

# Premier pas avec JupyterLab

J. Seiler

**BiGEst**



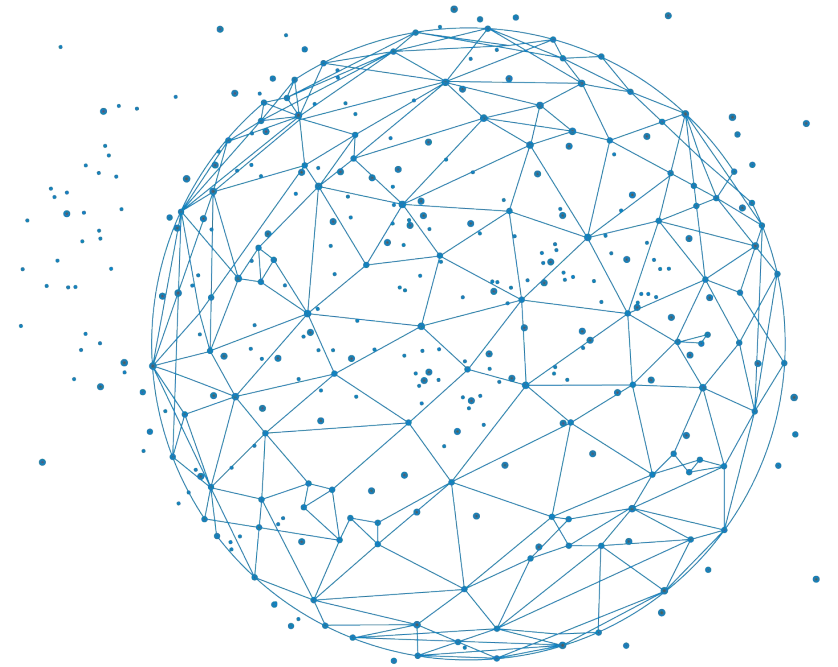


Literate programming

Jupyter Notebook

JupyterLab

# Literate programming





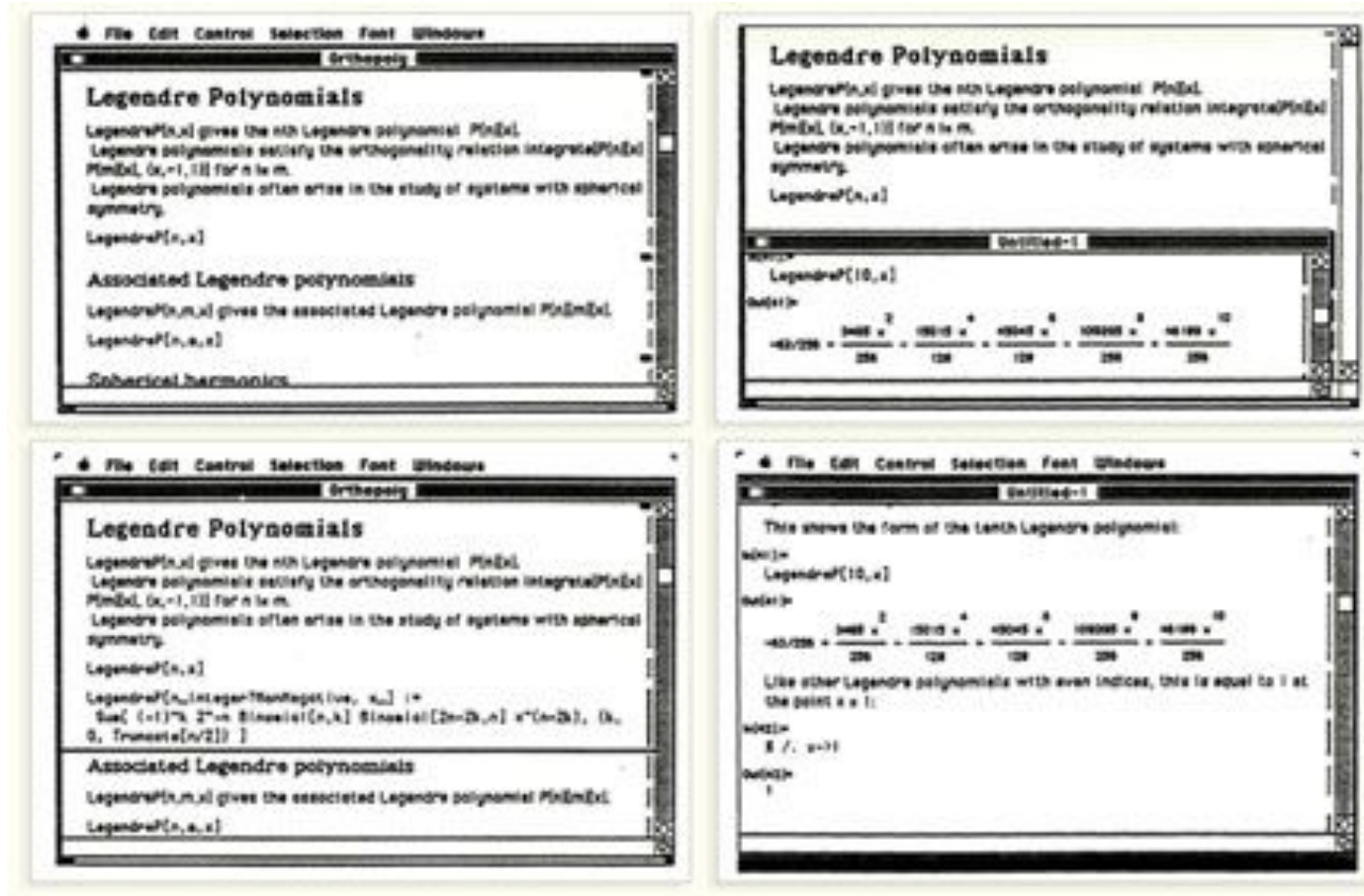
Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.

— Donald E. Knuth, *Literate Programming*, 1984



A literate computing environment is one that allows users not only to execute commands interactively, but also to store in a literate document the results of these commands along with figures and free-form text.

- Millman KJ and Perez F (2014)



Wolfram Mathematica notebook (1987)



What does it look like ?

**jupyter** Lorenz Differential Equations (autosaved)

File Edit View Insert Cell Kernel Help Python 3

Code Cell Toolbar: None

### Exploring the Lorenz System

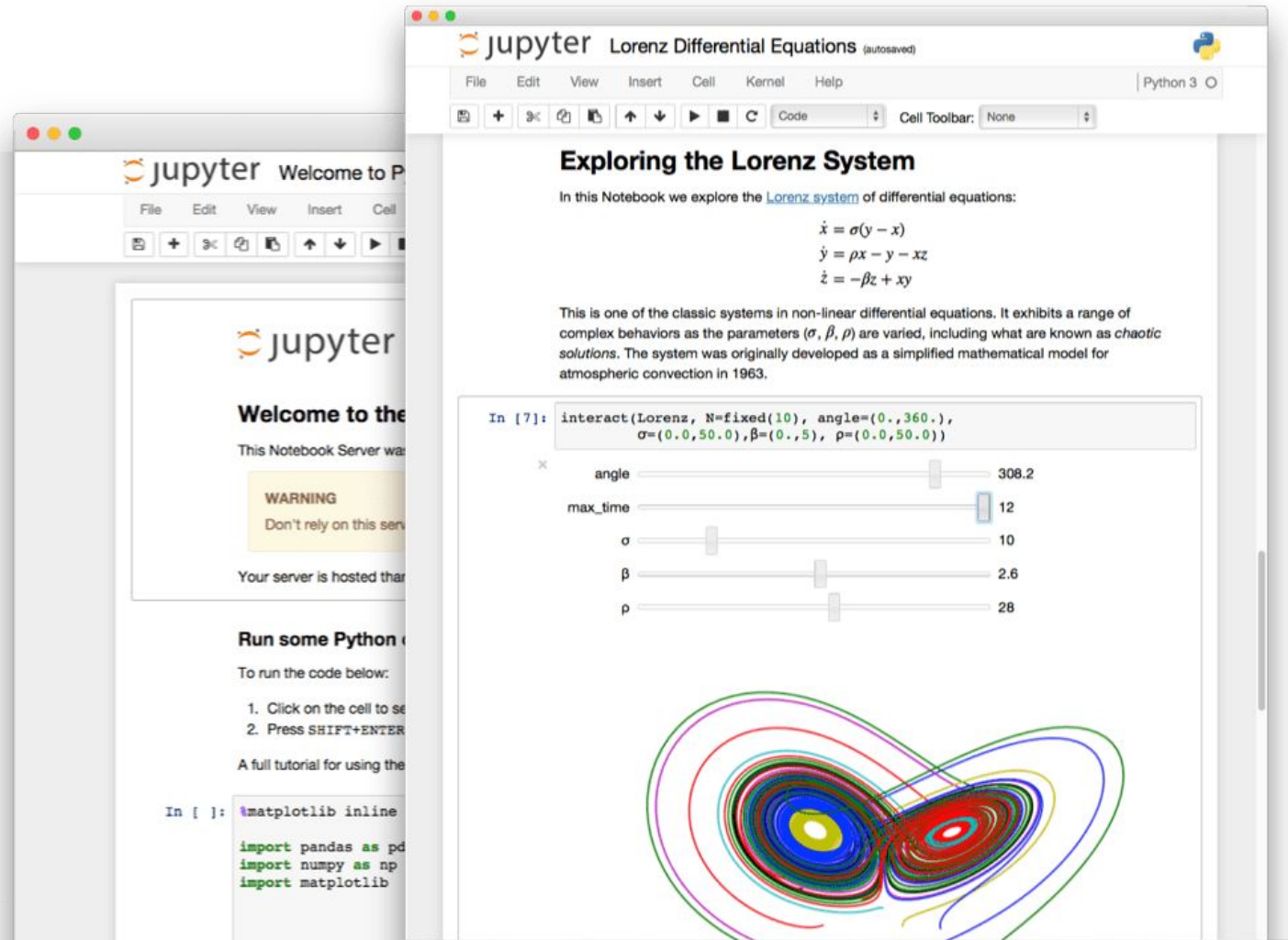
In this Notebook we explore the [Lorenz system](#) of differential equations:

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

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters  $(\sigma, \beta, \rho)$  are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

```
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
                sigma=(0.0,50.0), beta=(0.,5), rho=(0.0,50.0))
```

angle 308.2  
max\_time 12  
 $\sigma$  10  
 $\beta$  2.6  
 $\rho$  28



Interactive programming interface

allowing to combine both natural and computer languages.

In one file:

- Explanations
- Code
- Results
- Graphs and plots





## Why using literate programming frameworks ?

### Use cases:

- Labbook
- Day to day analyses
- Analysis reports
- Writing scientific article

# Example of an article entirely written using a notebook



## File (on a repository)

The screenshot shows a GitHub repository for 'colomoto-docker'. The file 'invasion.ipynb' is selected, showing its commit history and content. The file content is the title and author information of the article: 'Prediction of Mutations to Control Pathways Enabling Tumour Cell Invasion with the CoLoMoTo Interactive Notebook (Tutorial)'. The authors listed are Nicolas Levy, Aurélien Naldi, Céline Hernandez, Gautier Stoll, Denis Thieffry, Andrei Zinovyev, Laurence Calzone, and Loïc Paulevé.

## Published article

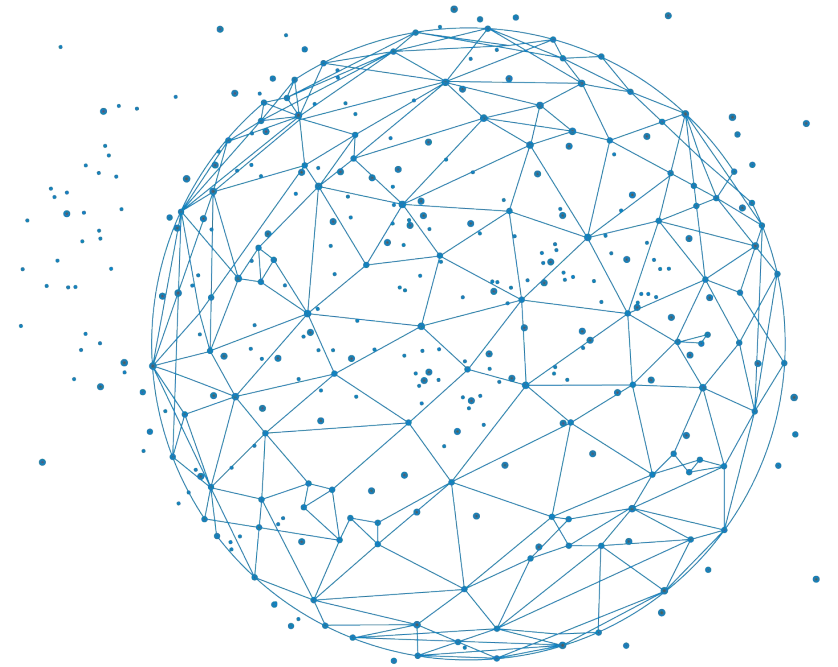
The screenshot shows the published article on the Frontiers website. The article title is 'Prediction of Mutations to Control Pathways Enabling Tumor Cell Invasion with the CoLoMoTo Interactive Notebook (Tutorial)'. The authors are listed as Nicolas Levy, Aurélien Naldi, Céline Hernandez, Gautier Stoll, Denis Thieffry, Andrei Zinovyev, Laurence Calzone, and Loïc Paulevé. The article has 1,866 total views and is a protocols article published in Frontiers in Physiology on 06 July 2018. The DOI is 10.3389/fphys.2018.00787.

## Executable file

The screenshot shows the executable file page on nbviewer. The article title is 'Prediction of Mutations to Control Pathways Enabling Tumour Cell Invasion with the CoLoMoTo Interactive Notebook (Tutorial)'. The authors are listed as Nicolas Levy, Aurélien Naldi, Céline Hernandez, Gautier Stoll, Denis Thieffry, Andrei Zinovyev, Laurence Calzone, and Loïc Paulevé. The abstract is visible, starting with 'Boolean and multi-valued logical formalisms are increasingly used to model complex cellular networks. To ease the development and analysis of logical models, a series of software tools have been proposed, often with specific assets. However, combining these tools typically implies a series of cumbersome software installation and model conversion steps. In this respect, the CoLoMoTo Interactive Notebook provides a joint distribution of several logical modelling software tools, along with an interactive web Python interface easing the chaining of complementary analyses.'

DOI:10.3389/fphys.2018.00787

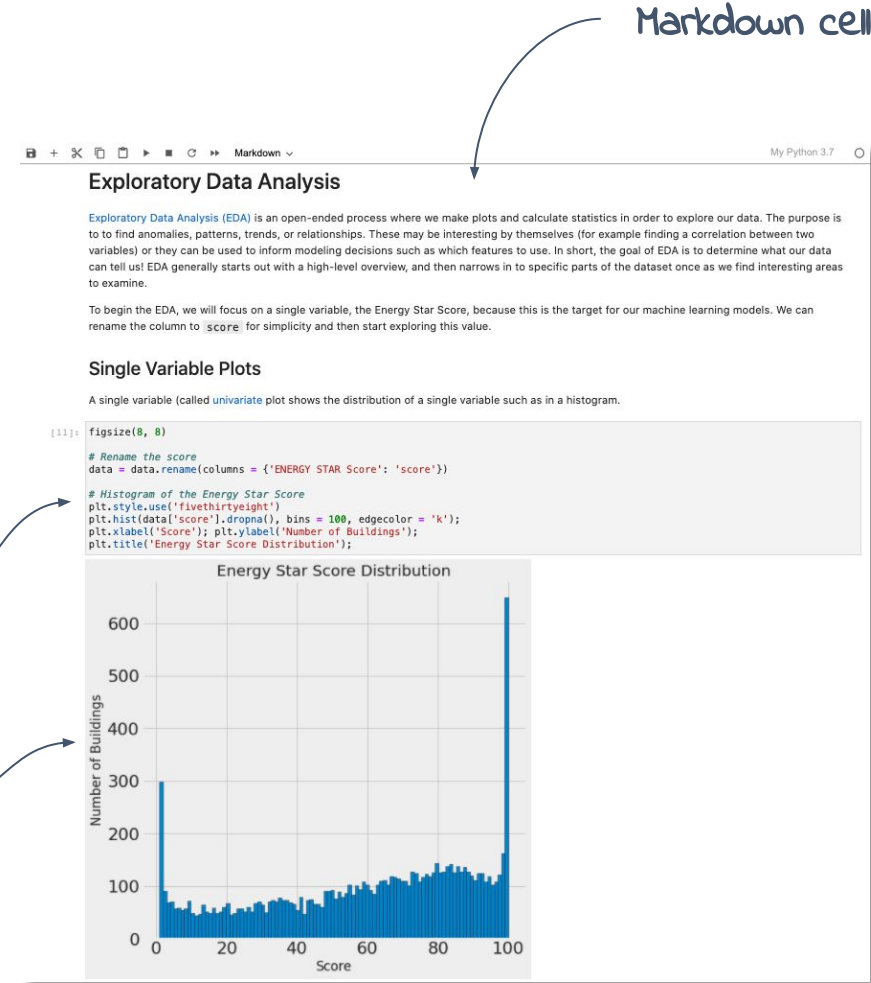
# Jupyter Notebook



# What is a Jupyter Notebook

- A special file with the .ipynb extension
- A combination of Markdown and code
- Code can be executed in the notebook
- Code output is integrated directly into the notebook

Markdown cell



Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an open-ended process where we make plots and calculate statistics in order to explore our data. The purpose is to find anomalies, patterns, trends, or relationships. These may be interesting by themselves (for example finding a correlation between two variables) or they can be used to inform modeling decisions such as which features to use. In short, the goal of EDA is to determine what our data can tell us! EDA generally starts out with a high-level overview, and then narrows in to specific parts of the dataset once we find interesting areas to examine.

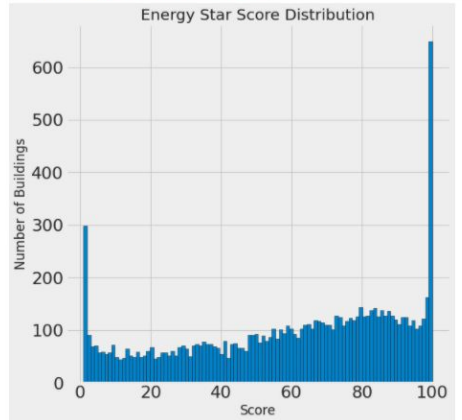
To begin the EDA, we will focus on a single variable, the Energy Star Score, because this is the target for our machine learning models. We can rename the column to `_score` for simplicity and then start exploring this value.

Single Variable Plots

A single variable (called univariate plot shows the distribution of a single variable such as in a histogram.

```
[11]: figsize(8, 8)
# Rename the score
data = data.rename(columns = {'ENERGY STAR Score': 'score'})

# Histogram of the Energy Star Score
plt.style.use('fivethirtyeight')
plt.hist(data['score'], dropna(), bins = 100, edgecolor = 'k');
plt.xlabel('Score'); plt.ylabel('Number of Buildings');
plt.title('Energy Star Score Distribution');
```



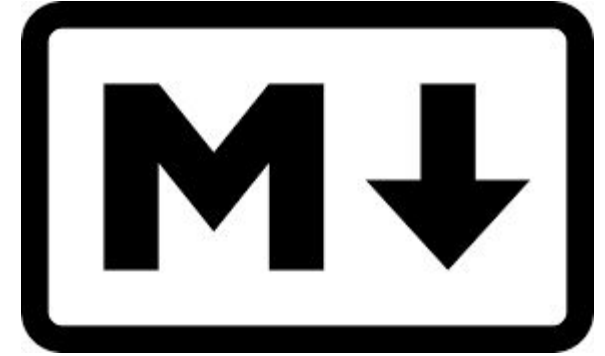
Code cell

Result

Markdown is a Lightweight markup language

Designed to be :

- easy to write using any generic text editor (plain-text-formatting syntax)
- easy to read in its raw form



From GitHub's help page

<https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax>

#### Example in markdown

```
# Heading
```

```
## Sub-heading
```

```
### Another deeper heading
```

```
A [link](http://example.com).
```

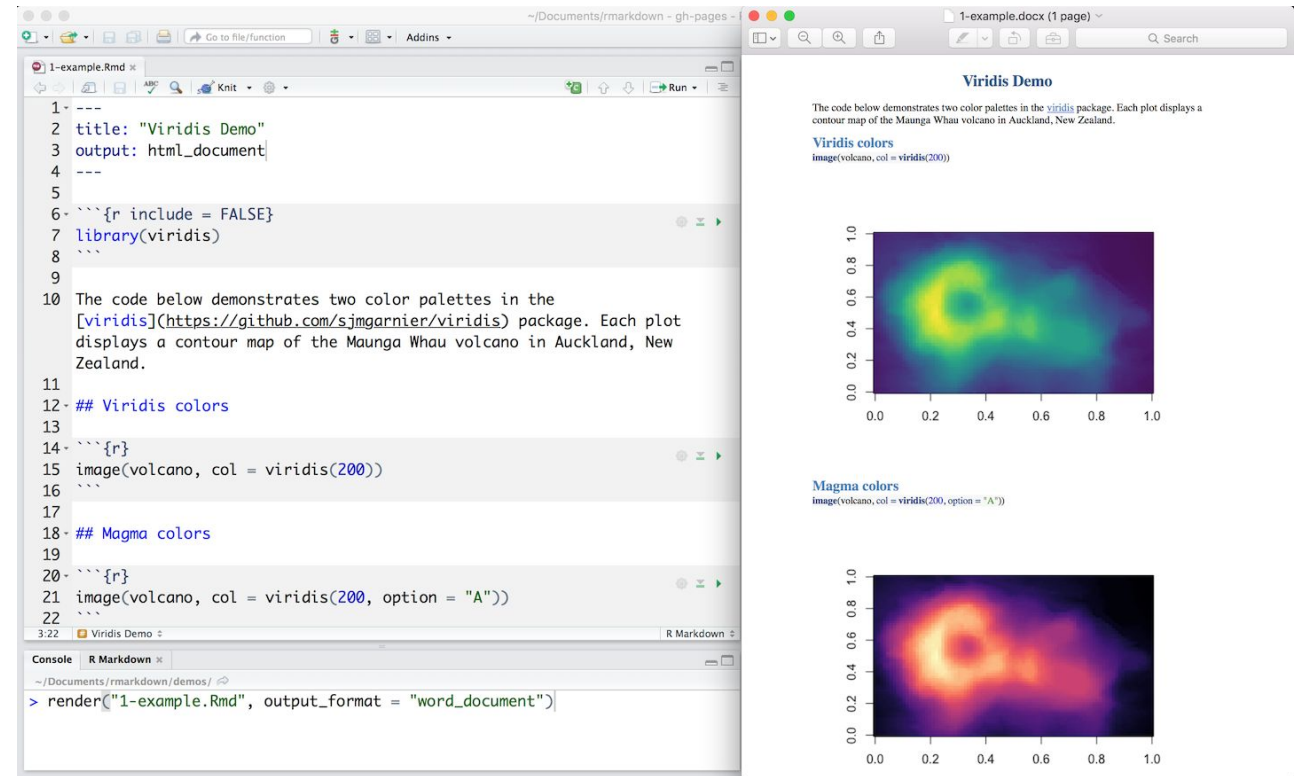
```
Text attributes _italic_, *italic*, **bold**, `monospace`.
```

```
Bullet list:
```

- \* apples
- \* oranges
- \* pears

But how is this useful for literate programming?

When you want to weave both code (to be interpreted) and formatting information, you precisely need a lightweight language for the formatting part.



The screenshot shows an R Markdown document being rendered into a document format. The left pane displays the source code, and the right pane shows the rendered output.

```
1 ---
2 title: "Viridis Demo"
3 output: html_document
4 ---
5
6 ```{r include = FALSE}
7 library(viridis)
8 ```
9
10 The code below demonstrates two color palettes in the
11 [viridis](https://github.com/sjmgarnier/viridis) package. Each plot
12 displays a contour map of the Maunga Whau volcano in Auckland, New
13 Zealand.
14
15 ## Viridis colors
16 ```{r}
17 image(volcano, col = viridis(200))
18 ```
19
20 ## Magma colors
21 ```{r}
22 image(volcano, col = viridis(200, option = "A"))
23 ```
```

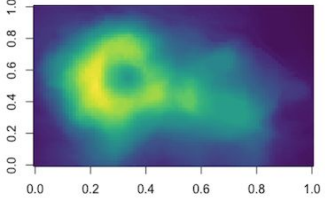
The rendered output shows the following structure:

### Viridis Demo

The code below demonstrates two color palettes in the `viridis` package. Each plot displays a contour map of the Maunga Whau volcano in Auckland, New Zealand.

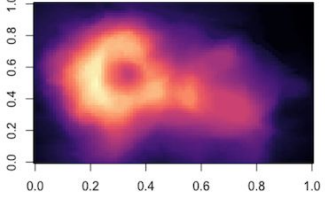
#### Viridis colors

```
image(volcano, col = viridis(200))
```



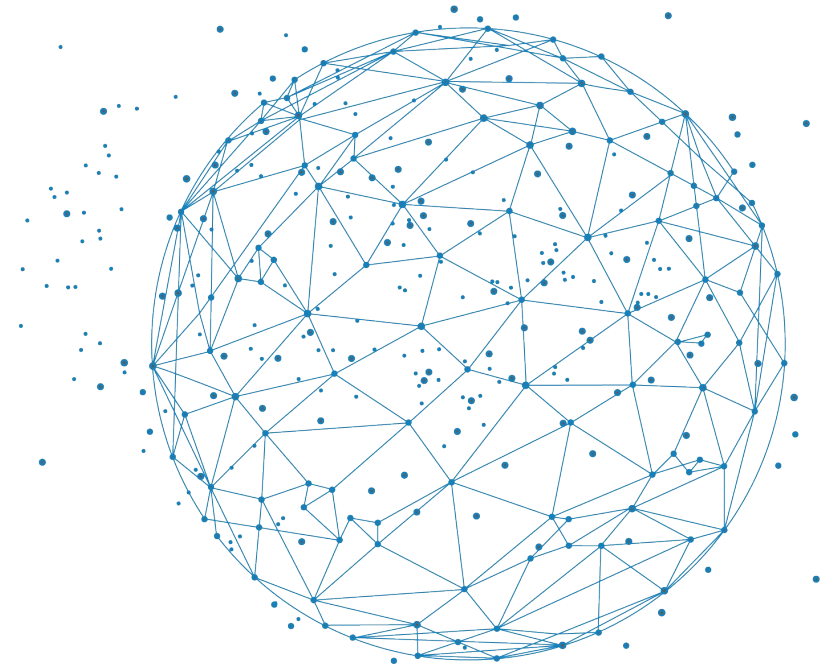
#### Magma colors

```
image(volcano, col = viridis(200, option = "A"))
```





# JupyterLab

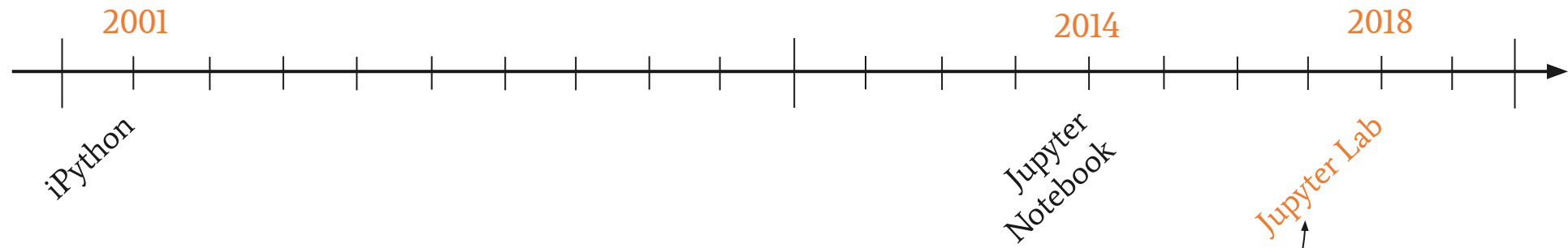




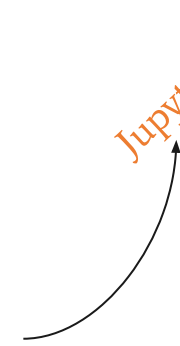


## How to create and use notebooks ?

You need a **Notebook server**



A complete environment for **online** data analysis



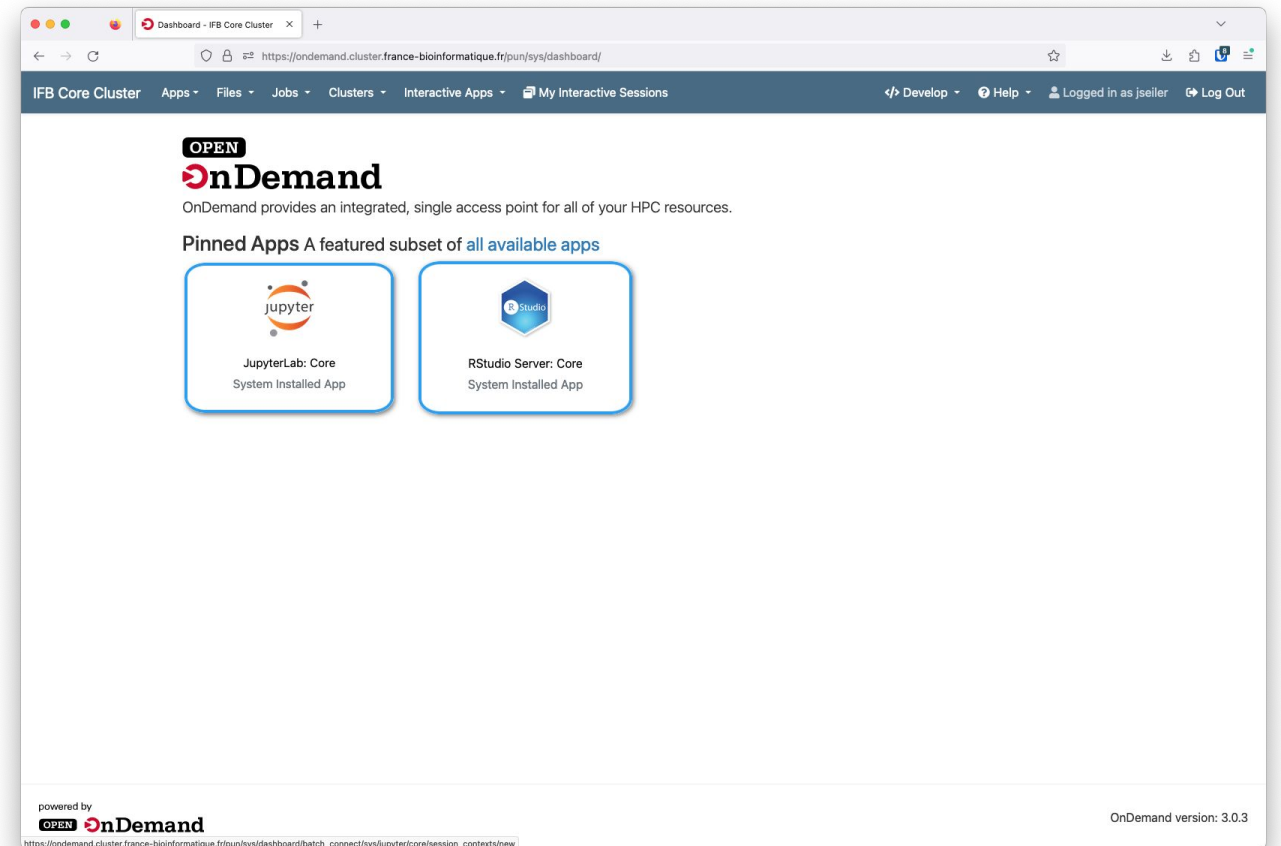


On the IFB Core Cluster, you can launch your own JupyterLab server from the Open OnDemand portal : <https://ondemand.cluster.france-bioinformatique.fr>





Use your IFB Core Cluster account to connect to the IFB Open OnDemand portal



<https://ondemand.cluster.france-bioinformatique.fr>



## Start a **JupyterLab Core** server

Reservation : 2412\_fairbioinfo

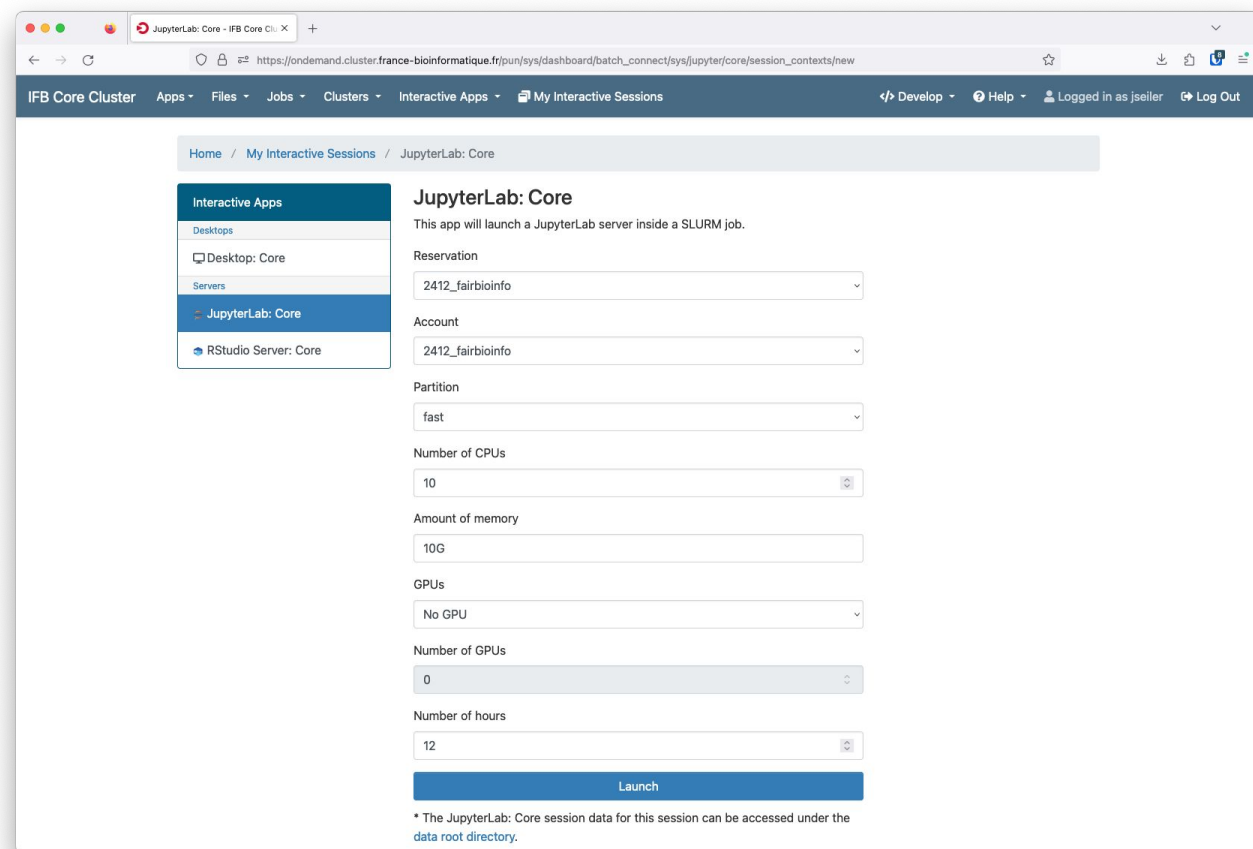
Number of CPUs : 10

Amount of memory : 10G

Number of hours : 6 hours

Click **Launch**

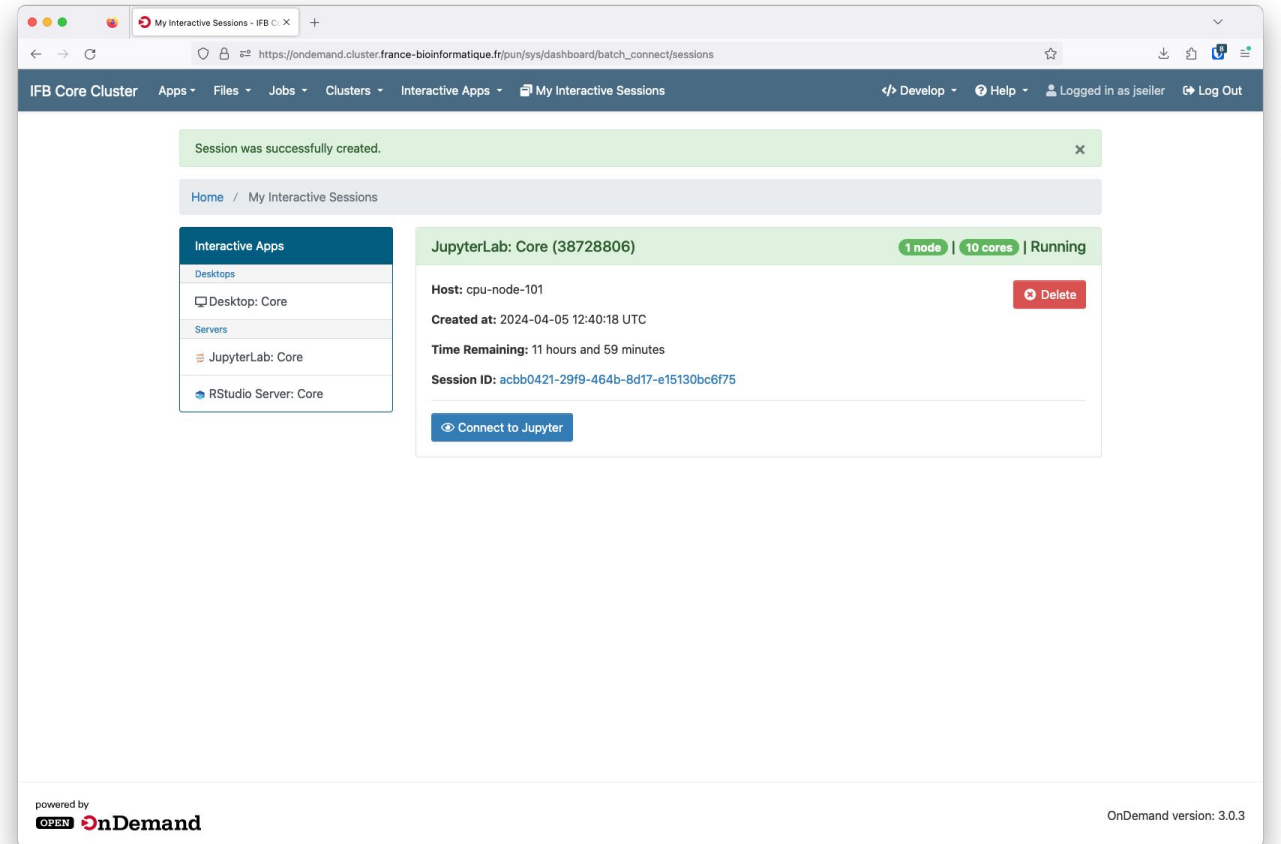
and wait of few seconds...



`https://ondemand.cluster.france-bioinformatique.fr`

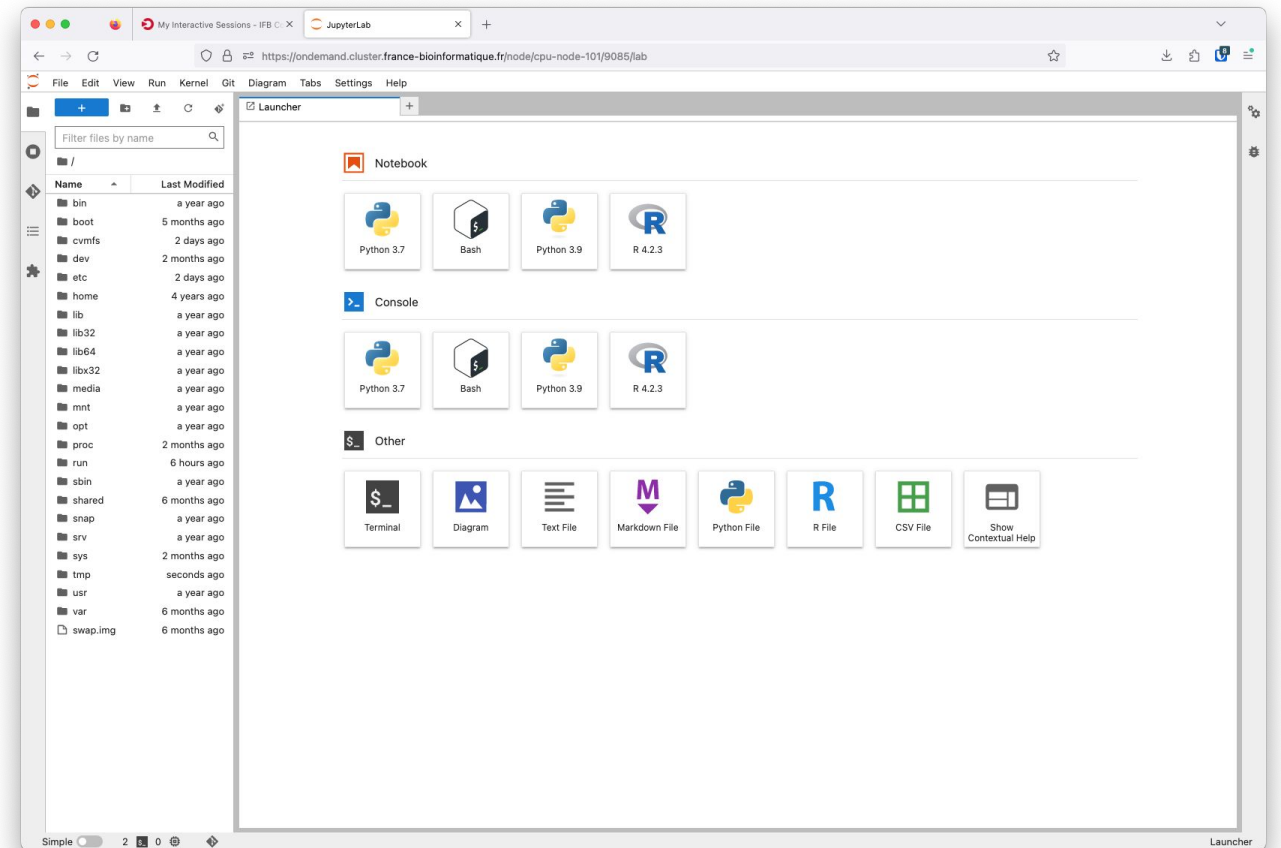


When your Jupyter app is ready  
Click **Connect to Jupyter**



`https://ondemand.cluster.france-bioinformatique.fr`

## Welcome to JupyterLab



<https://ondemand.cluster.france-bioinformatique.fr>