

Laboratory Notebooks

What is this?

Année : _____

Numéro : _____

Cahier de laboratoire

Cahier de laboratoire

Laboratory notebook

Numéro :
Number: _____

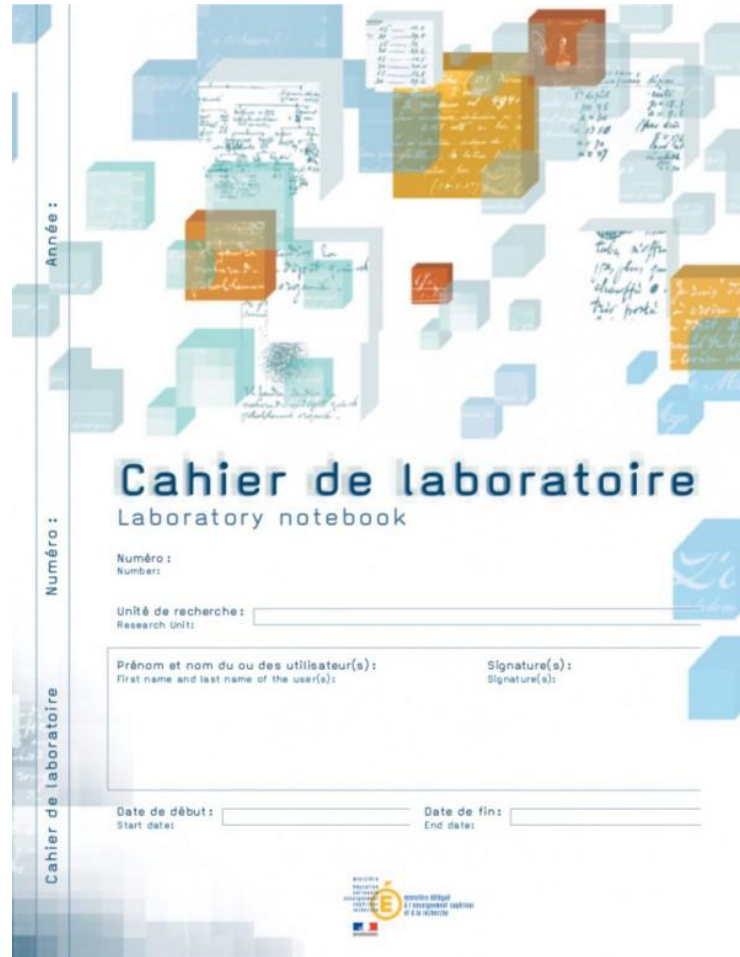
Unité de recherche : _____
Research Unit:

Prénom et nom du ou des utilisateur(s) : _____
First name and last name of the user(s):

Signature(s) : _____
Signature(s):

Date de début : _____ Date de fin : _____
Start date: End date:

Ministère de l'Enseignement Supérieur et de la Recherche
Université de la Méditerranée



What is it for ?



Année : _____

Numéro : _____

Cahier de laboratoire
Laboratory notebook

Unité de recherche :
Research Unit: _____

Prénom et nom du ou des utilisateur(s):
First name and last name of the user(s): _____

Signature(s):
Signature(s): _____

Date de début :
Start date: _____

Date de fin :
End date: _____

Logo of the French Republic and the logo of the Institut National de Recherche pour l'Exploitation de la Mer (INRM) are visible at the bottom.

What is it for ?

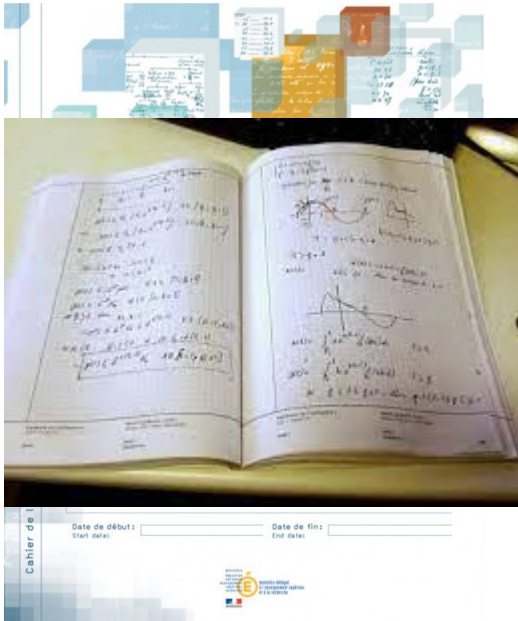


The laboratory notebook allows :

- day-to-day recording of the details of the work
- to report on the progress and scientific experimentation, from the idea to the conclusion
- to keep knowledge in a lab

Also very useful for drafting a patent or for proving anteriority.

What is it for ?



A legal tool :

- Each notebook and the pages are numbered.
- On the cover page, we find on each notebook the mentions of the owner of the results.
- Each page has two parts at the bottom intended to be dated and signed: two signatures = two people, user and third party (witness),
 - ideally a third party not involved in the research work but capable of understanding it

<https://www.curie.asso.fr/-Cahier-de-laboratoire-national-.html>
<https://slideplayer.fr/slide/3817405/>

Who is it for ?



For all those who carry out research work :

- researchers,
- engineers,
- technicians,
- doctoral students,
- trainees,
- etc.

Are you using it ?



Année : _____

Numéro : _____

Cahier de laboratoire

Cahier de laboratoire
Laboratory notebook

Numéro :
Number: _____

Unité de recherche :
Research unit: _____

Prénom et nom du ou des utilisateur(s):
First name and last name of the user(s): _____

Signature(s):
Signature(s): _____

Date de début :
Start date: _____

Date de fin :
End date: _____

Logo of the French Republic and the European Union.

Electronic Laboratory Notebooks (ELN)

Modern LN since 2009 (C.U.R.I.E. Network)

But less and less adapted to recent evolutions of our work

- Increased data quantity
- Change in the nature of data
- Dematerialization
- Security

We need an electronic tool for individual traceability.

Electronic Laboratory Notebook



Le rapport du groupe de travail « Cahier de laboratoire électronique » (ELN) présente une vision partagée sur la définition, le cadrage, les usages et le périmètre fonctionnel de l'ELN, qui doit pouvoir s'intégrer dans les environnements informatiques et institutionnels existants. Il émet un ensemble de recommandations sur les critères de choix d'un outil et intègre une liste comparative d'outils existants.

Electronic Laboratory Notebook

Calendrier du projet



Note de synthèse

- Caractéristiques techniques souhaitées
- Inventaire des solutions existantes sur le marché

Janvier 2021

Mise en œuvre de la stratégie adoptée

- Mise en œuvre de la procédure d'achat (licences, etc.)
- Cahier des charges de la solution à développer

Mars-Juillet 2021

Mise à disposition de ou des solutions sélectionnées

- Outils déployés sous licences CNRS
- Solution électronique développée en interne

Janvier 2022 – Juin 2023



This session

- Introduction to Laboratory Notebooks
- Literate programming
- Markdown
- Notebooks for bioinformatics
 - R Notebooks (Rmarkdown)
 - Jupyter(Lab)
- Practical session (JupyterLab)



Literate programming

Literate programming

What is literate programming ?

“ Let us change our traditional attitude to the construction of programs:

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

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

Literate programming

”Literate programming is a programming paradigm introduced by Donald Knuth in which a computer program is given an explanation of its logic in a natural language, such as English, interspersed with snippets of macros and traditional source code, from which compilable source code can be generated.” Donald Knuth, 1984.

Wikipedia, 18/08/2020

https://en.wikipedia.org/wiki/Literate_programming#Workflow

Literate programming

What does it look like ?

The image displays two overlapping Jupyter Notebook windows. The foreground window, titled "Lorenz Differential Equations (autosaved)", shows a notebook with the following content:

- 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 (σ, β, ρ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.
- Code cell (In [7]):

```
interact(Lorenz, N=fix(10), angle=(0.,360.),
         sigma=(0.0,50.0), beta=(0.,5), rho=(0.0,50.0))
```
- Interactive sliders for parameters: angle (0 to 308.2), max_time (0 to 12), σ (0 to 10), β (0 to 2.6), and ρ (0 to 28).
- A plot of the Lorenz attractor showing a complex, chaotic trajectory.

The background window, titled "Welcome to Jupyter", shows the notebook's introductory text and a warning message:

- Welcome to the Jupyter Notebook**
- This Notebook Server was started by you.
- WARNING**
Don't rely on this server for production use.
- Your server is hosted that...
- Run some Python code**
- To run the code below:
- 1. Click on the cell to see the code.
- 2. Press SHIFT+ENTER
- A full tutorial for using the notebook is available at [http://jupyter-notebook.github.io](#).
- Code cell (In []):

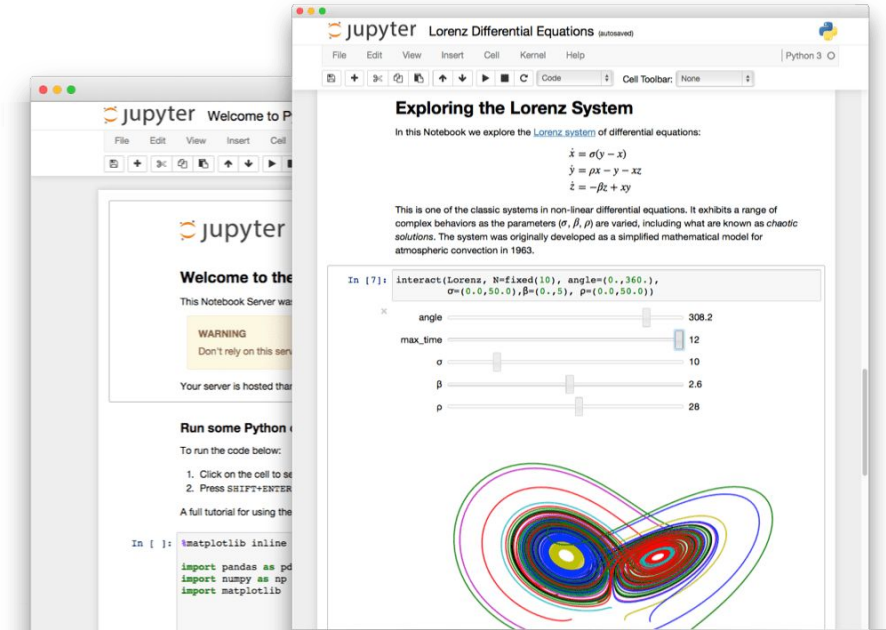
```
!matplotlib inline
import pandas as pd
import numpy as np
import matplotlib
```

Literate programming

Interactive programming interface
allowing to combine both natural and
computer languages.

In one file:

- Explanations
- Code
- Results
- Graphs and plots



Literate programming

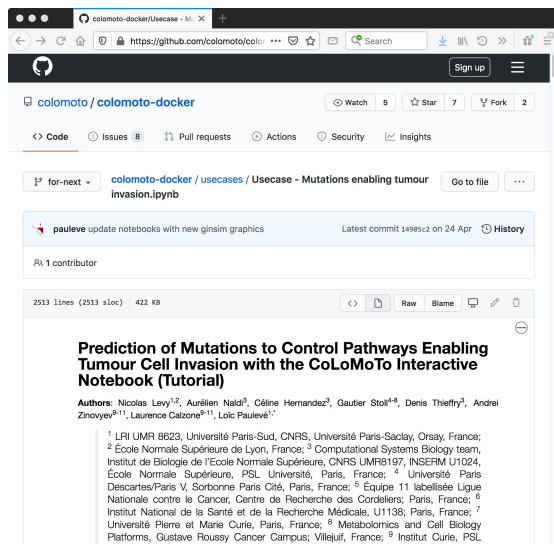
Why using literate programming frameworks ?

Use cases:

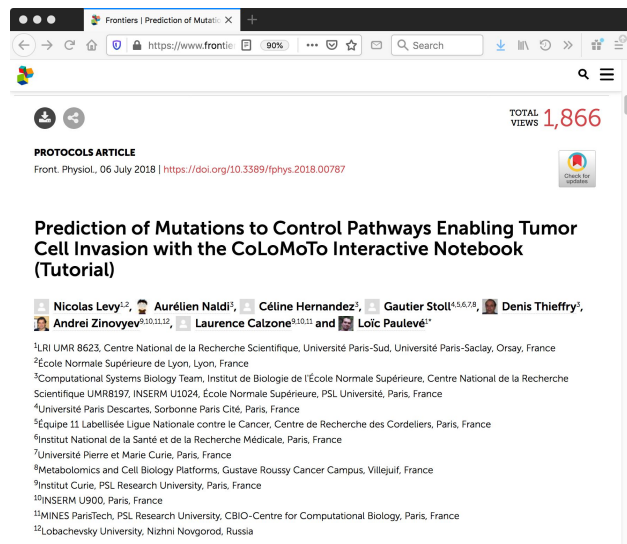
- Labbook
- Day to day analyses
- Analysis reports
- Writing scientific article

Example of an article entirely written using a notebook

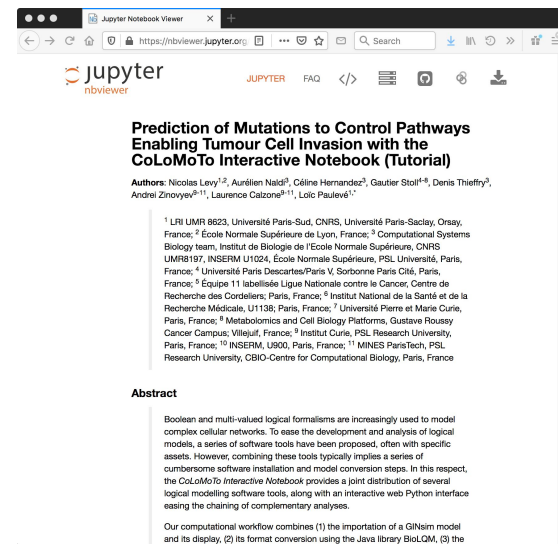
File (on a repository)



Published article



Executable file



DOI:10.3389/fphys.2018.00787

Markup / Markdown

Markup

Definition

A markup language uses tags to define elements within a document.

Three different types and usage :

- Presentational (used by traditional word-processing systems)
 - Markup is invisible
- Procedural, provides instructions to process the text (e.g. TeX, PostScript)
 - Markup is visible and can be directly manipulated by the author.
- Descriptive, to label documents parts (e.g. LaTeX, HTML, XML...)
 - Emphasizes the document structure.

Markdown

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

Markdown

You've probably seen it already on Wikipedia, GitHub (README)...

```
# Heading

## Sub-heading

### Another deeper heading

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

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

Bullet list:

* apples
* oranges
* pears
```

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>

Markdown

But how is this useful for literate programming?

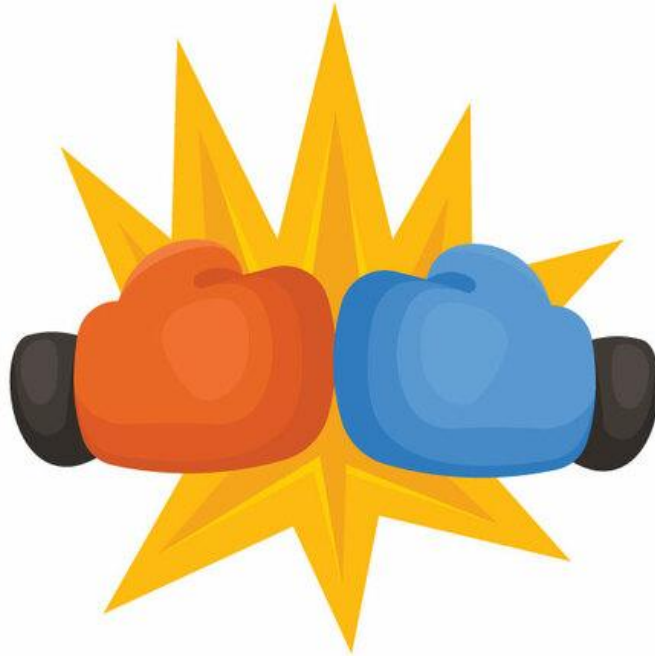
When you want to weave both code (to be interpreted) and formatting information, you precisely need a lightweight language for the formatting part.

Notebooks for bioinformatics

Notebooks for bioinformatics

R Notebooks

Jupyter(Lab)



R Notebooks : Sweave (2002)

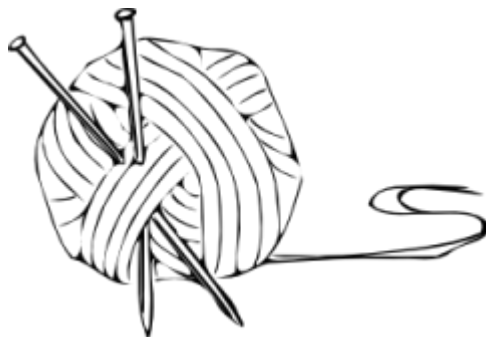
At the beginning, there was nothing.

Then came Sweave.

Leisch, Friedrich (2002). "Sweave, Part I: Mixing R and LaTeX: A short introduction to the Sweave file format and corresponding R functions"

And people saw that the path would be long...

R Notebooks : knitR (2011)



”The knitr package was designed to be a transparent engine for dynamic report generation with R, solve some long-standing problems in Sweave, and combine features in other add-on packages into one package”

<https://yihui.org/knitr/>

R Notebooks : RMarkdown (2012)

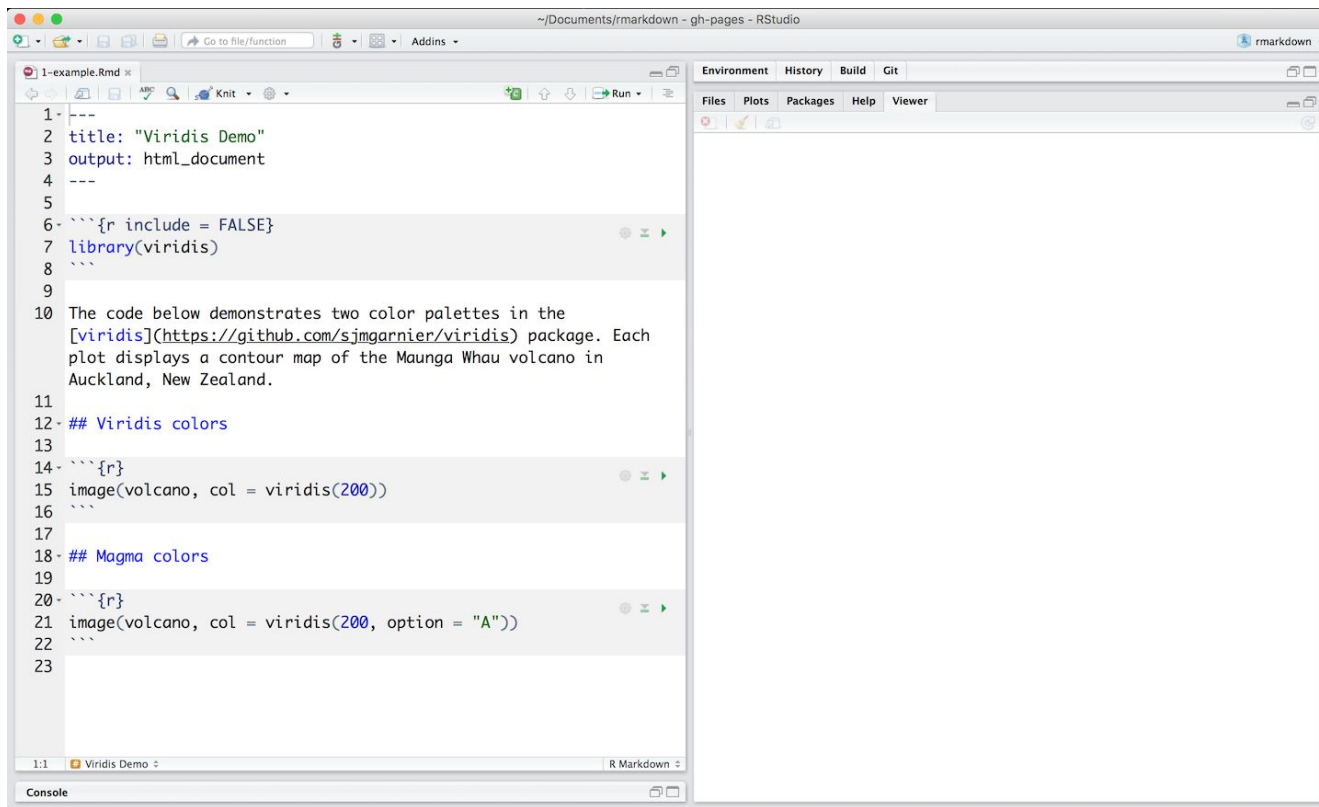


”When you run render, R Markdown feeds the .Rmd file to knitr, which executes all of the code chunks and creates a new markdown (.md) document which includes the code and its output.

The markdown file generated by knitr is then processed by pandoc which is responsible for creating the finished format.”

<https://rmarkdown.rstudio.com>

R Notebooks

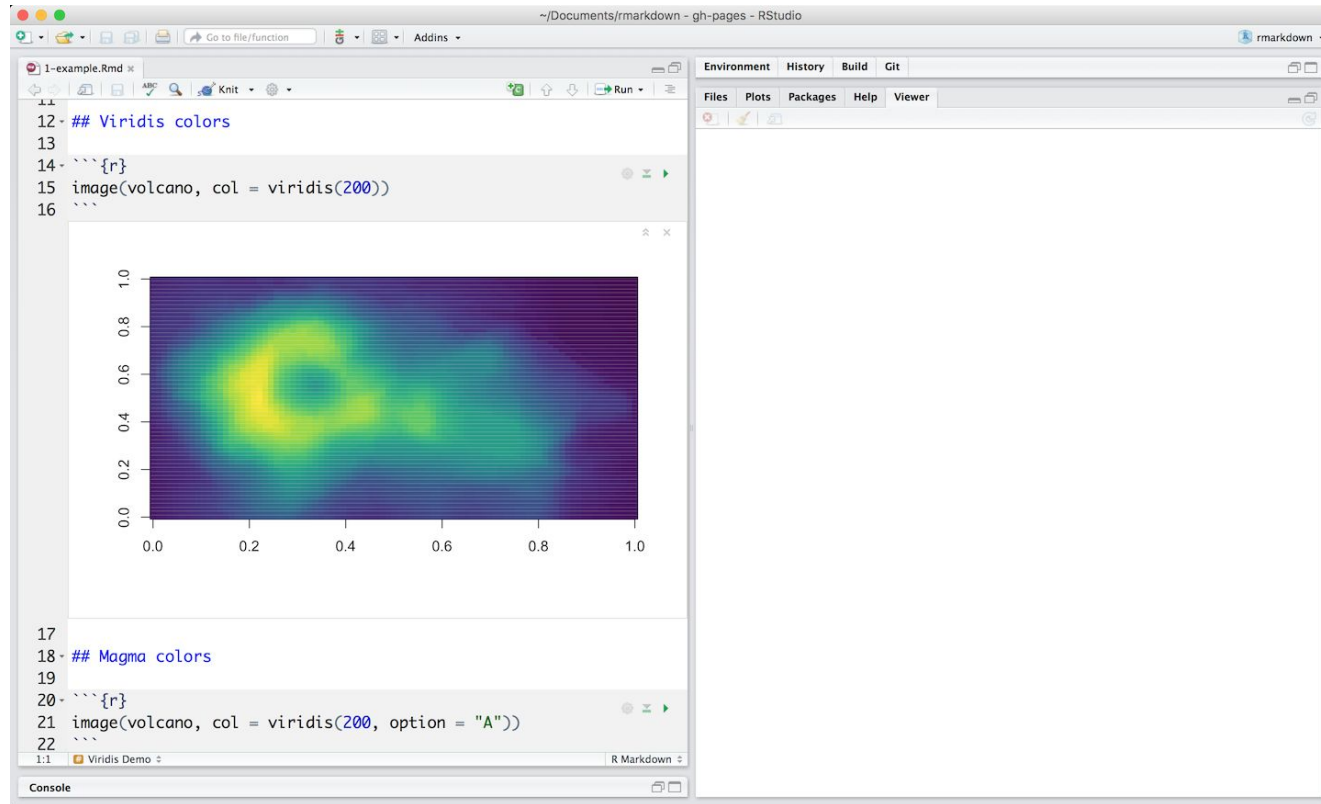


The screenshot shows the RStudio interface with an R Markdown document open. The document content is as follows:

```
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
12 plot displays a contour map of the Maunga Whau volcano in
13 Auckland, New Zealand.
14
15 ## Viridis colors
16
17 ```{r}
18 image(volcano, col = viridis(200))
19 ```
20
21 ## Magma colors
22
23 ```{r}
24 image(volcano, col = viridis(200, option = "A"))
25 ```
26
```

The right-hand pane of RStudio is currently empty, showing the Environment, History, Build, and Git tabs. The bottom status bar indicates the current position is 1:1 in the 'Viridis Demo' R Markdown document.

R Notebooks



The screenshot shows the RStudio interface with an R Notebook open. The notebook content is as follows:

```
12- ## Viridis colors
13
14- ```{r}
15 image(volcano, col = viridis(200))
16 ```
```

The output of the first code chunk is a heatmap of the 'volcano' dataset, rendered using the 'viridis' color palette. The x and y axes both range from 0.0 to 1.0. The plot shows a central bright yellow-green region that transitions to dark purple at the edges.

```
17
18- ## Magma colors
19
20- ```{r}
21 image(volcano, col = viridis(200, option = "A"))
22 ```
```

The second code chunk is partially visible but has not been executed. The RStudio interface includes a top toolbar with 'Knit' and 'Run' buttons, a right-hand pane with 'Environment', 'History', 'Build', and 'Git' tabs, and a bottom 'Console' pane.

R Notebooks and more...

Markdown Basics

Output Formats

Notebooks

Slide Presentations

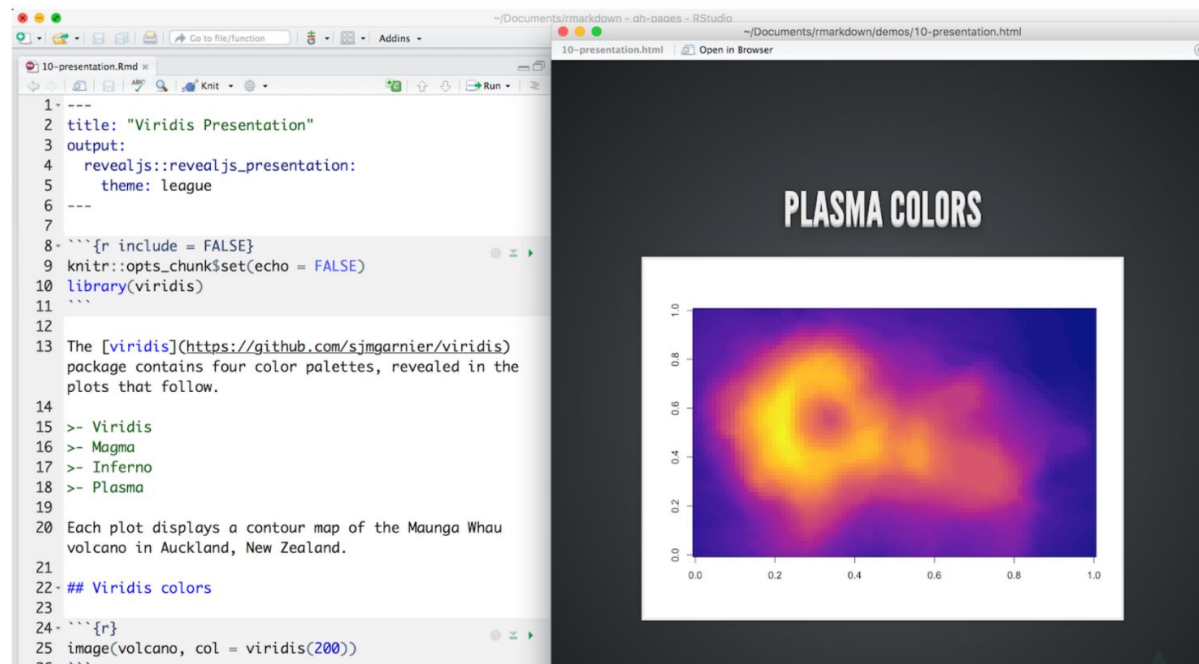
Dashboards

Websites

Interactive Documents

Cheatsheets

file below, which is available [here](#) on RStudio Cloud.



The screenshot displays the RStudio interface. On the left, the R script editor shows the following code:

```
1 ---
2 title: "Viridis Presentation"
3 output:
4   revealjs::revealjs_presentation:
5     theme: league
6 ---
7
8 ```{r include = FALSE}
9 knitr::opts_chunk$set(echo = FALSE)
10 library(viridis)
11 ```
12
13 The [viridis](https://github.com/sjmgarnier/viridis)
14 package contains four color palettes, revealed in the
15 plots that follow.
16
17 >- Viridis
18 >- Magma
19 >- Inferno
20 >- Plasma
21
22 Each plot displays a contour map of the Maunga Whau
23 volcano in Auckland, New Zealand.
24
25 ## Viridis colors
26
27 ```{r}
28 image(volcano, col = viridis(200))
29 ```
```

On the right, the rendered HTML output is shown in a browser window. It features a dark background with the text "PLASMA COLORS" in white at the top. Below the text is a square contour plot of the Maunga Whau volcano, rendered using the Plasma color palette. The plot's x and y axes both range from 0.0 to 1.0, with tick marks every 0.2 units.

Jupyter

- A bit of history...
- 2011 : IPython (interactive Python shell) with notebook functionalities
- 2014 : Spin-off project called *Project Jupyter* a non-profit, open-source project maintained by a strong Community
 - "Jupyter will always be 100% open-source software, free for all to use and released under the liberal terms of the modified BSD license"
 - A reference to the three core programming languages supported by Jupyter (Julia, Python and R)

<https://jupyter.org/>

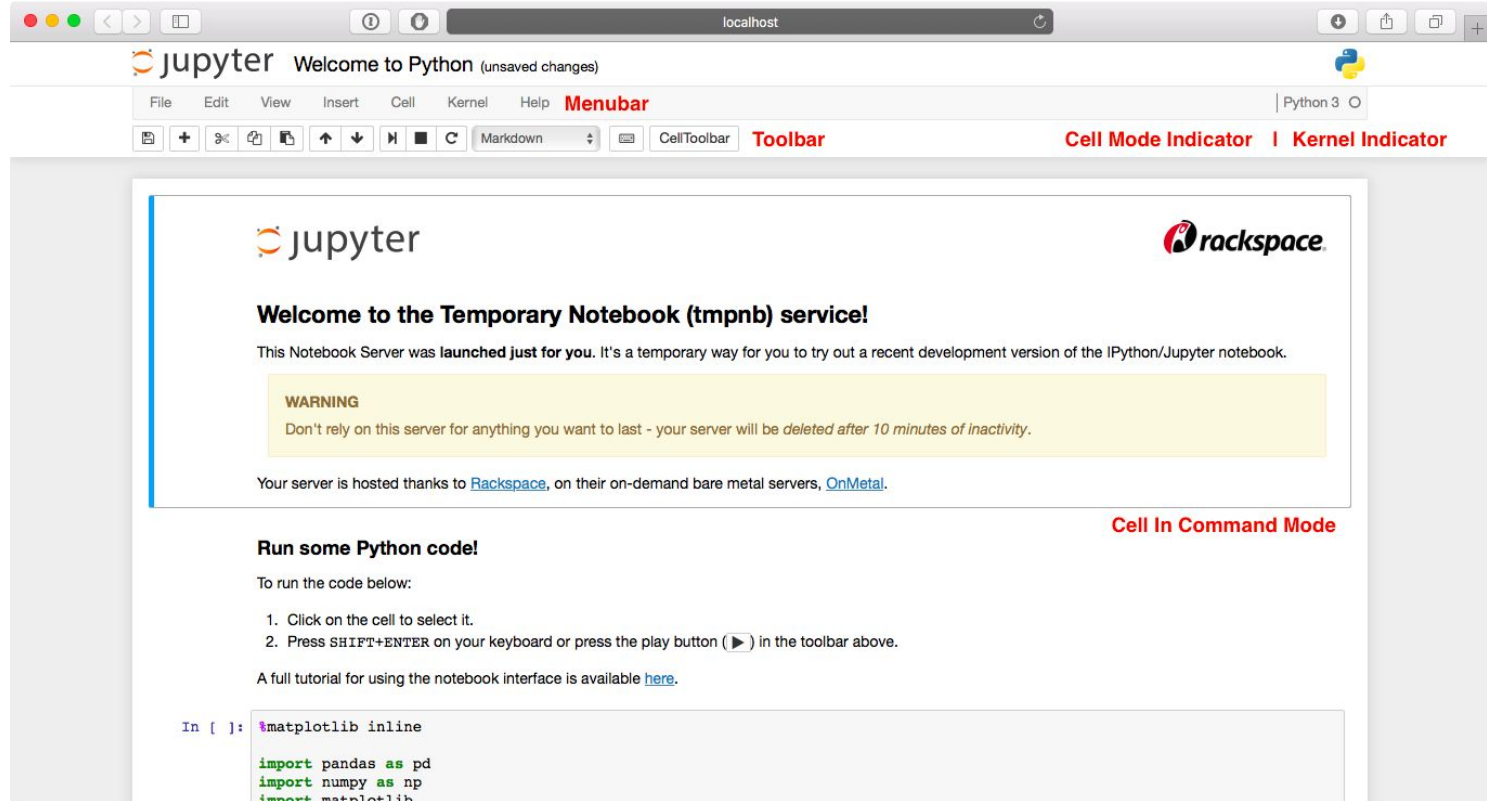
Jupyter

What is it exactly ?

Web-based interactive computational environment.

- Web-based : client/server
- Interactive : notebook system
- Computational environment : console, many kernels available...

Jupyter : Notebook editor




The screenshot shows a web browser window with the URL `localhost`. The page title is "jupyter Welcome to Python (unsaved changes)". The browser's address bar shows "localhost". The page content includes the Jupyter logo, a "Welcome to the Temporary Notebook (tmpnb) service!" message, a warning box, and instructions on how to run Python code. The interface also features a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Help" options, and a toolbar with various icons for cell operations. The page is styled with a light gray background and a white content area.

localhost

jupyter Welcome to Python (unsaved changes)

File Edit View Insert Cell Kernel Help **Menubar** Python 3

Cell Mode Indicator | Kernel Indicator

jupyter 

Welcome to the Temporary Notebook (tmpnb) service!

This Notebook Server was **launched just for you**. It's a temporary way for you to try out a recent development version of the IPython/Jupyter notebook.

WARNING
Don't rely on this server for anything you want to last - your server will be *deleted after 10 minutes of inactivity*.

Your server is hosted thanks to [Rackspace](#), on their on-demand bare metal servers, [OnMetal](#).

Run some Python code! Cell In Command Mode

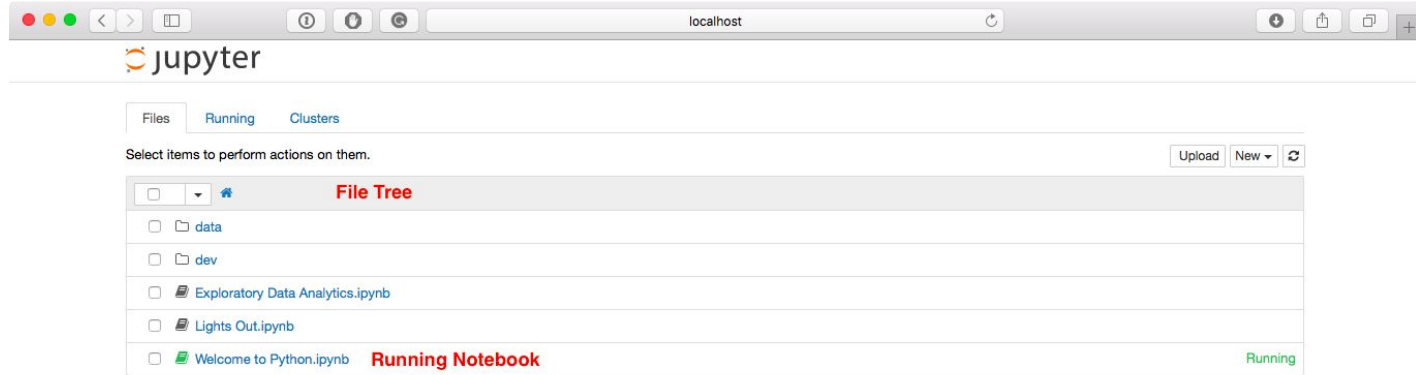
To run the code below:

1. Click on the cell to select it.
2. Press **SHIFT+ENTER** on your keyboard or press the play button (▶) in the toolbar above.

A full tutorial for using the notebook interface is available [here](#).

```
In [ ]: %matplotlib inline
import pandas as pd
import numpy as np
import matplotlib
```

Jupyter : Dashboard (Project Jupyter only)



JupyterLab : Dashboard

The image shows the JupyterLab Launcher interface. At the top, there is a menu bar with 'File', 'Edit', 'View', 'Run', 'Kernel', 'Tabs', 'Settings', and 'Help'. Below the menu bar is a toolbar with a search icon and a refresh icon. The left sidebar contains a file browser with a search bar labeled 'Filter files by name' and a list of files and folders:

Name	Last Modified
/	
courses	a day ago
labbook	a day ago
meetings	a day ago
Dockerfile	6 days ago
Untitled.ip...	6 days ago

The main area is titled 'Launcher' and is divided into three sections:

- Notebook**: Contains three icons for 'Python 3 (ipykernel)', 'Julia 1.7.2', and 'R'.
- Console**: Contains three icons for 'Python 3 (ipykernel)', 'Julia 1.7.2', and 'R'.
- Other**: Contains five icons for 'Terminal', 'Text File', 'Markdown File', 'Julia File', and 'Python File'. Below these are two more icons for 'R File' and 'Show'.

At the bottom left, there is a 'Simple' toggle switch and a status bar showing '0' and '0'. At the bottom right, the word 'Launcher' is displayed.