

Principles of Computer Science II

Python Data Visualization Library

Ioannis Chatzigiannakis

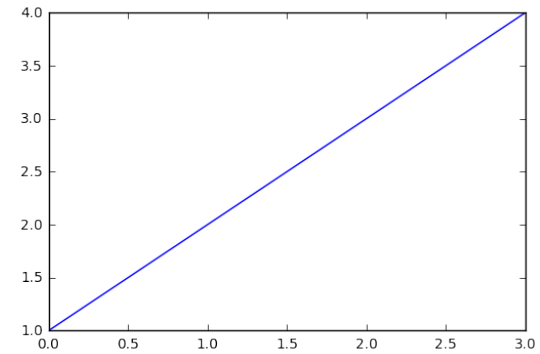
Sapienza University of Rome

Lecture 12



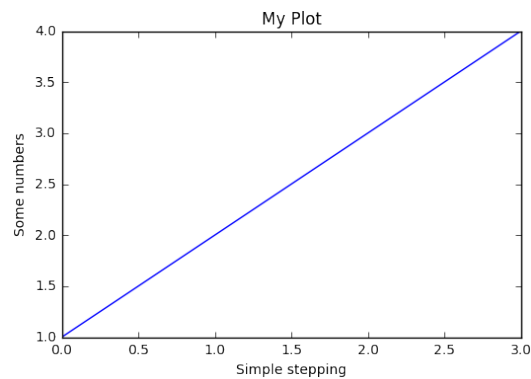
Simple Plotting

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

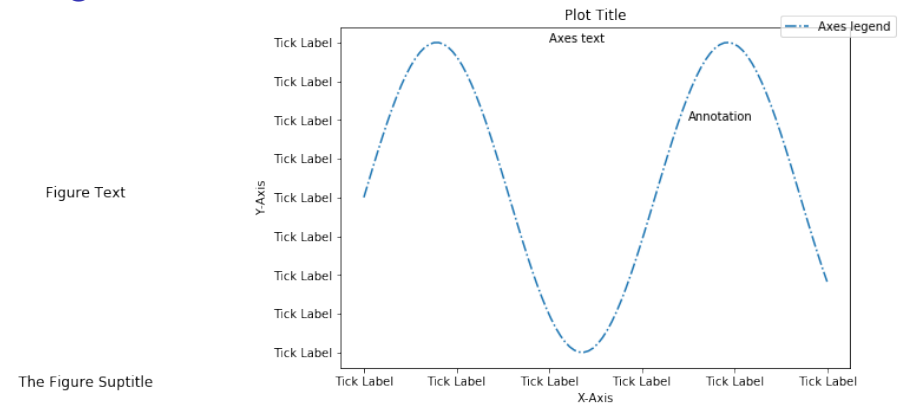


Axis Labels & Title

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



Plotting Areas

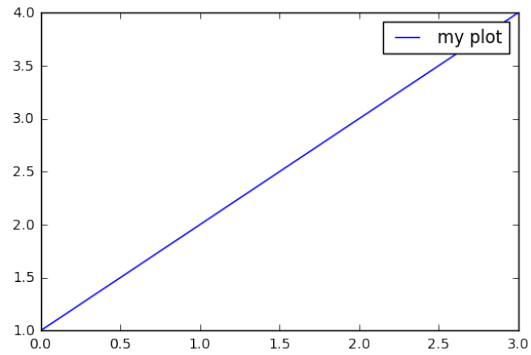


- ▶ A range of options are provided to parameterize the look&feel of the figure.
- ▶ `savefig("filename")` – plotting + store to file.



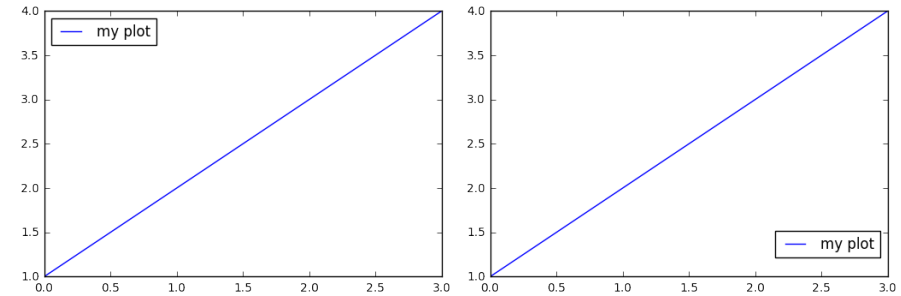
Plot Labels & Legend

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



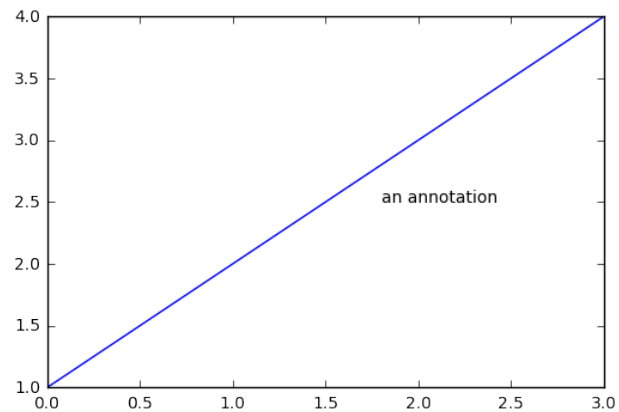
Legend Location

```
1 plt.plot([1,2,3,4], label='my plot')
2 plt.legend(bbox_to_anchor=(0.3, 1))
3 plt.show()
4
5 plt.plot([1,2,3,4], label='my plot')
6 plt.legend(bbox_to_anchor=(1, 0.2))
7 plt.show()
```



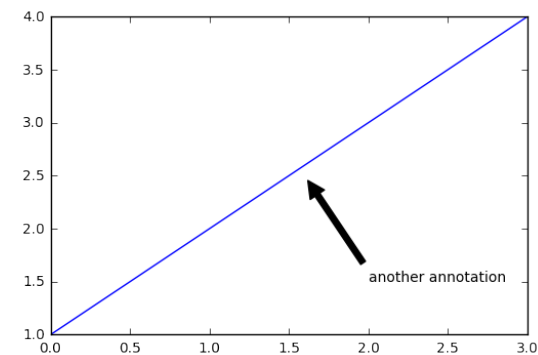
Simple Text

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



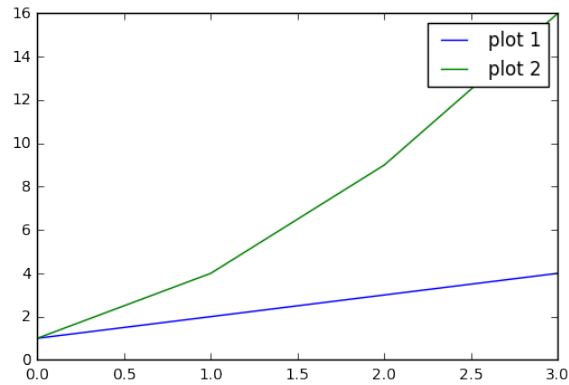
Text Annotations

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



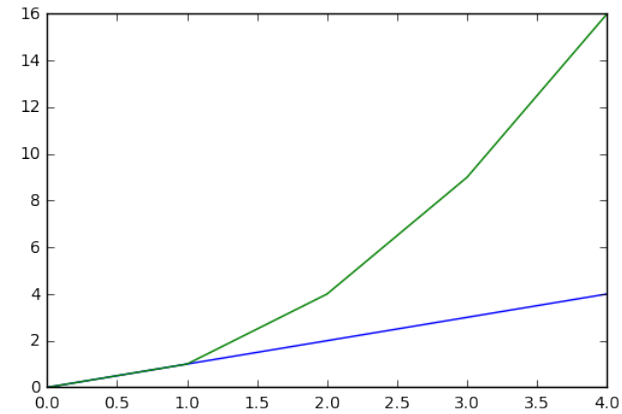
Multiple Plots

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



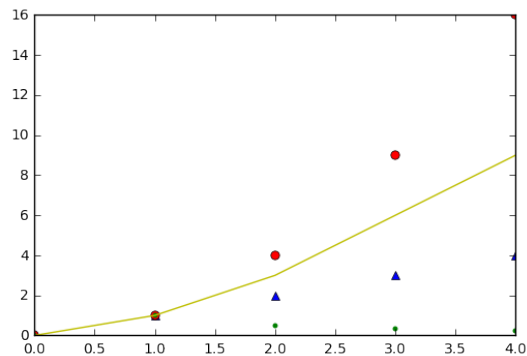
Plot X-axis Values

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



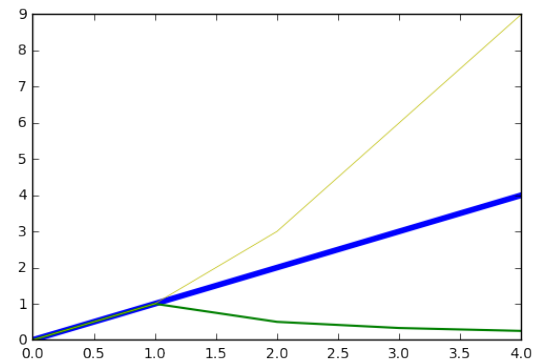
Marker Style

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



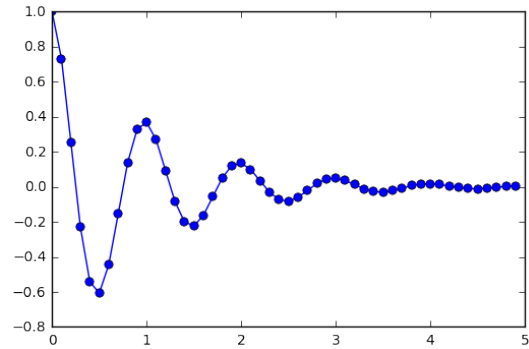
Line Style

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



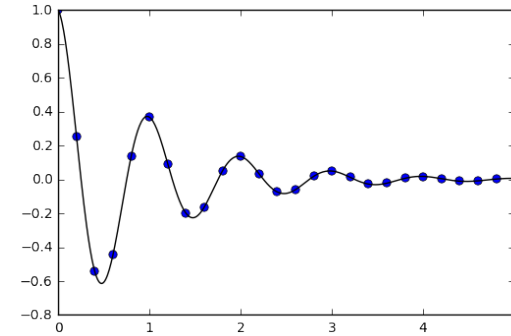
Line Style & Markers Style

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



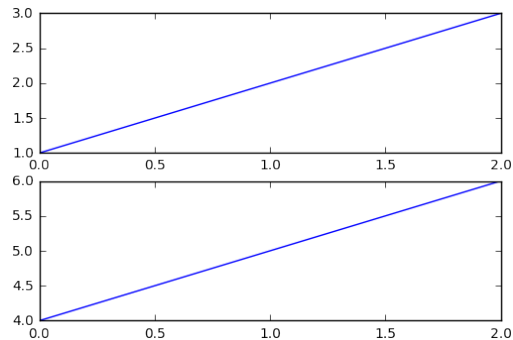
Separate Line Style, Marker Style + Marker positioning

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



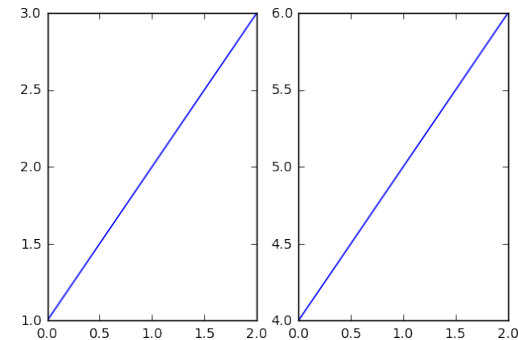
Multiple Subplots – Vertical

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



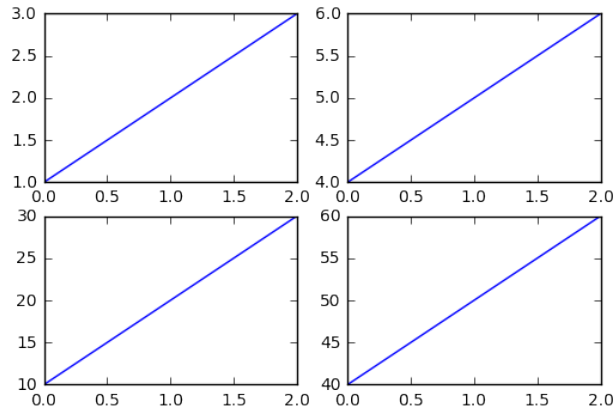
Multiple Subplots – Horizontal

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



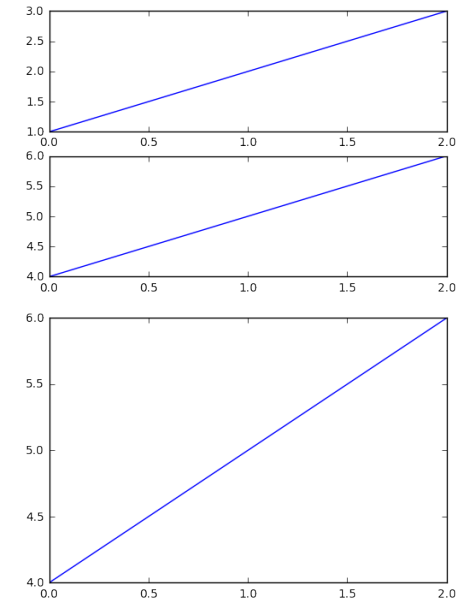
Multiple Subplots – Grid

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



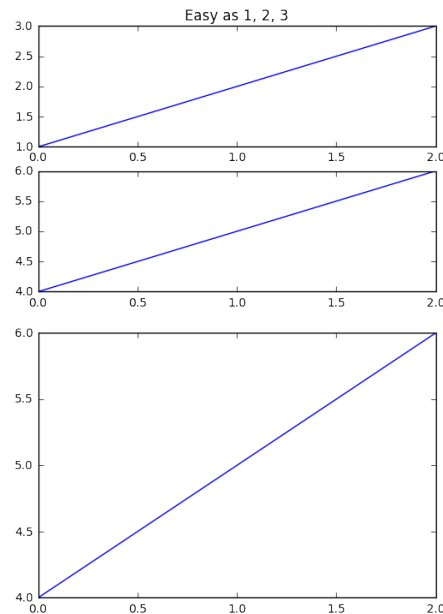
Multiple Figures

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



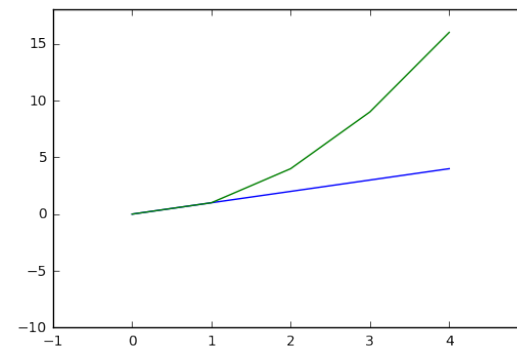
Multiple Figures – Concurrent Access

```
1 plt.figure(1)
2 plt.subplot(211)
3 plt.plot([1, 2, 3])
4 plt.subplot(212)
5 plt.plot([4, 5, 6])
6
7
8 plt.figure(2)
9 plt.plot([4, 5, 6])
10
11 plt.figure(1)
12 plt.subplot(211)
13 plt.title('Easy as 1, 2, 3')
14 plt.show()
```



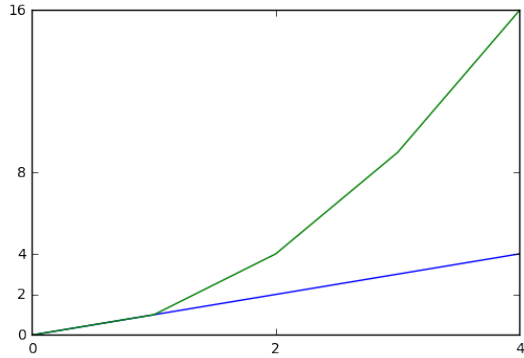
Controlling Axis Values

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



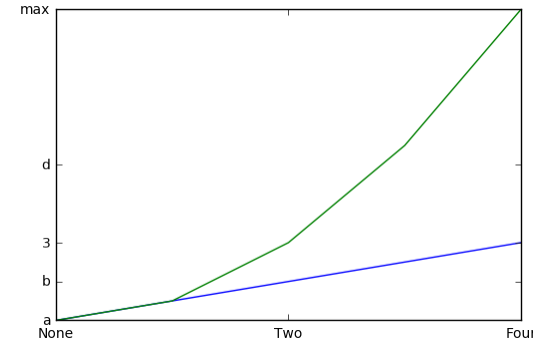
Axis Tick Positioning

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



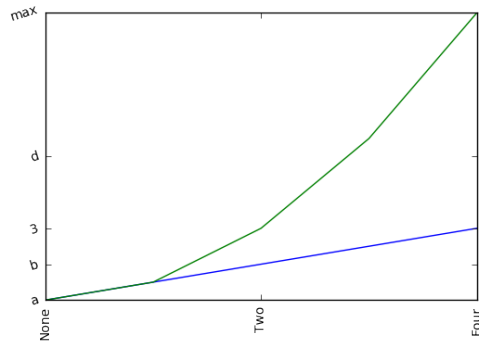
Axis Tick Positioning & Labels

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



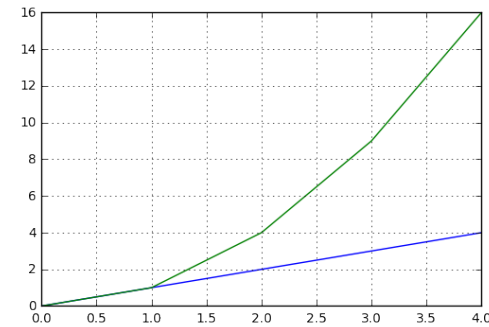
Axis Tick Positioning, Labels & Orientation

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



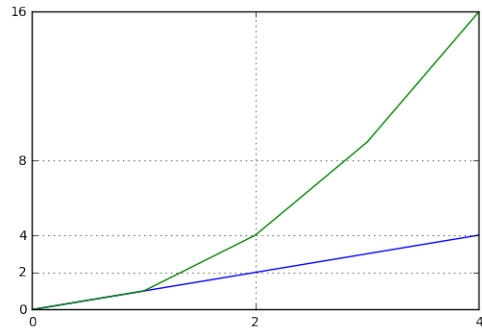
Simple Grid

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



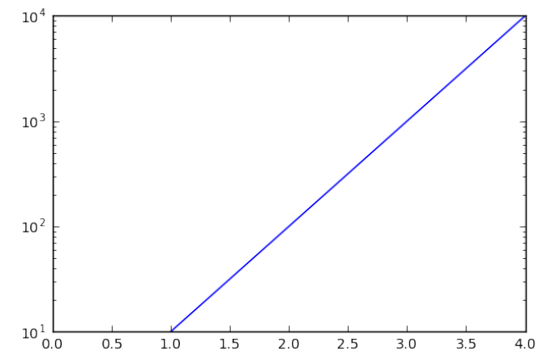
Ticks & Grid

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



Logarithmic Scale

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



Logarithmic Scale & Grid

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

