

Plots in Python

2 main modules are required `numpy` and `matplotlib`.

NumPy

Numpy toolkit is important and fundamental package for scientific computing with Python. NumPy is a standard python module that is part of the standard python library.

NumPy's main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In NumPy dimensions are called *axes*.

A 2-D array has 2 axes. The first axis has a length of 2, the second axis has a length of 3.

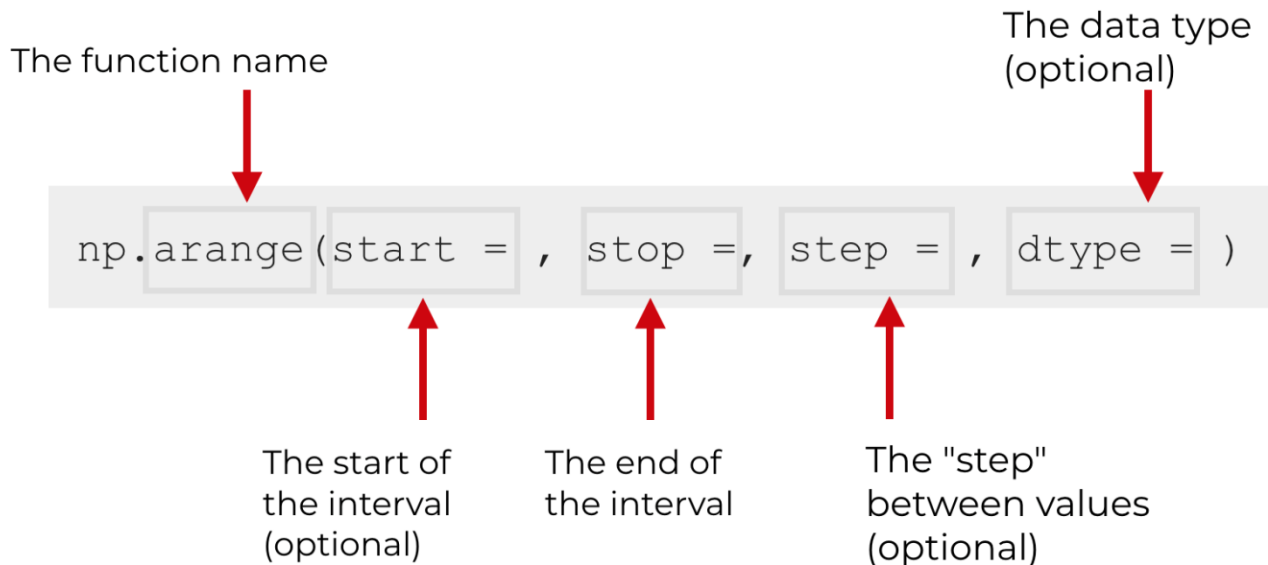
```
[[ 1., 0., 0.], [ 0., 1., 2.]]
```

The Numpy `arange` function (sometimes called `np.arange`) is a tool for creating numeric sequences in Python.

You will see the `np.arange` function in a lot of data science code.

The NumPy `arange` function returns evenly spaced numeric values within an interval, stored as a NumPy array (i.e., an `ndarray` object).

Example: `numpy.arange(5)` produces the array 0,1,2,3,4 # This is very similar to the range function.



- **start:** optional parameter and indicates the starting value on your range. Default is 0.
- **stop:** This is a required parameter. End of the range. As in all other examples of python indexing, this value is not included in the resulting range (up to but not including the stop value)
- **step:** Optional and it specifies the spacing. Default is 1
- **dtype:** Optional and specifies the datatype. If you do not specify, python will decide based on other values in the function.

```
import numpy as np # IMPORT NUMPY
```

```
np.arange(stop = 5) # same as np.arange(5) 0,1,2,3,4
```

```
np.arange(start = 0, stop = 8, step = 2) # increments of 2 0,2,4,6
```

for the dtype, you can specify one of the python date types:

dtype= 'float' dtype = 'integer'

matplotlib Module:

matplotlib module is not part of the standard Python modules and has to be installed.

Installing matplotlib in windows: (similar instructions apply for MAC)

- Start by opening a command line. You can do this by going in the windows search window (bottom left corner of the screen) . Type cmd
- You should see and select the Command Prompt App to display the Command Prompt small window.
- Type the following code into the command line: `py -m pip install -U matplotlib --user`.
- It is possible that pip does not get installed by default. One potential fix before installing the module is to run the following command:
- `python -m ensurepip --default-pip`
- `pip3 install matplotlib`
<http://www.eg.bucknell.edu/~csci203/common-files/course-docs/Matplotlib-installing.html>

matplotlib.pyplot

<https://matplotlib.org/gallery/index.html>

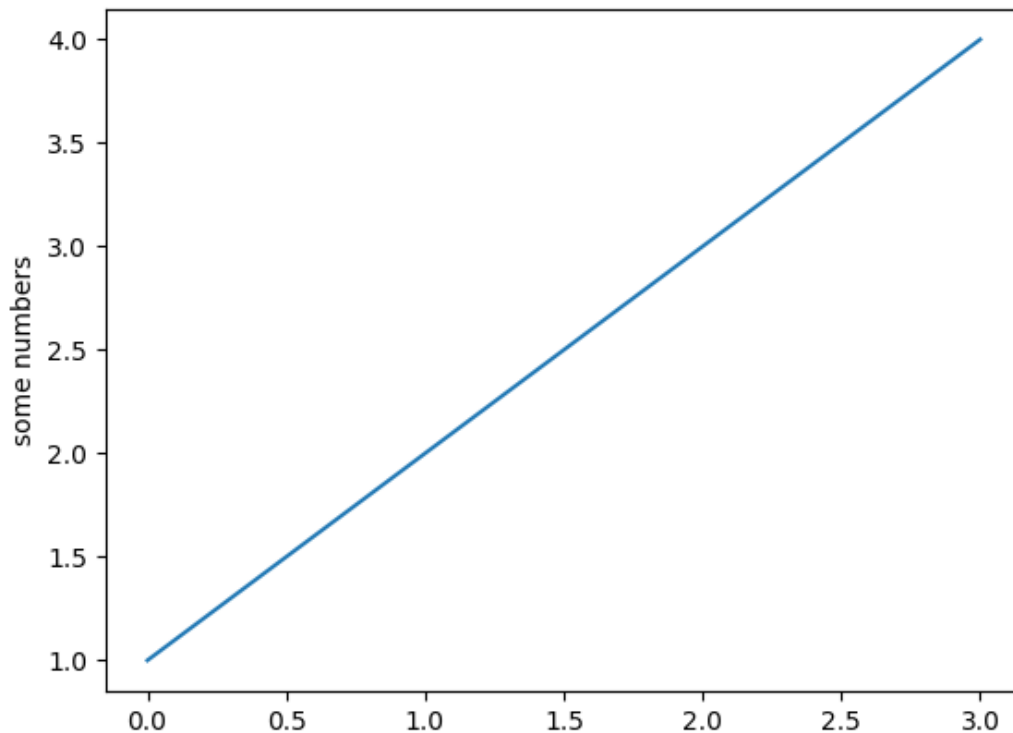
- **Matplotlib is a Python 2D plotting library**
- **You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc.**
- **All this ... with just a few lines of code.**
- **For simple plotting, which we are going to do here, the pyplot module is sufficient.**
- **matplotlib.pyplot is a collection of command-style functions.**
- **Each function adds or does a small change or decoration to a figure.**

Simple Example:

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

```
plt.plot([1, 2, 3, 4])  
plt.ylabel('some numbers')  
plt.show()
```

Note that a single input list is taken as y, the x values are automatically generated

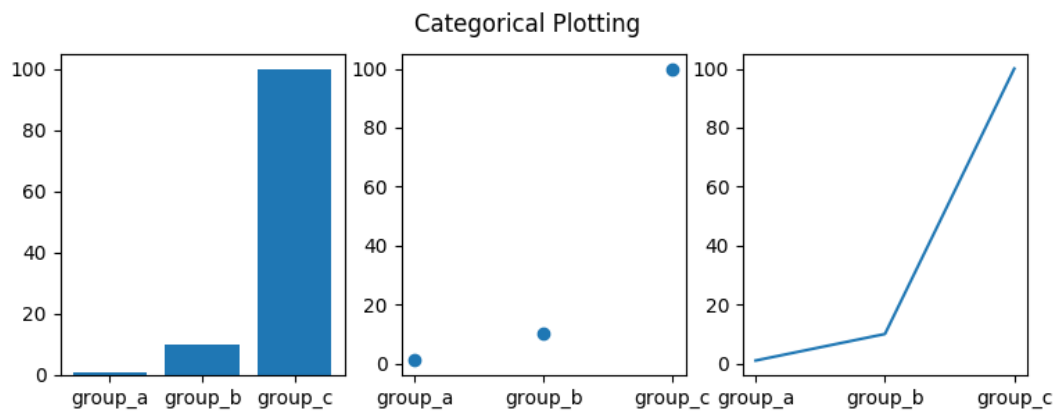


Categorical Plotting:

```
names = ['group_a', 'group_b', 'group_c']  
values = [1, 10, 100]
```

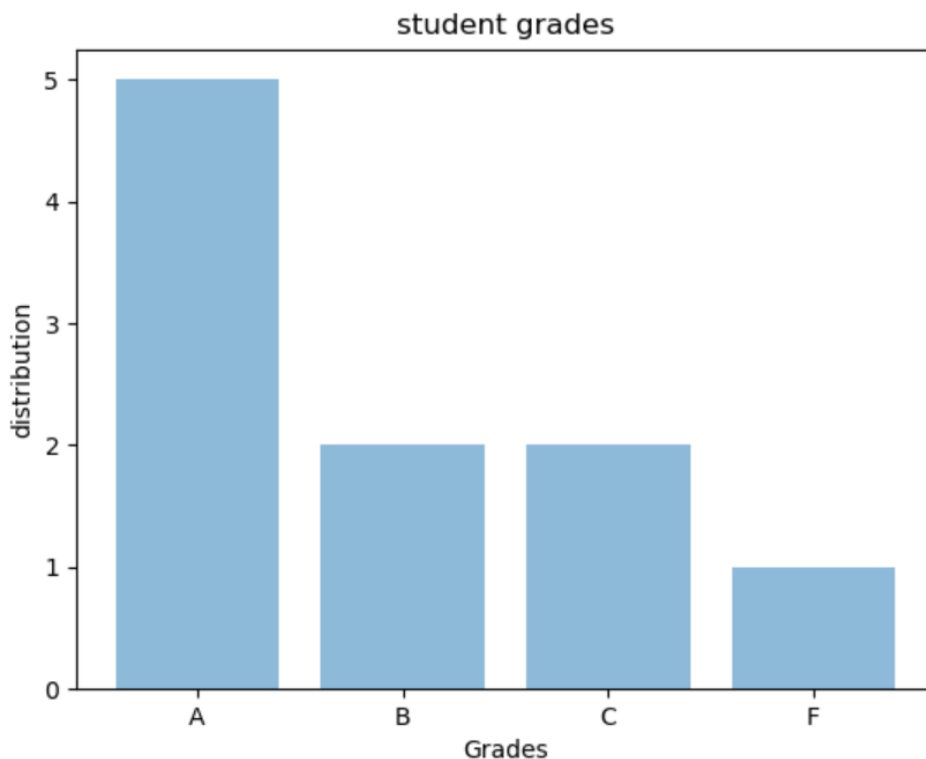
```
plt.figure(1, figsize=(9, 3))  
plt.subplot(131)
```

`plt.bar(names, values)`
`plt.subplot(132)`
`plt.scatter(names, values)`
`plt.subplot(133)`
`plt.plot(names, values)`
`plt.suptitle('Categorical Plotting')`
`plt.show()`



A Simpler Example Using barchart plot Only:

```
grades = ("A","B","C","F")
markings = np.arange(len(grades))
distribution = [5,2,2,1]
plt.bar(markings,distribution, align='center',alpha=0.5)
plt.xticks(markings,grades) #x axis markings
plt.title("student grades")
plt.xlabel("Grades")
plt.ylabel("distribution")
plt.show()
```



Some Additional Properties For Barchart:

```
plt.bar(x=,height=,width=,bottom=,align='center',alpha=0.5)
```

x : sequence of scalars. The x coordinates of the bars.

height : scalar or sequence of scalars. The height(s) of the bars.

width : scalar or array-like, optional The width(s) of the bars (default: 0.8).

bottom : scalar or array-like, optional. The y coordinate(s) of the bars bases (default: 0).

align : {'center', 'edge'}, optional, default: 'center'. Alignment of the bars to the x coordinates:

- 'center': Center the base on the x positions.
- 'edge': Align the left edges of the bars with the x positions.

Alpha: float value $0.0 \leq \alpha \leq 1.0$ (0.0 transparent through 1.0 opaque) Default is 1.0

<https://matplotlib.org/tutorials/introductory/pyplot.html#sphx-glr-tutorials-introductory-pyplot-py>