

[Table of contents \(../toc.ipynb\)](#)

Plotting packages

- Until now, we were just able to return results via `print()` command.
- However, as humans depend very much on vision, data visualization is urgently needed.

Matplotlib



- Very common and widely used is the [matplotlib \(https://matplotlib.org/\)](https://matplotlib.org/) package.
- Matplotlib provides with `pyplot` module very similar plotting commands as Matlab.
- Matplotlib is packaged and available through pip and conda.
- Let's create our first plot.

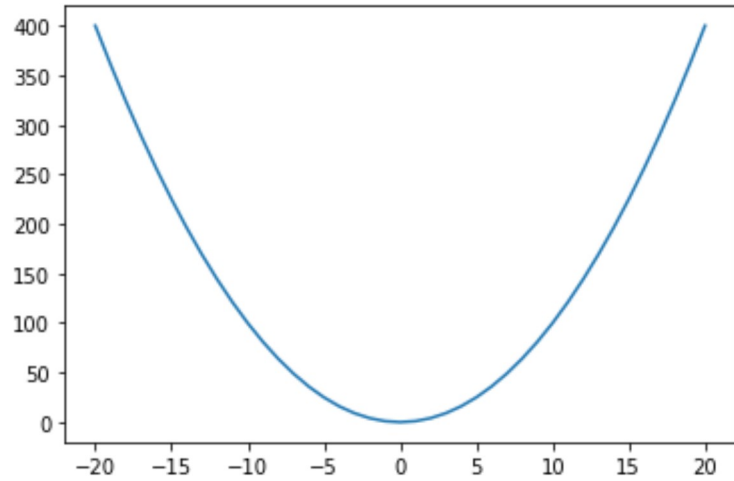
```
In [16]: from matplotlib import pyplot as plt
%matplotlib inline
```

- The first line of code imports the pyplot module, which is very convenient for most simple plots.
- The second line is a special Jupyter magic command to present the plots instantly.

Basic plotting

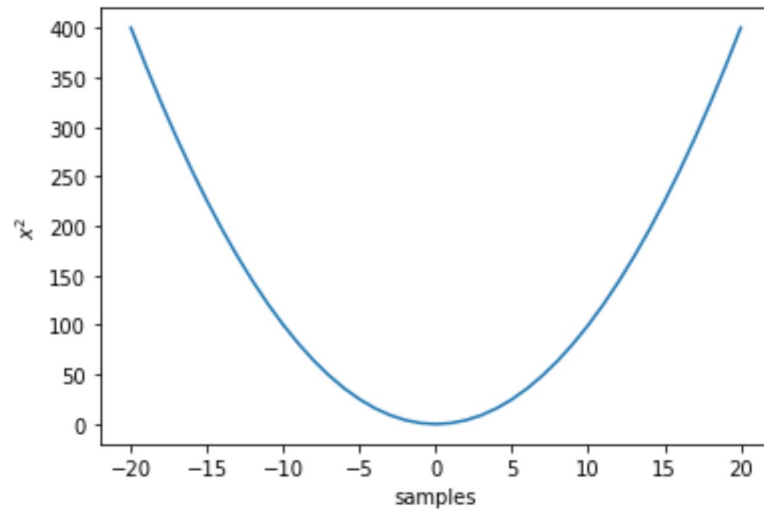
```
In [17]: # First, we create some data to plot, which is not that handy with lists.  
x = [i for i in range(-20, 21)]  
y = [x_i**2 for x_i in x]
```

```
In [18]: # Now the plot command  
plt.plot(x, y)  
  
plt.show()
```



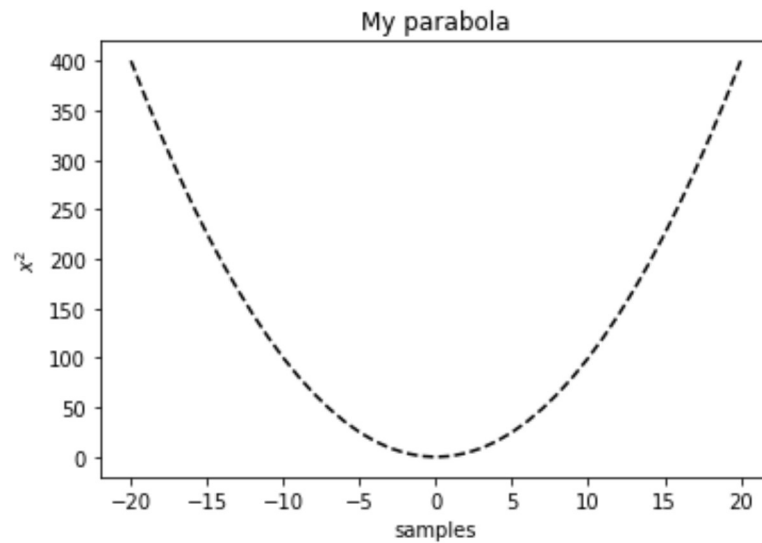
```
In [19]: # Here the same plot with axis labels
plt.plot(x, y)
plt.xlabel("samples")
plt.ylabel("$x^2$") # You can use LaTeX's math support

plt.show()
```



```
In [20]: # Some line style arguments and a title
plt.plot(x, y, "k--")
plt.xlabel("samples")
plt.ylabel("$x^2$") # You can use LaTeX's math support
plt.title("My parabola")

plt.show()
```



Multiple plots in one figure

To present two time series, we will create an additional list of values.

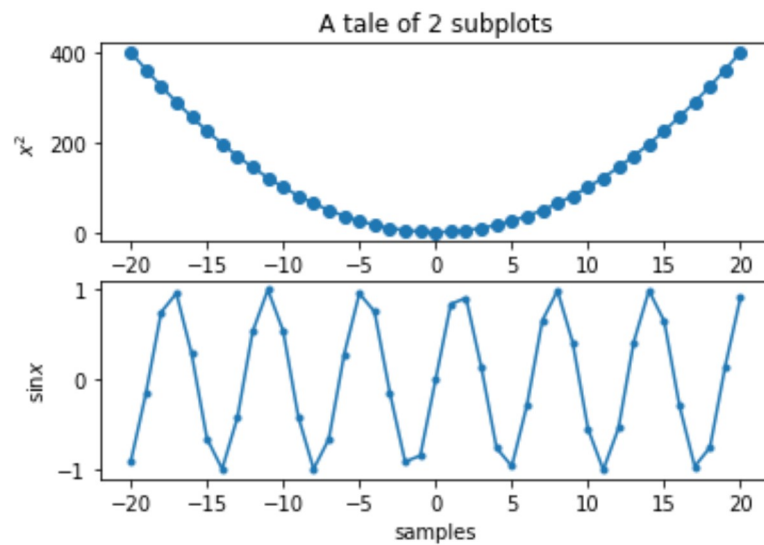
```
In [21]: import math  
y2 = [math.sin(x_i) for x_i in x]
```



```
In [22]: plt.subplot(2, 1, 1)
plt.plot(x, y, 'o-')
plt.title('A tale of 2 subplots')
plt.ylabel('$x^2$')

plt.subplot(2, 1, 2)
plt.plot(x, y2, '.-')
plt.xlabel('samples')
plt.ylabel('$\sin{\mathbf{x}}$')

plt.show()
```

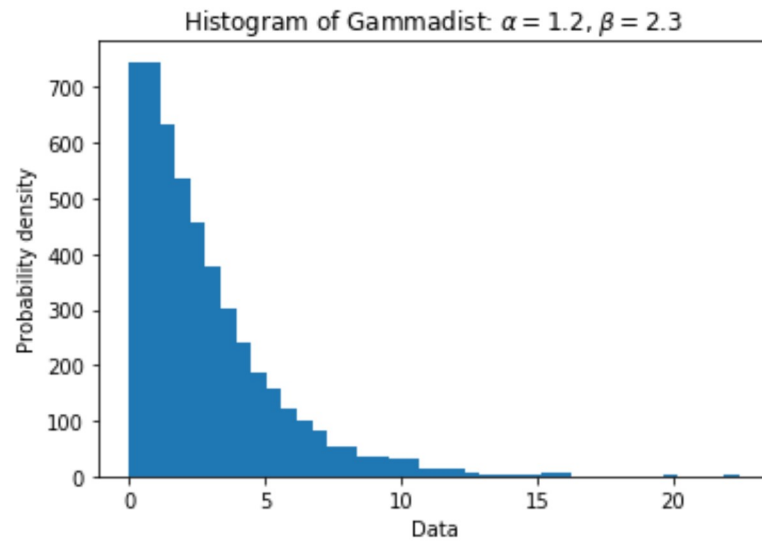


- I guess you got the concept here.
- Also basic statistical plots are simple to generate.

```
In [23]: # Generate random numbers in a list  
import random  
  
y3 = [random.gammavariate(alpha=1.2, beta=2.3) for i in range(0, 5000)]
```

```
In [24]: plt.hist(y3, bins=40)
plt.xlabel('Data')
plt.ylabel('Probability density')
plt.title(r'Histogram of Gammadist:  $\alpha=1.2$ ,  $\beta=2.3$ ')

plt.show()
```



Exercise: Matplotlib (10 minutes)



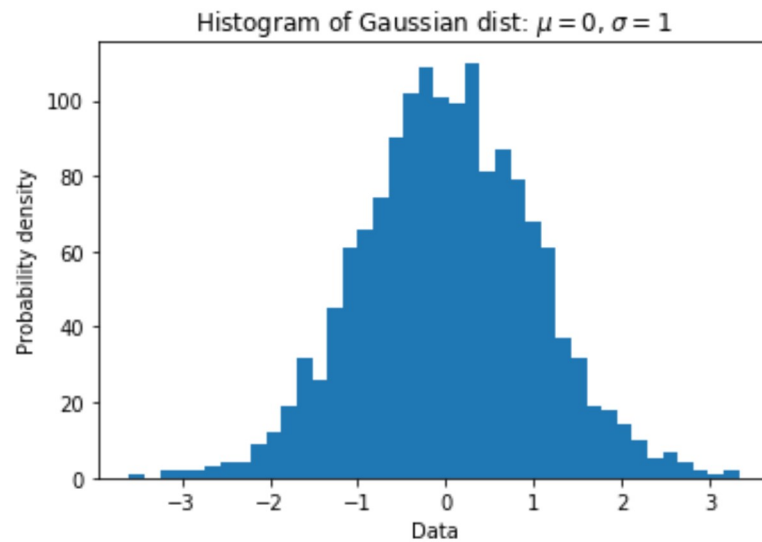
Here the task:

- Activate your local Python environment and install matplotlib with `conda install matplotlib`.
- Either work in Jupyter notebook, in Ipython shell, or Pycharm.
- Create a list of 1500 data points.
- Add a second list of 1500 data points with Gaussian distribution $\mathcal{N} \sim (0, 1)$.
- Create a figure with a histogram of the data.

Solution

Please find one possible solution in [solution_plotting.py](#) [\(solution_plotting.py\)](#) file.

```
In [25]: %run solution_plotting.py  
plt.show()
```



<Figure size 432x288 with 0 Axes>

More plot types

- Many other plots are as simple as the examples above.
- Other typical plots are
 - Bar plots
 - 3D plots
 - Pie charts
 - Scatter plots
 - ...
- You can find many more plot examples in the [Matplotlib gallery \(https://matplotlib.org/gallery/index.html\)](https://matplotlib.org/gallery/index.html).

```
In [26]: IFrame(src='https://matplotlib.org/gallery/index.html', width=700, height=600)
```

Out[26]:

John Hunter Excellence in Plotting Contest 2020 **submissions are open!** Entries are due June 1, 2020.



Fork me on GitHub

Version 3.2.1

[Installation](#) [Documentation](#) [Examples](#) [Tutorials](#) [Contributing](#) [Search](#)

[home](#) | [contents](#) »

[modules](#) | [ir](#)

Gallery

This gallery contains examples of the many things you can do with Matplotlib. Click on any image to see the full image and source code.

For longer tutorials, see our [tutorials page](#). You can also find [external resources](#) and a [FAQ](#) in our [user guide](#).

Lines, bars and markers



Table of Contents

Gallery

- [Lines, bars and markers](#)
- [Images, contours and fields](#)
- [Subplots, axes and figures](#)
- [Statistics](#)
- [Pie and polar charts](#)
- [Text, labels and annotation](#)
- [Pyplot](#)
- [Color](#)

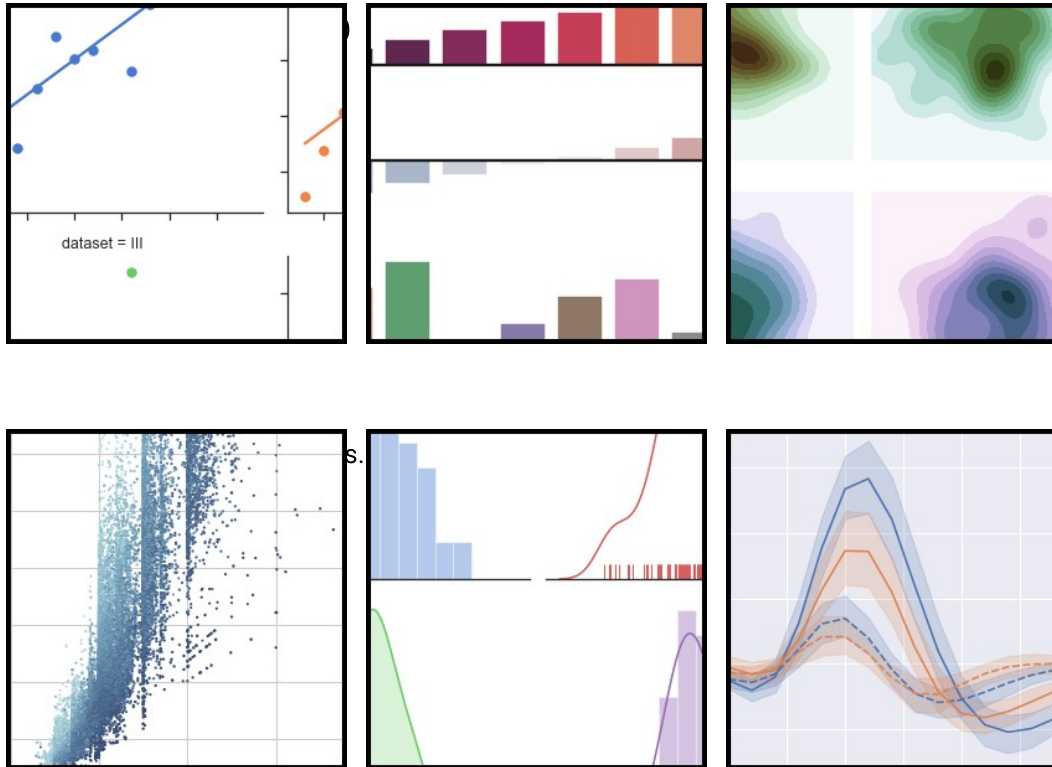
Seaborn

- Seaborn is a statistical plot library for Python.
- It is based on matplotlib.
- Please find it's documentation here <https://seaborn.pydata.org/> (<https://seaborn.pydata.org/>).
- You can install it with `conda install seaborn`.
- Next comes a snapshot of seaborn's example gallery <https://seaborn.pydata.org/examples/index.html> (<https://seaborn.pydata.org/examples/index.html>).


```
In [27]: IFrame(src='https://seaborn.pydata.org/examples/index.html', width=700, height=600)
```

Out[27]:

Example gallery



Bokeh



- Add to "static" plots, interactive plots and dashboards can be build with Bokeh library.
- Interactive plots are ideal if you want to visualize large data sets.
- Real time information is visible like server load,...
- Boekh is easy to install via conda and pip.
- Please find here the Bokeh gallery <https://docs.bokeh.org/en/latest/docs/gallery.html> (<https://docs.bokeh.org/en/latest/docs/gallery.html>).
- And in next cell, an interactive plot of bokeh website is presented.

```
In [28]: IFrame(src='https://demo.bokeh.org/crossfilter', width=700, height=600)
```

Out[28]:

Plotly

- Plotly is another very simple to use interactive plot library.
- You can find more detail in <https://plot.ly/python/> (<https://plot.ly/python/>).
- Next, some examples from plotly gallery are presented.

```
In [29]: IFrame(src='https://plot.ly/python/', width=700, height=600)
```

Out [29]:

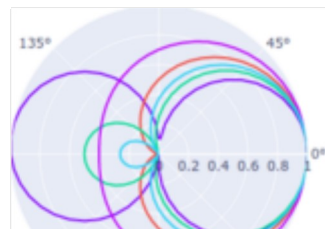


Plotly Python Open Source Graphing Library

Plotly's Python graphing library makes interactive, publication-quality graphs. Examples of how to make line plots, scatter plots, area charts, bar charts, error bars, box plots, histograms, heatmaps, subplots, multiple-axes, polar charts, and bubble charts.

Plotly.py is [free and open source](https://plot.ly/python/is-plotly-free) (/python/is-plotly-free) and you can [view the source, report issues or contribute on GitHub](https://github.com/plotly/plotly.py) (https://github.com/plotly/plotly.py).

Fundamentals



In a nutshell

- There are many very powerful plot libraries available for Python.
- The chance that someone wrote a template for your plotting task is very high.
- Just look in the galleries and you will find a great starting point.