Summary of Used Code in the Tutorial

Base

Objects

Data Type	When to use it	Methods/Operators	Example
Bool		Declare with True or False	`x = True`
Integer		Declare with integer number	`x = 1`
Float		Declare with float number	`x = 1.0`
String		Declare with commas	`x = "I am a string"`
		+	`>>> "I am" + " a string"` `"I am a string"`
List	To contain a list of objects	Declare with brackets	`x = [1,"3",4.,"elephant"]`
		Access by element	`>>> x[1]` `1`
		Access by slice	`>>> x[1:3]` `["3",4.]`
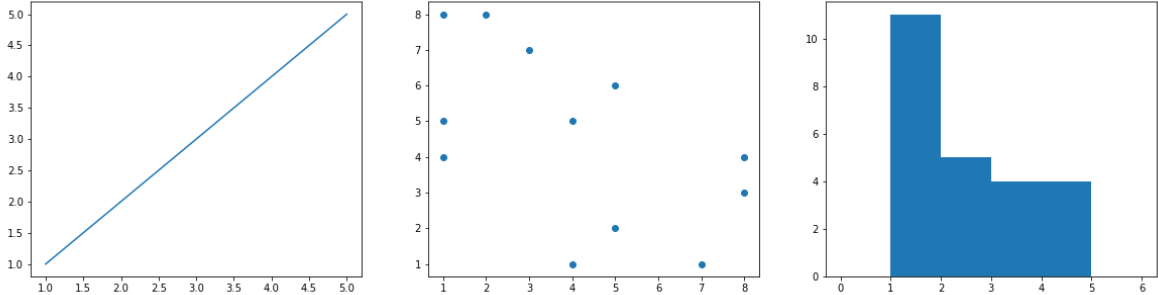
| | | | ```
>>> x[1:]
["3",4.,"elephant"]
>>> x[:3]
[1,"3",4.]
>>> x[:]
[1,"3",4.,"elephant"]
``` |
| | | Add elements | ```
>>> x.append(5.)
[1,"3",4.,"elephant",5.]
``` |
| Dictionary | To contain a list of objects to be accessed by a key | Declare with curly brackets and key:value separated by commas | ```
x = {"Var1":1,"Var2":"hi"}
``` |
| | | Access by key | ```
>>> x["Var1"]
1
``` |
| | | Get keys | ```
>>> x.keys()
["Var1","Var2"]
``` |

Functions

Function	Description	Example
| len | Get the length of a container | ```
>>> len([1,2,3])
3
``` |
| help | Print description of a object | ```
>>> help(len)
len(obj, /)
    Return the number of items in a container.
``` |
| print | Print the value of a variable | ```
>>> x = 3
``` |
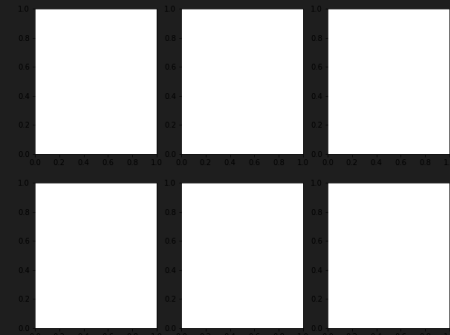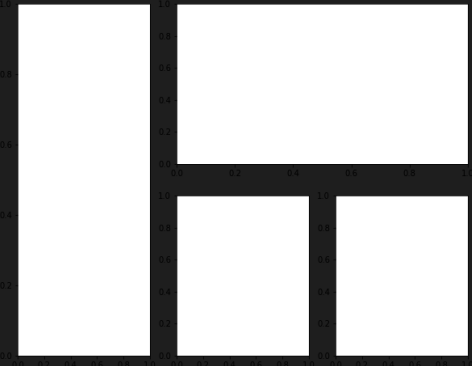
| | | ```
>>> print(x)
3
``` |
|---|---|---|

Matplotlib.Pyplot

Habitual abbreviation: plt

Objects

Data Type	When to use it	Methods/Operators	Example
Figure	To make an image	Constructor	`>>> fig = plt.figure()`
		Save	`>>> fig = plt.figure()` `>>> fig.savefig("Name.png")`
Axes	To make a panel in a figure	hist	`>>> ax.hist(x)`
		plot	`>>> ax.plot(x)`
		scatter	`>>> ax.scatter(x)`
		set_title	`>>> ax.set_title("Title",fontsize=20)`
		set_axis	`>>> ax.set_axis([0,1,0,2])`
		set_xlabel	`>>> ax.set_xlabel("Title",fontsize=20)`
		set_ylabel	`>>> ax.set_ylabel("Title",fontsize=20)`
		set_legend	`>>> ax.set_legend("Title",fontsize=20)`

Functions

Function	Description	Example
subplots	Make a figure and axis list with a grid of panels	```>>> fig, ax = plt.subplots(1,3,figsize=[20,5])```
subplots_mosaic	Make a figure and axis dictionary with a grid of panels	```>>> fig, ax = plt.subplot_mosaic([["A","B","B"],["A","D","E"]],figsize=[10,8])```

Combined examples

| Input code | ```
fig, ax = plt.subplots(1,3,figsize=[20,5])

ax[0].plot([1,2,3,4,5],[1,4,9,16,25])
ax[1].scatter([1,4,5,2,8,3,7,1,8,4,1,5],[4,5,2,8,3,7,1,8,4,1,5,6])
ax[2].hist([1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,3,3,3,3,4,4,4,4],bins=[0,1,2,3,4,5,6]);
#bins specify the ends of the bins
``` |
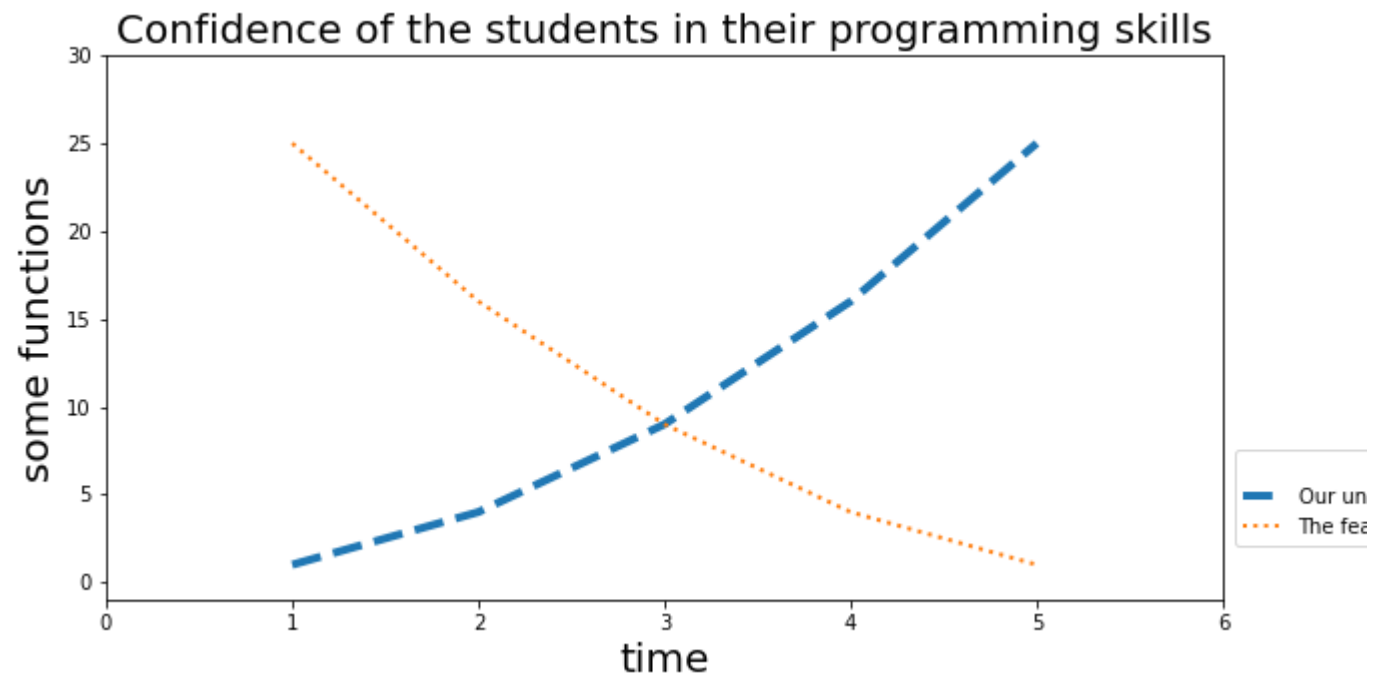|---|---|
| Output | |

| Input code | ```
fig, ax = plt.subplots(1,1,figsize=[10,5])

ax.plot([1,2,3,4,5],[1,4,9,16,25],linewidth=4,linestyle="dashed",label="Our understanding of Python")
ax.plot([1,2,3,4,5],[25,16,9,4,1],linewidth=2,linestyle="dotted",label="The fear of programming")

ax.set_title("Confidence of the students in their programming skills",fontsize=20)
ax.set_xlabel("time",fontsize=20)
ax.set_ylabel("some functions",fontsize=20)
``` |
|---|---|

```
ax.legend(title="Legend",loc=(1.01,.1))
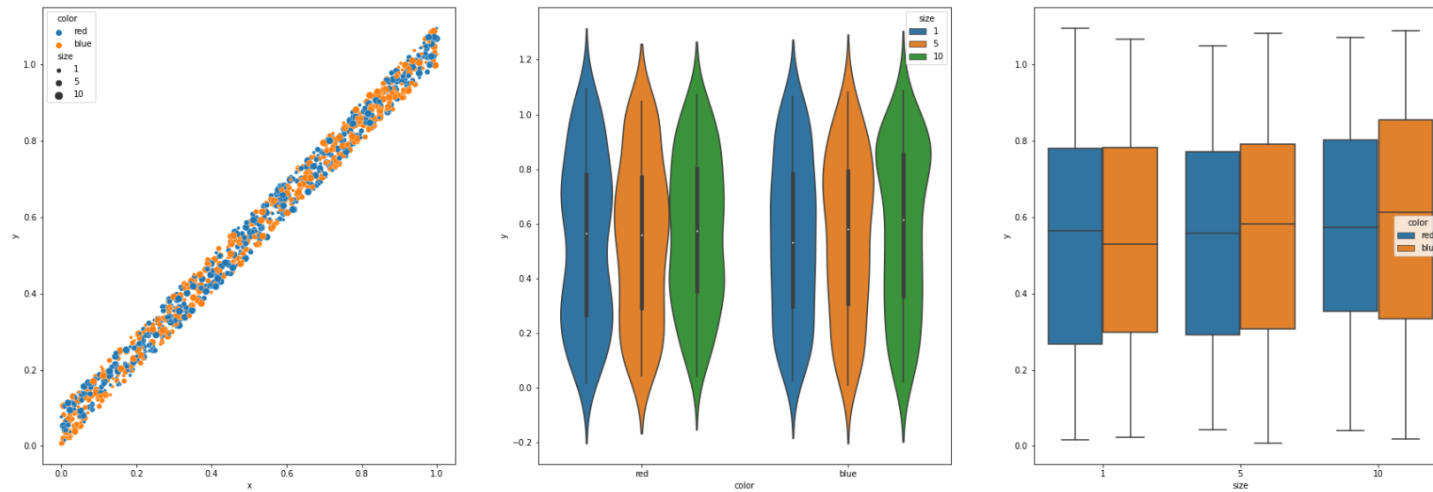ax.axis([0,6,-1,30]) #Set the limits of observation of the axis
```

Output

Confidence of the students in their programming skills

Seaborn

Habitual abbreviation: sns

Functions

Function	Description	Example
lineplot	Make a line plot	```>>> fig, ax = plt.subplots(1,1,figsize=[20,5])``` ```>>> sns.scatterplot(data=d,x="x",y="y",hue="color",ax=ax)```
scatterplot	Make a scatter plot	```>>> fig, ax = plt.subplots(1,1,figsize=[20,5])``` ```>>> sns.scatterplot(data=d,x="x",y="y",hue="color",size="size",ax=ax)```
violinplot	Make a violin plot	```>>> fig, ax = plt.subplots(1,1,figsize=[20,5])``` ```>>> sns.violinplot(data=d,x="color",y="y",hue="size",ax=ax)```
boxplot	Make a box plot	```>>> fig, ax = plt.subplots(1,1,figsize=[20,5])``` ```>>> sns.boxplot(data=d,x="size",y="y",hue="color",ax=ax)```

Combined example

Input code
 	
(assumes you have a pandas.DataFrame assigned to variable d)
	```fig,ax = plt.subplots(1,3,figsize=[30,10])```
	
	```sns.scatterplot(data=d,x="x",y="y",hue="color",size="size",ax=ax[0])```
	```sns.violinplot(data=d,x="color",y="y",hue="size",ax=ax[1])```
	```sns.boxplot(data=d,x="size",y="y",hue="color",ax=ax[2])```

Output	

Numpy

Habitual abbreviation: np

Objects

Data Type	When to use it	Methods/Operators	Example

Array	Container that has:
	- N dimensions
	- All the same data type
		Construct filled with zeros	```>>> a = np.zeros([2,3],int)```
			```array([[0, 0, 0],```
			``` [0, 0, 0]])```

		Construct filled with zeros	```>>> a = np.ones([2,3],int)``` ```array([[1, 1, 1],```

| | | | ```
    [1, 1, 1]])
``` |
| | | Get shape | ```
>>> x = np.array([1,2,3])
>>> x.shape
(3,)
``` |
| | | Get type | ```
>>> x = np.array([1,2,3])
>>> x.dtype
dtype('int64')
``` |
| | | Access by coordinate | ```
>>> x= np.array([[1,2,3],[4,5,6],[7,8,9]])
>>> x[0,0]
1
``` |
| | | Access by slicing | ```
>>> x= np.array([[1,2,3],[4,5,6],[7,8,9]])
>>> x[:,1:]
[[2 3]
 [5 6]
 [8 9]]
``` |
| | | Access by condition | ```
>>> x= np.array([[1,2,3],[4,5,6],[7,8,9]])
>>> get_even = (x % 2 > 0)
array([[ True, False,  True],
       [False,  True, False],
       [ True, False,  True]])
>>> x[get_even]
array([3, 5])
``` |
| | | Transpose | ```
>>> a = np.zeros([2,3],int)
array([[0, 0, 0],
       [0, 0, 0]])
>>> a.transpose()
``` |

```
array([[0, 0],
       [0, 0],
       [0, 0]])
```

Functions

Function	Description	Example
mean	Make mean of the array or along a specified axis	```>>> np.mean(a,axis=0)```
std	Compute the standard deviation of the array or of a specific axis	```>>> np.std(a,axis=0)```
corrcoef	Make the correlation matrix from a matrix of variables x observations	```>>> np.corrcoef(a)```

Pandas

Habitual abbreviation: pd

Objects

Data Type	When to use it	Methods/Operators	Example
DataFrame	Object to work with data tables	Construct	```>>> d = pd.DataFrame()```
		Construct from csv	```>>> d = pd.load_csv("data.csv", sep=",")```
		Save to csv	```>>> d.to_csv("data.csv")```
		Access by column	```>>> d["age"]```

		Access by location	```>>> d.loc[:3,"age"]```
		Access by index location	```>>> d.iloc[:3,:2]```
		Access by condition	```>>> get = d["color"] == "red"``` ```>>> d.loc[get,["x","y","color"]]```
		Get matrix in numpy array format	```>>> d.values```
		Compute mean	```>>> d.mean()```
		Compute standard deviation	```>>> d.std()```
		Group by condition	```>>> d.groupby("color").std()```