In this plot, I will explain how to draw a graph using the Jupyter notebook with the matplotlib plotting library, which is the standard 2D plotting library for Python.

 Although we will only introduce the very basic functions in this plot, matplotlib can produce various kinds of publication quality figures with minimal effort.

 If you are interested in obtaining more information, you should visit their gallery page to see examples of the various types of plots can be generated.

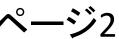
- First, we should import the libraries needed to make the graphs.
- For this, type the following commands in a new cell and run them.

 Note that Python will ignore anything that comes after the # symbol, it is just a comment that you add to make your code more readable.

 So you do not have to type them to get the examples to work.

- The 1st line is a 'magic' ipython command that controls the environment settings.
- Here we are instructing the system to output graphs inside the notebook itself.

 If you do not include this command, the graphs will be plotted in separate window.



 The 2nd line is used to import the "numpy" library, which we have already used previously. • The 3rd line is to import the "pyplot" library, which is a subset of "matplotlib" and provides a MATLABlike plotting framework, use it with a shorter name The 4th line is to use a customized style sheet called "ggplot" which modifies the default plot settings to produce more beautiful plots.

 These libraries will be extensively used in this course to plot all data and functions.

- As the 1st example, let us try to draw a simple sin function.
- Create a new cell and type the following commands and run them.

 You will finally obtain a graph of sin(x) for x equlas from -3 to +3 in a linear scale. Here the description of each command is not explained in detail but given as a short comment after #. If you are unsure about any commands, please refer to one of the many free online resources, such as the official matplotlib or Python websites.

 For the 2nd example we will draw several functions, which are simple powers of x in a log-log scale.

 First, define the functional form to be plotted using a "def" block as shown here.

ページ4

 Create a new cell and type the following commands and run them.

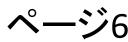
- The "def" keyword defines a new function, called "func," which takes two parameters x and n.
- The function will return x to the power of n.

- Then, type the following commands in a new cell and run them to draw the functions in a log-log scale.
- The description of each command is given after #.

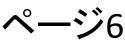
 The main part is between the 2nd and 4th lines where the three power functions, x, x^2 , and x^3 are plotted versus x.

 By plotting them on a log-log scale, all three functions appear as straight lines, with slopes of one, two, and three respectively.

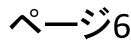
 In this example, the labels and legends are also plotted together with the functions on the same graph.



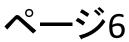
 The 3rd example is to draw a histogram of a collection of random-data points. As you can see in the following cell, the 3rd line generates a sequence of 100,000 random numbers, uniformly distributed between 0 and 1, and then stores them in an array R.



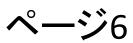
 The 4th line is the main part of this example, where the normalized histogram of R is calculated using 100 bins and plotted as a graph using a single command.



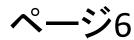
• Like this, the "hist" function allows us to calculate the histogram of any array data very easily.



 Finally by plotting the histogram, we can confirm that the distribution of generated random numbers is really uniformly distributed between 0 and 1.



As you have seen here,
Python has very powerful and easy-to-use graphical capabilities.



 In particular, we will repeatedly use the "hist" function in this course. The 4th example is to draw trajectories of random steps, which describes a process called a 1-dimensional random walk.

 A trajectory means the sequence of temporal positions, usually as a function of time or number of steps.

 In the code example shown here, the 6th line generates ten trajectories of 10,000 random steps of +1 or -1., and then stores them in an array named step.

 The positions of ten independent random walkers are calculated at each step by accumulating the individual values of step, from zero to the number of steps.

 Finally by plotting the position as a function of number of steps, we can visualize the trajectories of the random walkers.