Introduction to Matplotlib

MRE/EME 5983 Robot Operating Systems

Overview

- What is matplotlib?
- matplotlib overview

What Is matplotlib?

- Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python
- We will focus on matplotlib.pyplot
- matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB
- Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

Source: https://matplotlib.org/stable/index.html

Introduction to matplotlib.pyplot

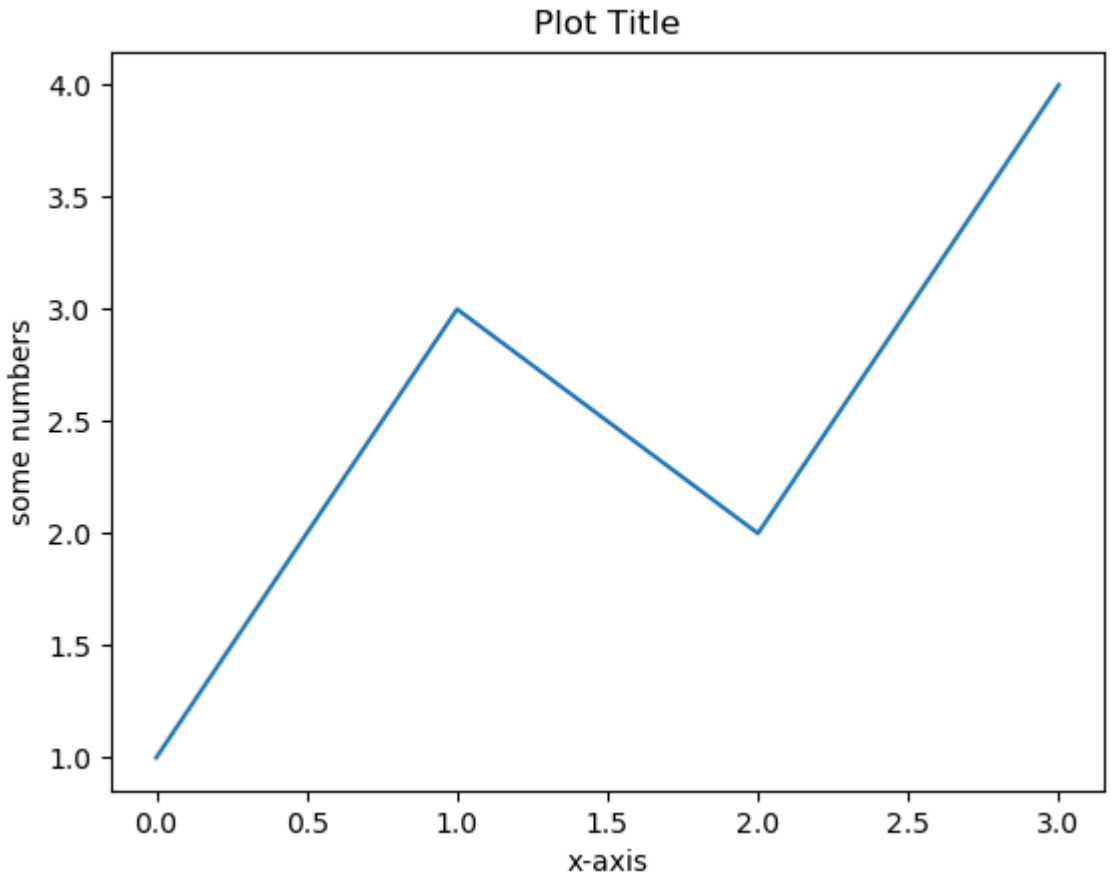
- In order to use matplotlib.pyplot, we first need to import the library

```
import matplotlib.pyplot as plt
```

- Once, we can quickly create plots

```
plt.plot([1, 3, 2, 4])
plt.ylabel('some numbers')
plt.xlabel('x-axis')
plt.title('Plot Title')
plt.show()
```

Note: If you provide a single list or array to plot, matplotlib assumes it is a sequence of y values, and automatically generates the x values for you. Since python ranges start with 0, the default x vector has the same length as y but starts with 0. Hence the x data are [0, 1, 2, 3].



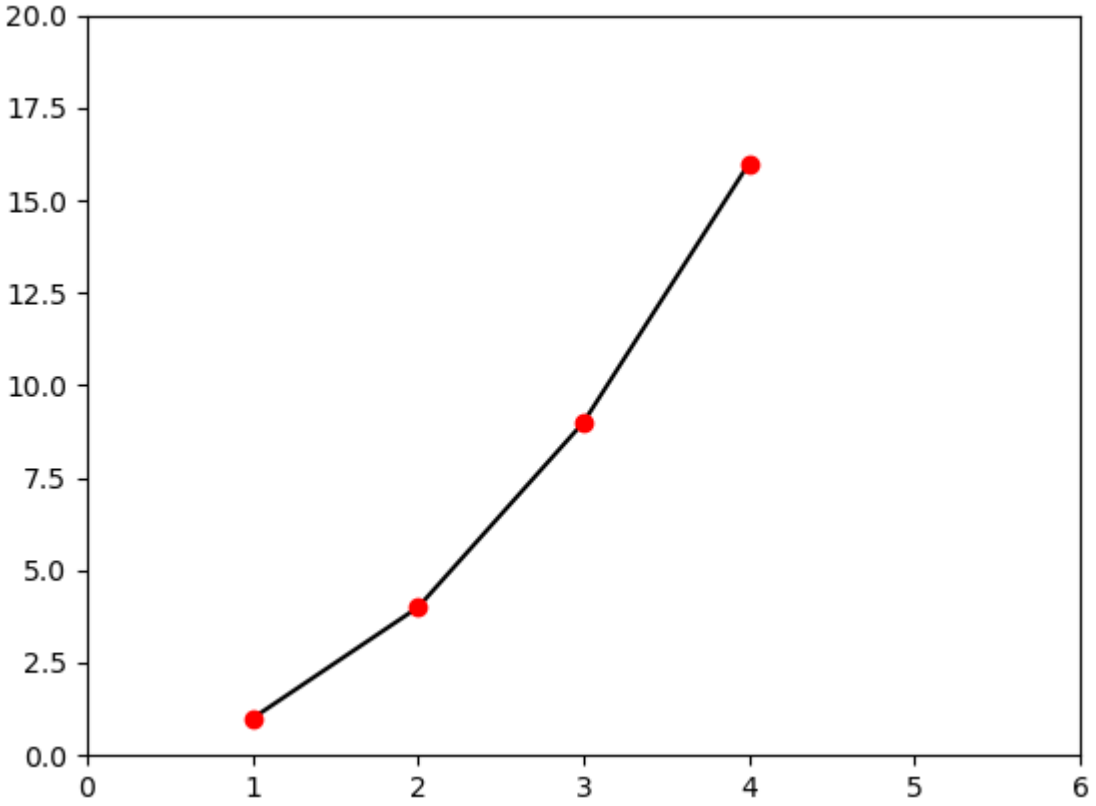
Source: https://matplotlib.org/stable/tutorials/introductory/pyplot.html

Introduction to matplotlib.pyplot

- Adding line and marker formatting

```
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'k-')
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'ro')
plt.axis([0, 6, 0, 20])
plt.show()
```

Similar to MATLAB, you can control the plot line and symbol colors and type through a format string. For example ‘k-’ = black line while ‘ro’ red filled circles.



Source: https://matplotlib.org/stable/tutorials/introductory/pyplot.html

Introduction to matplotlib.pyplot

Line Styles or Marker Control

character	description
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
':'	dotted line style
'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker
's'	square marker
'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin_diamond marker
'|'	vline marker
'_'	hline marker

Color Control

character	color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white

Introduction to matplotlib.pyplot

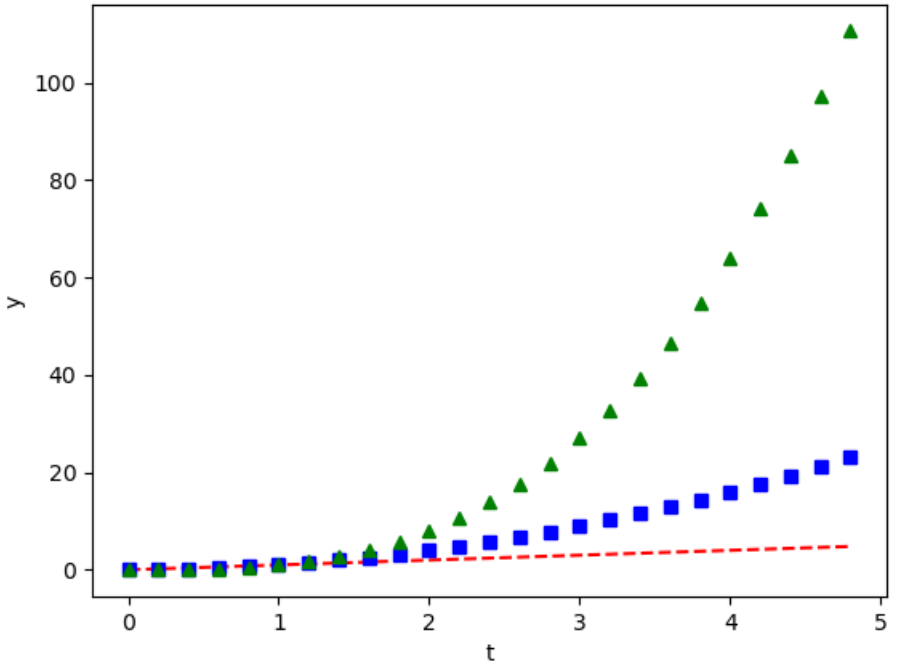
- Incorporating NumPy with matplotlib

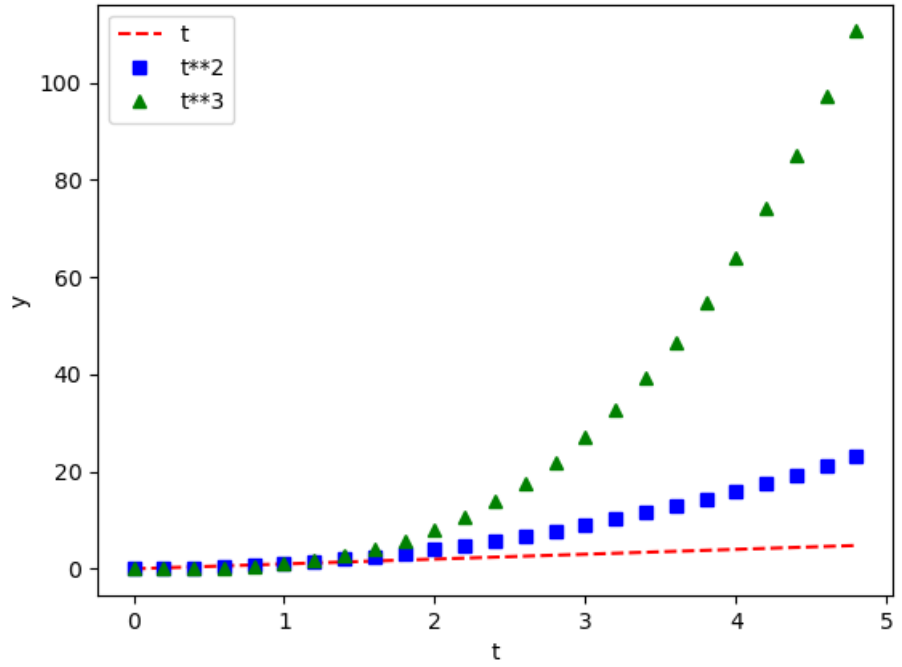
```
import numpy as np

# evenly sampled time at 200ms intervals
t = np.arange(0., 5., 0.2)

# red dashes, blue squares and green triangles
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.xlabel('t')
plt.ylabel('y')
plt.show()
```

```
# Add labels for a legend
plt.plot(t, t, 'r--', label='t')
plt.plot(t, t**2, 'bs', label='t**2')
plt.plot(t, t**3, 'g^', label='t**3')
plt.xlabel('t')
plt.ylabel('y')
plt.legend()
plt.show()
```





Source: https://matplotlib.org/stable/tutorials/introductory/pyplot.html

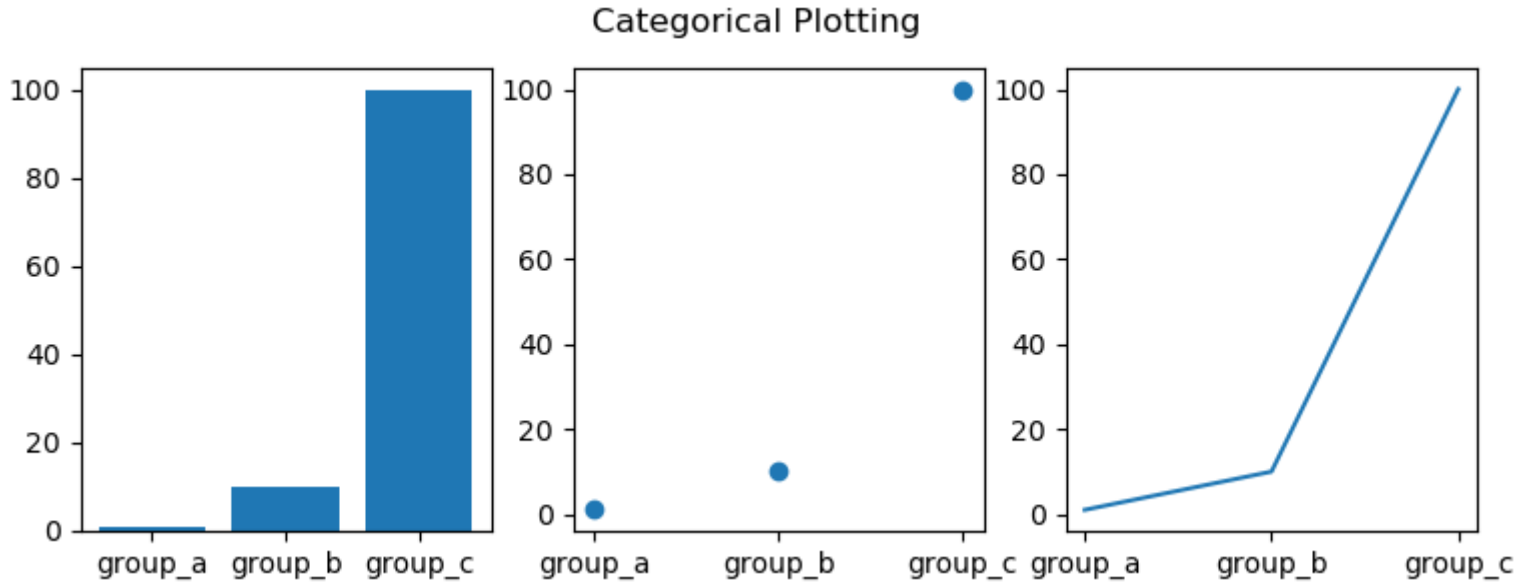
Introduction to matplotlib.pyplot

- Plotting with categorical variables

```
names = ['group_a', 'group_b', 'group_c']
values = [1, 10, 100]

plt.figure(figsize=(9, 3))

plt.subplot(131)
plt.bar(names, values)
plt.subplot(132)
plt.scatter(names, values)
plt.subplot(133)
plt.plot(names, values)
plt.suptitle('Categorical Plotting')
plt.show()
```



This example also demonstrates subplots. Subplots allows us to have multiple plots in one figure. The subplot command takes a three digit argument plt.subplot($digit_1 digit_2 digit_3$)

$digit_1$: number of rows
$digit_2$: number of columns
$digit_3$: plot number (numbered by column then row)

Source: https://matplotlib.org/stable/tutorials/introductory/pyplot.html

Introduction to matplotlib.pyplot

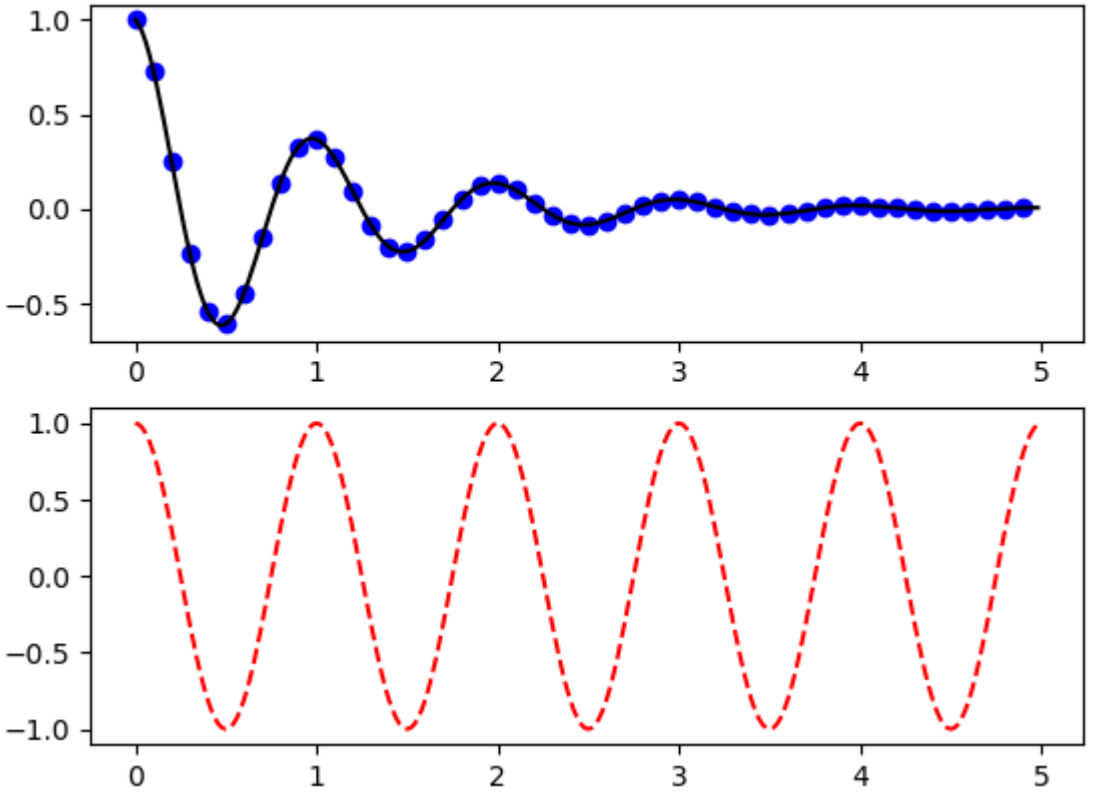
- Controlling multiple figures and axes

```
def f(t):
    return np.exp(-t) * np.cos(2*np.pi*t)

t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.02)

plt.figure()
plt.subplot(211)
plt.plot(t1, f(t1), 'bo', t2, f(t2), 'k')

plt.subplot(212)
plt.plot(t2, np.cos(2*np.pi*t2), 'r--')
plt.show()
```



Source: https://matplotlib.org/stable/tutorials/introductory/pyplot.html

Introduction to matplotlib.pyplot

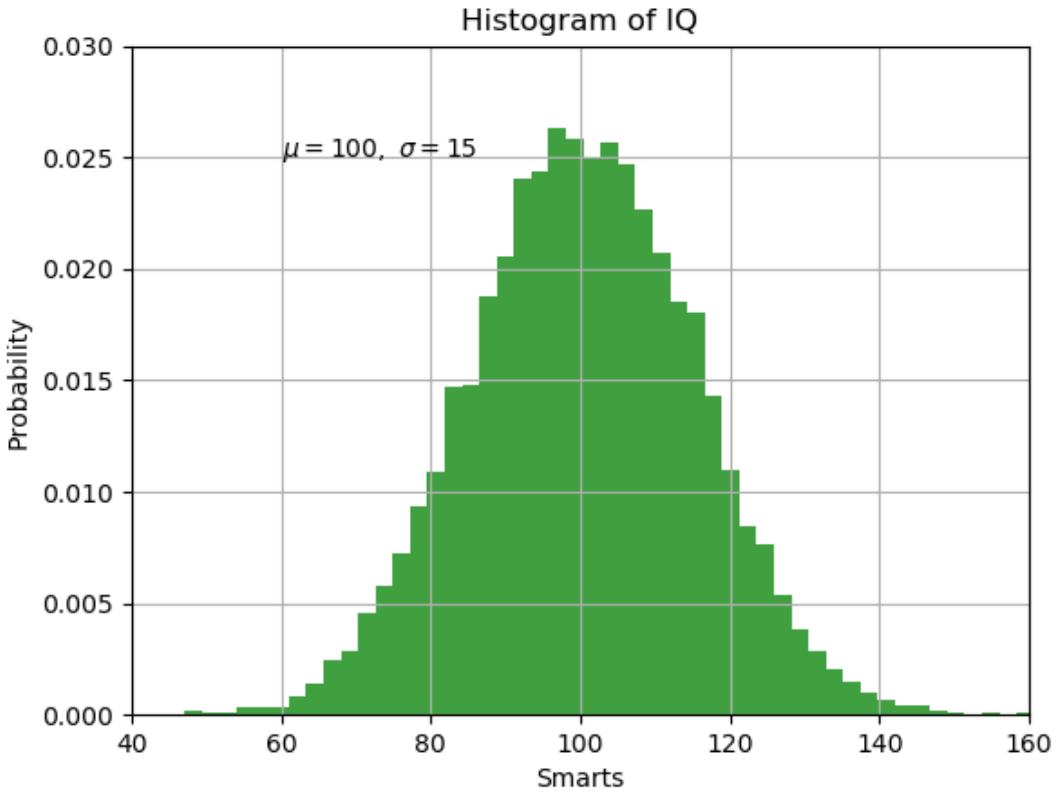
- Placing text annotations

```
mu, sigma = 100, 15
x = mu + sigma * np.random.randn(10000)

# the histogram of the data
n, bins, patches = plt.hist(x, 50, density=True, facecolor='g', alpha=0.75)


plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title('Histogram of IQ')
plt.text(60, .025, '$\mu=100,\ \sigma=15$')
plt.axis([40, 160, 0, 0.03])
plt.grid(True)
plt.show()
```

You at use $ delimiters to enter LaTeX formatted text



Source: https://matplotlib.org/stable/tutorials/introductory/pyplot.html

Summary

- We were introduced to Matplotlib
- We will leverage this library to support plotting needs in upcoming assignments