Installation or Setup



Numpy

**Installing numpy with Anaconda** a cross-platform (Linux, Mac OS X, Windows) Python distribution for data analytics and scientific computing.

**Command for installing numpy:**

conda install numpy

**To update numpy to the latest version in anaconda or miniconda use:**

conda update numpy

**Installing numpy with pip:**

pip install numpy

**Import numpy library**:

Creating ID Array from List

My list=|1, 2,3,â,5 ]

np.array(hy\_list}

array((I, Z, 3, 4, 5])

Creating 2D Array from Li6t

np . array { { 1, 2 , â ] , { 4 , S , 6 ] , { i , 8 , 9 ] j

» rray ( ( t 1, 2, 3 ] ,

[ 4, s , s j ,

Creating Array with help of arange method.

ng.arange{ 2,12)

array({ 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]}

Create Zero's matric s

array((}0., 0.],

Creating unit marries np . one a I ( 3 , 2 j }

np. eye ‹ ‹ I

a rray t ( t I . , 0 . , 0 . , 0 . J ,

[ 0 . , 1,. 0 . , 0 . j .

t 0 . , 0 . , I . , 0 . ] ,

arcay{[0.81626231, 0.62Q89421, 0.36626622, Q.77068519]}

np.random.rand(4,4)

$ rr$y{[[0•1893J i 55, 0.96255073, 0 - 5 56152) z , 0 - )524S9 ] ,

|0.95720688, 0.95990808, 0. 3 g1 I 9 86 I , 0. BI 332 14 9 ] ,

[0.53726093, 0.44967789, 0. ? B056 94 1, 0 . 16 14 192 2 ] ,

|0.15594651, 0.23578237, 0. 50208 904, 0. 8925098 5 ] ) )

mat=ng.random.rand(Z0}.reshape{5, 4 j

array{[[0.15]99962, 0.64669223, 0 . 9 5 i 0 9 93 I , 0 . 53 775 9S 8 ] ,

[0.12832676, 0.71483939, 0.8620B107, 0.4120665 ],

[0.8660€395, 0.96338 22, Q.55607608, 0.748Q5943 ],

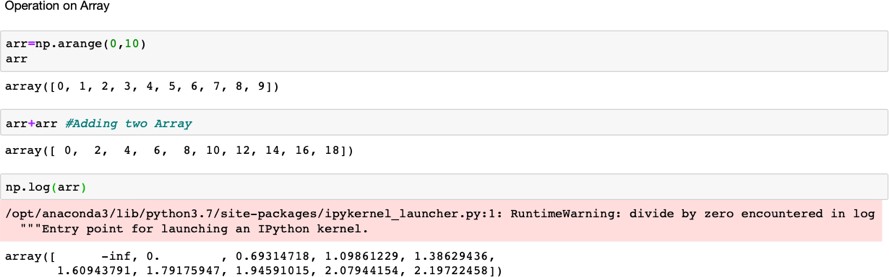
[0.19223759, 0.92543899, Q.5B36lJ02, Q. 5l4670Z9] ,

[0.67558236, 0.74893868, 0.07982758, 0.02191947]]

type{mat} numpy.ndarray

0 . 96 35B42 L90 3Bg BB





# Pandas

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labelled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in python. It is open source library built on top numpy. It also built- in visualization features. It can work with data from wide variety of sources.

Installation or Setup

**Installing pandas with Anaconda** a cross-platform (Linux, Mac OS X, Windows) Python distribution for data analytics and scientific computing.

**Command for installing pandas:**

conda install pandas

**To update pandas to the latest version in anaconda or miniconda use:**

conda update pandas

**Installing pandas with pip:**

pip install pandas

**Import pandas library**:

Data Structure in pandas:

* Series
* DataFrame

**Series:**

Series is a one-dimensional labelled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called index.

tape r t panaa s a « p‹t



Initialise Series

labels={ z', ’b','c']

T# ddtd]

d’f ’'

Passing Array as parameter

pd.Seriesiaata=orr!.

1 20

2

dtyge: int64

Passing List as parameter

pd.Scrics{my\_data,labels)

b Z0

dtype: int64

Passing dictionary as parameter

gd.series(d)

a 10

b 20

30

atype: int64

Deleting in Series

aci ps t r. 1

30

Updating in Series

•[\*i

O. t.

d 4O

dtype: int64

Iteration of Series

gs--gd.Seriea d) for number in ps:

p-int!numberi

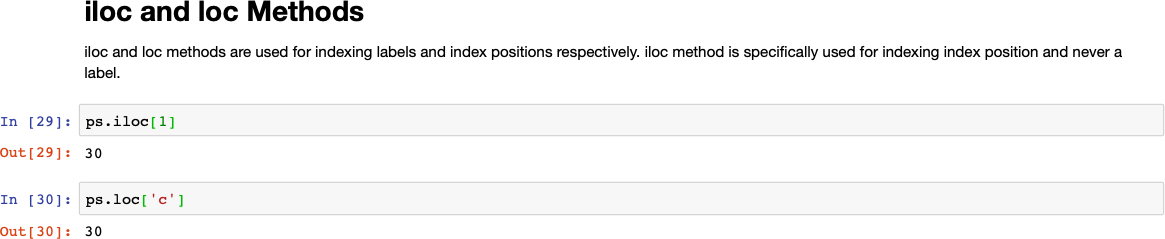
20

30

Iteration of Series along with their indexes

( ' b ' , 2 O )

( ' c ' , *5 0 )*

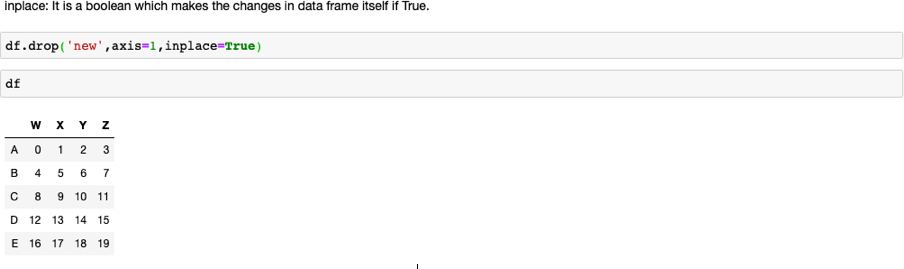
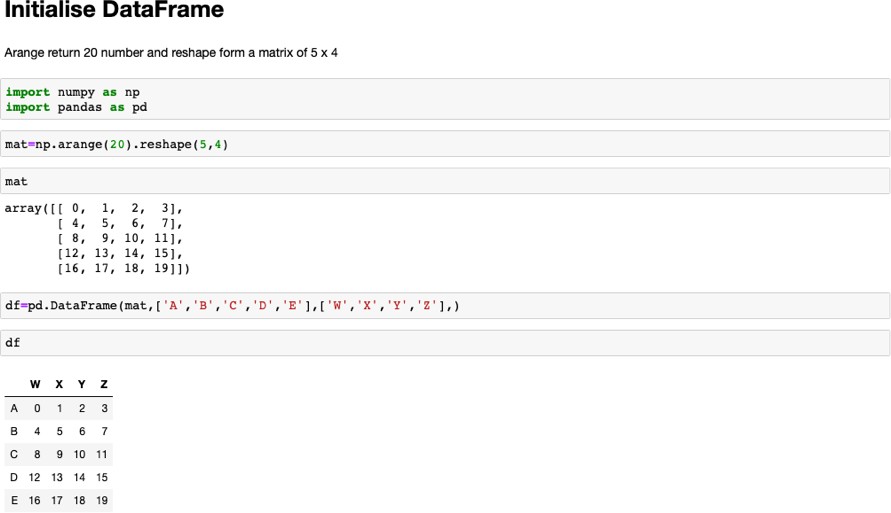


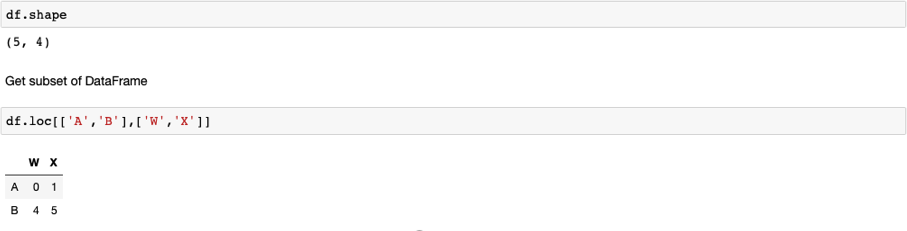
DataFrame

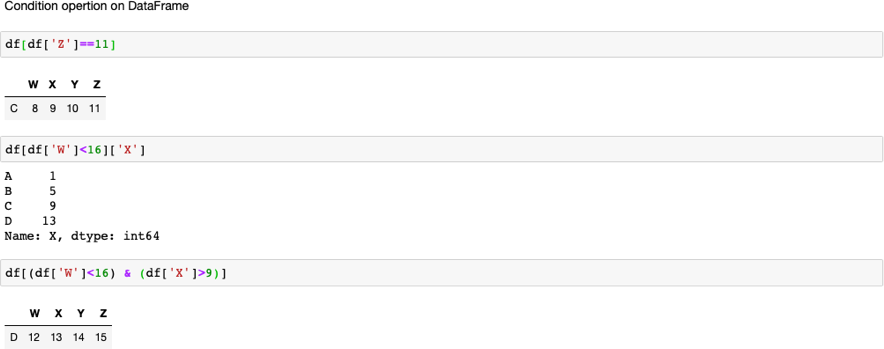
A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.

Features of DataFrame

* Potentially columns are of different types
* Size – Mutable
* Labelled axes (rows nd columns)
* Can Perform Arithmetic operations on rows and columns



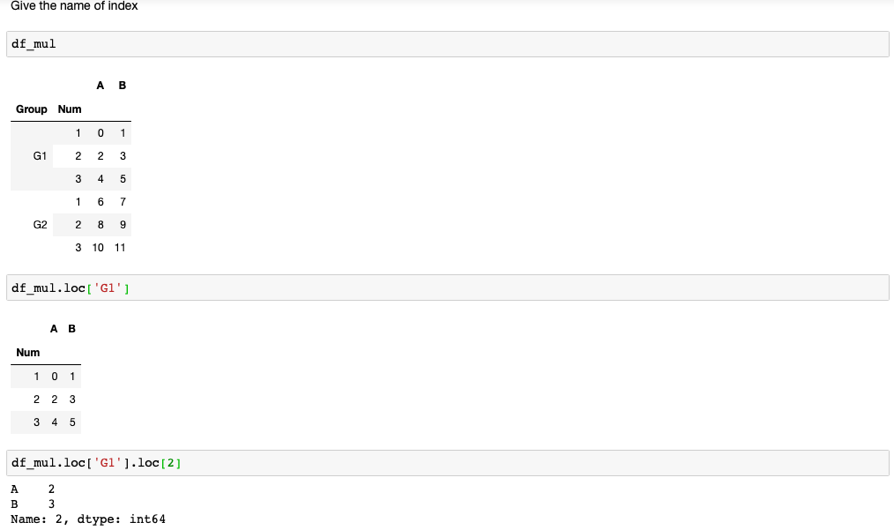






Multiple DataFrame





Matplotlib



**Introduction**

Matplotlib is the "grandfather" library of data visualization with Python. It was created by John Hunter. He created it to try to replicate MatLab's (another programming language) plotting capabilities in Python. So if you happen to be familiar with matlab, matplotlib will feel natural to you.

It is an excellent 2D and 3D graphics library for generating scientific figures. Some of the major Pros of Matplotlib are:

* Generally easy to get started for simple plots
* Support for custom labels and texts
* Great control of every element in a figure
* High-quality output in many formats
* Very customizable in general

Matplotlib allows you to create reproducible figures programmatically. Let's learn how to use it! Before continuing this lecture, I encourage you just to explore the official Matplotlib web page: <http://matplotlib.org/>

**Installation or Setup**

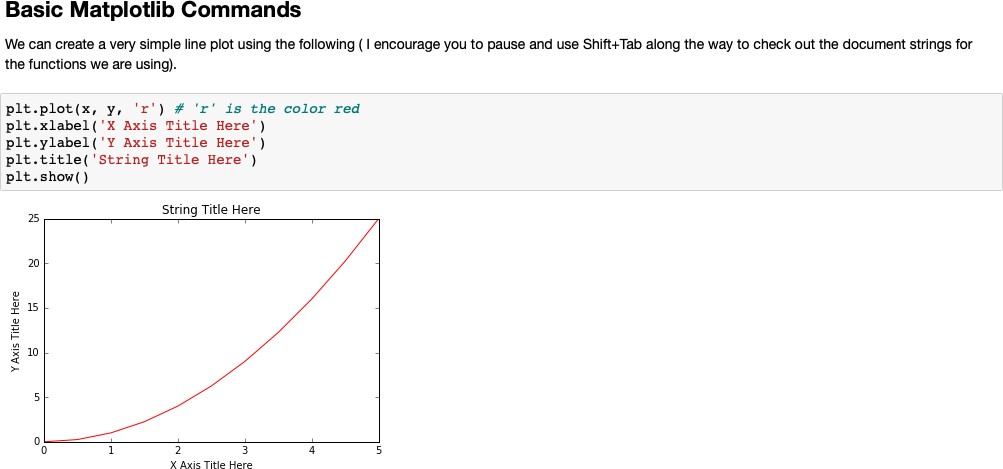
Install it with pip or conda at command line or terminal

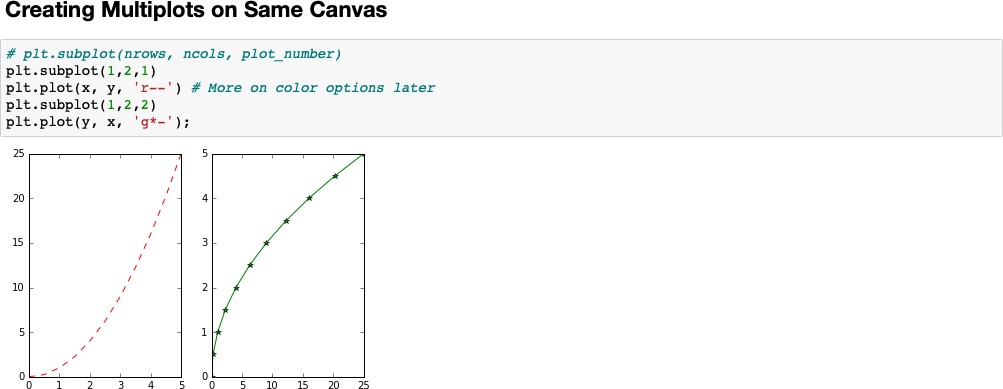
* conda install matplotlib
* pip install matplotlib

**Importing**

Import the matplotlib.pyplot module under the name plt:





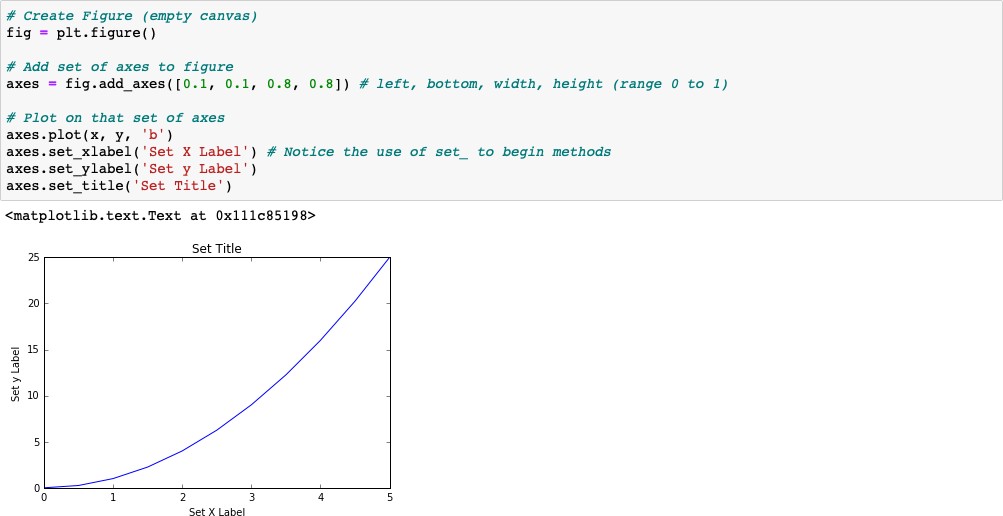


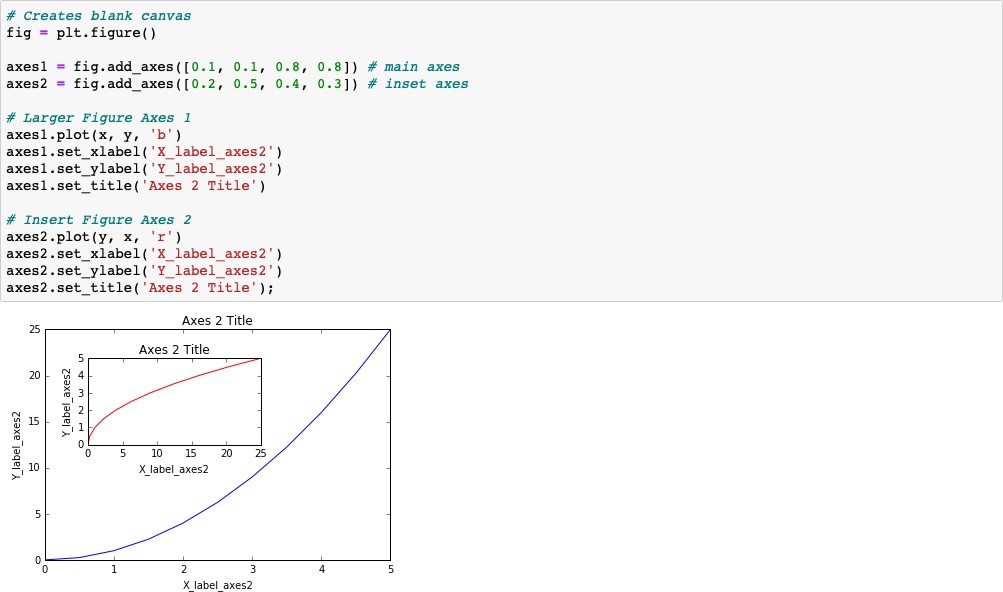
**Matplotlib Object Oriented Method**

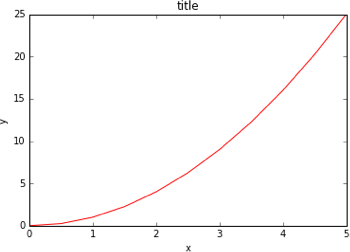
Now that we've seen the basics, let's break it all down with a more formal introduction of Matplotlib's Object Oriented API. This means we will instantiate figure objects and then call methods or attributes from that object.

**Introduction to the Object Oriented Method**

The main idea in using the more formal Object Oriented method is to create figure objects and then just call methods or attributes off of that object. This approach is nicer when dealing with a canvas that has multiple plots on it.



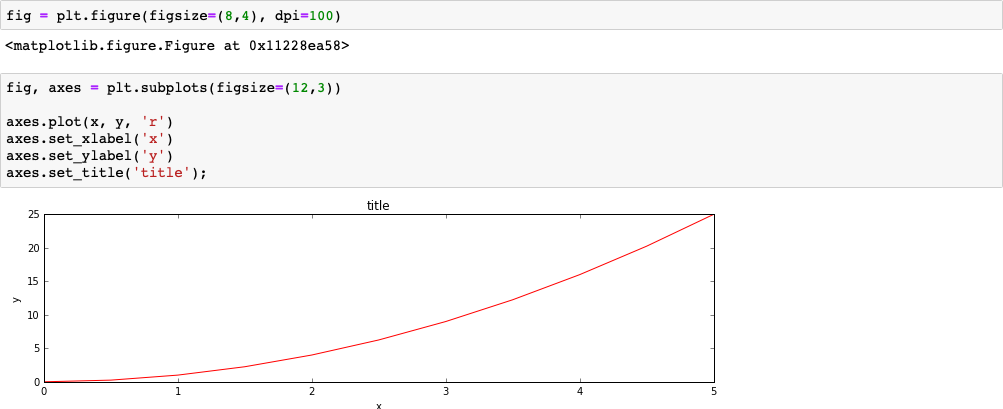




**Figure size, aspect ratio and DPI**

Matplotlib allows the aspect ratio, DPI and figure size to be specified when the Figure object is created. You can use the figsize and dpi keyword arguments.

* figsize is a tuple of the width and height of the figure in inches
* dpi is the dots-per-inch (pixel per inch).



**Saving figures**

Matplotlib can generate high-quality output in a number formats, including PNG, JPG, EPS, SVG, PGF and PDF.

To save a figure to a file we can use the savefig method in the Figure class:



**Legends, labels and titles**

Now that we have covered the basics of how to create a figure canvas and add axes instances to the canvas, let's look at how decorate a figure with titles, axis labels, and legends.

**Figure titles**

A title can be added to each axis instance in a figure. To set the title, use the set\_title method in the axes instance:

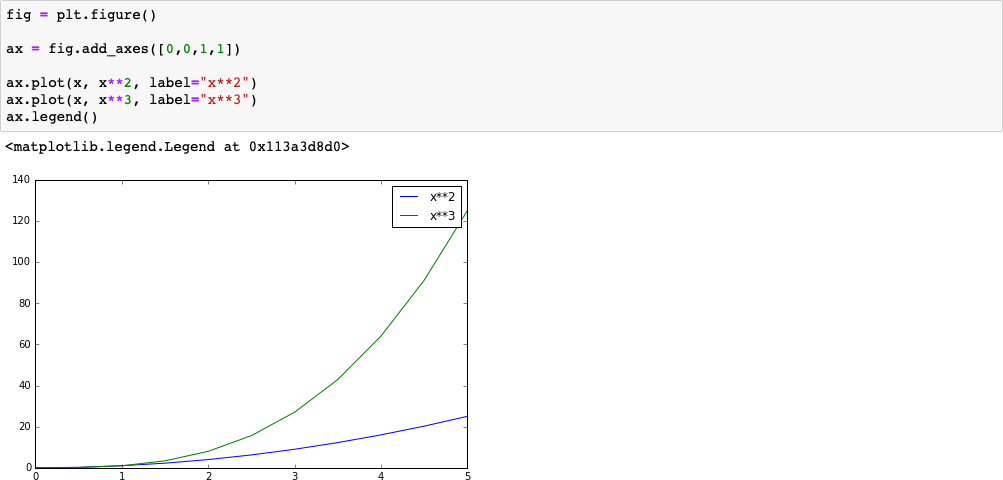
**Axis labels**

Similarly, with the methods set\_xlabel and set\_ylabel, we can set the labels of the X and Y axes:



Legends

You can use the label="label text" keyword argument when plots or other objects are added to the figure, and then using the legend method without arguments to add the legend to the figure:



Line and marker styles

To change the line width, we can use the linewidth or lw keyword argument. The line style can be selected using the linestyle or ls keyword arguments:



