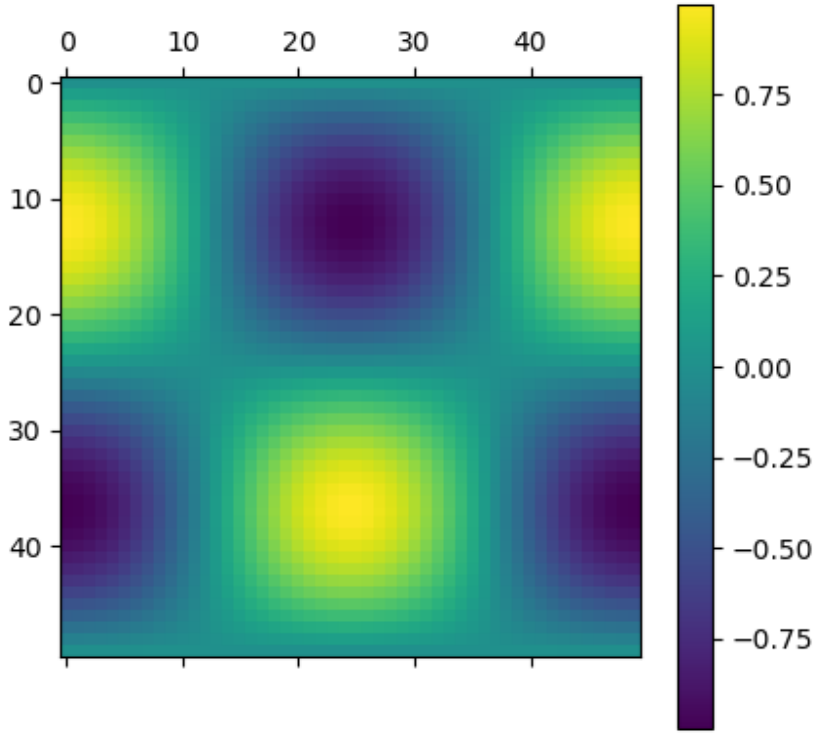


Heatmap, colormap

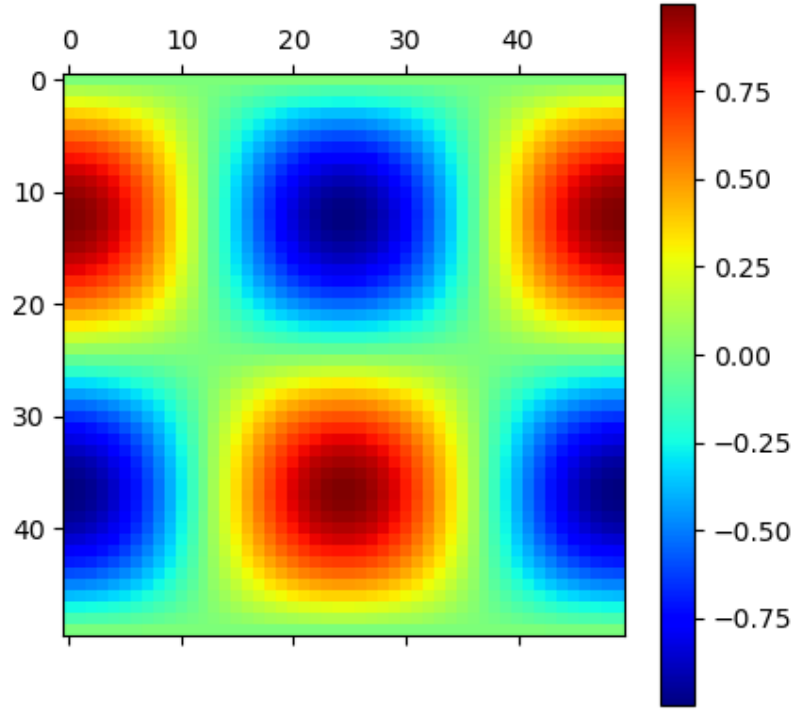
- colormap

```
t = np.linspace(0, 2, 50)
f = 0.5
x = np.sin(2*np.pi*f*t)
y = np.cos(2*np.pi*f*t)
z = x[:,None]@y[None,:]
```

```
plt.matshow(z)
plt.colorbar()
plt.show()
```

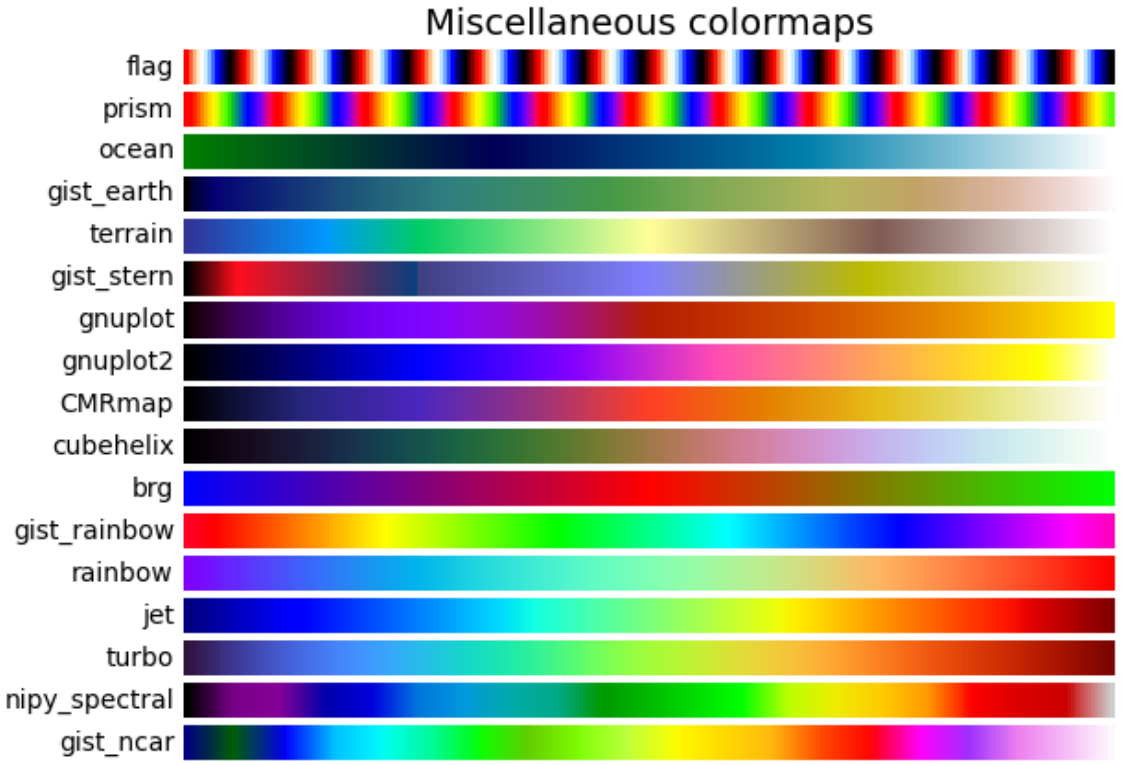


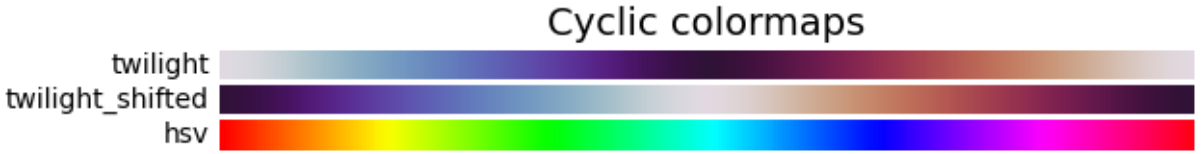

```
plt.matshow(z, cmap='jet’)
plt.colorbar()
plt.show()
```

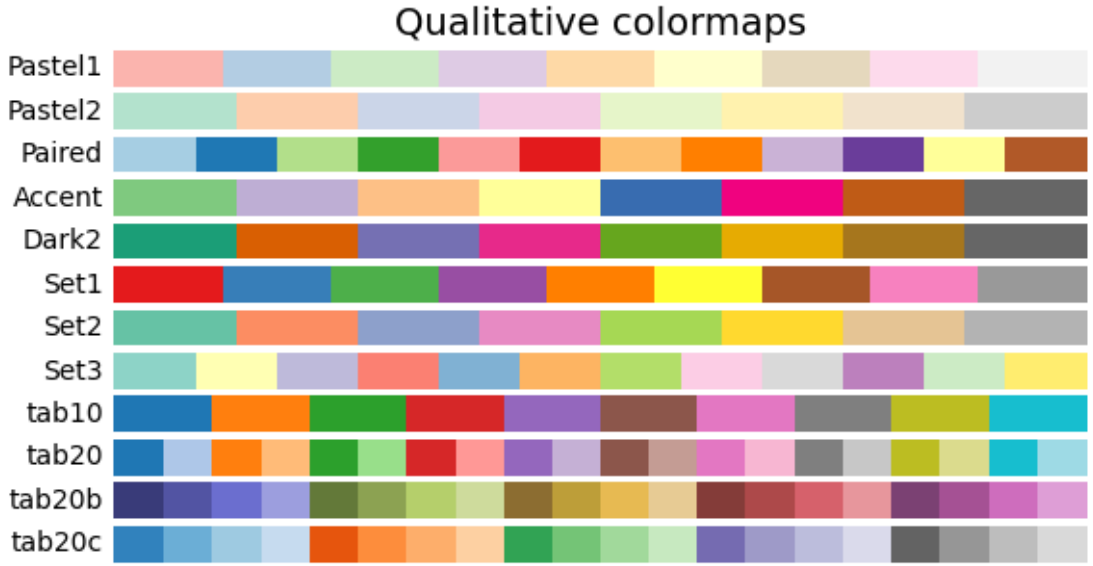


Colormaps

- Colormaps
- For more colormaps, check on: https://matplotlib.org/stable/tutorials/colors/colormaps.html







Image

- Image

```
#Read Image


file = 'china_wall.jpg'
I = plt.imread(file)


#Show Image
plt.figure()
plt.imshow(I)
plt.show()


plt.figure()
plt.imshow(I)
plt.axis('off')
plt.show()
```

Lecture Outline

Visualisation: Introduction to Matplotlib

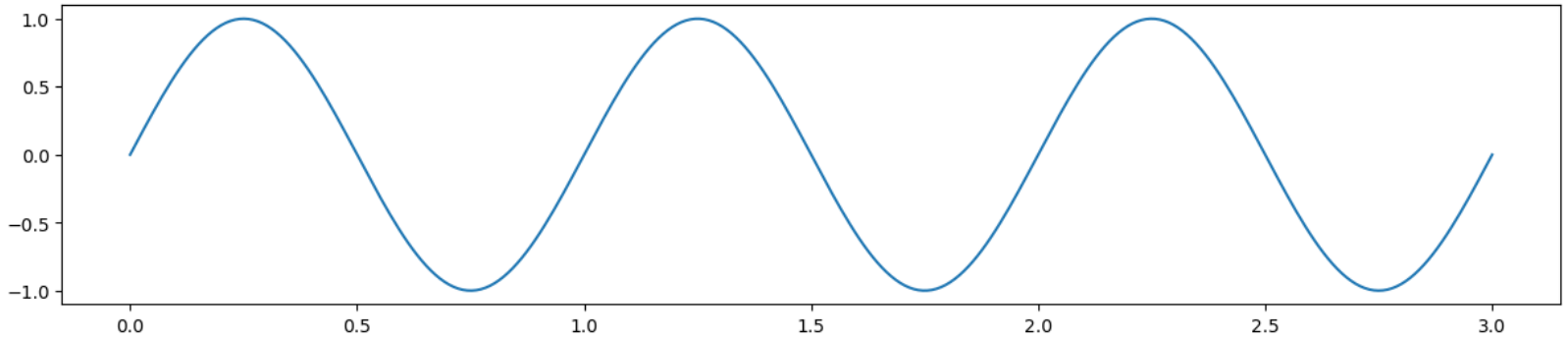
- Basic Plots: Line Plot: Labels, title, colour, Legend
- Scatter, Bar, Stem
- Matrix, Heatmap, Colormap, Image
- Figure Size and Multiple plots: Subplots
- Statistics: Histogram, Pie-Chart
- Decorating plots with labels, title, colour, markers, texts
- More on Figures and Plots

Figure Size

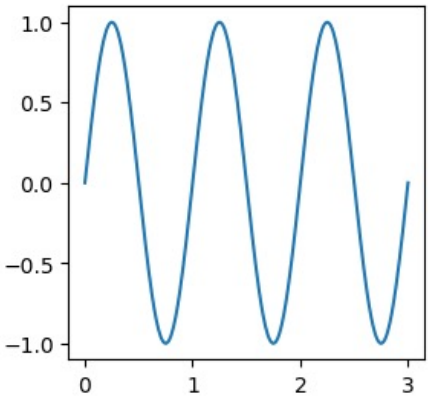
- Figure-size

```
t = np.linspace(0, 3, 300)
f = 1
x = np.sin(2*np.pi*f*t)

plt.figure(figsize=(15,3)
plt.plot(t,x)
plt.show()

plt.figure(figsize=(3,3)
plt.plot(t,x)
plt.show()
```

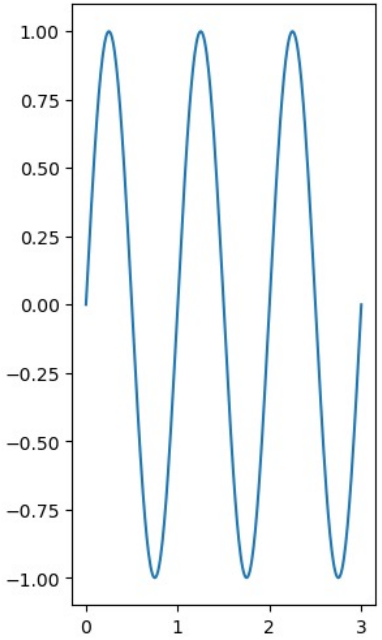
(15, 3)



(3, 3)



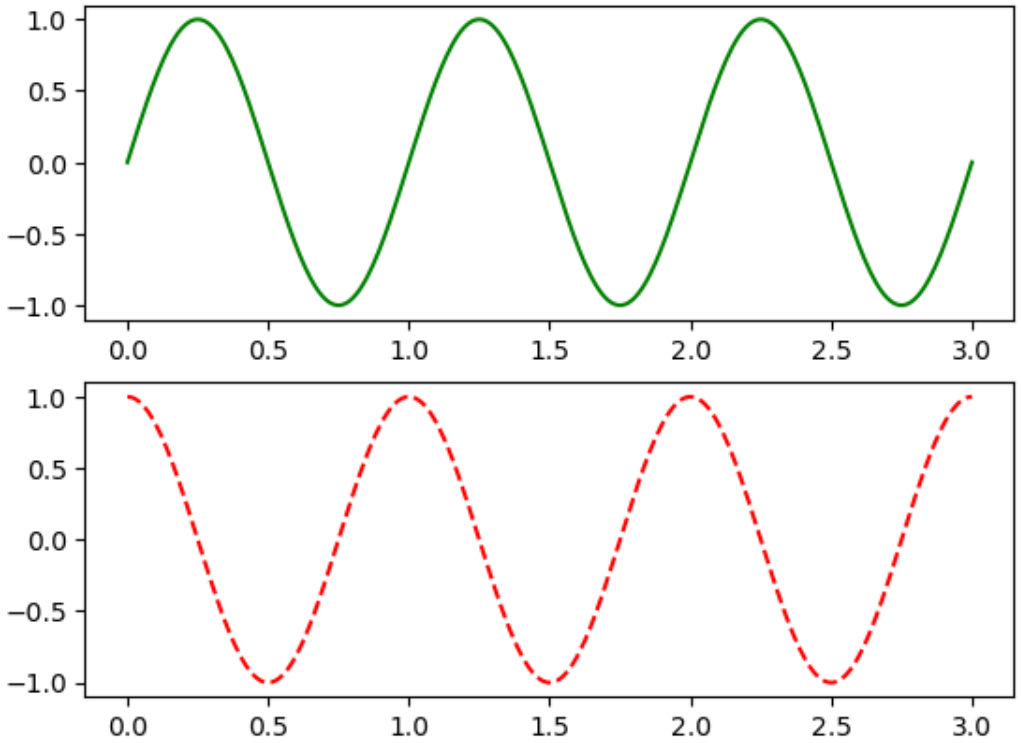
(3, 6)



Multiple plots

- Subplot

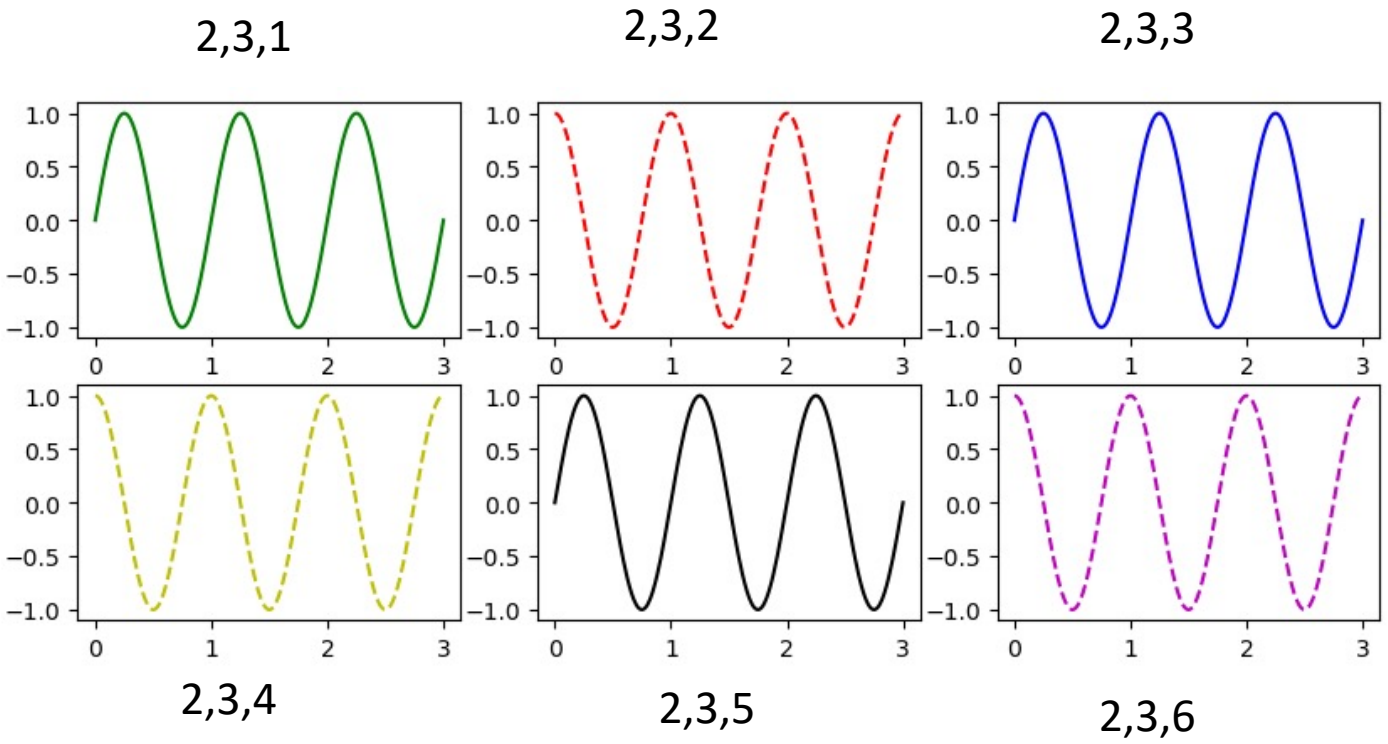
```
t = np.linspace(0, 3, 300)
f = 1
x = np.sin(2*np.pi*f*t)
y = np.cos(2*np.pi*f*t)

plt.figure()
plt.subplot(2,1,1)
plt.plot(t,x, color='g')
plt.subplot(2,1,2)
plt.plot(t,y,color='r', ls='--')
plt.show()
```



Multiple plots

- Subplot (2-rows, 3-columns)

```
plt.figure(figsize=(10,4))
plt.subplot(2,3,1)
plt.plot(t,x, color='g')
plt.subplot(2,3,2)
plt.plot(t,y,color='r', ls='--')
plt.subplot(2,3,3)
plt.plot(t,x, color='b')
plt.subplot(2,3,4)
plt.plot(t,y,color='y', ls='--')
plt.subplot(2,3,5)
plt.plot(t,x, color='k')
plt.subplot(2,3,6)
plt.plot(t,y,color='m', ls='--')
plt.show()
```



Lecture Outline

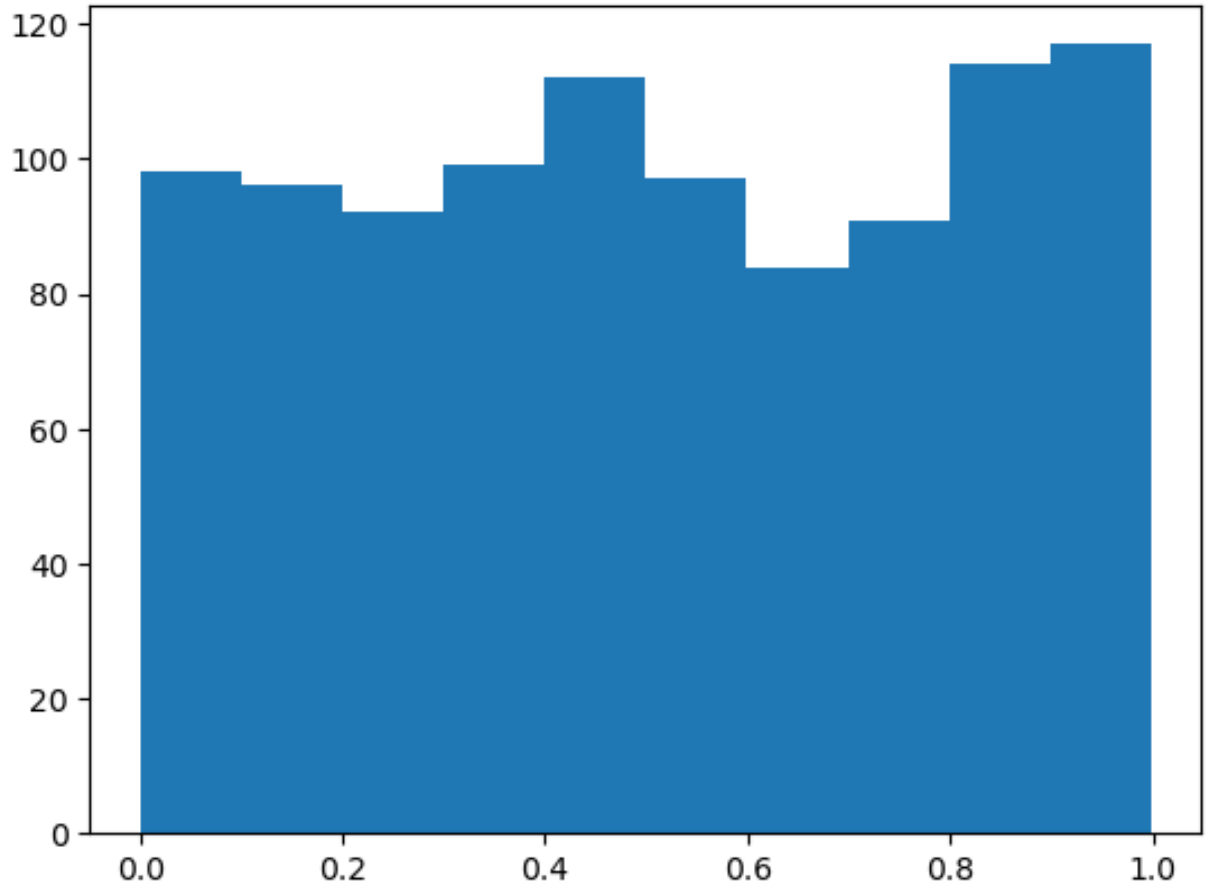
Visualisation: Introduction to Matplotlib

- Basic Plots: Line Plot: Labels, title, colour, Legend
- Scatter, Bar, Stem
- Matrix, Heatmap, Colormap, Image
- Figure Size and Multiple plots: Subplots
- Statistics: Histogram, Pie-Chart
- Decorating plots with labels, title, colour, markers, texts
- More on Figures and Plots

Histogram

- Histogram

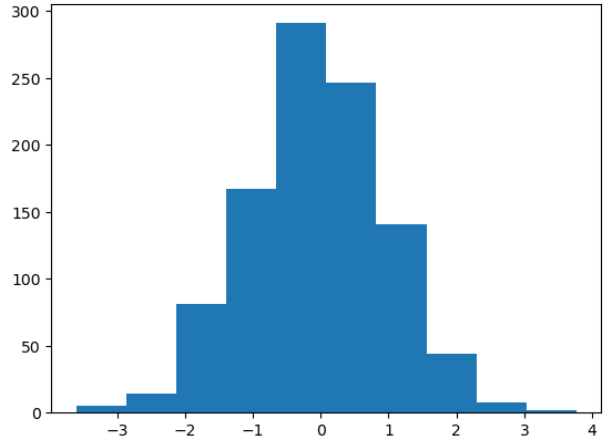
```
#Uniform Distribution
x = np.random.rand(1000)


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

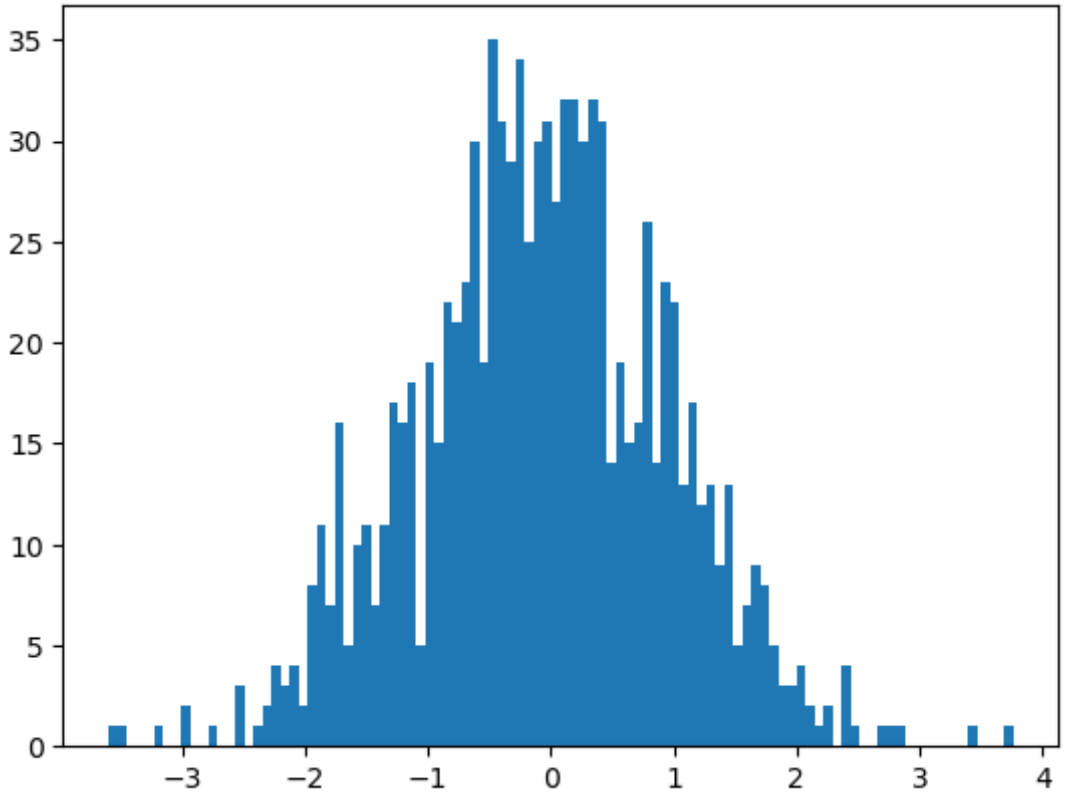


Histogram

- Histogram

```
#Gaussian Distribution
#Normal Distribution
x = np.random.randn(1000)

plt.figure()
plt.hist(x)
plt.show()

plt.figure()
plt.hist(x, bins=100)
plt.show()
```

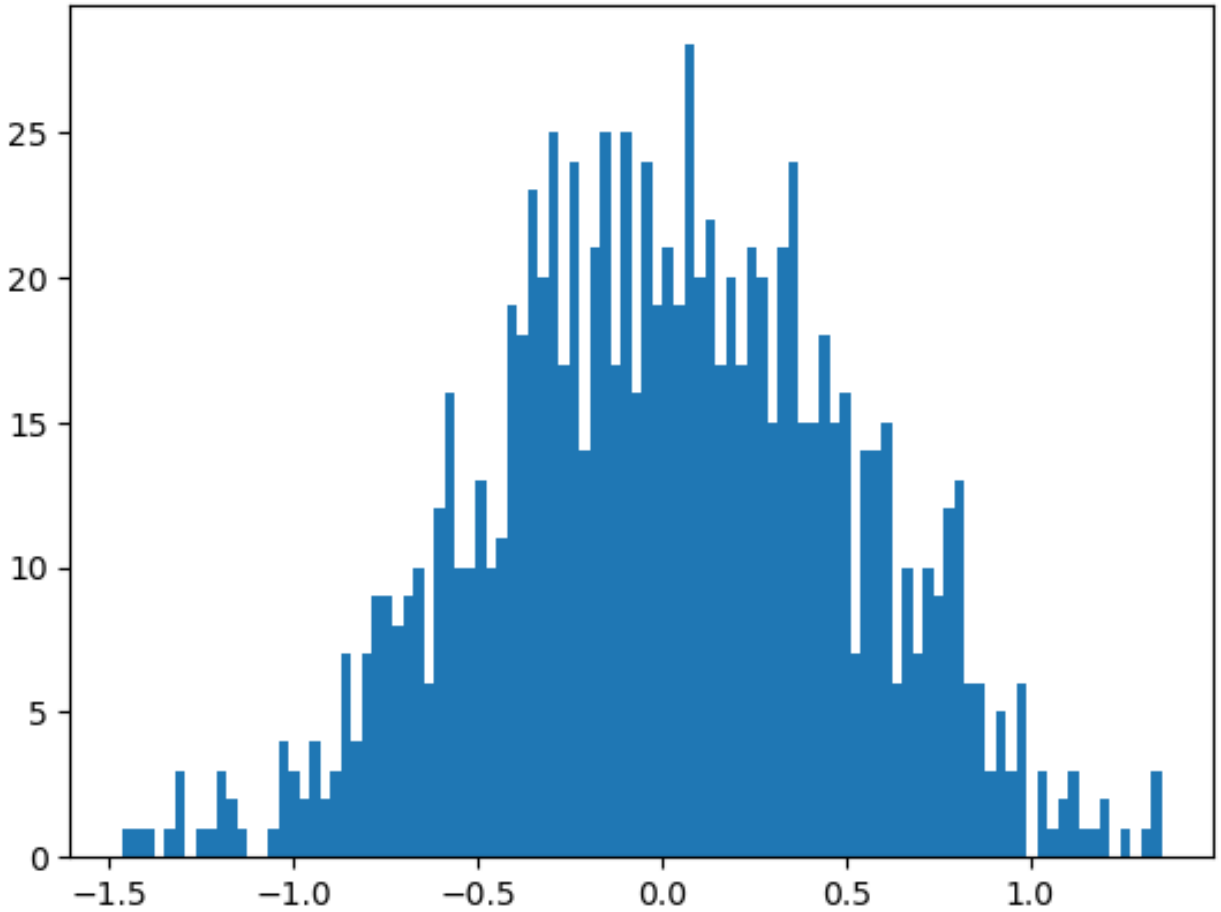




Histogram

- Histogram

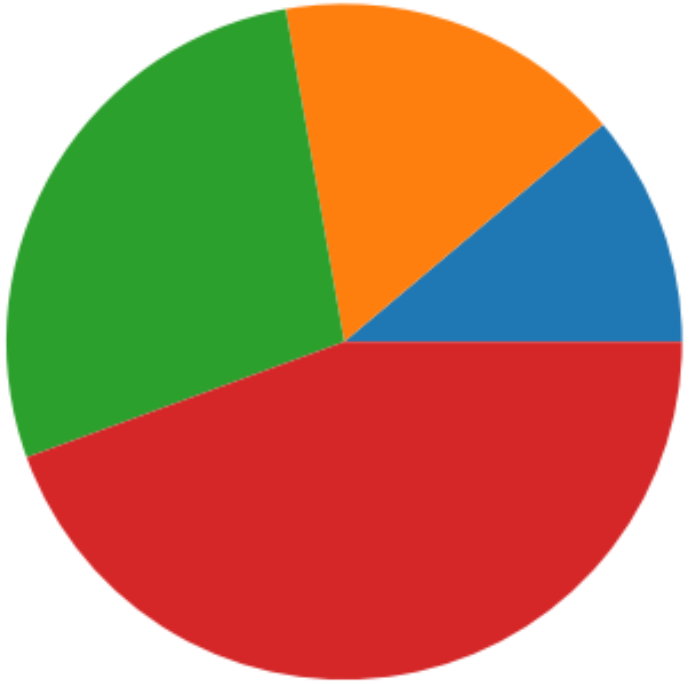
```
#Normal Distribution
x = np.random.randn(1000)*0.5


plt.figure()
plt.hist(x, bins=100)
plt.show()
```



Pie-Chart

- Pie-Chart

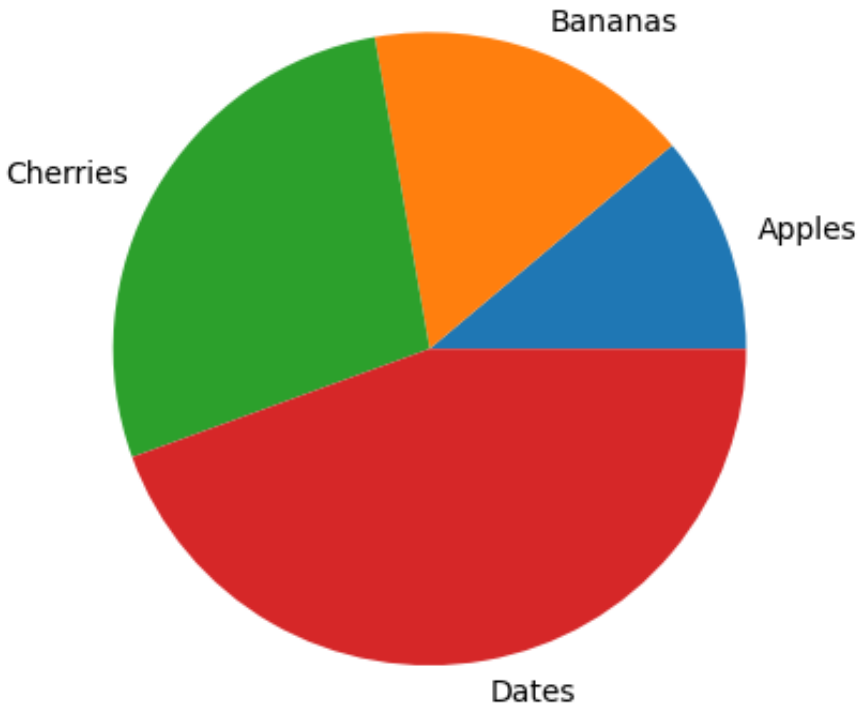
```
y = np.array([10, 15, 25, 40])

plt.pie(y)
plt.show()
```



Pie-Chart

- Pie-Chart

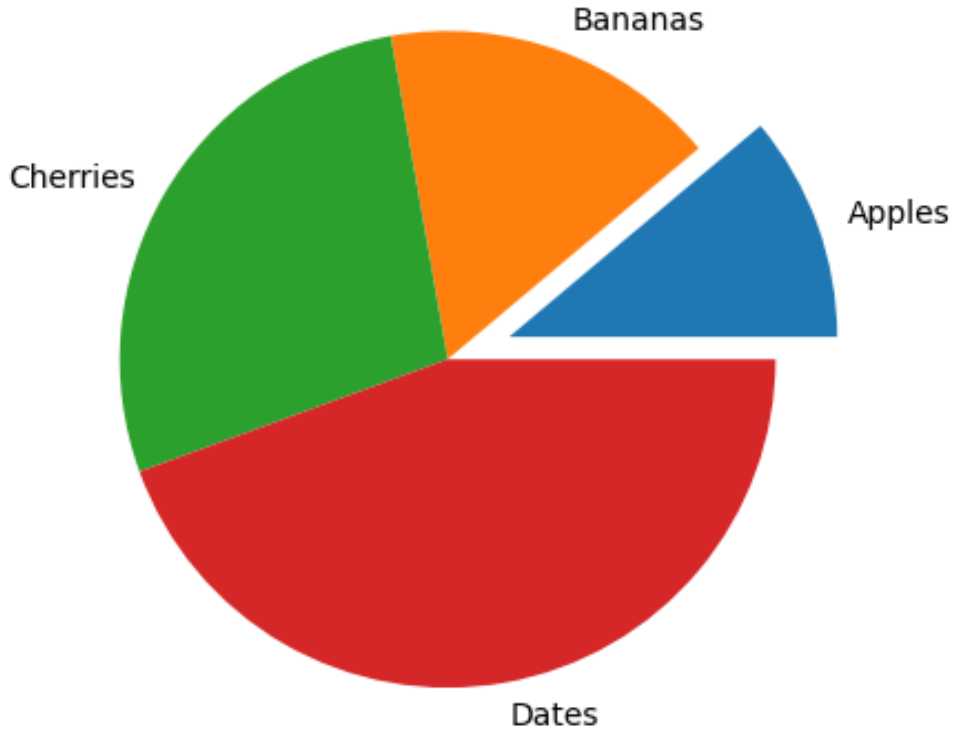
```
y = np.array([10, 15, 25, 40])
mylabels = ["Apples", "Bananas",
"Cherries", "Dates"]

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



Pie-Chart

- Pie-Chart

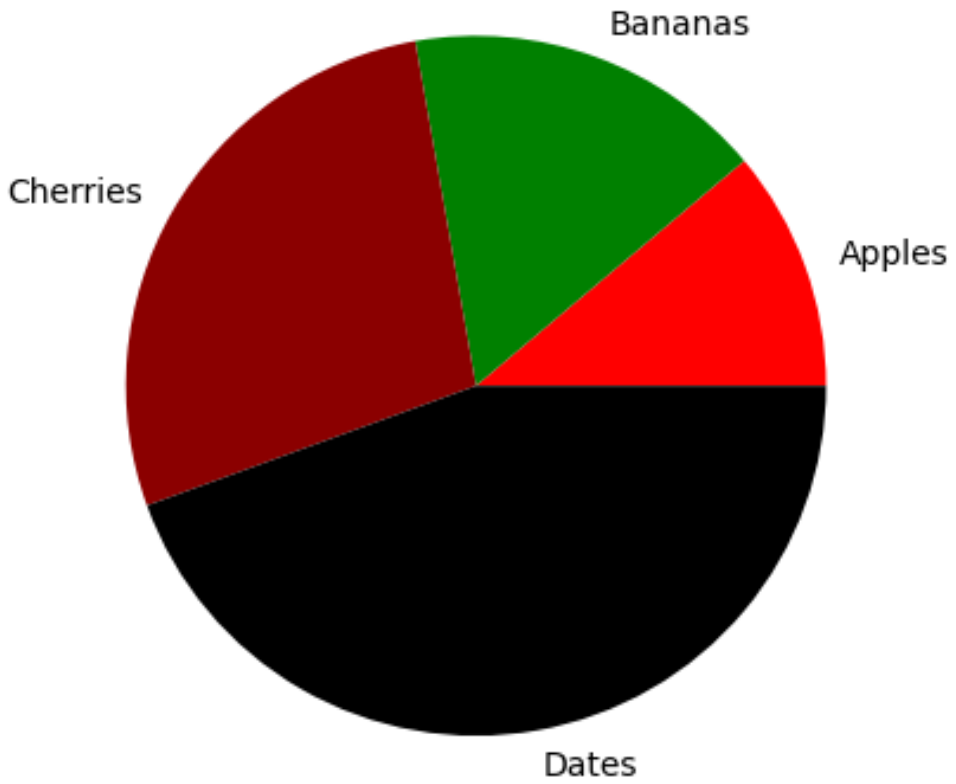
```
y = np.array([10, 15, 25, 40])
mylabels = ["Apples", "Bananas",
"Cherries", "Dates"]
myexplode = [0.2, 0, 0, 0]

plt.pie(y, labels = mylabels,
explode = myexplode)
plt.show()
```



Pie-Chart

- Pie-Chart

```
y = np.array([10, 15, 25, 40])
mylabels = ["Apples", "Bananas",
"Cherries", "Dates"]
mycolors = ["red", "green",
"darkred", "k"]

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



Lecture Outline

Visualisation: Introduction to Matplotlib

- Basic Plots: Line Plot: Labels, title, colour, Legend
- Scatter, Bar, Stem
- Matrix, Heatmap, Colormap, Image
- Figure Size and Multiple plots: Subplots
- Statistics: Histogram, Pie-Chart
- Decorating plots with labels, title, colour, markers, texts
- More on Figures and Plots

Decorating Figures

- Following

plt.title

plt.xlabel, plt.ylabel

plt.xlim, plt.ylim

plt.xticks, plt.yticks

Figure-size

Subplots

Color

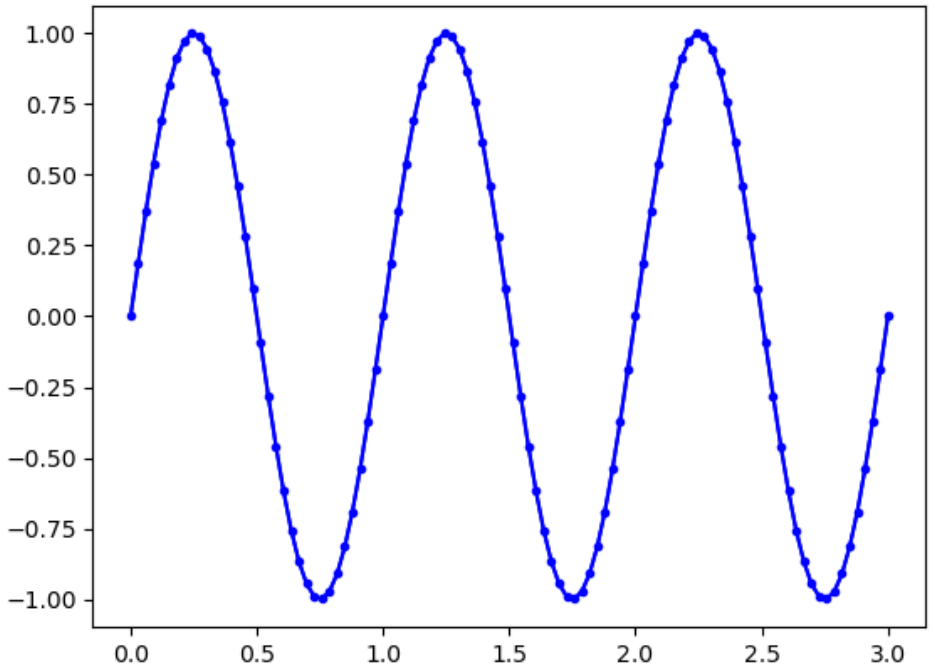
Linestyle

Markertype

Markersize

Alpha

```
t = np.linspace(0, 3, 100)
f = 1
x = np.sin(2*np.pi*f*t)
y = np.cos(2*np.pi*f*t)
plt.plot(t,x, color='b',marker='.')
plt.plot(t,x, color='b')
plt.show()
```



Decorating Figures

- Following

plt.title

plt.xlabel, plt.ylabel

plt.xlim, plt.ylim

plt.xticks, plt.yticks

Figure-size

Subplots

Color

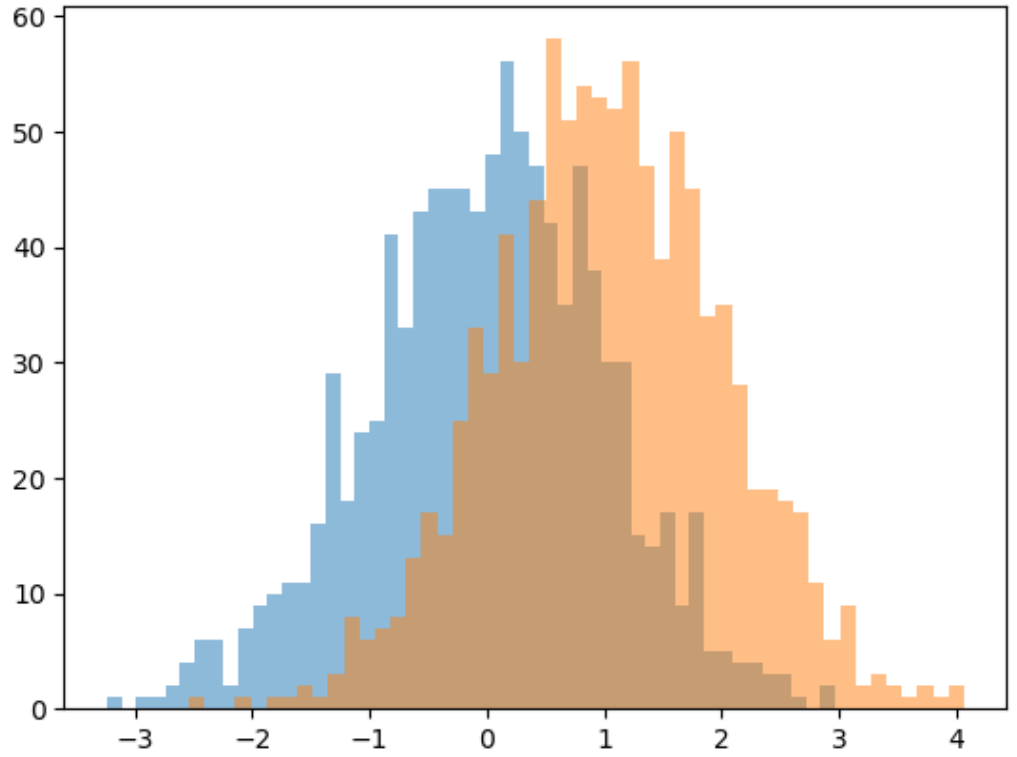
Linestyle

Markertype

Markersize

Alpha

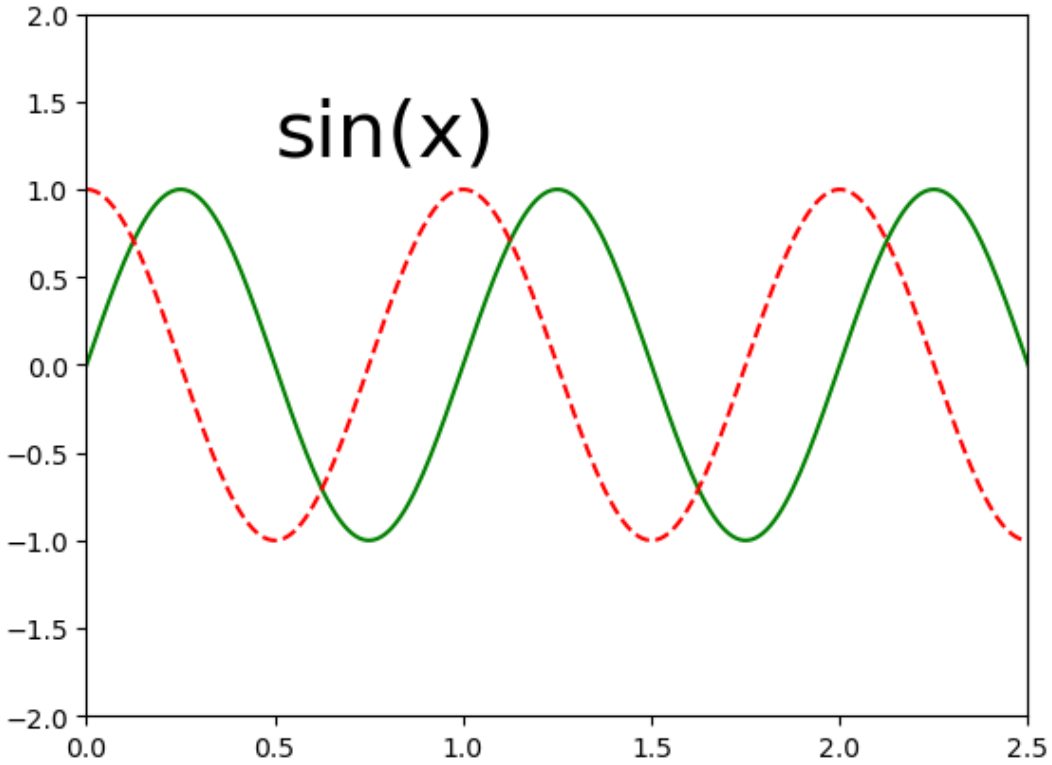
```
x = np.random.randn(1000)
y = np.random.randn(1000)+1

plt.figure()
plt.hist(x,alpha=0.5,bins=50)
plt.hist(y,alpha=0.5,bins=50)
plt.show()
```



Text in Figure

- Text in Figure

```
t = np.linspace(0, 3, 300)
f = 1
x = np.sin(2*np.pi*f*t)
y = np.cos(2*np.pi*f*t)
plt.plot(t,x, color='g')
plt.plot(t,y,color='r', ls='--')
plt.xlim([0,2.5])
plt.ylim([-2,2])
plt.text(0.5,1.2, 'sin(x)', fontsize=30)
plt.show()
```



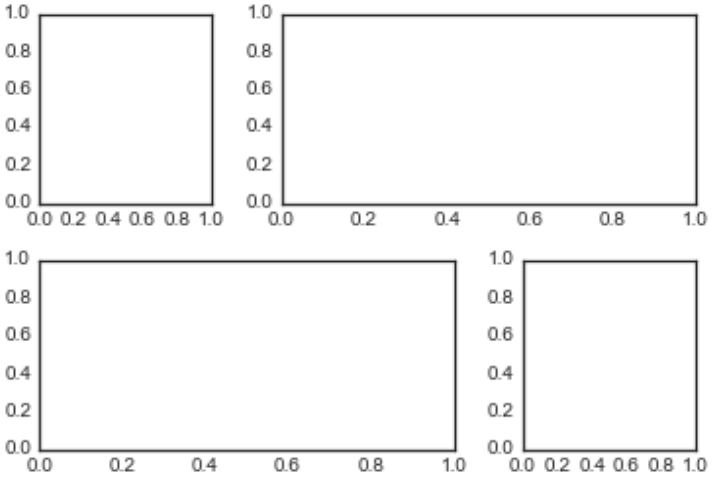
Lecture Outline

Visualisation: Introduction to Matplotlib

- Basic Plots: Line Plot: Labels, title, colour, Legend
- Scatter, Bar, Stem
- Matrix, Heatmap, Colormap, Image
- Figure Size and Multiple plots: Subplots
- Statistics: Histogram, Pie-Chart
- Decorating plots with labels, title, colour, markers, texts
- More on Figures and Plots

More on Figures/Plots

- Plotting with Axes
  - fig, ax = plt.subplots(2)
- Customising Image grid
  - grid = plt.GridSpec(2, 3, wspace=0.4, hspace=0.3)
- Customising Legend, Font, Line, Markers
- Boxplot, Error Plots, Density, Filling



- Plotting with Seaborn Library
  - **import seaborn as sns**

Ref: Python Data Science Handbook, 2nd Edition, Chapter 4

Link: https://jakevdp.github.io/PythonDataScienceHandbook/

- Next !!!
  - 4.1: Pandas Library

Queen Mary

University of London