



for Beginners

Ana Karla Díaz-Rodríguez



INSTITUTO DE
ASTROFÍSICA DE
ANDALUCÍA



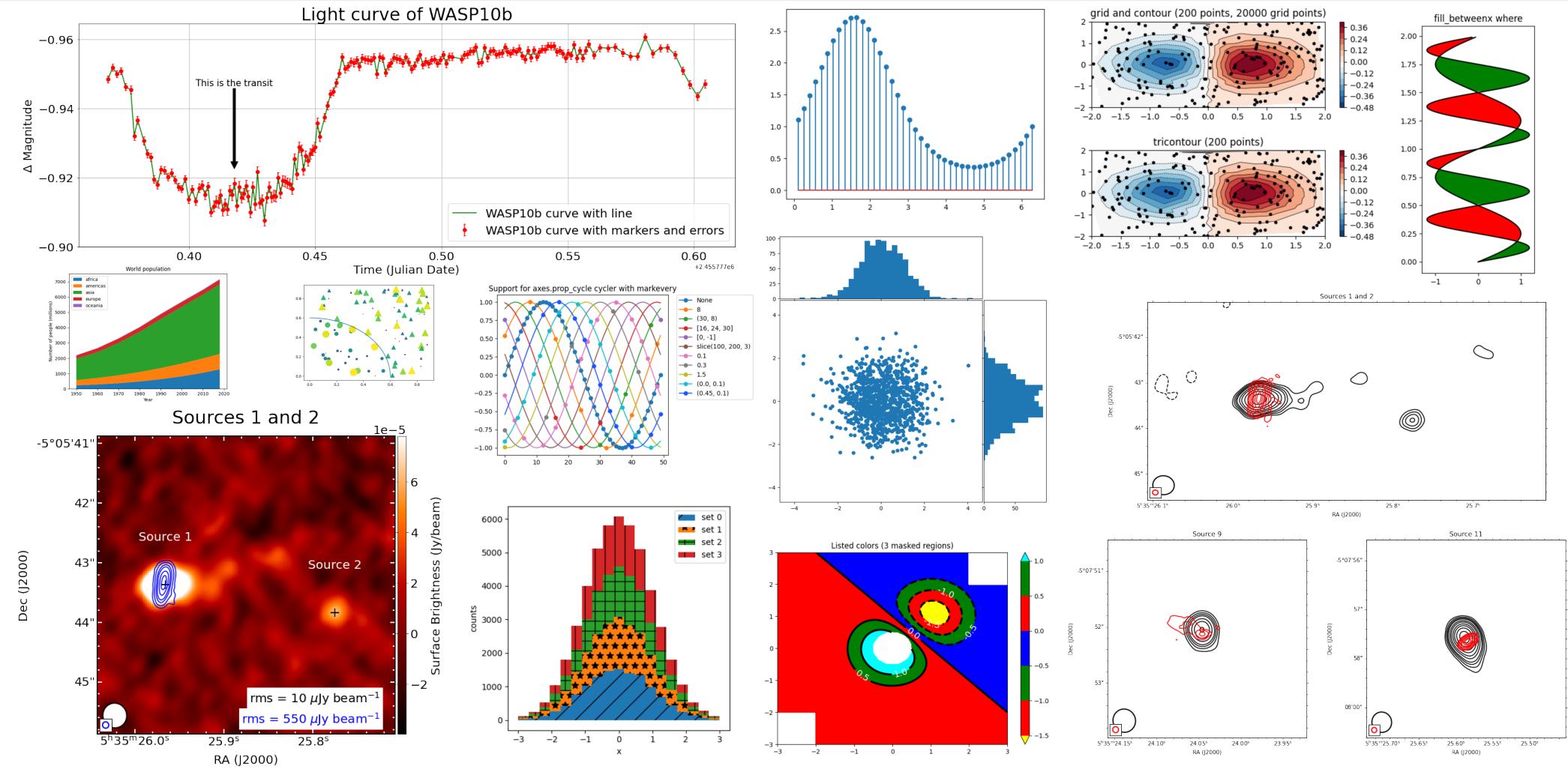
EXCELENCIA
SEVERO
OCHOA



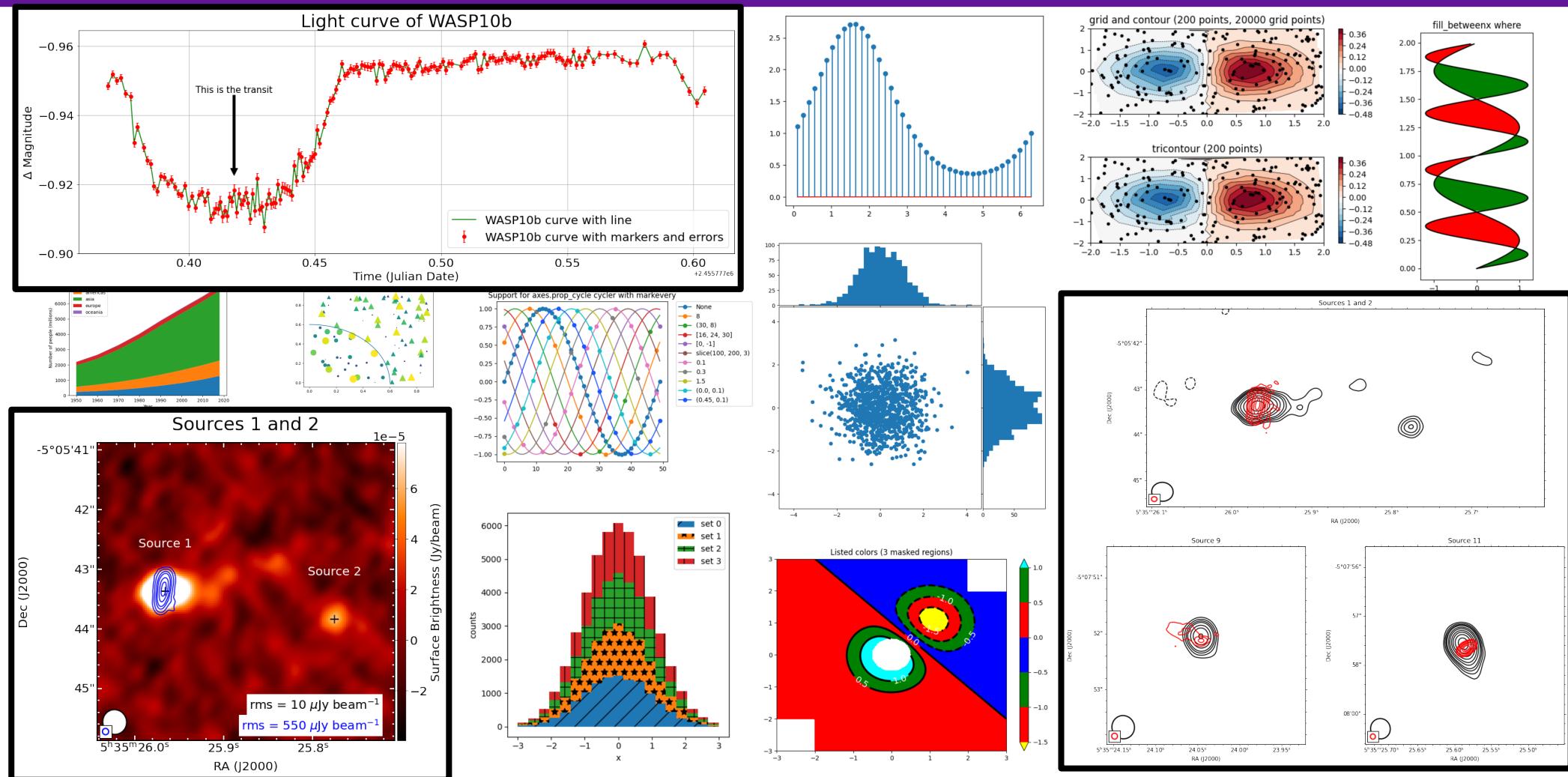
EUROPEAN ARC
ALMA Regional Centre || UK



“Making easy things easy and hard things possible.”



“Making easy things easy and hard things possible.”



What is Matplotlib?

- Python library for creating static, animated, and interactive visualizations

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

- Open source. Please [cite!](#)

J. D. Hunter, "Matplotlib: A 2D Graphics Environment",
Computing in Science & Engineering,
vol. 9, no. 3, pp. 90-95, 2007

[[pdf](#)]

- Extensive [documentation](#). Practical [cheat-sheets](#).

How to use it?

Always inside Python



- Terminal
- Scripts
- Notebooks (e.g. Jupyter, Google Colab)

How to use it?

Always inside Python



- **Terminal**
- Scripts
- Notebooks (e.g. Jupyter, Google Colab)

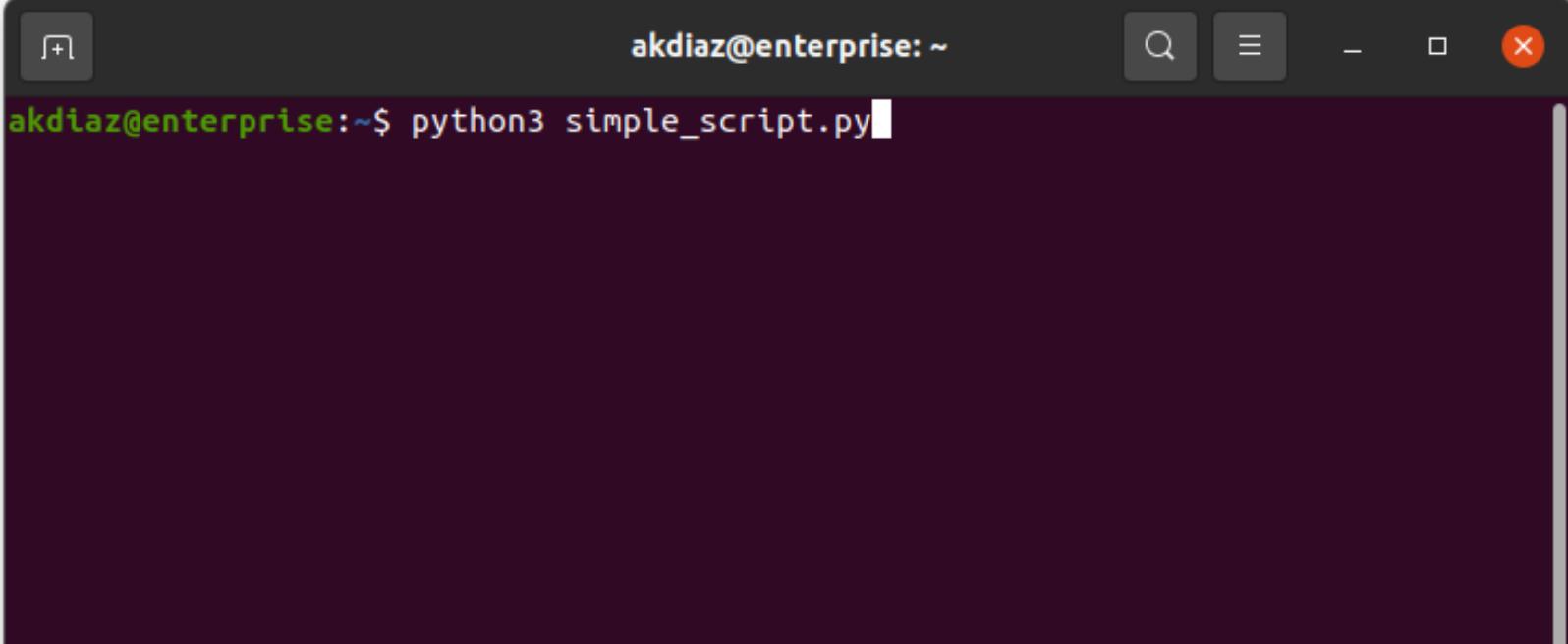
A screenshot of a terminal window titled "akdiaz@enterprise: ~". The window shows a Python 3.8.5 session. The user imports matplotlib.pyplot, creates some data, and makes a plot. The plot is displayed in the terminal window.

```
akdiaz@enterprise:~$ python3
Python 3.8.5 (default, Jan 27 2021, 15:41:15)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import matplotlib.pyplot as plt
>>> # Create some data
>>> x = range(0,50)
>>> y = range(50,100)
>>> # Make plot
>>> fig, ax = plt.subplots()
>>> ax.plot(x,y)
[<matplotlib.lines.Line2D object at 0x7f0fff796b80>]
>>> fig.show()
>>> █
```

How to use it?

Always inside Python 😎

- Terminal
- **Scripts**
- Notebooks (e.g. Jupyter, Google Colab)



A screenshot of a terminal window titled "akdiaz@enterprise: ~". The window has a dark theme with light-colored text. In the terminal, the command "python3 simple_script.py" is typed and visible at the bottom of the screen.

```
akdiaz@enterprise:~$ python3 simple_script.py
```

How to use it?

Always inside Python



- Terminal
- Scripts
- **Notebooks** (e.g. Jupyter, Google Colab)

The image shows a desktop environment with two browser windows open. The left window is a Firefox instance displaying a Jupyter Notebook titled "Annular average". The notebook contains Python code for astronomical data analysis, specifically for calculating annular averages. The right window is also a Firefox instance displaying a Google Colab notebook titled "Matplotlib for Beginners.ipynb". This notebook includes a table of contents with links to various exercises and a code cell for installing the aplpy library.

In [1]:

```
from astropy.io import fits
from astropy.wcs import WCS
import matplotlib.pyplot as plt
from astropy import units as u
from astropy.wcs import utils
from astropy import stats
from photutils import CircularAnnulus, CircularAperture
from photutils.aperture import aperture_photometry as aper_phot
import numpy as np
import astropy.io.fits as pyfits
from astropy.time import Time
# import matplotlib.colors as colors
```

Function definitions

In [126]: def import_fits(fits image):

Table of contents

- Initialize
- Basics
 - Exercise 1
 - Exercise 2
 - Exercise 3
 - Exercise 4
 - Exercise 5
 - Exercise 6
 - Excercise 7
 - Excercise 8

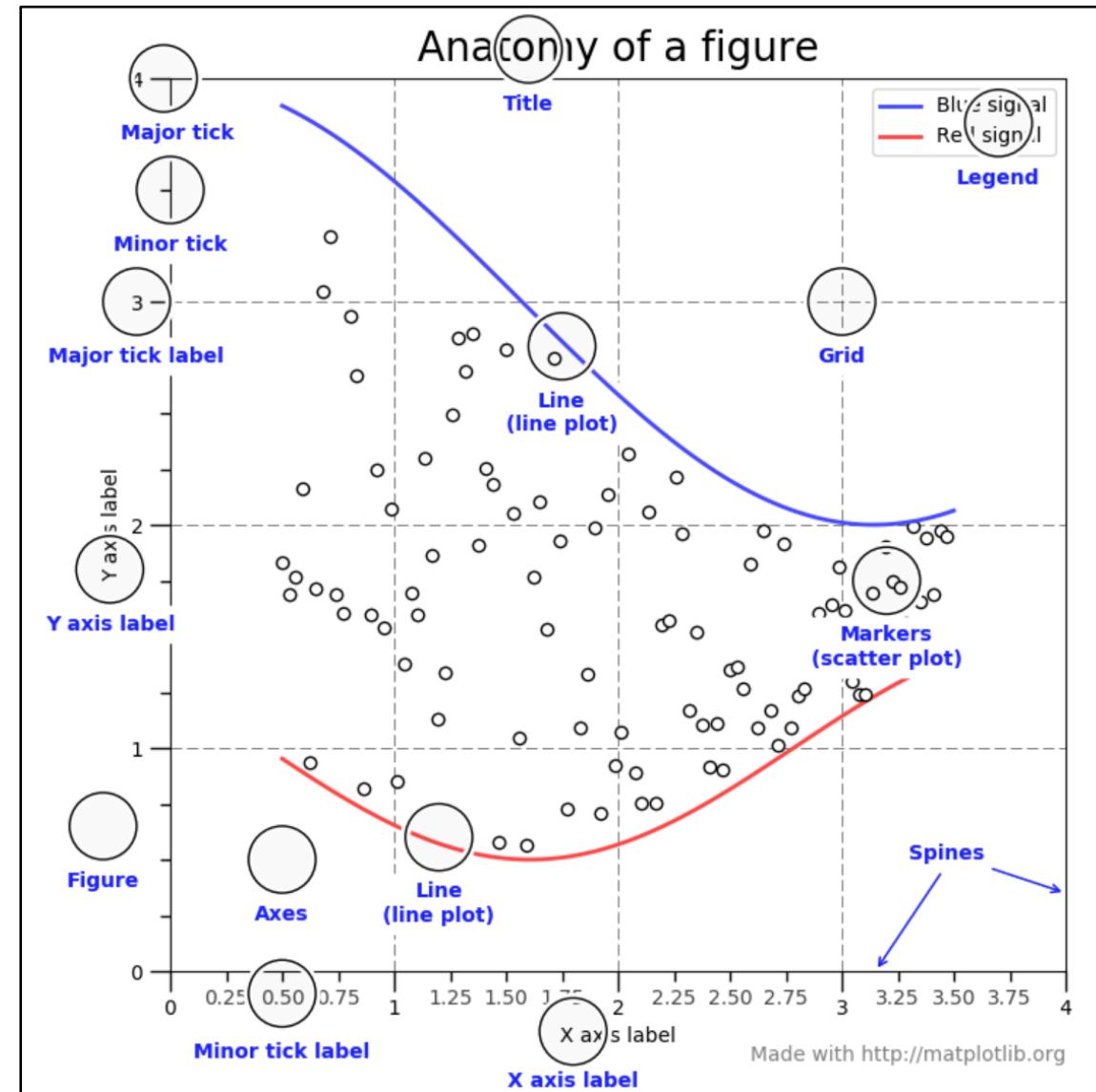
Initialize

```
[ ] %pip install aplpy
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec #to make a
from mpl_toolkits.axes_grid1.inset_locator import
import aplpy #to work with fits
from astropy import units as u #to work with uni
import numpy as np
import random
```

[] #Import data files to use (don't worry about this
from google.colab import files

Figure

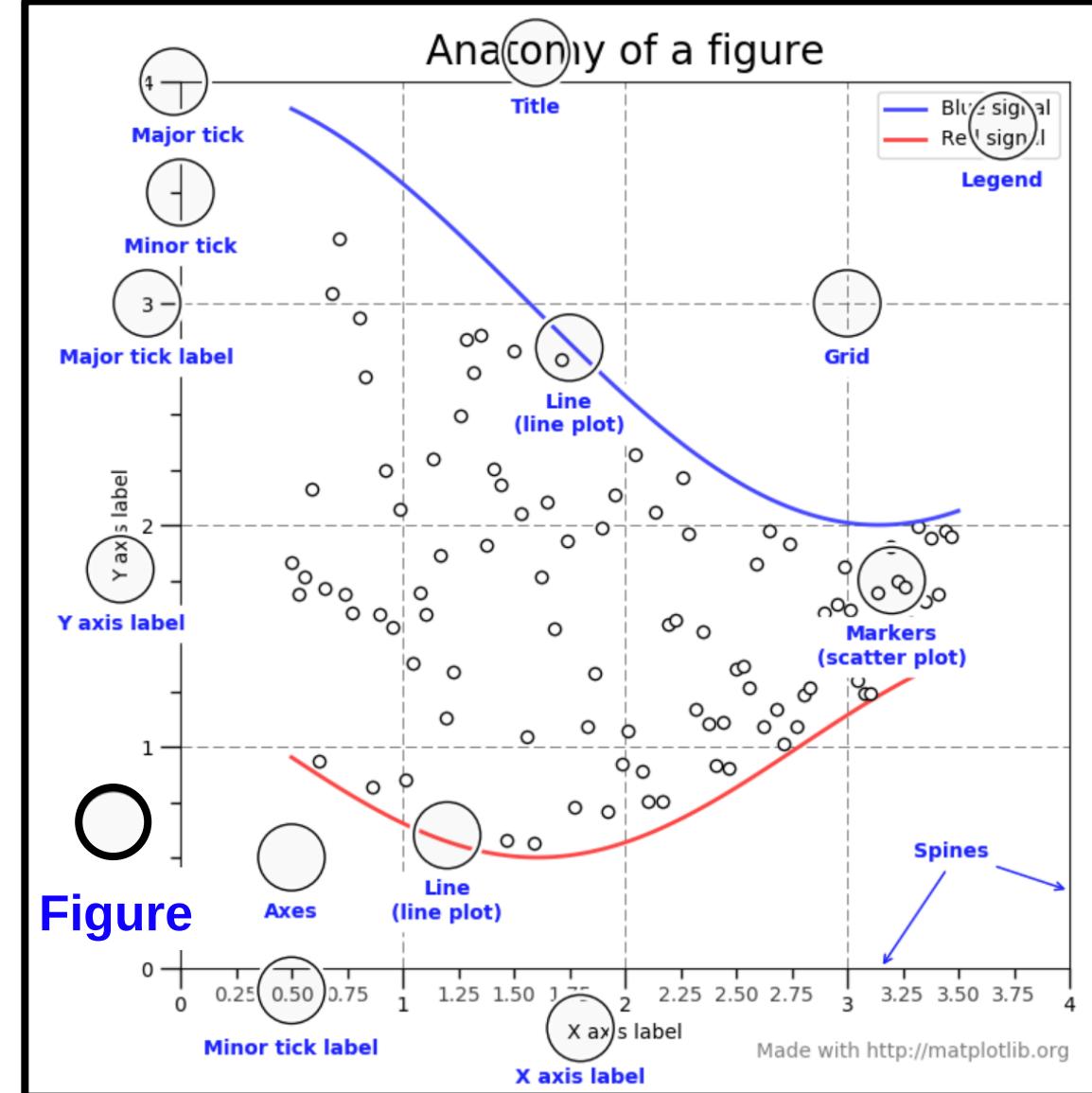
```
>>> fig, ax = plt.subplots()
```



Figure

```
>>> fig, ax = plt.subplots()
```

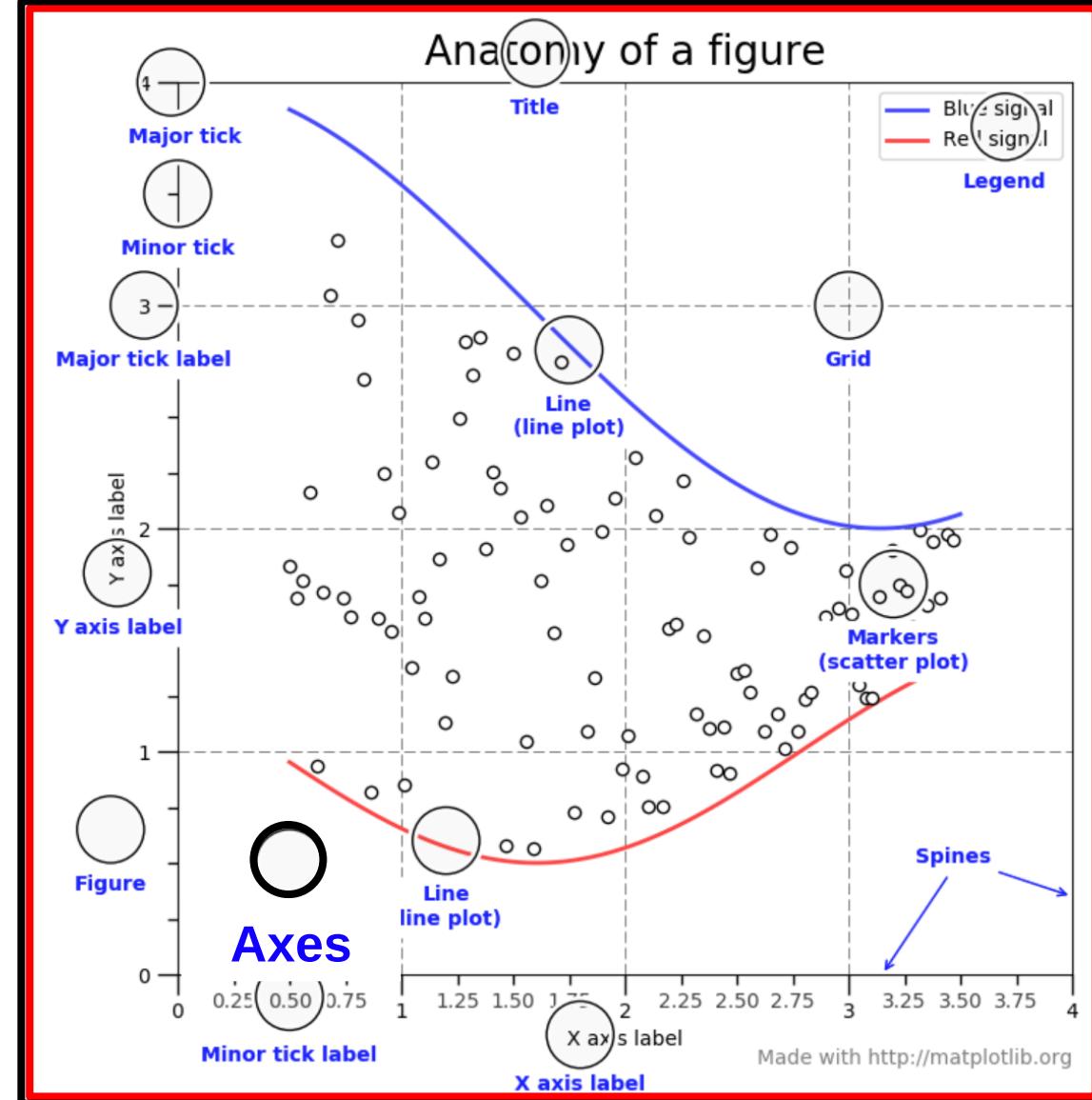
- The **Figure** is the whole figure



Figure

```
>>> fig, ax = plt.subplots()
```

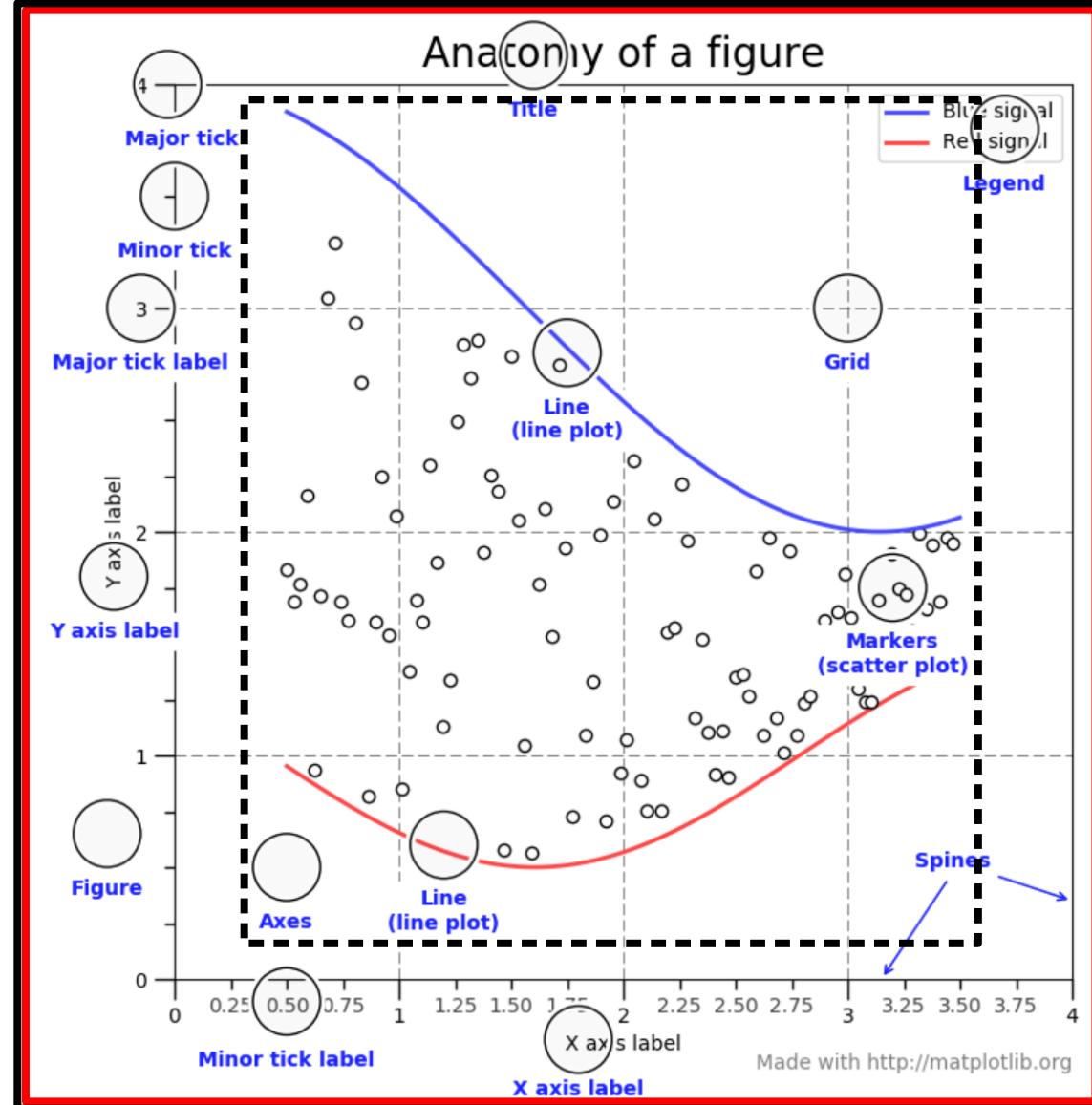
- The **Figure** is the whole figure
- It can contain several **Axes**



Figure

```
>>> fig, ax = plt.subplots()
```

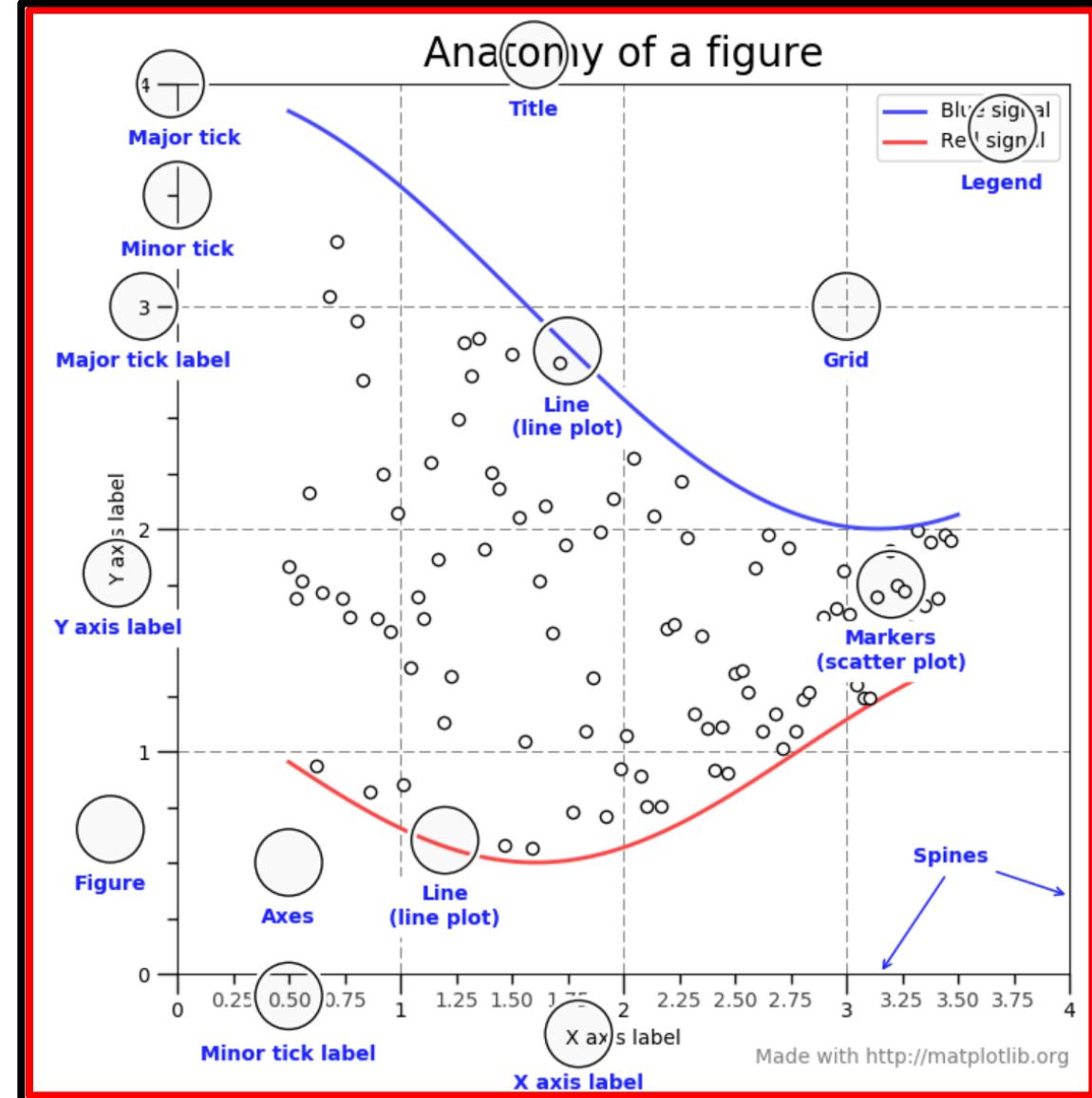
- The **Figure** is the whole figure
- It can contain several **Axes**
- The **Axes** is where you actually draw your plot



Figure

```
>>> fig, ax = plt.subplots()
```

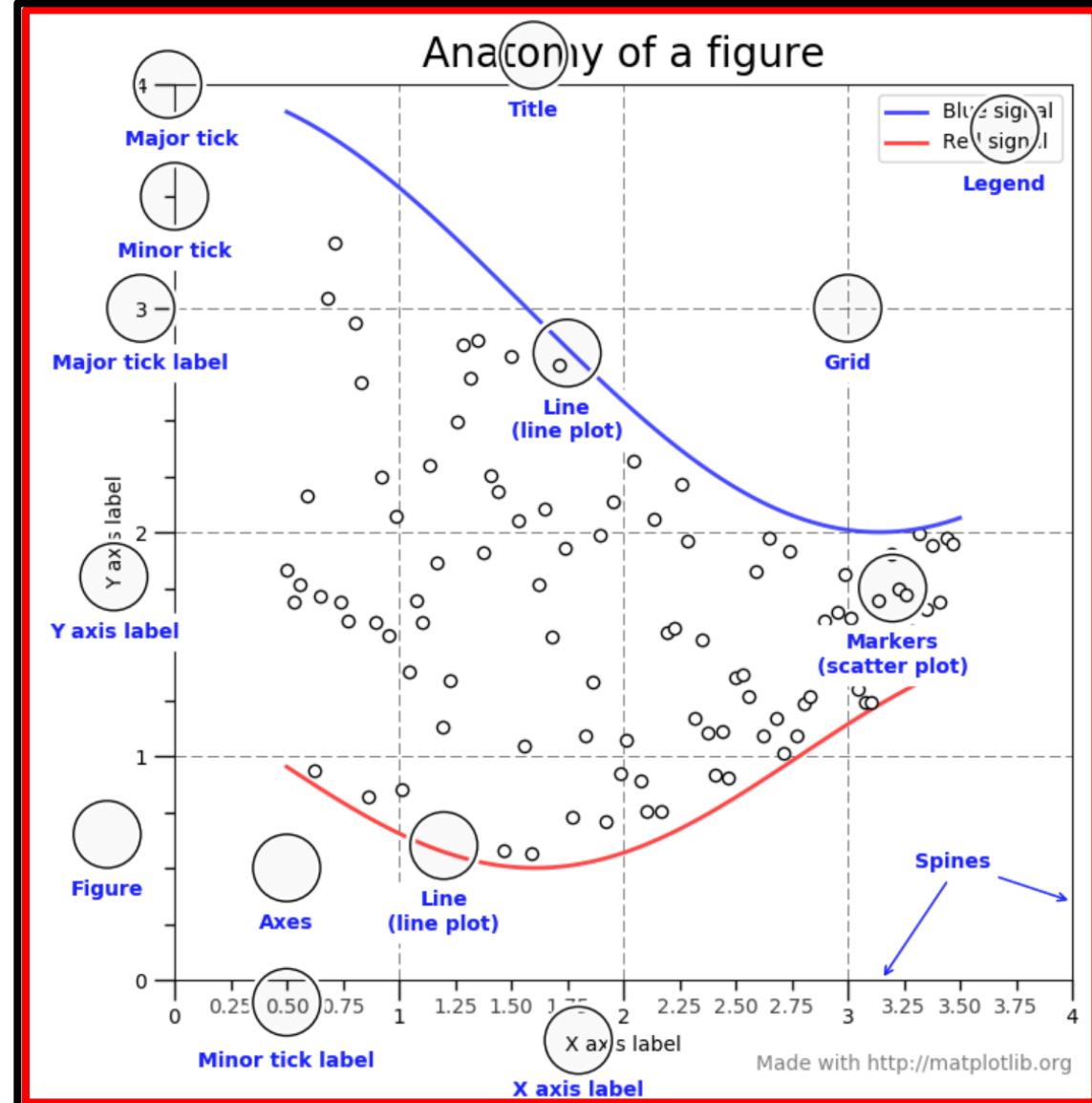
- The **Figure** is the whole figure
- It can contain several **Axes**
- The **Axes** is where you actually draw your plot
- Each **Axes** has its own **Artists** (axis, labels, legends, etc)



Figure

```
>>> fig, ax = plt.subplots()
```

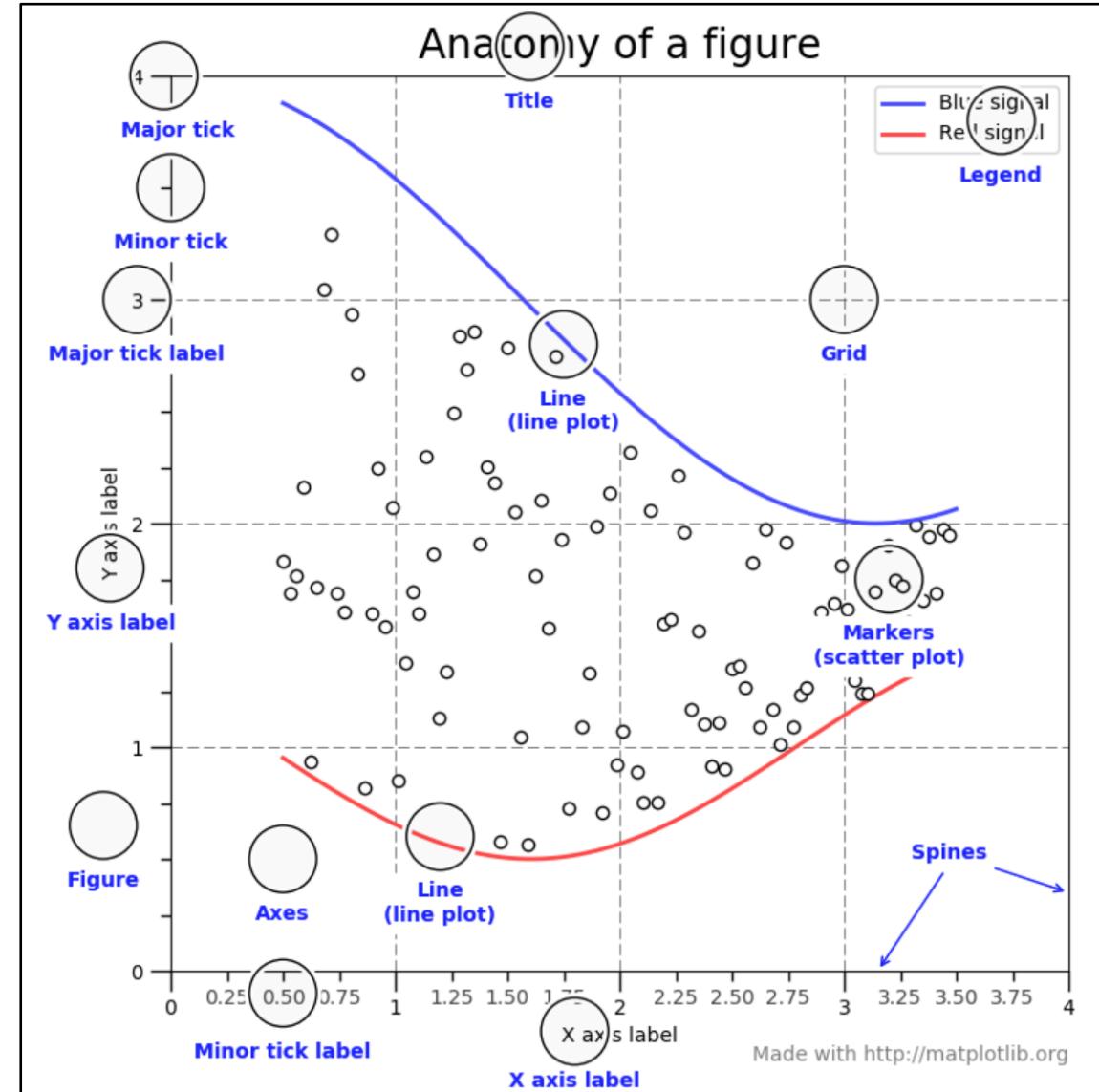
- The **Figure** is the whole figure
- It can contain several **Axes**
- The **Axes** is where you actually draw your plot
- Each **Axes** has its own **Artists** (axis, labels, legends, etc)
- Every bit is customizable



Figure

```
>>> fig, ax = plt.subplots()
```

Customization



Figure

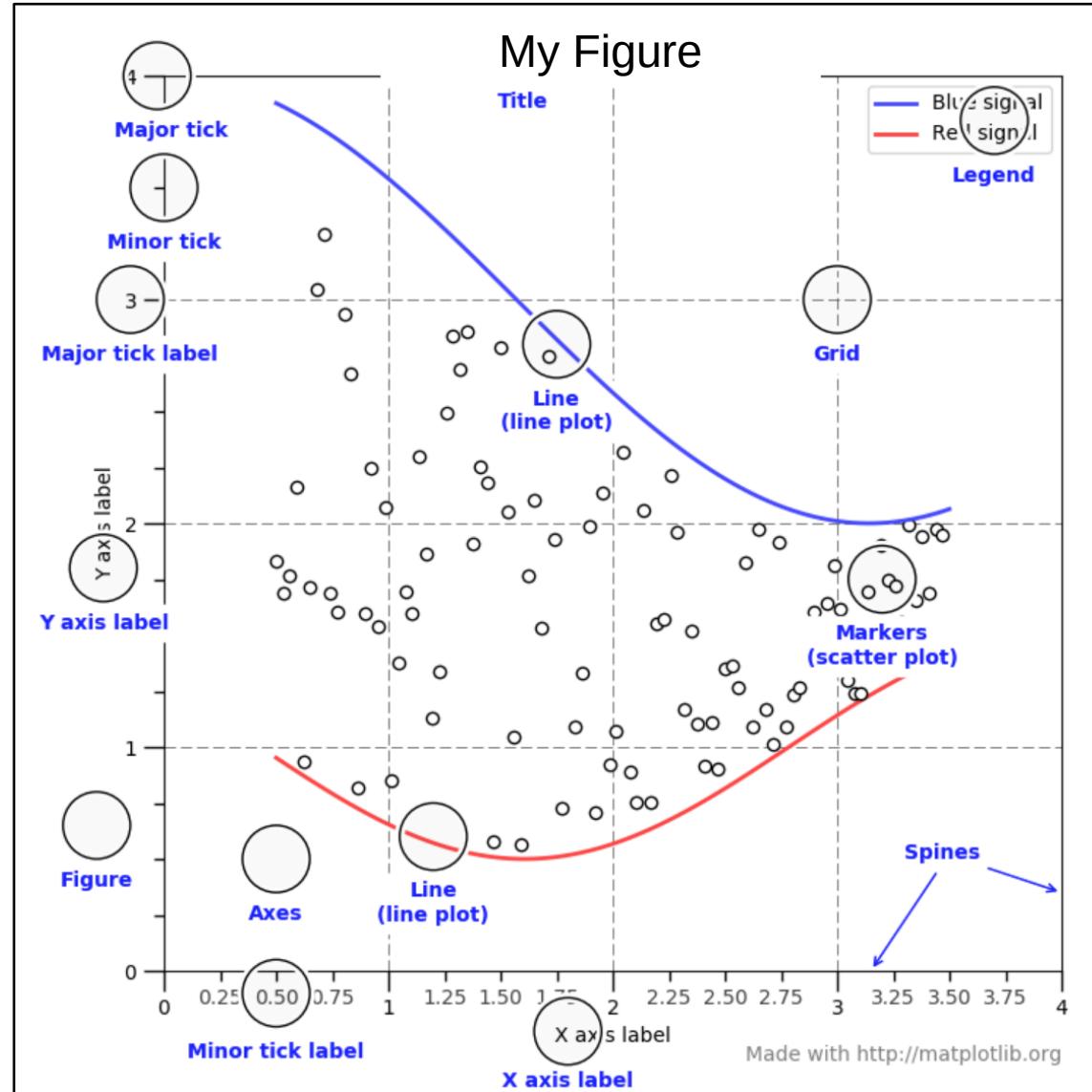
```
>>> fig, ax = plt.subplots()
```

Customization

```
>>> ax.set_title('My Figure')
```



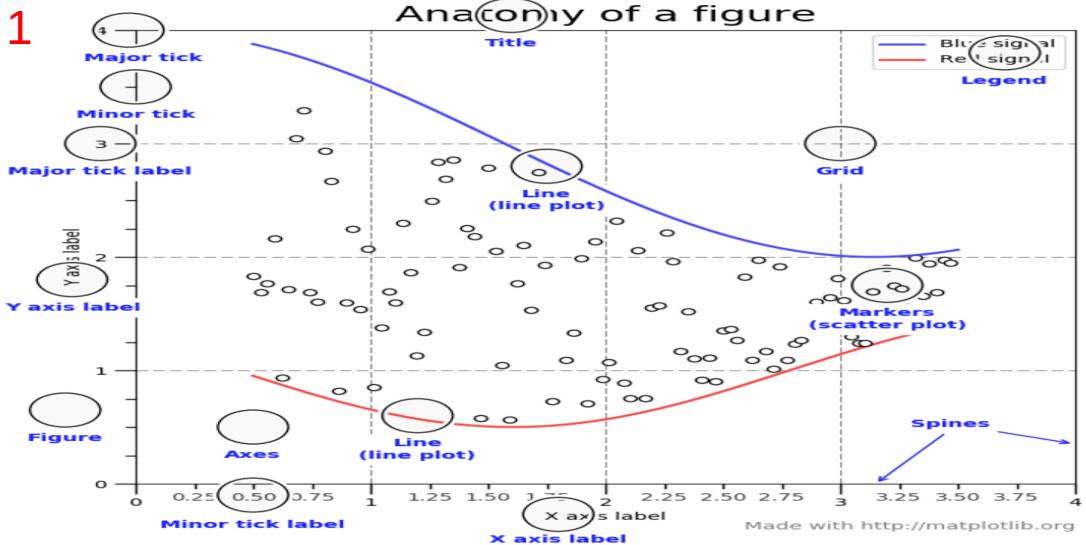
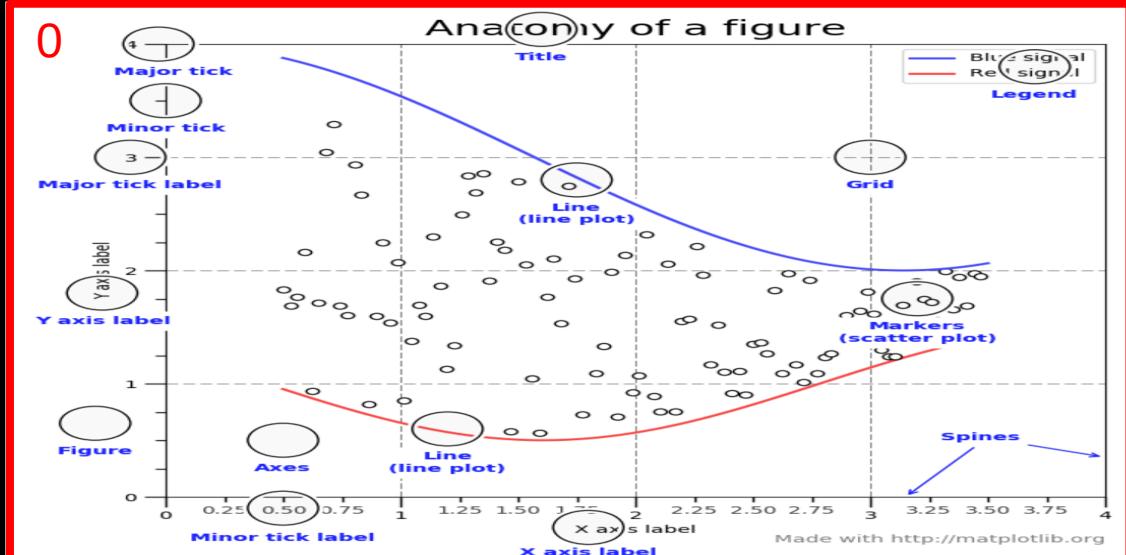
There are several functions for controlling the appearance of each artist
[\(axes doc\)](#)



Figure

(rows)

```
>>> fig, ax = plt.subplots(2)
```



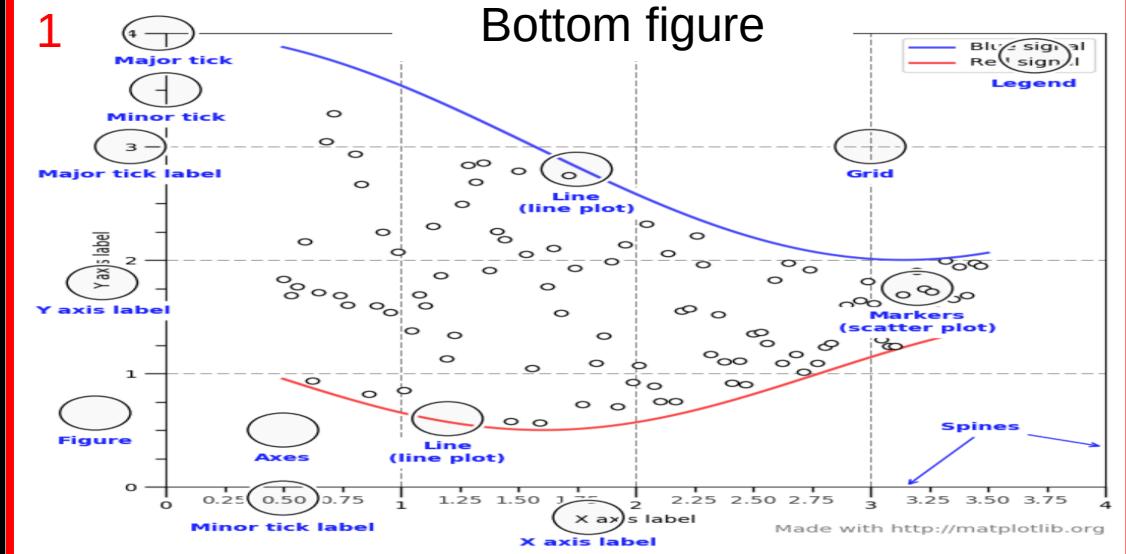
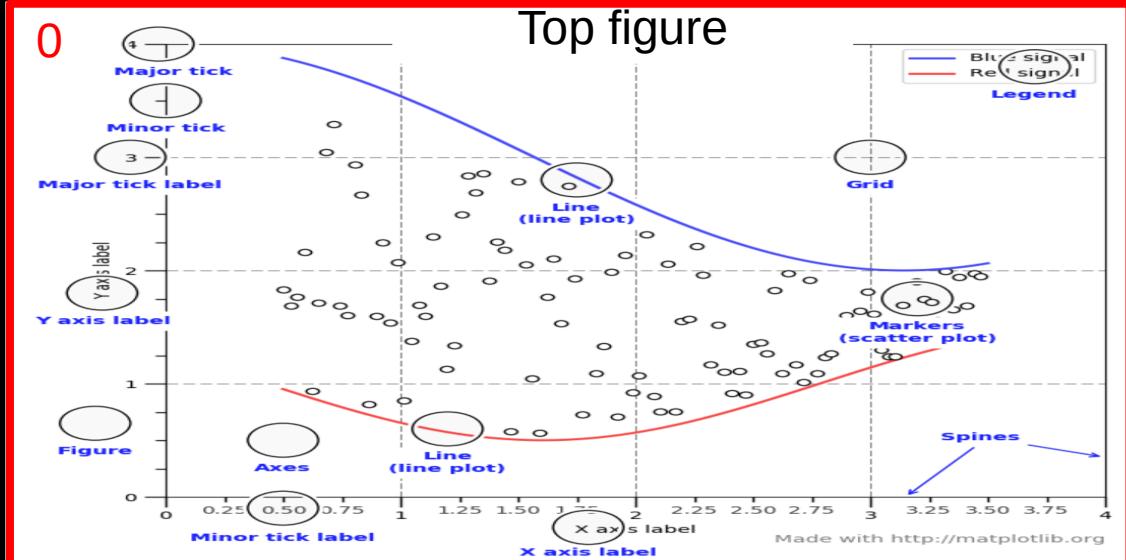
Figure

(rows)

```
>>> fig, ax = plt.subplots(2)
```

Customization

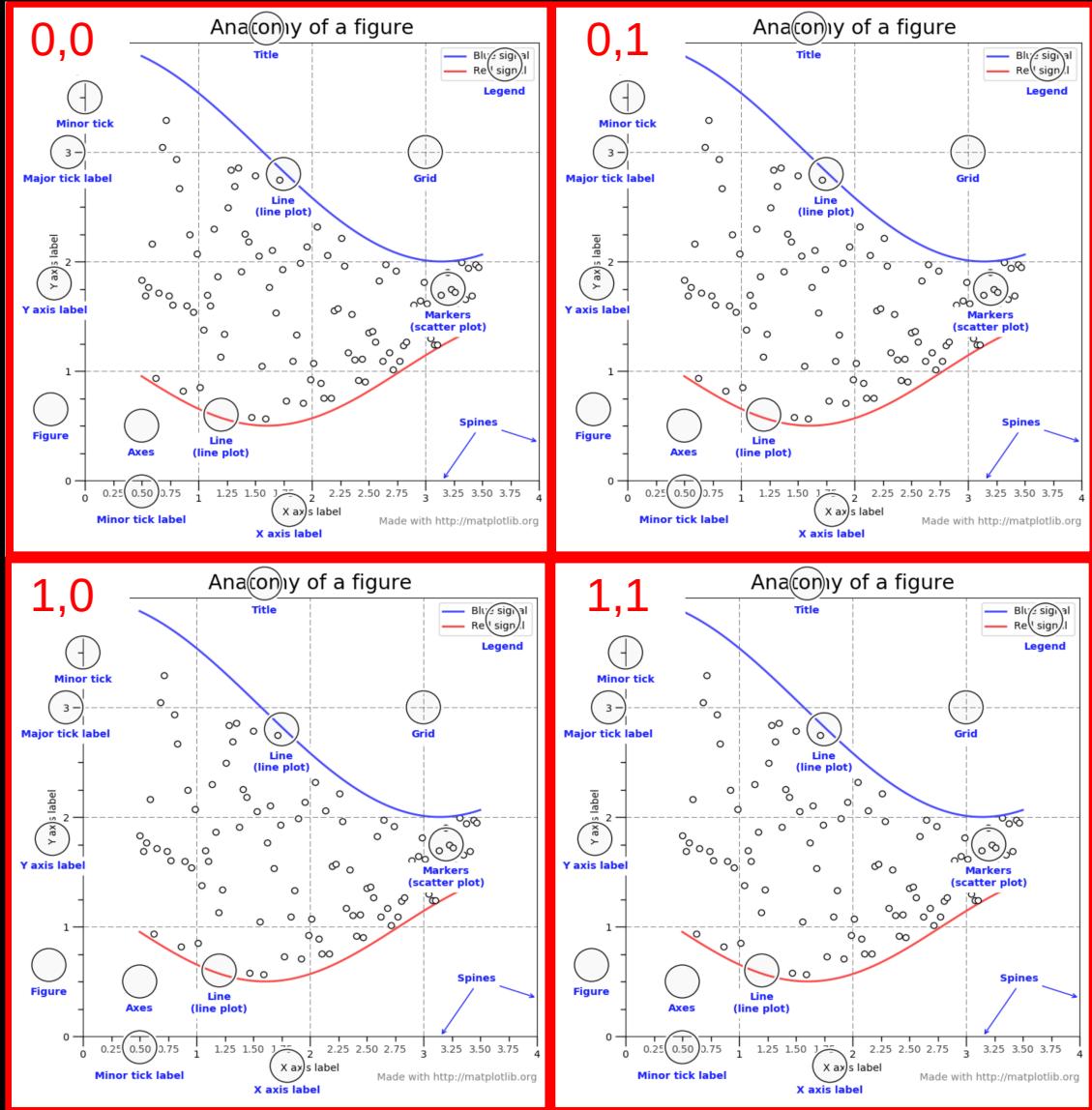
```
>>> ax[0].set_title('Top figure')  
>>> ax[1].set_title('Bottom figure')
```



Figure

(rows,columns)

```
>>> fig, ax = plt.subplots(2, 2)
```



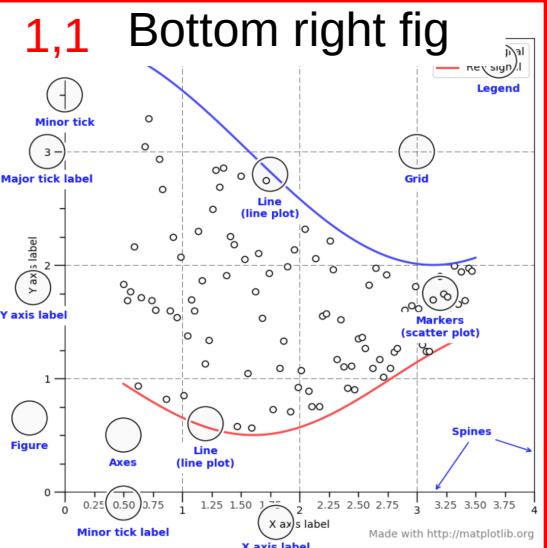
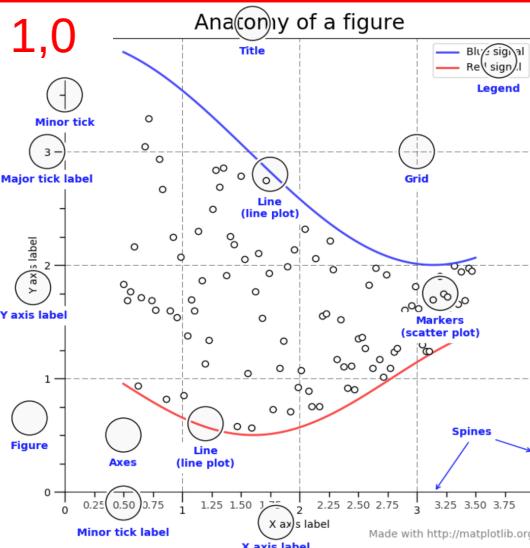
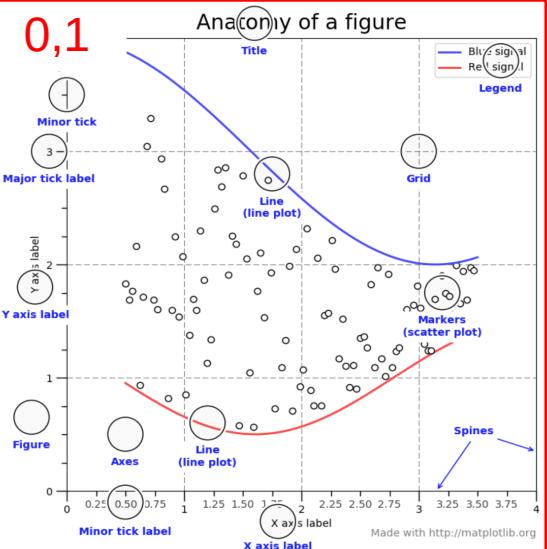
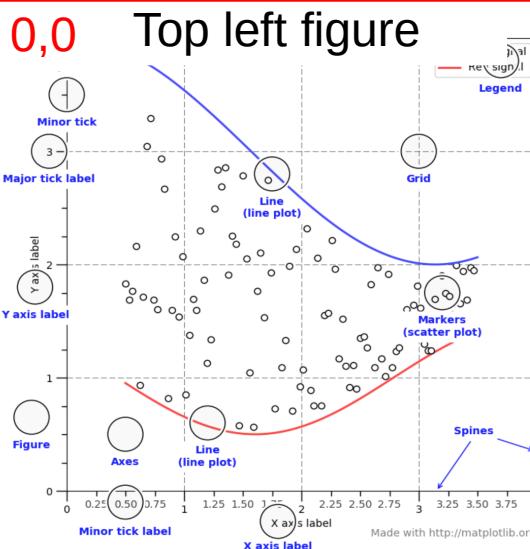
Figure

(rows,columns)

```
>>> fig, ax = plt.subplots(2, 2)
```

Customization

```
>>> ax[0,0].set_title('Top left figure')
>>> ax[1,1].set_title('Bottom right fig')
```



For Astronomers



Astronomical Plotting Library in Python

APLPy
(pronounced *apple pie*)

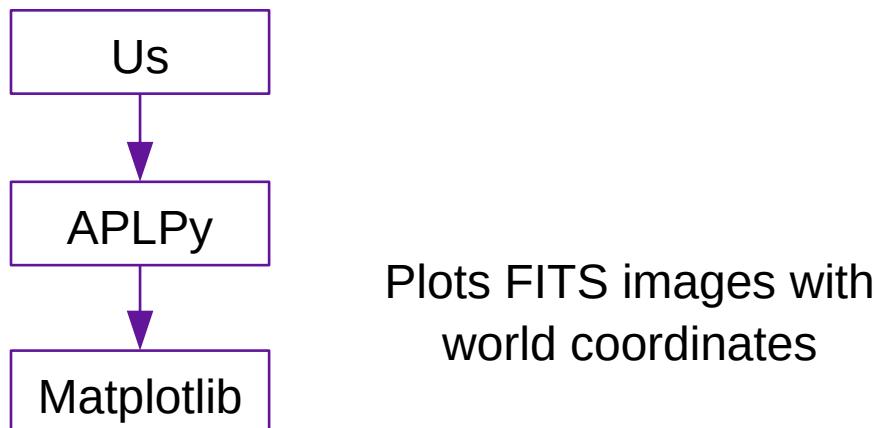
For Astronomers



Astronomical Plotting Library in Python

APLPy

(pronounced *apple pie*)

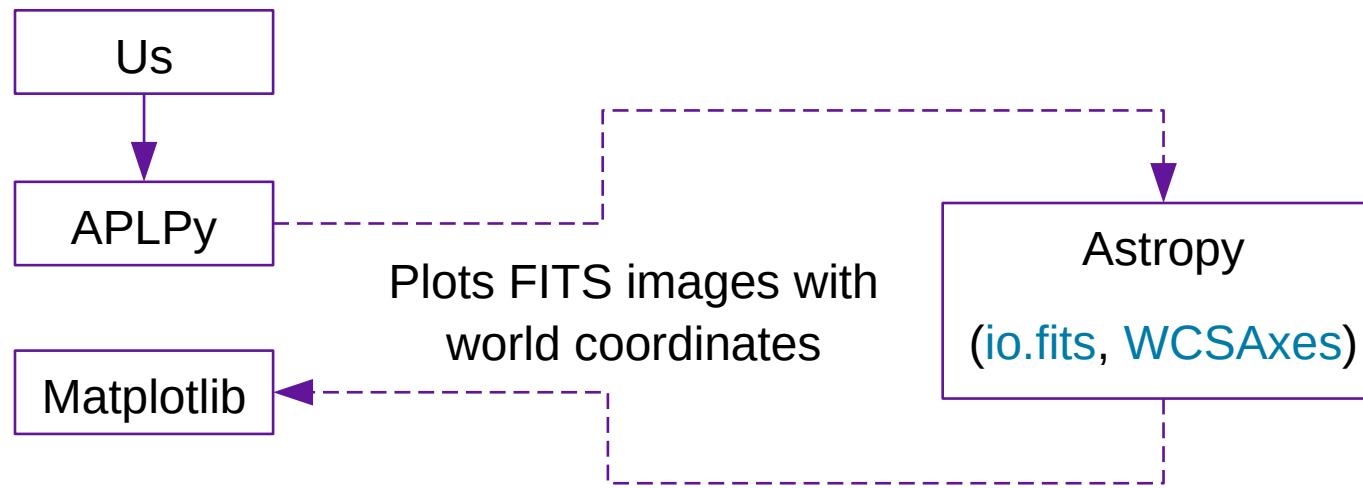


For Astronomers



Astronomical Plotting Library in Python

APLPy
(pronounced *apple pie*)



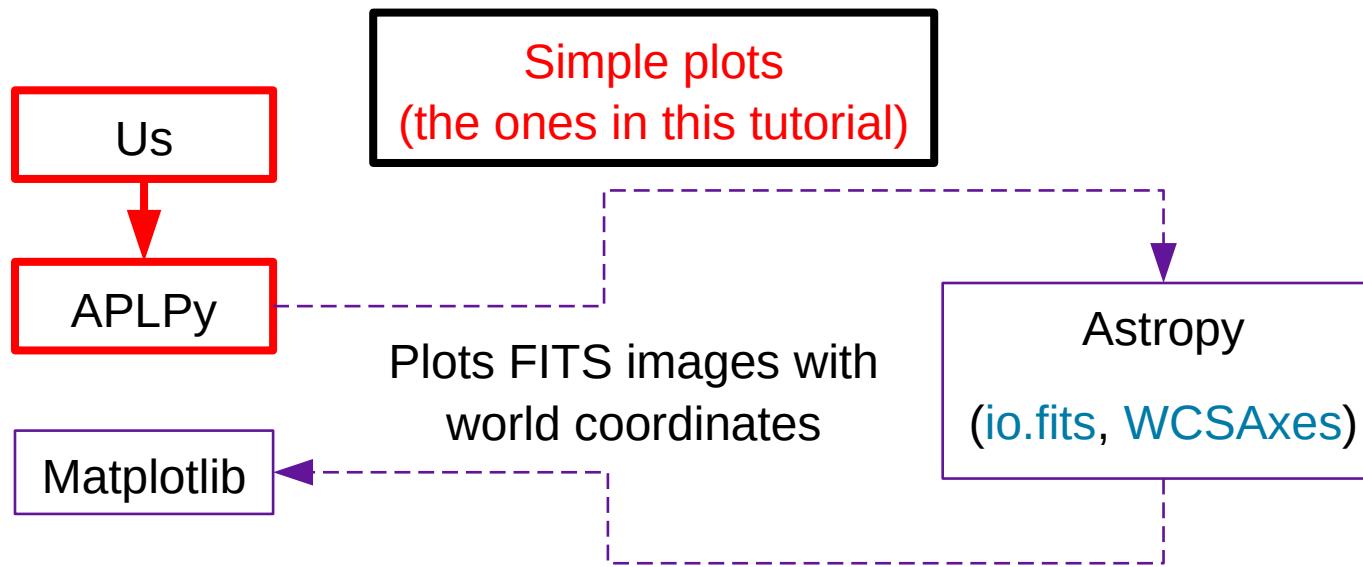
For Astronomers



Astronomical Plotting Library in Python

APLPy

(pronounced *apple pie*)



For Astronomers



Astronomical Plotting Library in Python

APLPy

(pronounced *apple pie*)

Us

Complex plots with total
control of their appearance

APLPy

Plots FITS images with
world coordinates

Matplotlib

Astropy

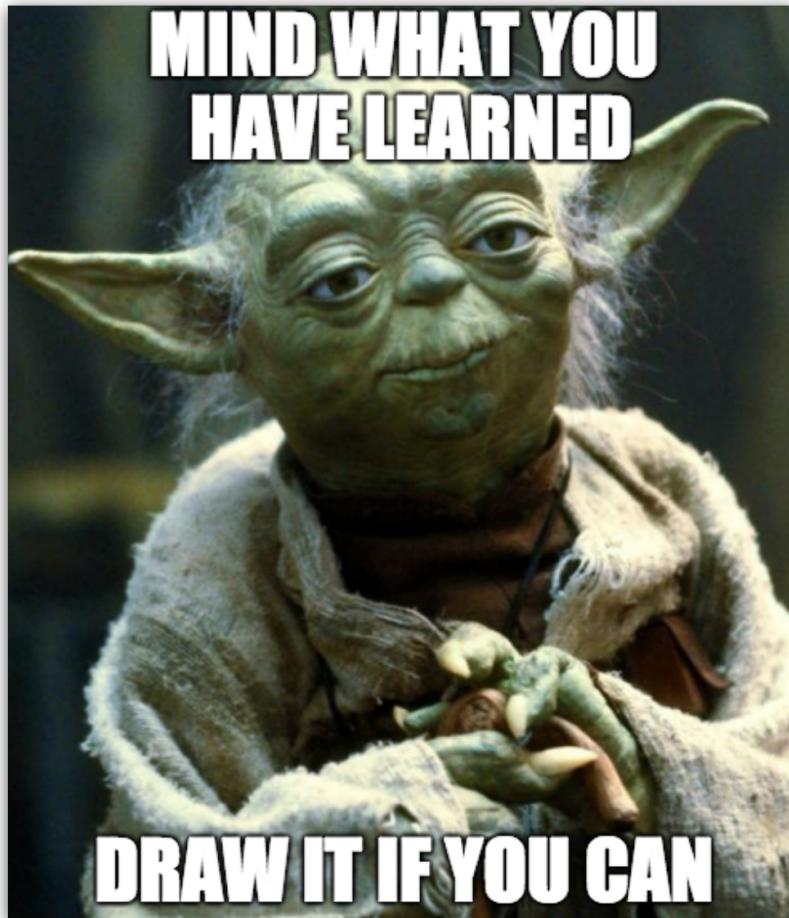
(`io.fits`, `WCSAxes`)

Hands-on Tutorial



1. Open the [Notebook](#)
2. Copy it to your Drive
(if you want to save the changes)
3. Follow along

Optional Assignment



1. Make a plot related to your work (e.g. for a paper you are writing)
2. Upload the plot, (the data) and the code to this [shared folder](#)
3. Check out what other participants did