

Principles of Computer Science II

Python Data Visualization Library

Ioannis Chatzigiannakis

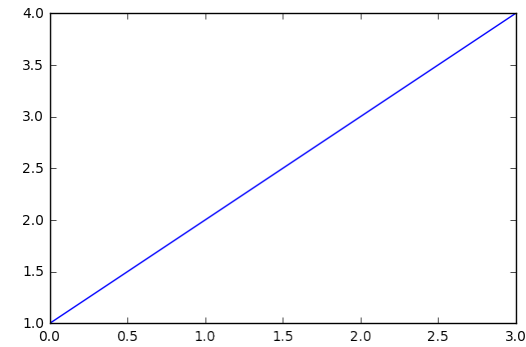
Sapienza University of Rome

Lecture 10



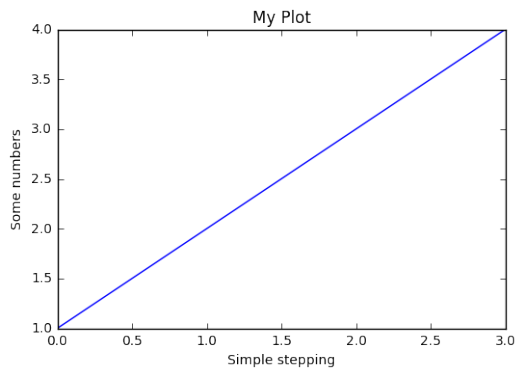
Simple Plotting

```
import matplotlib.pyplot as plt  
  
plt.plot([1,2,3,4])  
  
plt.show()
```

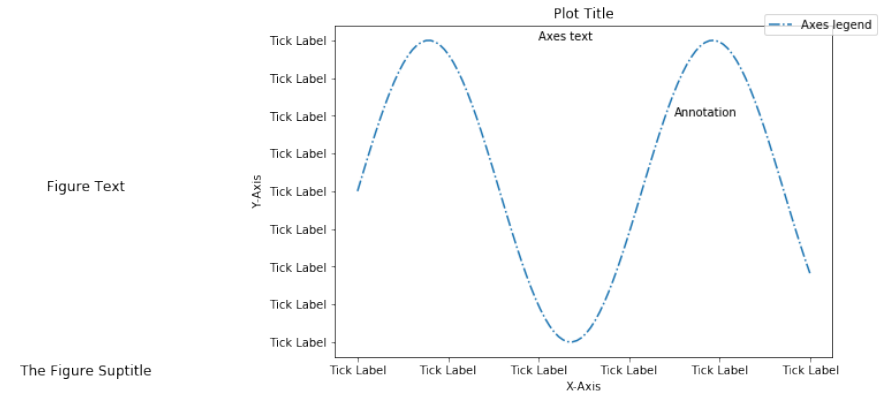


Axis Labels & Title

```
plt.plot([1,2,3,4])  
plt.ylabel('Some numbers')  
plt.xlabel('Simple stepping')  
plt.title('My Plot')  
plt.show()
```



Plotting Areas

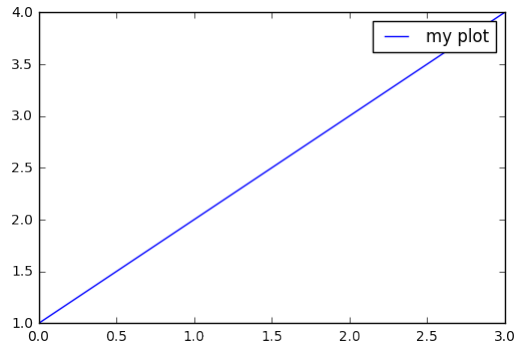


- ▶ A range of options are provided to parameterize the look&feel of the figure.
- ▶ `savefig("filename")` – plotting + store to file.



Plot Labels & Legend

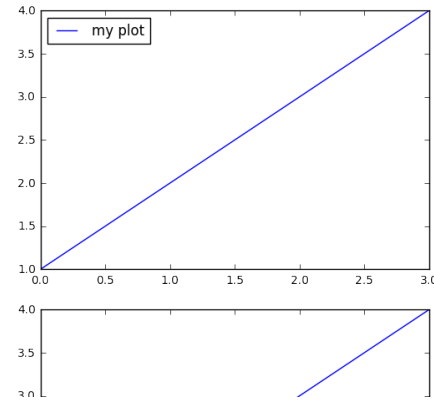
```
plt.plot([1,2,3,4], label='my plot')  
plt.legend()  
plt.show()
```



Legend Location

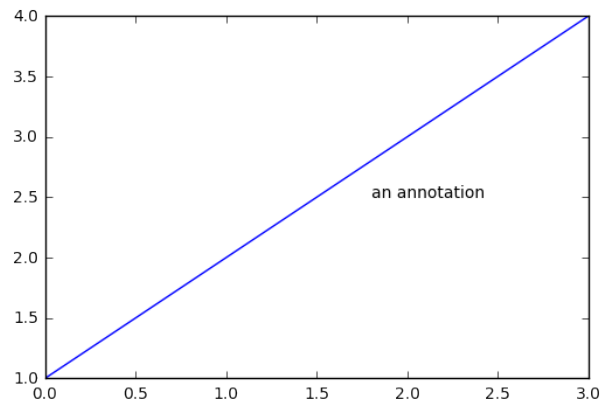
```
plt.plot([1,2,3,4], label='my plot')  
plt.legend(bbox_to_anchor=(0.3, 1))  
plt.show()
```

```
plt.plot([1,2,3,4], label='my plot')  
plt.legend(bbox_to_anchor=(1, 0.2))  
plt.show()
```



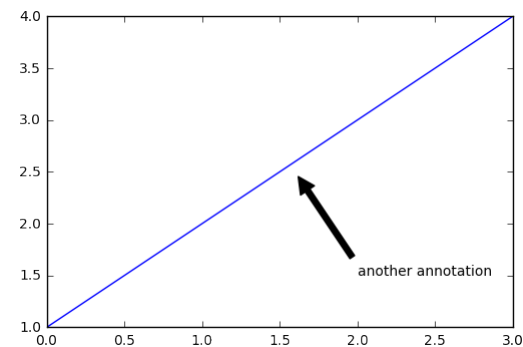
Simple Text

```
plt.plot([1,2,3,4], label='my plot')  
plt.text(1.8, 2.5, 'an annotation')  
plt.show()
```



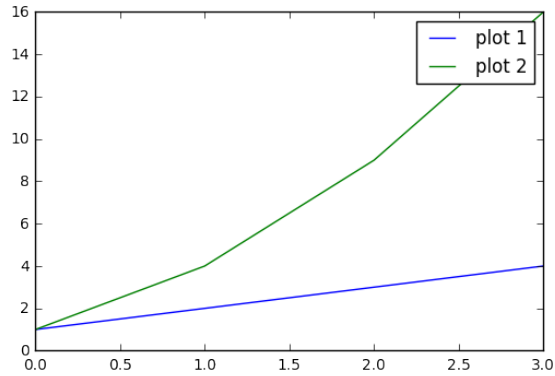
Text Annotations

```
plt.plot([1,2,3,4], label='my plot')  
plt.annotate('another annotation', xy=(1.6, 2.5),  
            xytext=(2, 1.5),  
            arrowprops=dict(facecolor='black', shrink=0.05))  
plt.show()
```



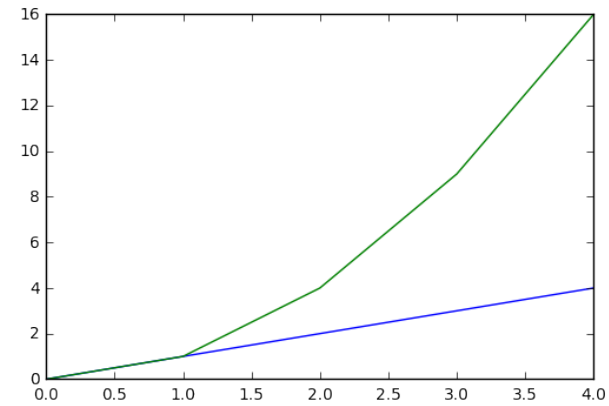
Multiple Plots

```
plt.plot([1,2,3,4], label='plot 1')  
plt.plot([1,4,9,16], label='plot 2')  
plt.legend()  
plt.show()
```



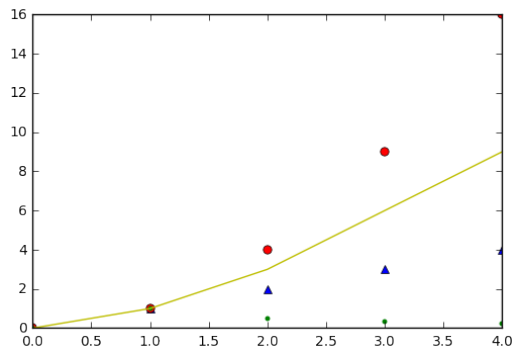
Plot X-axis Values

```
plt.plot([0,1,2,3,4], [0,1,2,3,4])  
plt.plot([0,1,2,3,4], [0,1,4,9,16])  
plt.show()
```



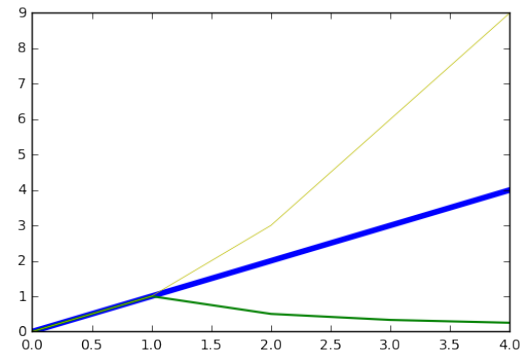
Marker Style

```
plt.plot([0,1,2,3,4], [0,1,2,3,4], 'b^')  
plt.plot([0,1,2,3,4], [0,1,4,9,16], 'ro')  
plt.plot([0,1,2,3,4], [0,1,.5,.33,.25], 'g.')  
plt.plot([0,1,2,3,4], [0,1,3,6,9], 'y-')  
plt.show()
```



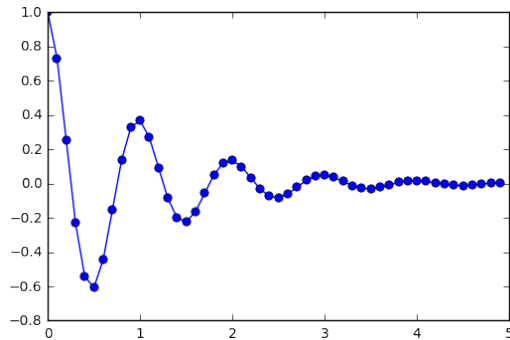
Line Style

```
plt.plot([0,1,2,3,4], [0,1,2,3,4], 'b', linewidth=4.0)  
plt.plot([0,1,2,3,4], [0,1,.5,.33,.25], 'g', linewidth=1.5)  
plt.plot([0,1,2,3,4], [0,1,3,6,9], 'y', linewidth=0.5)  
plt.show()
```



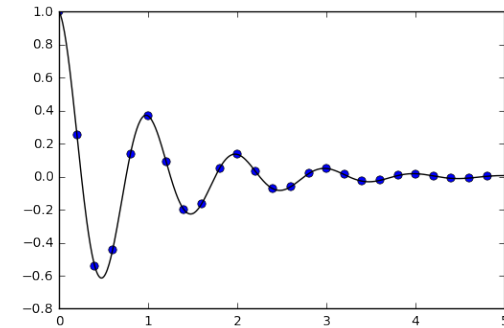
Line Style & Markers Style

```
import math
x = [i * 0.1 for i in range(0, 50)]
y = [math.exp(-i) * math.cos(2*math.pi*i) for i in x]
plt.plot(x, y, 'bo-', linewidth=1.0)
plt.show()
```



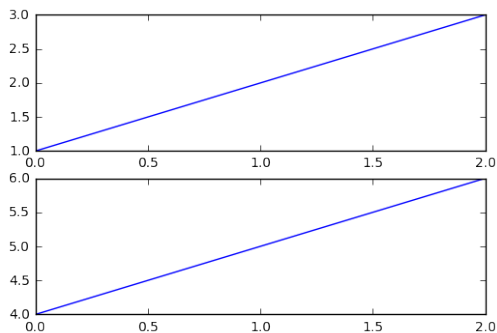
Separate Line Style, Marker Style + Marker positioning

```
x1 = [i * 0.2 for i in range(0, 25)]
y1 = [math.exp(-i) * math.cos(2*math.pi*i) for i in x1]
x2 = [i * 0.01 for i in range(0, 500)]
y2 = [math.exp(-i) * math.cos(2*math.pi*i) for i in x2]
plt.plot(x1, y1, 'bo', x2, y2, 'k')
plt.show()
```



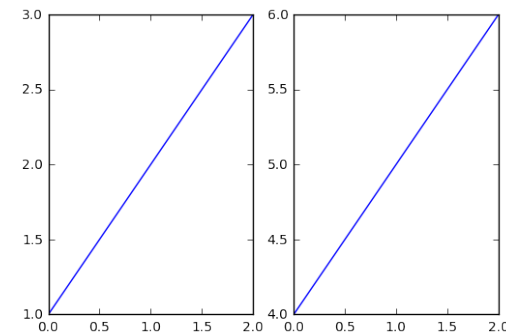
Multiple Subplots – Vertical

```
plt.figure()
plt.subplot(211)
plt.plot([1, 2, 3])
plt.subplot(212)
plt.plot([4, 5, 6])
plt.show()
```



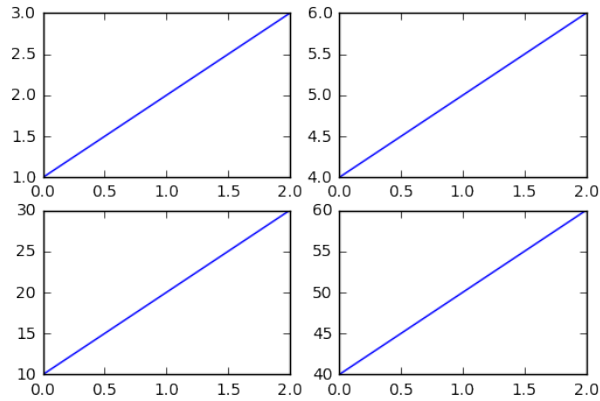
Multiple Subplots – Horizontal

```
plt.figure()
plt.subplot(121)
plt.plot([1, 2, 3])
plt.subplot(122)
plt.plot([4, 5, 6])
plt.show()
```



Multiple Subplots – Grid

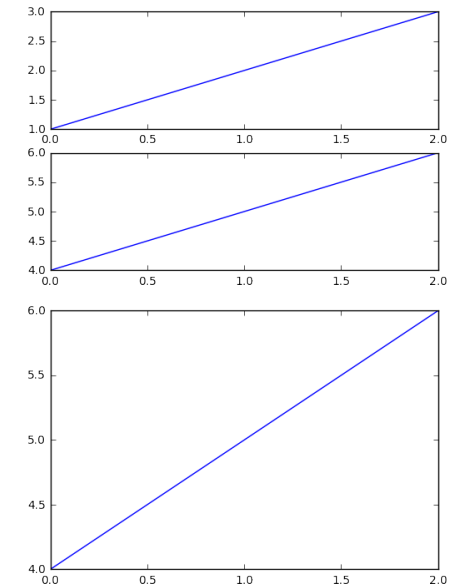
```
plt.figure()
plt.subplot(221)
plt.plot([1, 2, 3])
plt.subplot(222)
plt.plot([4, 5, 6])
plt.subplot(223)
plt.plot([10, 20, 30])
plt.subplot(224)
plt.plot([40, 50, 60])
plt.show()
```



Multiple Figures

```
plt.figure(1)
plt.subplot(211)
plt.plot([1, 2, 3])
plt.subplot(212)
plt.plot([4, 5, 6])
```

```
plt.figure(2)
plt.plot([4, 5, 6])
plt.show()
```

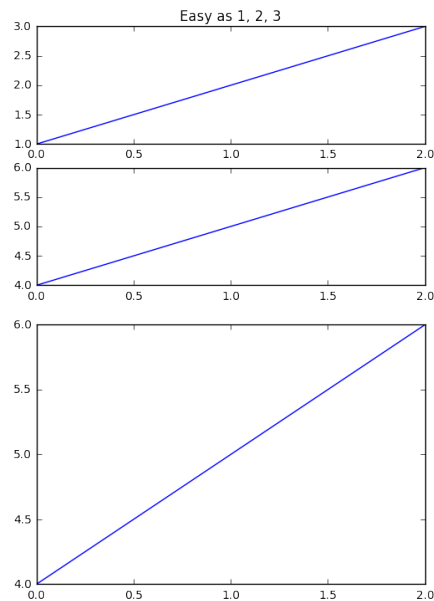


Multiple Figures – Concurrent Access

```
plt.figure(1)
plt.subplot(211)
plt.plot([1, 2, 3])
plt.subplot(212)
plt.plot([4, 5, 6])
```

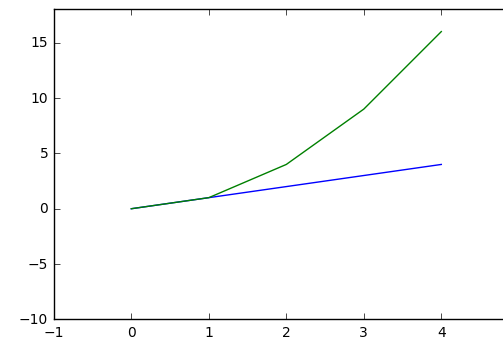
```
plt.figure(2)
plt.plot([4, 5, 6])
```

```
plt.figure(1)
plt.subplot(211)
plt.title('Easy as 1, 2, 3')
plt.show()
```



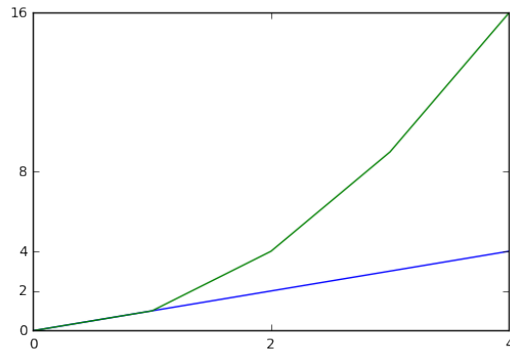
Controlling Axis Values

```
plt.plot([0,1,2,3,4], [0,1,2,3,4])
plt.plot([0,1,2,3,4], [0,1,4,9,16])
plt.xlim(-1,5)
plt.ylim(-10,18)
plt.show()
```



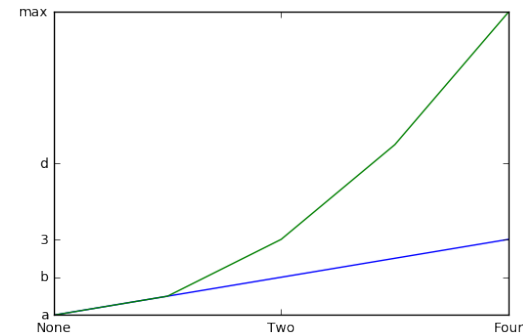
Axis Tick Positioning

```
plt.plot([0,1,2,3,4], [0,1,2,3,4])
plt.plot([0,1,2,3,4], [0,1,4,9,16])
plt.xticks([0,2,4])
plt.yticks([0,2,4,8,16])
plt.show()
```



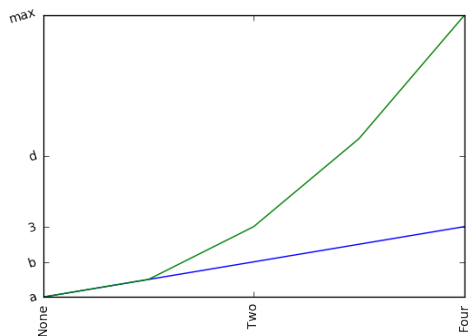
Axis Tick Positioning & Labels

```
plt.plot([0,1,2,3,4], [0,1,2,3,4])
plt.plot([0,1,2,3,4], [0,1,4,9,16])
plt.xticks([0,2,4], ['None', 'Two', 'Four'])
plt.yticks([0,2,4,8,16], ['a', 'b', '3', 'd', 'max'])
plt.show()
```



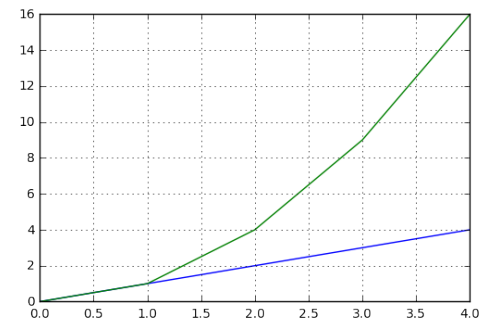
Axis Tick Positioning, Labels & Orientation

```
plt.plot([0,1,2,3,4], [0,1,2,3,4])
plt.plot([0,1,2,3,4], [0,1,4,9,16])
plt.xticks([0,2,4], ['None', 'Two', 'Four'], rotation='vertical')
plt.yticks([0,2,4,8,16], ['a', 'b', '3', 'd', 'max'], rotation=70)
plt.show()
```



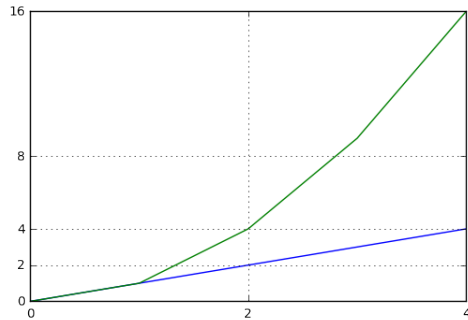
Simple Grid

```
plt.plot([0,1,2,3,4], [0,1,2,3,4])
plt.plot([0,1,2,3,4], [0,1,4,9,16])
plt.grid()
plt.show()
```



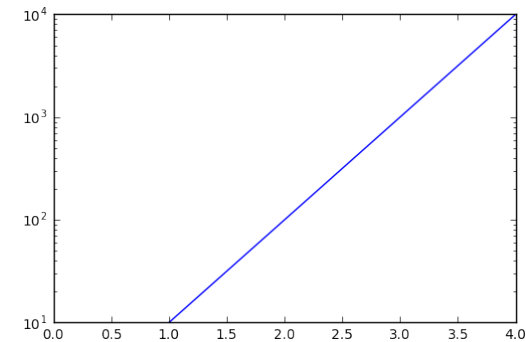
Ticks & Grid

```
plt.plot([0,1,2,3,4], [0,1,2,3,4])
plt.plot([0,1,2,3,4], [0,1,4,9,16])
plt.xticks([0,2,4])
plt.yticks([0,2,4,8,16])
plt.grid()
plt.show()
```



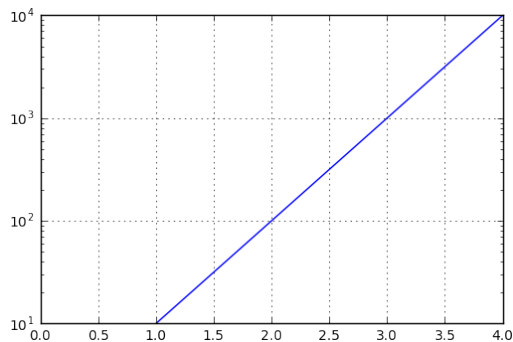
Logarithmic Scale

```
plt.plot([0,10,100,1000,10000])
plt.yscale('log')
plt.show()
```



Logarithmic Scale & Grid

```
plt.plot([0,10,100,1000,10000])
plt.yscale('log')
plt.grid()
plt.show()
```



3rd Assignment

- ▶ <https://www.rosalind.info/>
 - ▶ Complete the following **challenges**:
dna, rna, revc, iprb, gc, subs, lia, iev, cons, prob
 - ▶ <http://rosalind.info/problems/{challenge}>
- ▶ Create a GitHub repository and upload the code for each exercise.
- ▶ Email ichatz@diag.uniroma1.it
Subject: [PCS2] Homework 3
Your GitHub repository with your solutions, for all challenges.
Also send your account user account link:
<http://rosalind.info/users/{username}>

Deadline: 21 November 2022

