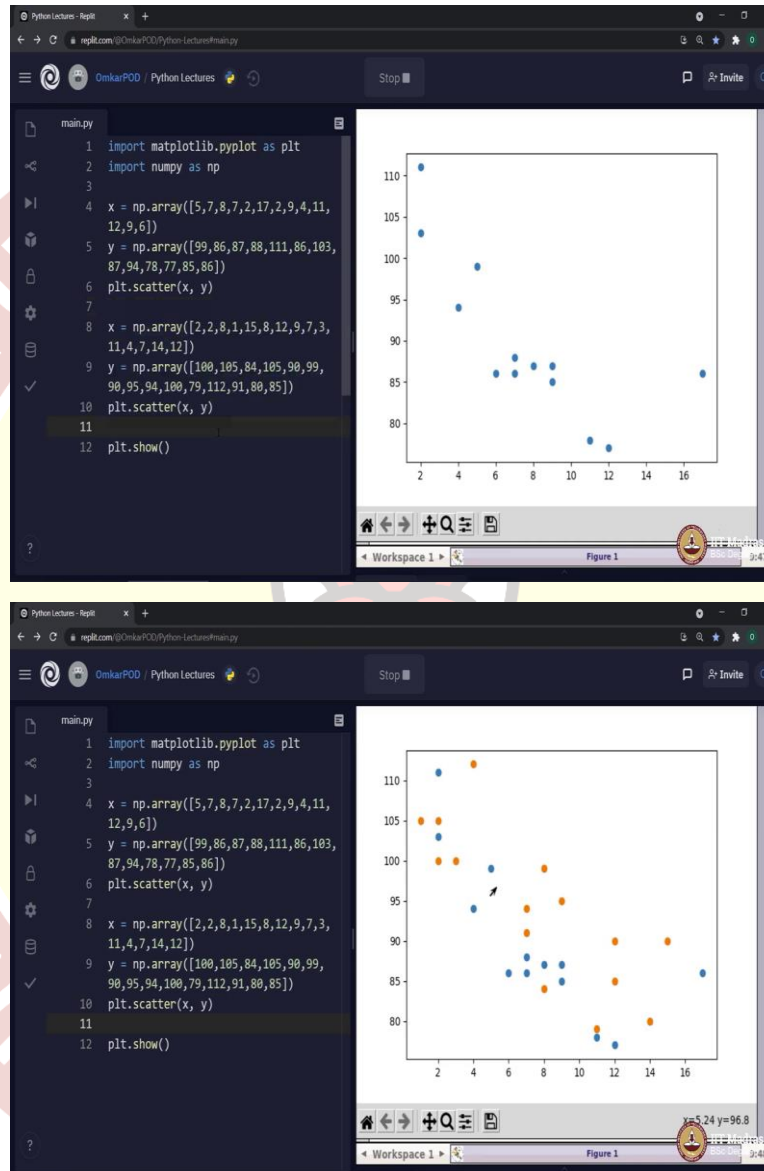


IIT Madras

ONLINE DEGREE

Programming in Python
Professor Sudarshan Iyengar
Department of Computer Science and Engineering
Indian Institute of Technology, Ropar
Introduction to Matplotlib library

(Refer Slide Time: 0:16)



Hello python student. Pandas was the first external library which we saw earlier and that provided us a powerful tool for data manipulation. Then we briefly introduced another external library called NumPy which is popular for its functional support in the scientific computing.

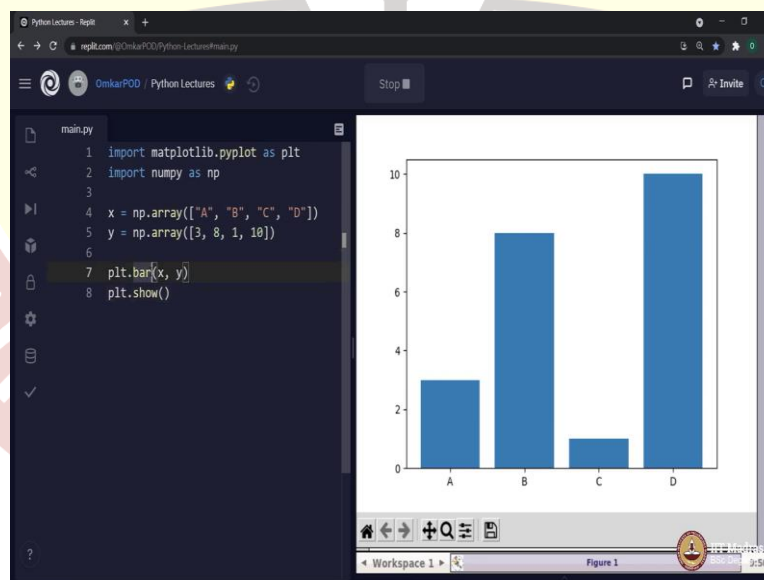
And now in this lecture we will introduce third external library called Matplotlib. And as the name suggest Matplotlib provides a useful package for data visualization. So, let us look at

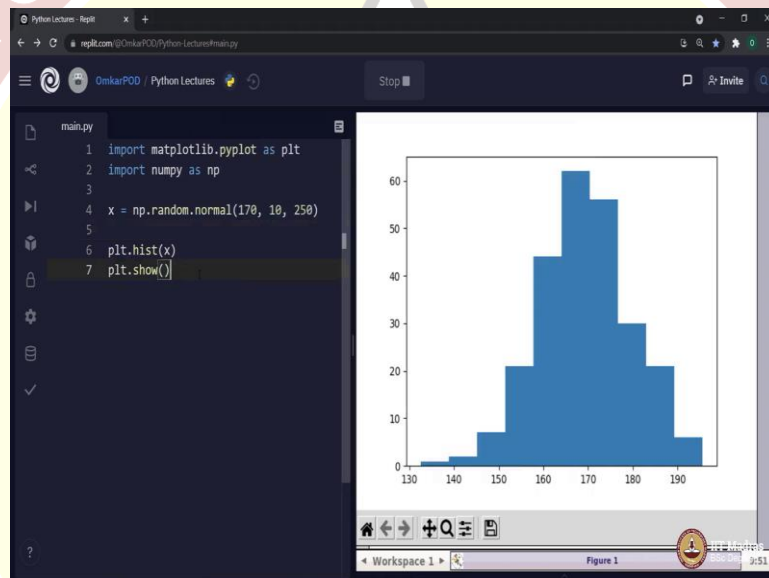
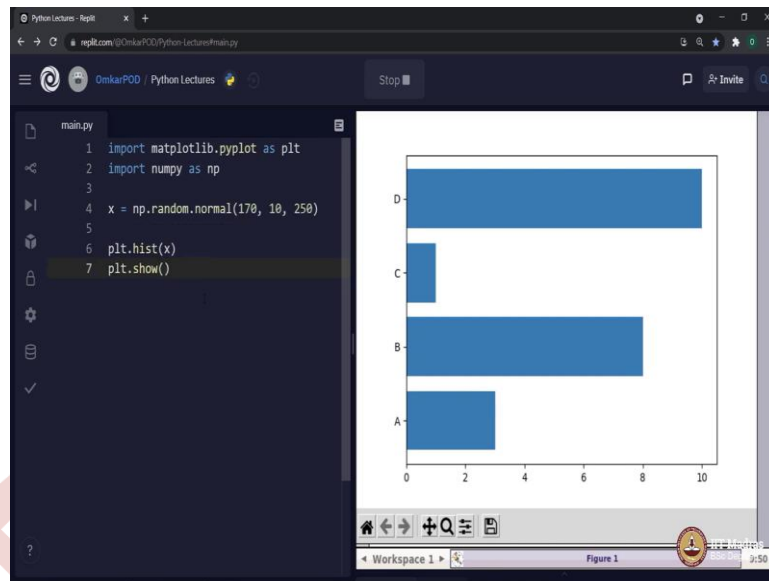
few commonly used functions. Look at this particular code. We have imported one module called pyplot from matplotlib as plt. And secondly, we are using this NumPy library in order to create NumPy arrays.

Here we have two arrays x and y with some numbers. Then using this particular function plt dot scatter and these two parameters x and y we should be able to get a scatter plot with these numbers. And in order to display this scattered plot as an output we are writing this particular function plt dot show. Let us execute this program. As you can see on the right-hand side of the screen which is R console you will see this particular message because of matplotlib.

And then at the top you will see that right hand side output window is split into two and you can simply drag it below in order to see the full-size output which is the scatter plot. Replete also allows you to shape this particular graph as an dot png file which you can use later. Let us add one more sample into this scattered plot and now let us execute this code again. As you can see now we have the same scattered plot with two different samples, one in blue and other in orange. These are the default colours provided for scattered plot but you can always configure your own colours and styles. Let us move to next function.

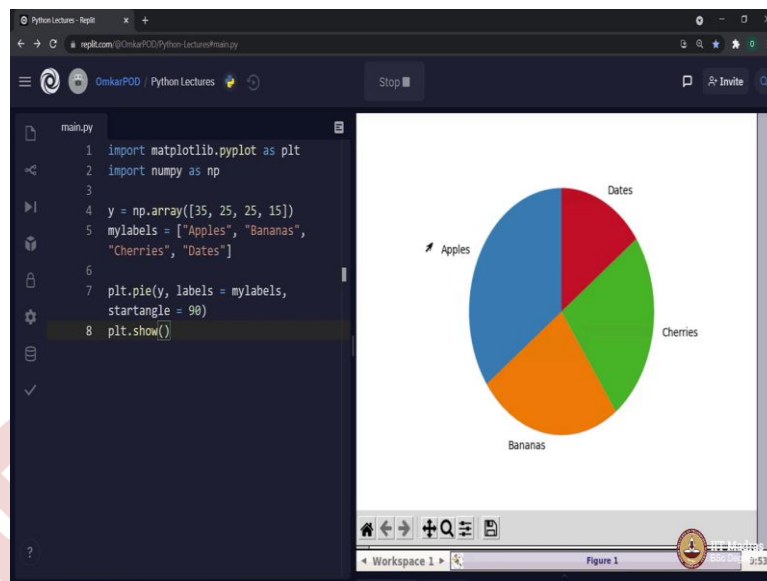
(Refer Slide Time: 3:01)





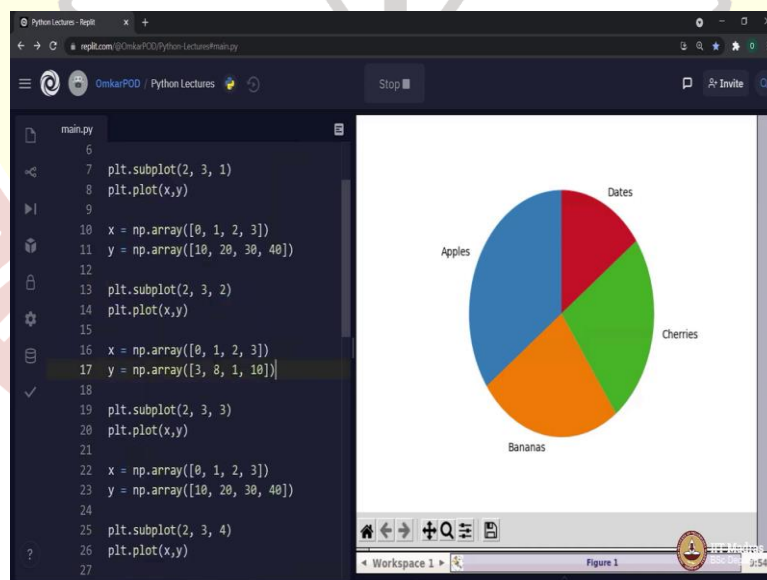
This is the function which we will use to get bar chart. Let us execute, in case if we want an horizontal bar chart, then the function is `barh`. Let us execute. Now let us look at third function which is histogram and the function used is `hist`. and the line number 4 might be new for you, but this is the way we generate random numbers using NumPy. Let us execute and we got the histogram.

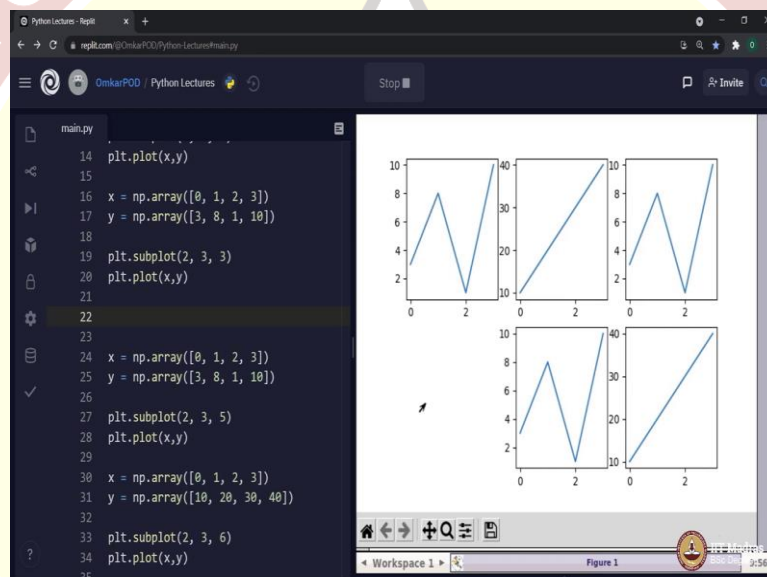
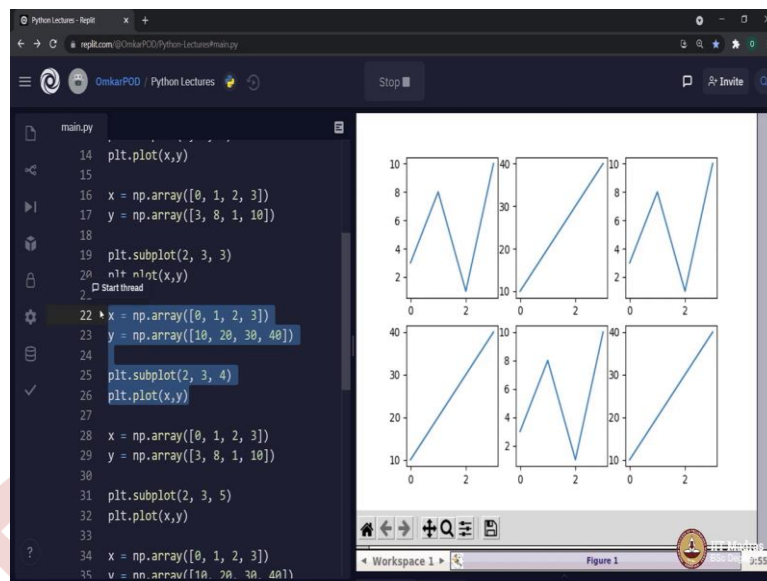
(Refer Slide Time: 3:56)



Next function is pie chart and for this the function is plt dot pie. Let us execute that is our pie chart. These are commonly used visualization techniques. In addition to that matplotlib also provides one additional feature something called as subplot which allows us to divide this particular visualization area into subgraphs and that each divided block can be used for a different graph. This feature is very useful in real world. Let us look at on such example.

(Refer Slide Time: 4:32)





This particular code might look very complicated and lengthy, but if you go into details it is very straight forward and simple Python program. The function which is used to get this subplot is `plt dot subplot` and it takes three parameters. First two parameters refer the number of columns and rows in which this particular area will be divided. Whereas, third parameter defines at which specific location this particular plot will go.

As you can see, we have divided this visualization area into 2 rows and 3 columns and this data will be used for our first graph. Then this data will be used for second graph. This one for third, this one for fourth, fifth and sixth. Once everything is ready, we will call this function `plt dot show`. Let us execute then you will understand it better. The same visualization area is now divided into 2 rows and 3 columns. Each subgraph is different based on the numbers which we gave in these NumPy arrays.

For example, if you remove this particular which is the fourth one and executed this program, then we will get all remaining plots at their respective position except the fourth one. So, it counts the plots from left to right from top to bottom. Let us add that one back and that is our full-size subplot. You must have noticed that is very simple and straight forward stuff.

These were only few samples. Apart from these, there are many other data visualization functions available in this library. So, explore this library and play around with many different parameters associated with these visualization techniques. Thank you for watching this lecture. Happy learning.

