



ESCAPE

European Science Cluster of Astronomy &
Particle physics ESFRI research Infrastructures

Jupyter Notebooks

Enrique Garcia LAPP/CNRS

ESCAPE data science summer School 2022 – 20 June 2022



Overview

- Python
- IPython
- Jupyter
 - Jupyter Notebooks
 - Basic usage
 - Useful features for presenting results
 - JupyterLab
- Reproducible computer environments
 - Binder
 - Docker



Python

- Multi purpose
 - Like C, C++ or Java, it is designed to build software in a variety of domains
- Interpreted
 - Code does not need to be previously compiled
- Object oriented (OOP)
 - Code is designed around data (objects) that contains properties and attributed
 - "Contrary" to Imperative (C), functional (SQL, Mathematica...) or logic programming
- High-level
 - Independent of type of computer

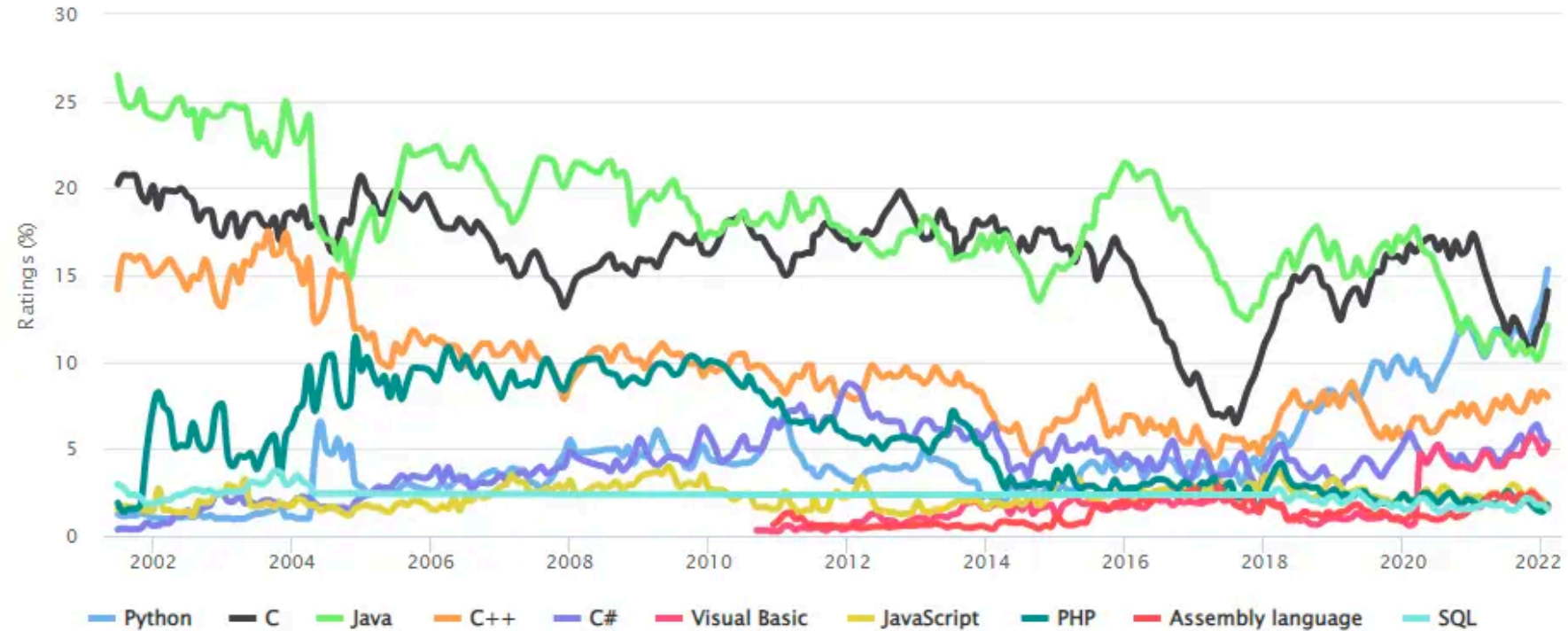


Python

- Released in 1991
- Python 1.0 released in 1994.
 - Guido Van Rossum
- Python 2 was released in 2000 and Python 3 in 2008.
 - Python2.7
 - Python3.10

TIOBE Programming Community Index

Source: www.tiobe.com



- Currently one of the most used programming languages.



Python

- Python interpreter

- The engine that runs python

- Operates like a shell

- Start it by typing

- \$ python

- Exit by typing

- >> exit()

- Scripts can be run too (f.ex)

- \$ python script_name.py

```
garciaenrique — -bash — 80x24
(base) garciaenrique@lappm-p841 ~ $ python
Python 3.7.4 (default, Aug 13 2019, 15:17:50)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> from math import pi
>>>
>>> r = 1.5
>>> vol = 4 / 3 * pi * r**3
>>>
>>> print('The volume is %f' % vol )
The volume is 14.137167
>>>
>>> exit() # or 'ctl + d'
(base) garciaenrique@lappm-p841 ~ $
(base) garciaenrique@lappm-p841 ~ $ python compute_volume.py
The volume is 14.137167
(base) garciaenrique@lappm-p841 ~ $
```



IPython

- More interactive shell and command line

- Code completion
- Highlights
- Visualisation tools (GUIs)
- "cells and scripts magics"

- Uses an IPython Kernel

- The python interpreter (backend)
- Default kernel in Jupyter project

- Launch it by typing
\$ ipython

```
Python 3.6.3 | packaged by conda-forge | (default, Nov 4 2017, 10:13:32)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.0.0.dev -- An enhanced Interactive Python. Type '?' for help.
```

```
In [1]: from numpy.fft import *
...: from numpy import arange
...: a = arange(32)
...: A = fft(a)
...: f = fftfreq(32)

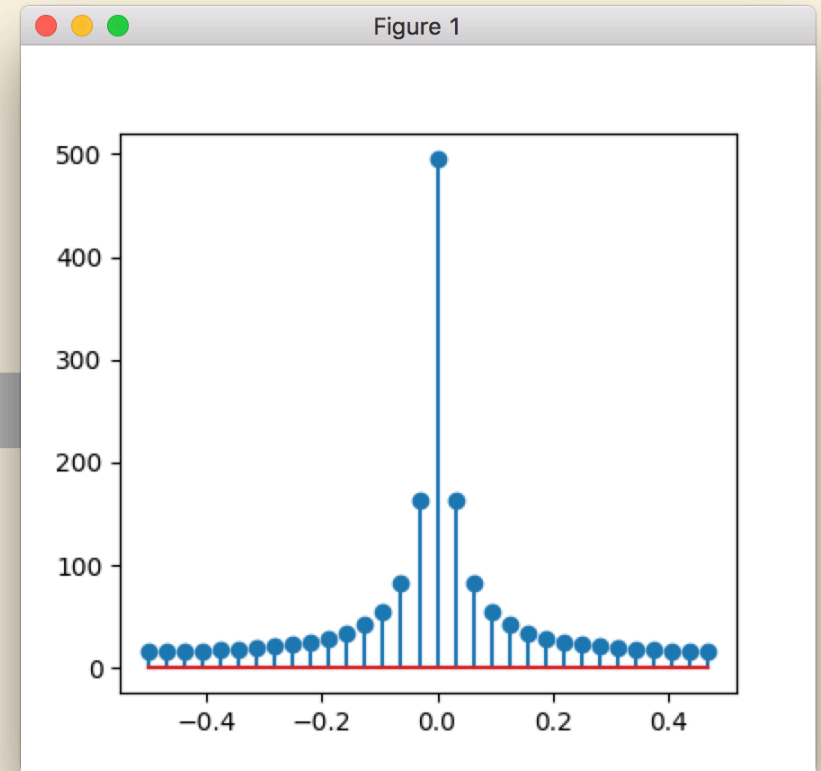
In [2]: %matplotlib tk

In [3]: from matplotlib.pyplot import stem

In [4]: stem(f, abs(A))
Out[4]: <Container object of 3 artists>

In [5]: _.
```

```
add_callback      eventson
baseline          get_children
count()           get_label
```



Jupyter (project)

- Side project of the IPython project

- Started on 2014

- Included support to

- **Julia, Python, R (Jupyter)**

- Web based computing platform that mixes

- Code and equations and text,

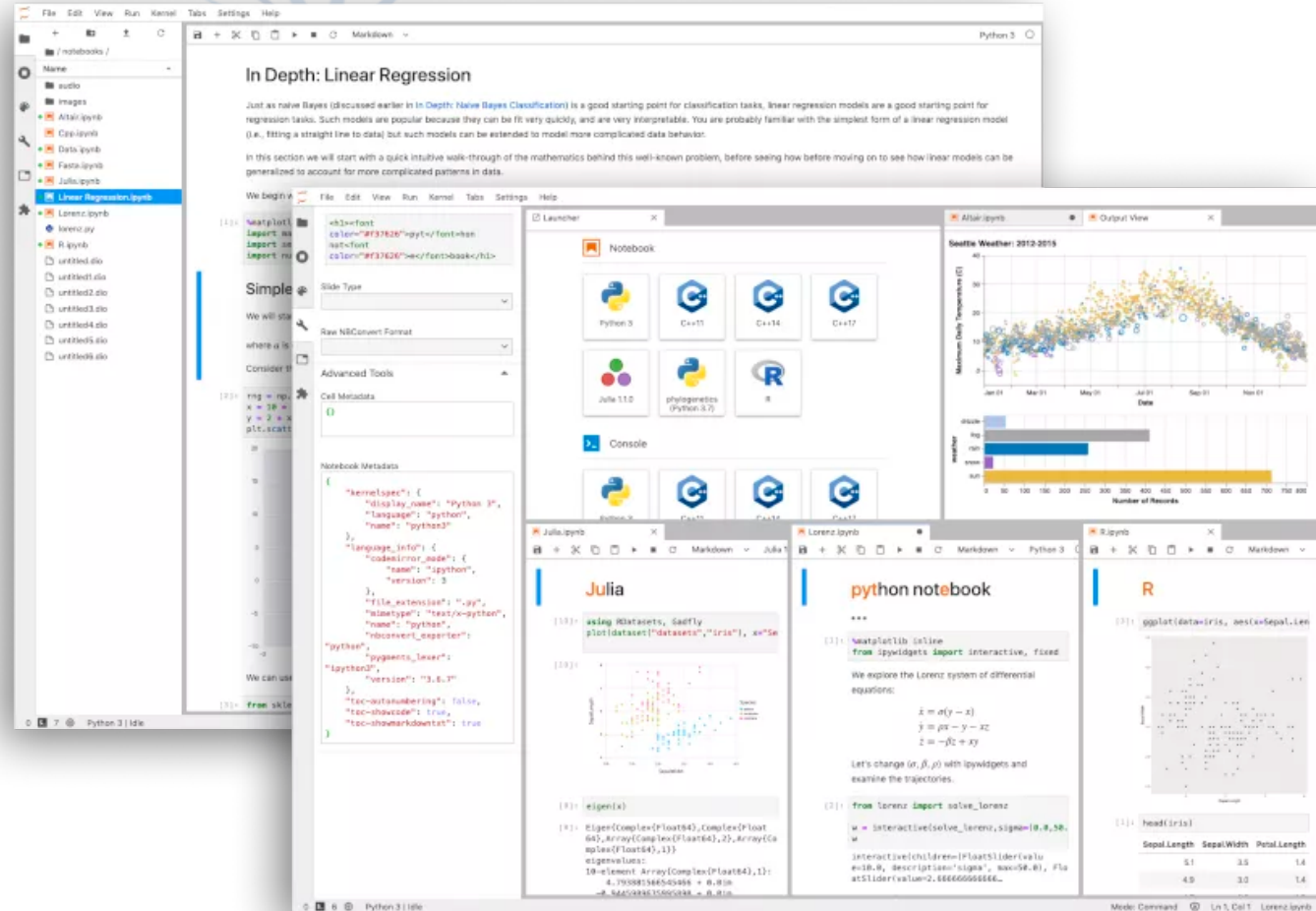
- Visualization environment and tools...

- File manager

- Interfaces:

- Jupyter Notebook

- JupyterLab



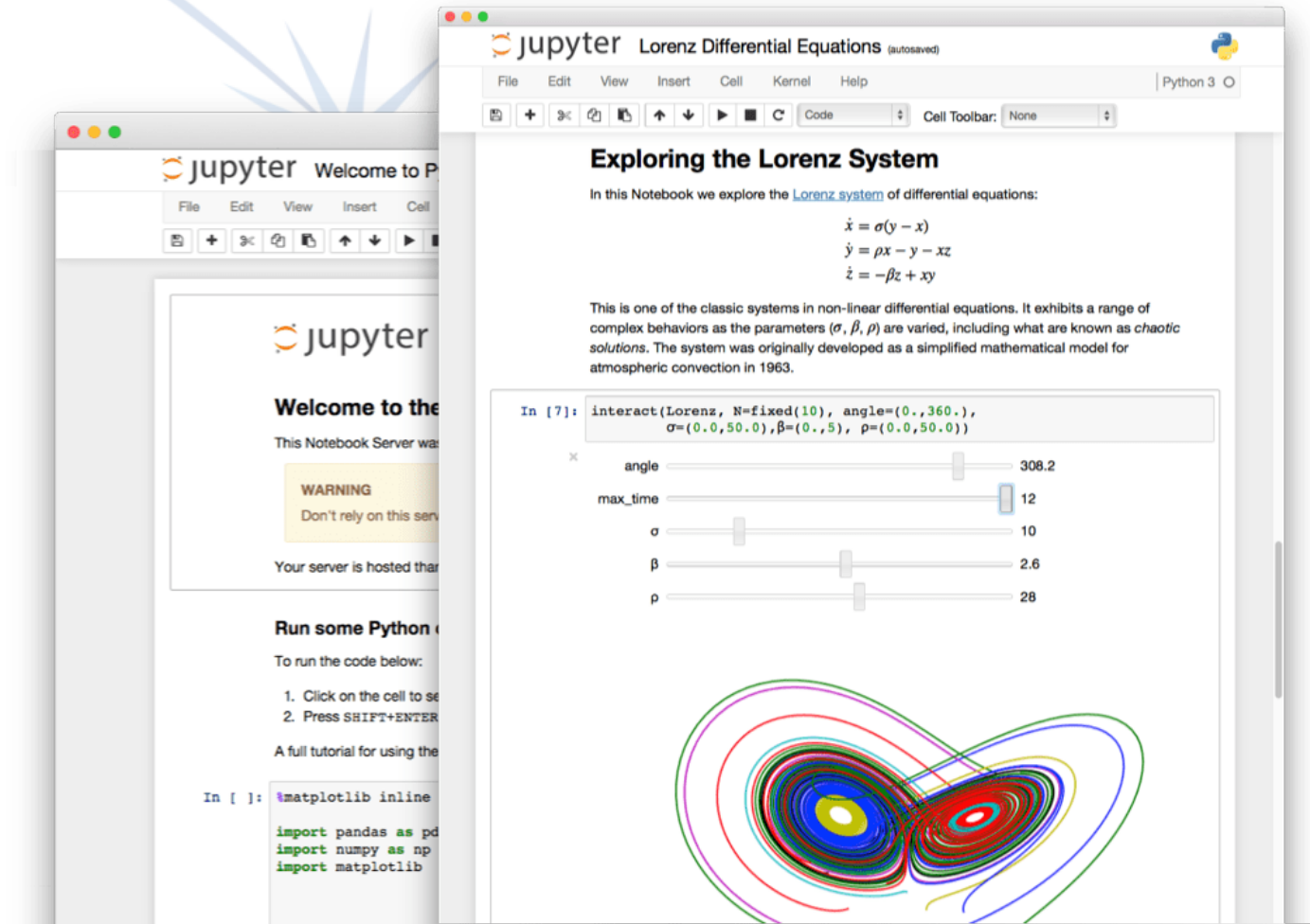
Jupyter Notebook

● Tutorial & Demo

- How to create and run a jupyter Notebook
- Interface: Cell types
- How to share/export a notebook
- How to install other Kernels

● Advanced features

- Jupyter Widgets
- Hiding the code
- Creating a presentation with Jupyter Notebooks



Jupyter Notebook

- Installing other kernels:

- Julia



- Download Julia

```
$ conda install julia
```

- Launch Julia

```
$ julia  
julia> using Pkg  
julia> Pkg.add("IJulia")  
julia> exit()
```

- Launch a new Jupyter session

- R



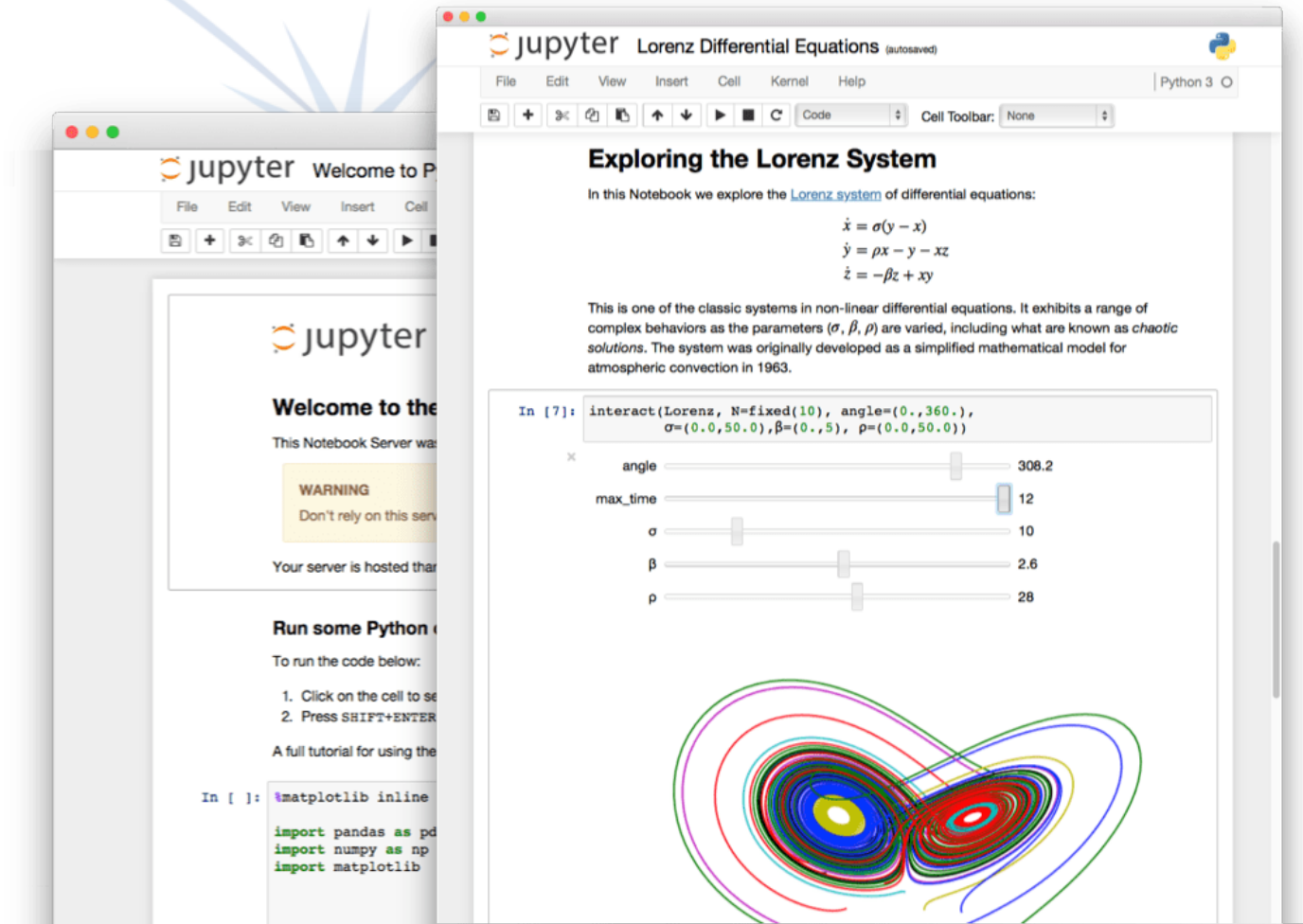
- Download R

- conda install -c r r-irkernel



Jupyter Notebook

- How to share a notebook:
 - GitHub/GitLab
 - NBViewer: <https://nbviewer.org/>
- An ipynb file can be exported to
 - HTML
 - LaTeX
 - PDF
 - Markdown
 - An executable script
 - ReStructured Text
 - RevealJS

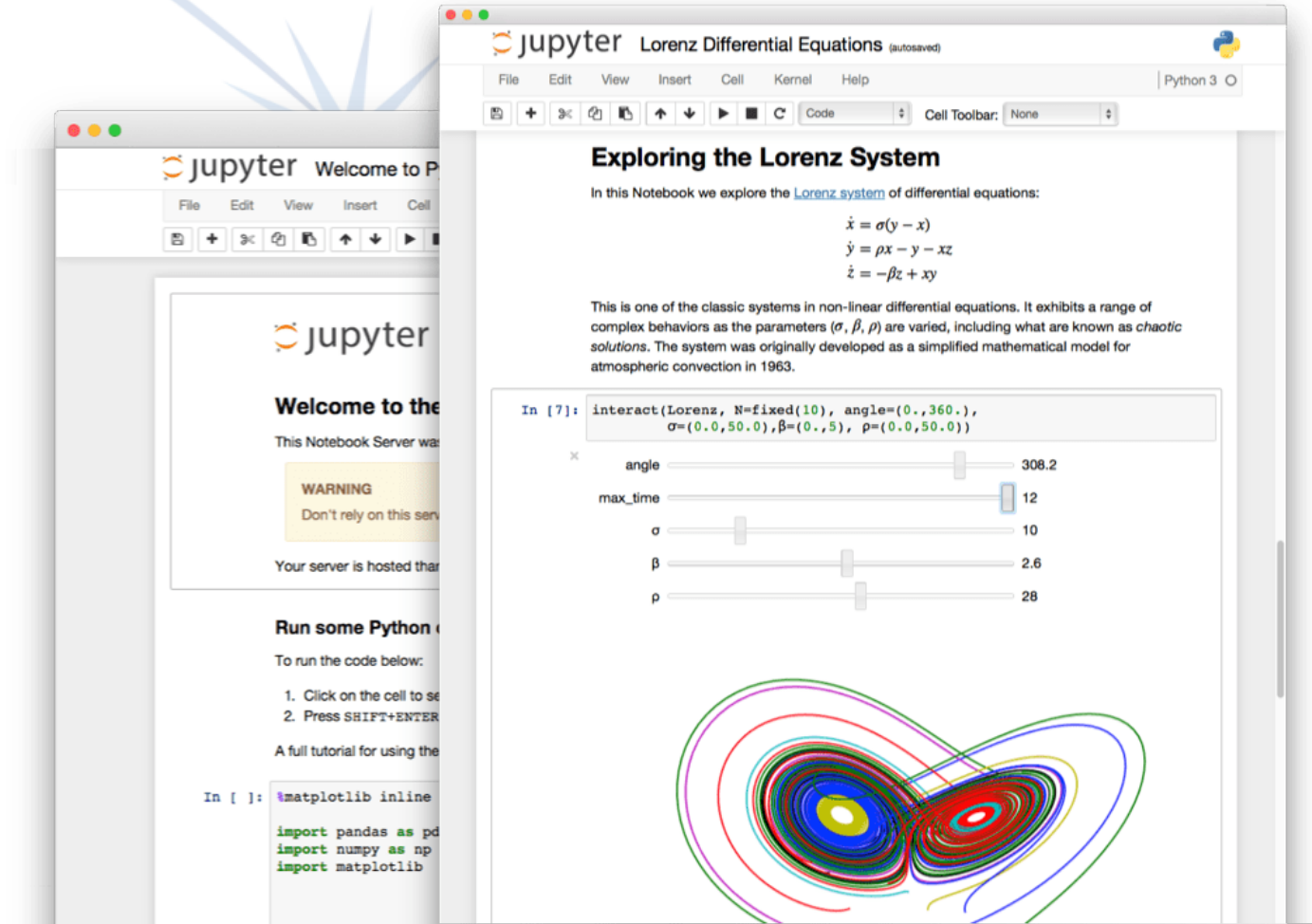


\$ jupyter nbconvert <input_notebook.ipynb> --to <output format>



Jupyter Notebook

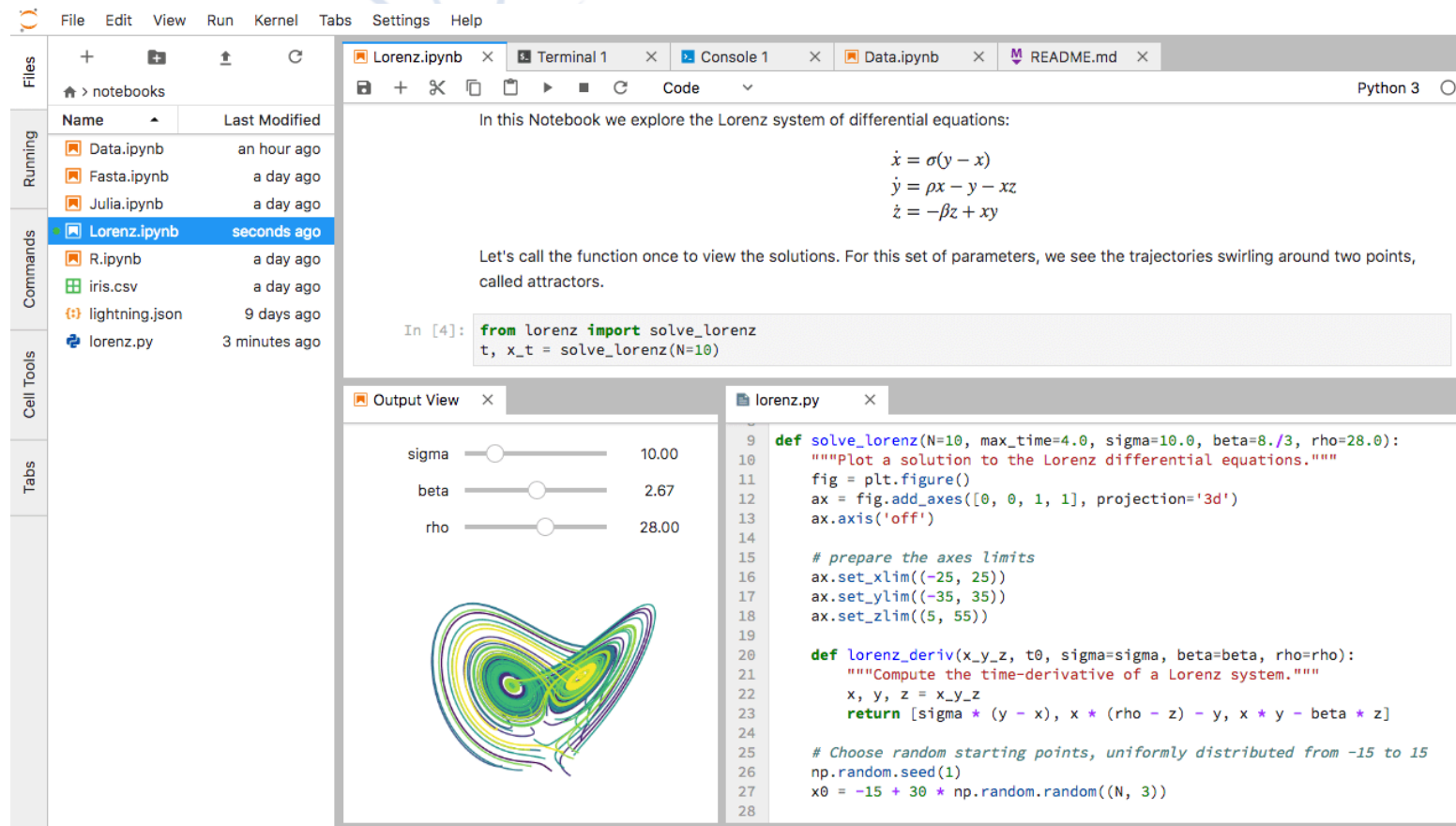
- Nice tool to
 - (quick) Exploratory analysis
 - Nice way to present results*
 - Markdown + code
- Maybe not that optimal to
 - Develop
 - Show results in between large amounts of code



- Interactive and modular web-based environment
 - Includes same functionalities Jupyter notebooks
 - File manager
 - Launcher of Jupyter interfaces

- Launch it


```
$ jupyter-lab
```



The screenshot displays the Jupyter Lab interface. On the left, a sidebar shows a file manager with a list of notebooks: Data.ipynb, Fasta.ipynb, Julia.ipynb, Lorenz.ipynb (selected), R.ipynb, iris.csv, lightning.json, and lorenz.py. The main area shows the 'Lorenz.ipynb' notebook. The code cell contains the following text:

In this Notebook we explore the Lorenz system of differential equations:

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy \end{aligned}$$

Let's call the function once to view the solutions. For this set of parameters, we see the trajectories swirling around two points, called attractors.

```
In [4]: from lorenz import solve_lorenz
t, x_t = solve_lorenz(N=10)
```

Below the code cell, there is an 'Output View' showing a 3D plot of the Lorenz attractor. The plot is a colorful, swirling trajectory in a 3D space. To the right of the plot, there are three sliders for parameters: sigma (set to 10.00), beta (set to 2.67), and rho (set to 28.00). Further to the right, a code cell shows the implementation of the Lorenz system:

```
9 def solve_lorenz(N=10, max_time=4.0, sigma=10.0, beta=8./3, rho=28.0):
10     """Plot a solution to the Lorenz differential equations."""
11     fig = plt.figure()
12     ax = fig.add_axes([0, 0, 1, 1], projection='3d')
13     ax.axis('off')
14
15     # prepare the axes limits
16     ax.set_xlim((-25, 25))
17     ax.set_ylim((-35, 35))
18     ax.set_zlim((5, 55))
19
20     def lorenz_deriv(x_y_z, t0, sigma=sigma, beta=beta, rho=rho):
21         """Compute the time-derivative of a Lorenz system."""
22         x, y, z = x_y_z
23         return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]
24
25     # Choose random starting points, uniformly distributed from -15 to 15
26     np.random.seed(1)
27     x0 = -15 + 30 * np.random.random((N, 3))
28
```



Reproducible computer environments



Turn a Git repo into a collection of interactive notebooks

<https://mybinder.org/>

- Builds and runs a docker container of the repository
 - by installing all the dependencies within the `environment.yml` file



Docker

- Application that virtualizes software packages into a "container".
 - The resulting file is called a **docker image**
 - When executing an image, you are running a **docker container**
 - Other virtualisation alternatives: Singularity, Podman...
- How to install Docker ?
 - Recommendation, use docker desktop:
<https://docs.docker.com/get-docker/>
- Pros:
 - Portability
 - Reproducibility
- Cons:
 - Learning curve





● Downloading and running the school image

```
$ docker pull ghcr.io/escape2020/escape-datascience-  
school2022:latest
```

```
$ docker run -p 8888:8888 ghcr.io/escape2020/escape-datascience-  
school2022:latest
```



Reproducible computer environments

- This was possible thanks to the use of **open science**, some of its tools and following FAIR principles

- An open github project – version control + Zenodo record (Findable)
(Accessible)
- A file describing the **computer environment** (Interoperable)
 - conda: environment.yml
 - Python: requirements.txt
- A **public license** + (Reusable)
The use of open source CI/CD tools



Thank you for your attention

Any question or comment ?

