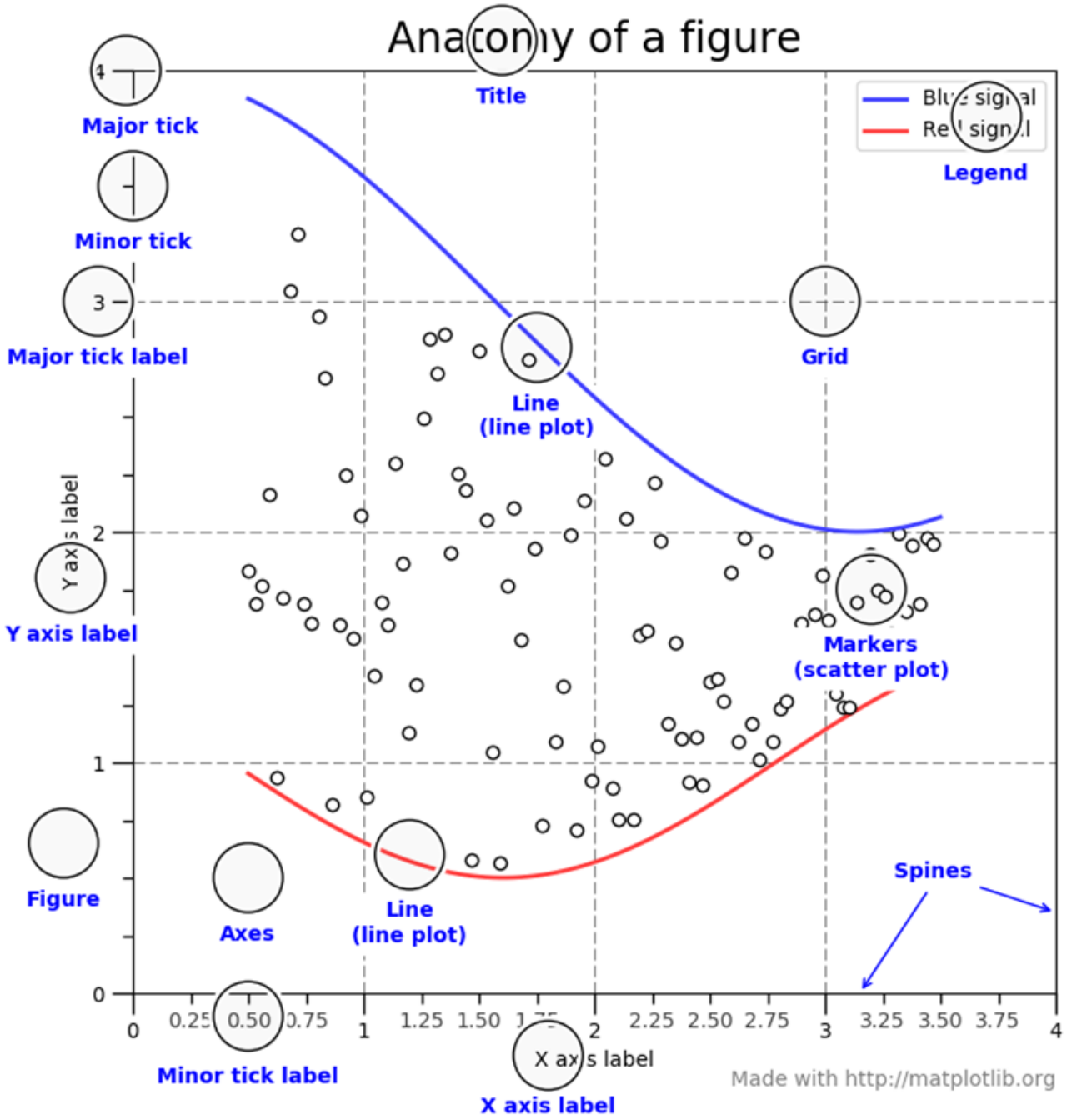
Python and Matplotlib

Matplotlib

A comprehension library for creating static, animated and interactive visualizations in Python.

Makes easy things easier and hard things possible.

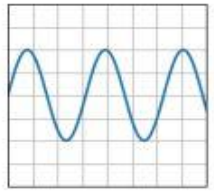
- Create publication quality plots
- Make interactive figures that can zoom, pan and update.
- Customize visual style and layout
- Export to many file formats
- Embed in JupyterLab and Graphical User Interfaces
- Use a rich array of third-party packages built on it.


Anatomy of a figure
Title
Major tick
Minor tick
Major tick label
Grid
Legend
Line
(line plot)
Y axis label
Markers
(scatter plot)
Figure
Axes
Line
(line plot)
Spines
Minor tick label
X axis label
Made with http://matplotlib.org

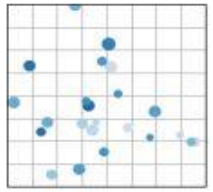
Types of Visuals

Basic

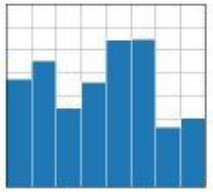
Basic plot types, usually y versus x.



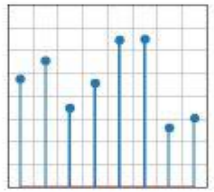
plot(x, y)



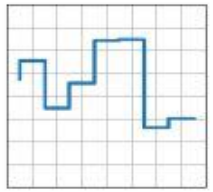
scatter(x, y)



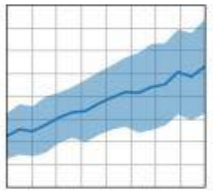
bar(x, height) / barh(y, width)



stem(x, y)



step(x, y)



fill_between(x, y1, y2)

Plots of arrays and fields

Plotting for arrays of data `Z(x, y)` and fields `U(x, y)`, `V(x, y)`.

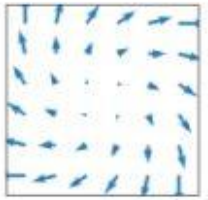
imshow(Z)

pcolormesh(X, Y, Z)

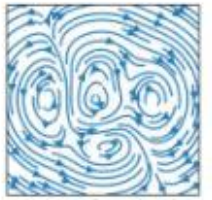
contour(X, Y, Z)

contourf(X, Y, Z)

barbs(X, Y, U, V)



quiver(X, Y, U, V)

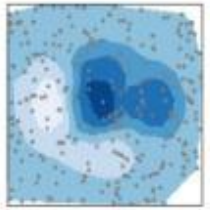


streamplot(X, Y, U, V)

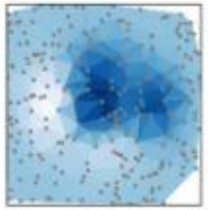
Unstructured coordinates

Sometimes we collect data `z` at coordinates `(x,y)` and want to visualize as a contour. Instead of gridding the data and then using `contour`, we can use a triangulation algorithm and fill the triangles.

tricontour(x, y, z)



tricontourf(x, y, z)

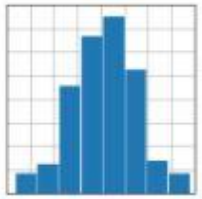


tripcolor(x, y, z)

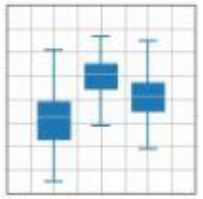
triplot(x, y)

Statistics plots

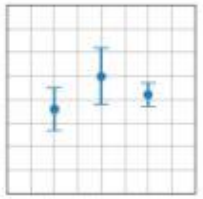
Plots for statistical analysis.



hist(x)

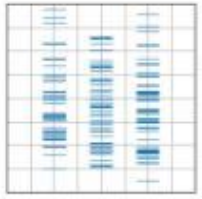


boxplot(X)



errorbar(x, y, yerr, xerr)

violinplot(D)



eventplot(D)

hist2d(x, y)

hexbin(x, y, C)

pie(x)

You need to install it

In your terminal, type

pip install matplotlib

You need to import it

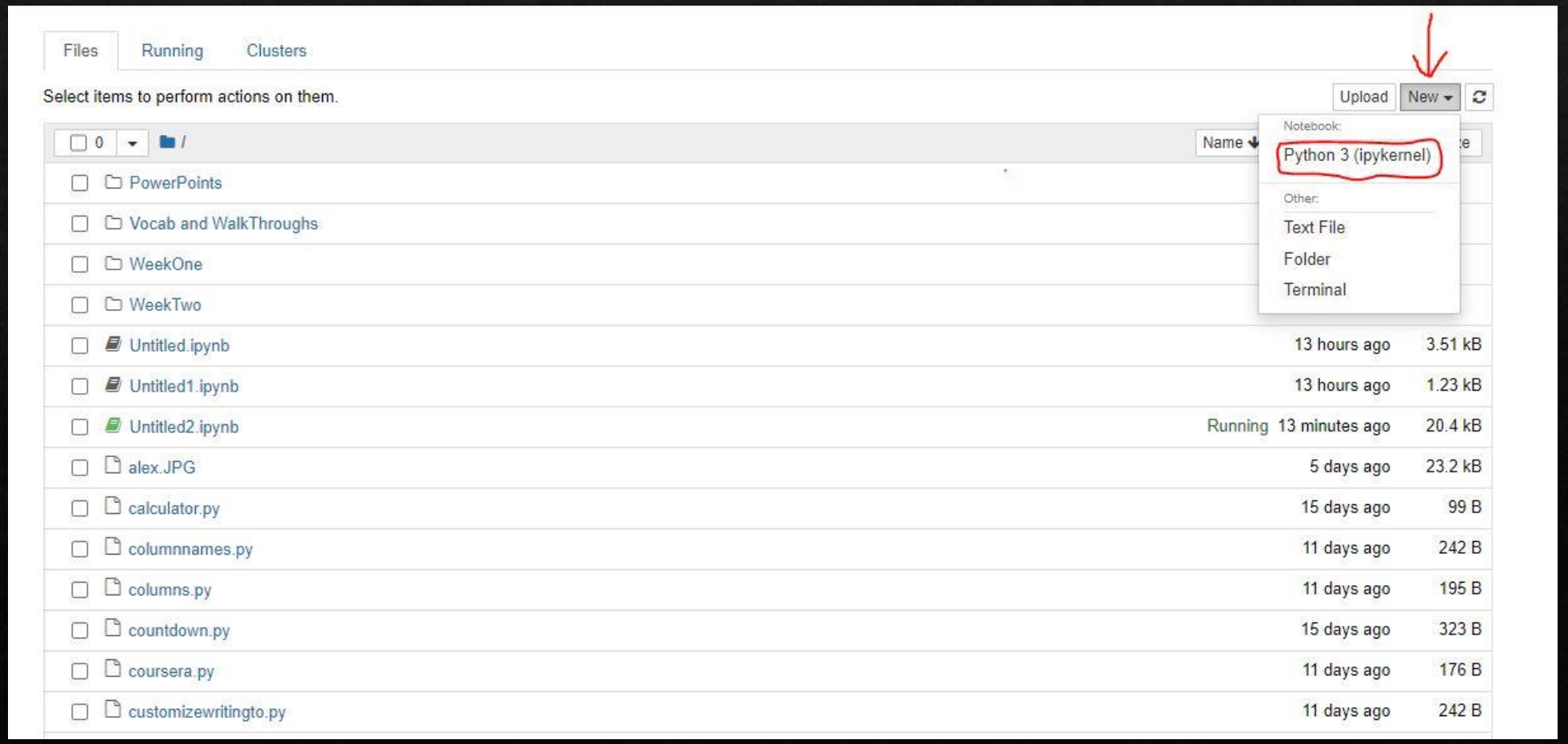
import matplotlib.pyplot as plt

import numpy as np

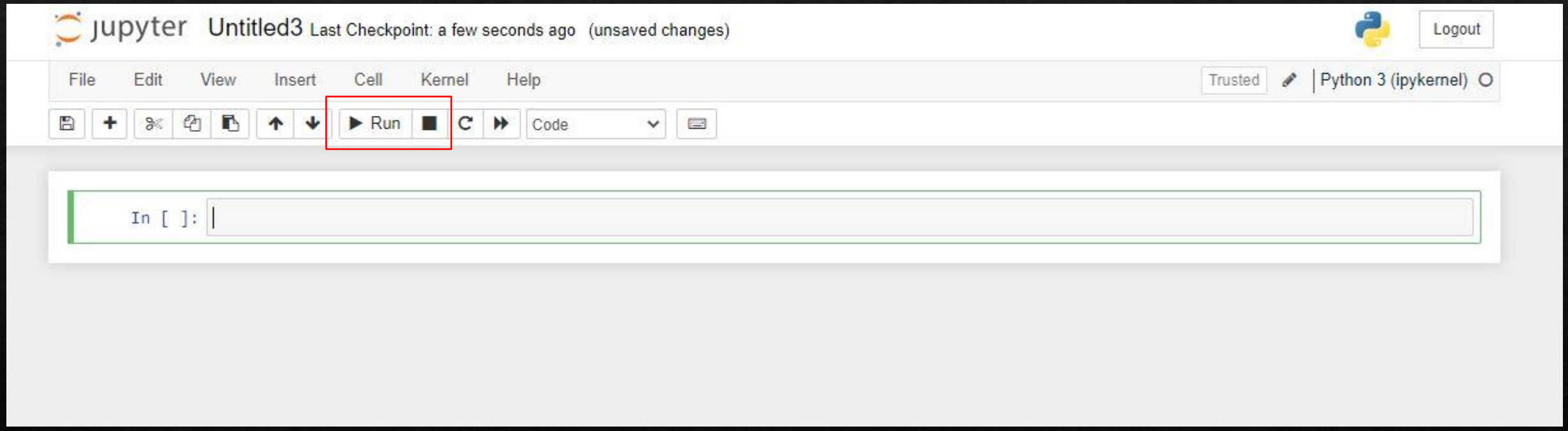
ALSO – pip install jupyter notebook

How to start the Jupyter Notebook Server

Type in jupyter notebook in your terminal and press enter



To execute commands click the RUN icon



Example One your first graph

```
import matplotlib.pyplot as plt

x = [1,2,3]   # x axis values
y = [2,4,1]   # corresponding y axis values

plt.plot(x, y)        # plotting the points

plt.xlabel('x - axis')       # naming the x axis

plt.ylabel('y - axis')       # naming the y axis

plt.title('My first graph!')        # giving a title to my graph

plt.show()   # function to show the plot
```

Example Two

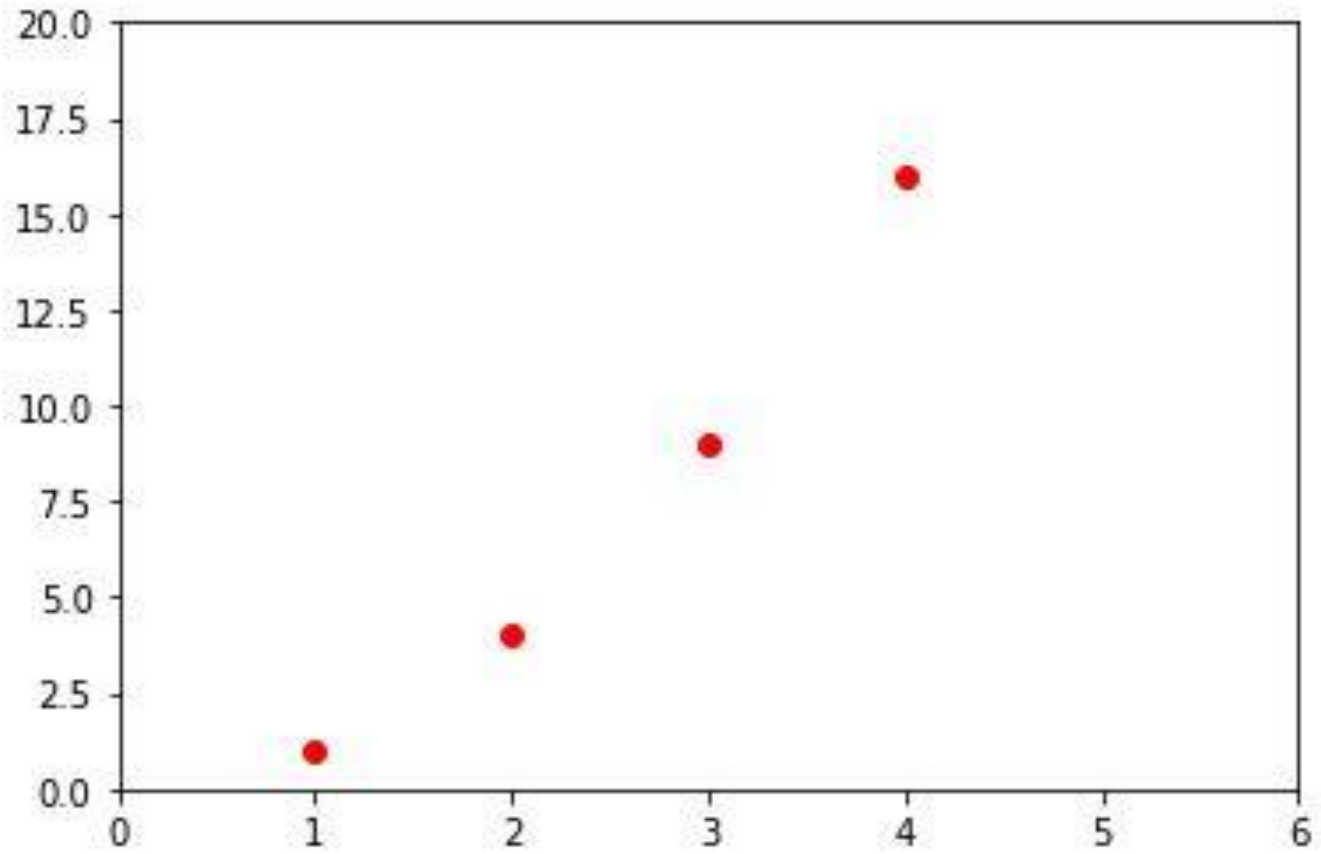
```
In [ ]: import matplotlib.pyplot as plt
        import numpy as np
```

```
In [13]: plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
         plt.ylabel('some numbers')
         plt.show()
```

Open VSCode
make sure you are in the right directory
initialize jupyter notebook

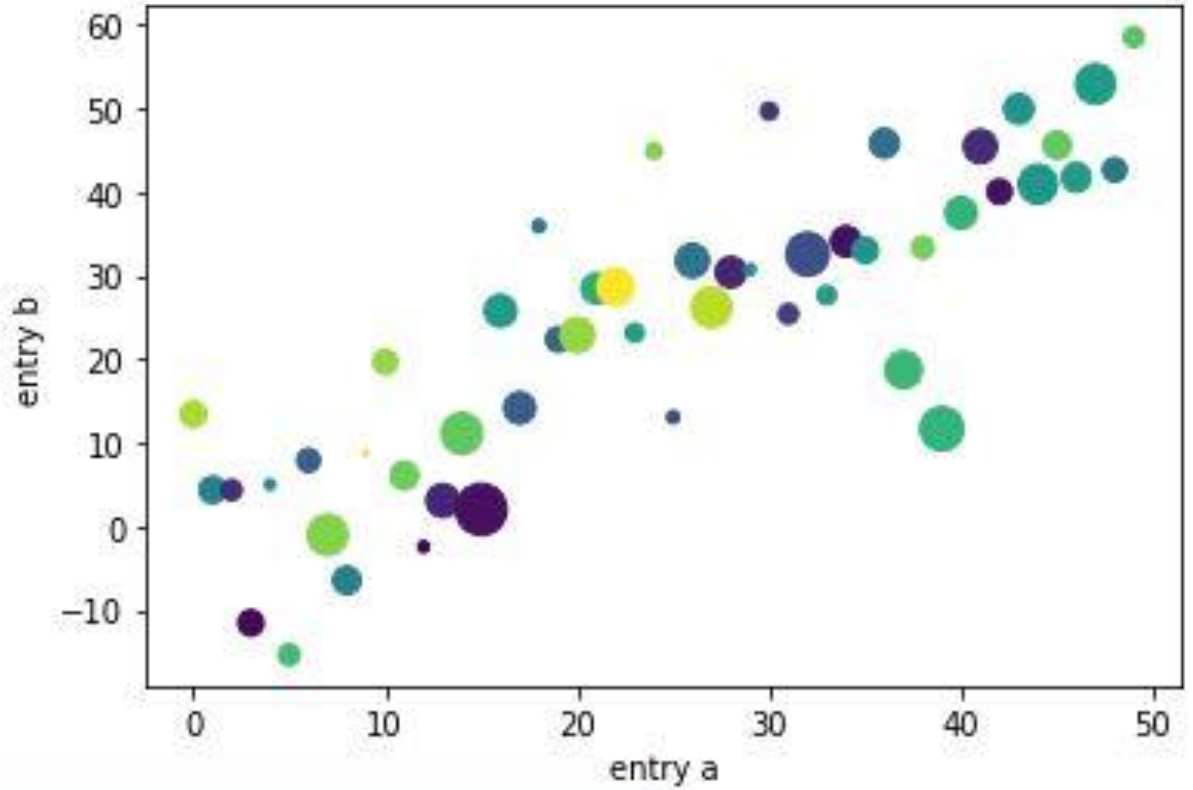
Example Three

In [14]:
```
import matplotlib.pyplot as plt
import numpy as np

plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'ro')
plt.axis([0, 6, 0, 20])
plt.show()
```



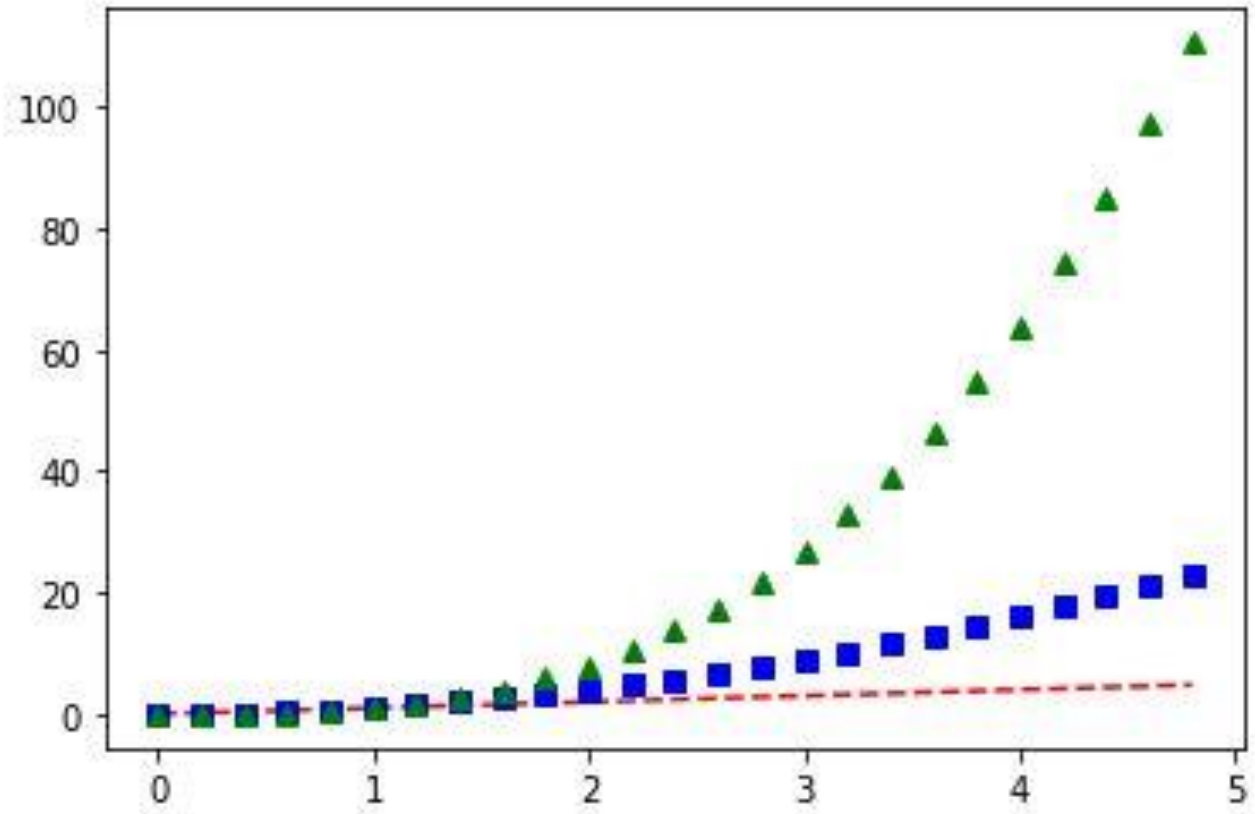
Example Four

```
In [17]: data = {'a': np.arange(50),
                 'c': np.random.randint(0, 50, 50),
                 'd': np.random.randn(50)}
         data['b'] = data['a'] + 10 * np.random.randn(50)
         data['d'] = np.abs(data['d']) * 100

         plt.scatter('a', 'b', c='c', s='d', data=data)
         plt.xlabel('entry a')
         plt.ylabel('entry b')
         plt.show()
```



Example Five

In [16]:
```
import numpy as np

# evenly sampled time at 200ms intervals
t = np.arange(0., 5., 0.2)

plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```



Customize – Customize - Customize

'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

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

Line Syntax	Description
'-'	Solid line
':'	Dotted line
'--'	Dashed line
'-.'	Dashed/dotted line

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

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

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

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

ms = markersize
mec = marker edge color
mfc = marker face color

Example Six

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


code = np.array([
    1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1,
    0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0,
    1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1,
    1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1])

pixel_per_bar = 4
dpi = 100

fig = plt.figure(figsize=(len(code) * pixel_per_bar / dpi, 2), dpi=dpi)
ax = fig.add_axes([0, 0, 1, 1])  # span the whole figure
ax.set_axis_off()
ax.imshow(code.reshape(1, -1), cmap='binary', aspect='auto',
          interpolation='nearest')
plt.show()
```

Example Seven

```
import matplotlib.pyplot as plt

a = [1, 2, 3, 4, 5]
b = [0, 0.6, 0.2, 15, 10, 8, 16, 21]

plt.plot(a)

plt.plot(b, "or")          # o is for circles and r is for red

plt.plot(list(range(0, 22, 3)))

plt.xlabel('Day ->’)     # naming the x-axis

plt.ylabel('Temp ->’) # naming the y-axis
```

Example Seven continued

```
c = [4, 2, 6, 8, 3, 20, 13, 15]
plt.plot(c, label = '4th Rep')

ax = plt.gca()  # get current axes command

# set command over the individual boundary line of the graph body
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)

# set the range or the bounds of the left boundary line to fixed range
ax.spines['left'].set_bounds(-3, 40)

# set the interval by  which the x-axis set the marks
plt.xticks(list(range(-3, 10)))
```

Example Seven continued

```
# set the intervals by which y-axis set the marks
plt.yticks(list(range(-3, 20, 3)))

# legend denotes what color signifies what
ax.legend(['1st Rep', '2nd Rep', '3rd Rep', '4th Rep'])

# annotate command helps to write ON THE GRAPH any text
# xy denotes the position on the graph
plt.annotate('Temperature V / s Days', xy = (1.01, -2.15))

# gives a title to the Graph
plt.title('All Features Discussed')

plt.show()
```

Example Eight

```
import matplotlib.pyplot as plt


a = [1, 2, 3, 4, 5]
b = [0, 0.6, 0.2, 15, 10, 8, 16, 21]
c = [4, 2, 6, 8, 3, 20, 13, 15]

# use fig whenever u want the
# output in a new window also
# specify the window size you
# want answer to be displayed
fig = plt.figure(figsize =(10, 10))
```

Example Eight continued

```
# creating multiple plots in a single plot
sub1 = plt.subplot(2, 2, 1)
sub2 = plt.subplot(2, 2, 2)
sub3 = plt.subplot(2, 2, 3)
sub4 = plt.subplot(2, 2, 4)

sub1.plot(a, 'sb')

# sets how the display subplot
# x axis values advances by 1
# within the specified range
sub1.set_xticks(list(range(0, 10, 1)))
sub1.set_title('1st Rep')

sub2.plot(b, 'or')
```

Example Eight continued

```
# sets how the display subplot x axis
# values advances by 2 within the
# specified range
sub2.set_xticks(list(range(0, 10, 2)))
sub2.set_title('2nd Rep')

# can directly pass a list in the plot
# function instead adding the reference
sub3.plot(list(range(0, 22, 3)), 'vg')
sub3.set_xticks(list(range(0, 10, 1)))
sub3.set_title('3rd Rep')
```

Example Eight continued

```
sub4.plot(c, 'Dm')

# similarly we can set the ticks for
# the y-axis range(start(inclusive),
# end(exclusive), step)
sub4.set_yticks(list(range(0, 24, 2)))
sub4.set_title('4th Rep')

# without writing plt.show() no plot will be visible
plt.show()
```

Example Nine

```
import matplotlib.pyplot as plt

# line 1 points
x1 = [1,2,3]
y1 = [2,4,1]

# plotting the line 1 points
plt.plot(x1, y1, label = "line 1")

# line 2 points
x2 = [1,2,3]
y2 = [4,1,3]

# plotting the line 2 points
plt.plot(x2, y2, label = "line 2")
```

Example Nine Continued

```
# naming the x axis
plt.xlabel('x - axis’)

# naming the y axis
plt.ylabel('y - axis’)

# giving a title to my graph
plt.title('Two lines on same graph!')

# show a legend on the plot
plt.legend()

# function to show the plot
plt.show()
```

Example Ten

```
import matplotlib.pyplot as plt

# x axis values
x = [1,2,3,4,5,6]
# corresponding y axis values
y = [2,4,1,5,2,6]

# plotting the points
plt.plot(x, y, color='green', linestyle='dashed', linewidth = 3,
         marker='o', markerfacecolor='blue', markersize=12)
```

```
# setting x and y axis range
plt.ylim(1,8)
plt.xlim(1,8)

# naming the x axis
plt.xlabel('x - axis')
# naming the y axis
plt.ylabel('y - axis')

# giving a title to my graph
plt.title('Some cool customizations!')

# function to show the plot
plt.show()
```

Example Eleven

```
import matplotlib.pyplot as plt

# x-coordinates of left sides of bars
left = [1, 2, 3, 4, 5]

# heights of bars
height = [10, 24, 36, 40, 5]

# labels for bars
tick_label = ['one', 'two', 'three', 'four', 'five']
```

```
# plotting a bar chart
plt.bar(left, height, tick_label = tick_label,
        width = 0.8, color = ['red', 'green'])

# naming the x-axis
plt.xlabel('x - axis')
# naming the y-axis
plt.ylabel('y - axis')
# plot title
plt.title('My bar chart!')

# function to show the plot
plt.show()
```

Example Twelve

```
import matplotlib.pyplot as plt

# frequencies
ages = [2,5,70,40,30,45,50,45,43,40,44,
        60,7,13,57,18,90,77,32,21,20,40]

# setting the ranges and no. of intervals
range = (0, 100)
bins = 10

# plotting a histogram
plt.hist(ages, bins, range, color = 'green',
         histtype = 'bar', rwidth = 0.8)
```

```
# x-axis label
plt.xlabel('age')
# frequency label
plt.ylabel('No. of people')
# plot title
plt.title('My histogram')

# function to show the plot
plt.show()
```

Example Thirteen

```
import matplotlib.pyplot as plt

activities = ['eat', 'sleep', 'work', 'play']   # defining labels
slices = [3, 7, 8, 6]   # portion covered by each label
colors = ['r', 'y', 'g', 'b']  # color for each label


# plotting the pie chart
plt.pie(slices, labels = activities, colors=colors,
        startangle=90, shadow = True, explode = (0, 0, 0.1, 0),
        radius = 1.2, autopct = '%1.1f%%')

plt.legend()  # plotting legend
plt.show()  # showing the plot
```

Example Fourteen

```
# importing the required modules
import matplotlib.pyplot as plt
import numpy as np

# setting the x - coordinates
x = np.arange(0, 2*(np.pi), 0.1)
# setting the corresponding y - coordinates
y = np.sin(x)

# plotting the points
plt.plot(x, y)

# function to show the plot
plt.show()
```

Example Fifteen

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

ypoints = np.array([3, 8, 1, 10, 5, 7])

plt.plot(ypoints)
plt.show()
```

Draw a line in a diagram from position (1, 3) to position (8,10)

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

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

plt.plot(xpoints, ypoints)
plt.show()
```

Draw 2 points in a diagram from position (1, 3) to position (8,10)

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

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

plt.plot(xpoints, ypoints, 'o')
plt.show()
```

Draw a line in a diagram from position (1, 3) to (2, 8) then to (6, 1) and finally to position (8,10)

```
import matplotlib.pyplot as plt

import numpy as np


xpoints = np.array([1, 8])

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


plt.plot(xpoints, ypoints, ‘o’)

plt.show()
```