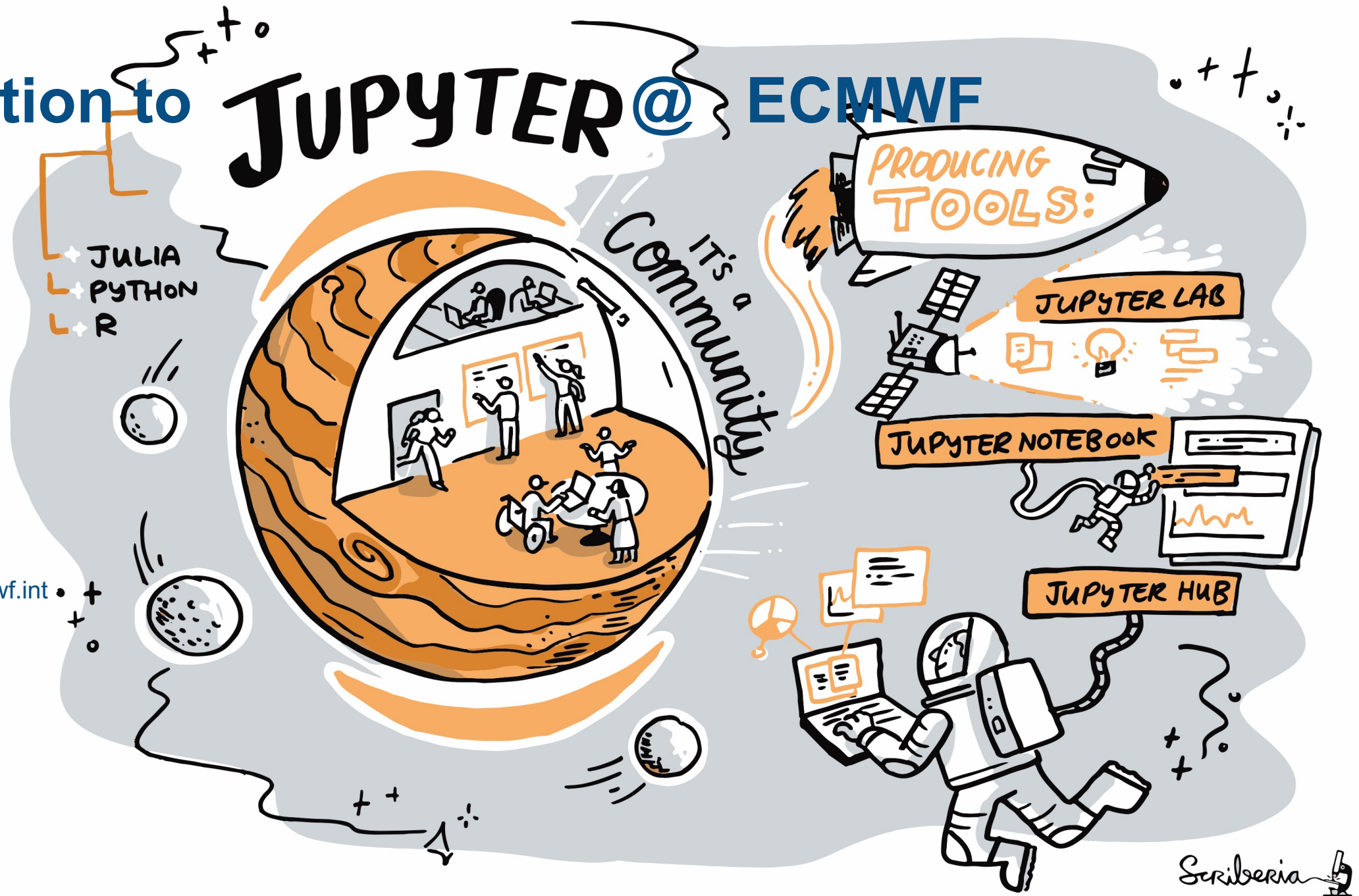


Introduction to JUPYTER@ECMWF



Milana Vučković

milana.vuckovic@ecmwf.int

TERMINOLOGY



JUPYTER

JULIA
PYTHON
R

IT'S A
Community

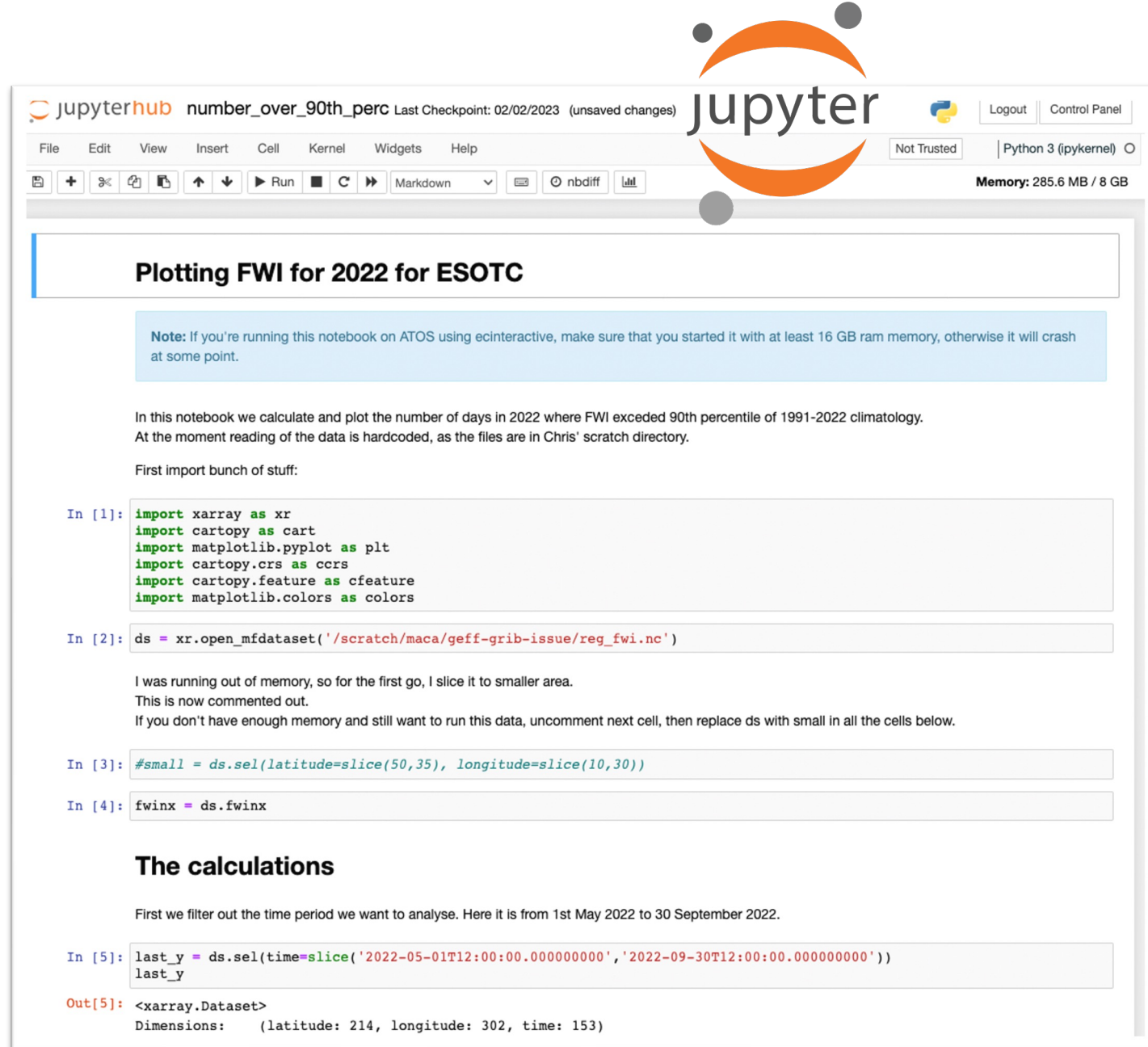


TERMINOLOGY

Jupyter Notebook

A notebook document (file extension **.ipynb**) is a document that can be rendered in a web browser

- It is a file, which stores your work in JSON format
- Based on a set of open standards for interactive computing
- Allows development of custom applications with embedded interactive computing.
- Can be extended by third parties



The screenshot shows a Jupyter Notebook interface with the title "number_over_90th_perc". The notebook content includes a title "Plotting FWI for 2022 for ESOTC", a note about memory requirements, and several code cells. The code cells contain Python code for importing libraries, opening a dataset, slicing it, and filtering it. The output of the last cell shows a dataset with dimensions (latitude: 214, longitude: 302, time: 153).

```
import xarray as xr
import cartopy as cart
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import matplotlib.colors as colors
```

```
ds = xr.open_mfdataset('/scratch/maca/geff-grib-issue/reg_fwi.nc')
```

```
#small = ds.sel(latitude=slice(50,35), longitude=slice(10,30))
```

```
fwinx = ds.fwinx
```

```
last_y = ds.sel(time=slice('2022-05-01T12:00:00.000000000', '2022-09-30T12:00:00.000000000'))
last_y
```

Out[5]: <xarray.Dataset>
Dimensions: (latitude: 214, longitude: 302, time: 153)

TERMINOLOGY

jupyterhub number_over_90th_perc Last Checkpoint: 02/02/2023 (unsaved changes) jupyter Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel) Memory: 285.6 MB / 8 GB

Plotting FWI for 2022 for ESOTC

Note: If you're running this notebook on ATOS using ecinteractive, make sure that you started it with at least 16 GB ram memory, otherwise it will crash at some point.

In this notebook we calculate and plot the number of days in 2022 where FWI exceeded 90th percentile of 1991-2022 climatology. At the moment reading of the data is hardcoded, as the files are in Chris' scratch directory.

First import bunch of stuff:

```
In [1]: import xarray as xr
import cartopy as cart
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import matplotlib.colors as colors
```

```
In [2]: ds = xr.open_mfdataset('/scratch/maca/geff-grib-issue/reg_fwi.nc')
```

I was running out of memory, so for the first go, I slice it to smaller area. This is now commented out. If you don't have enough memory and still want to run this data, uncomment next cell, then replace ds with small in all the cells below.

```
In [3]: #small = ds.sel(latitude=slice(50,35), longitude=slice(10,30))
```

```
In [4]: fwinx = ds.fwinx
```

The calculations

First we filter out the time period we want to analyse. Here it is from 1st May 2022 to 30 September 2022.

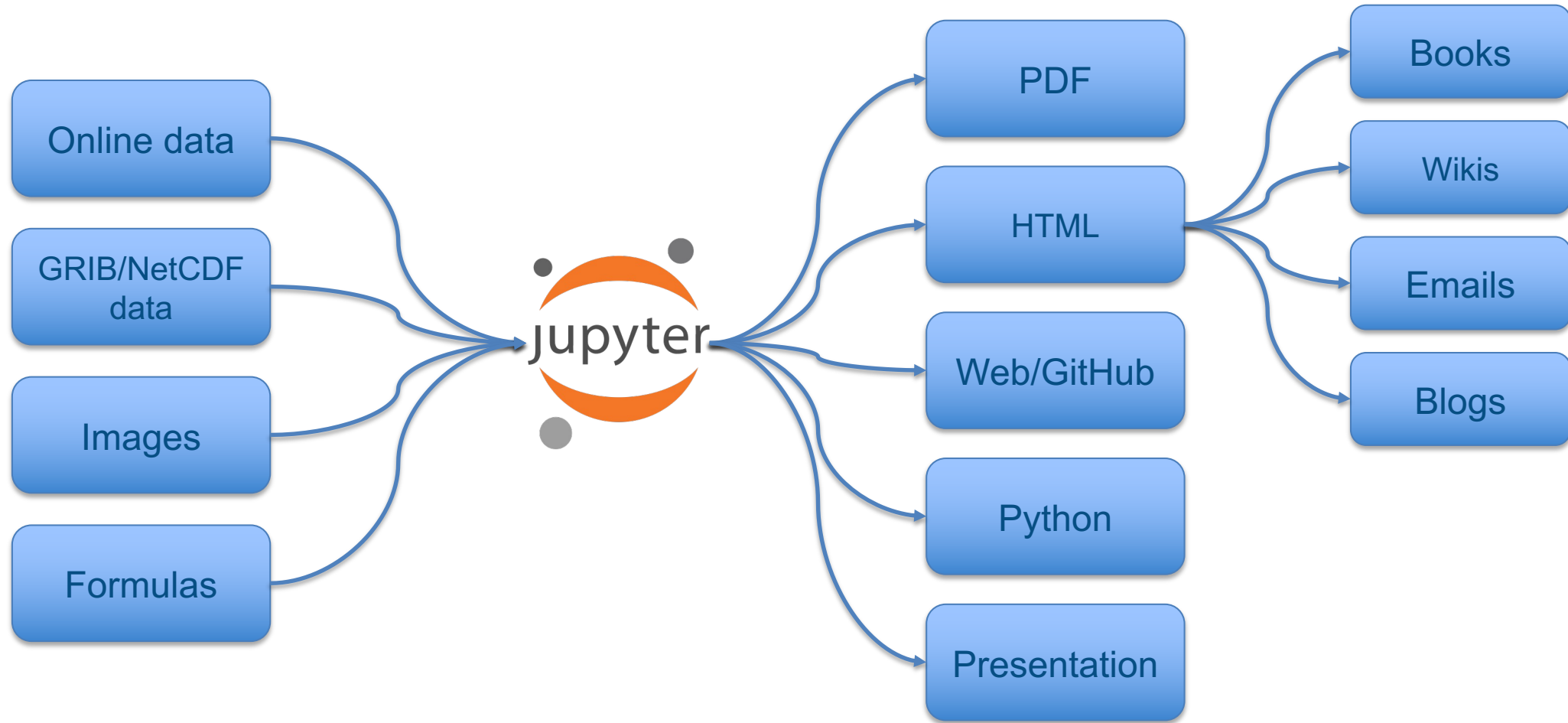
```
In [5]: last_y = ds.sel(time=slice('2022-05-01T12:00:00.000000000', '2022-09-30T12:00:00.000000000'))
last_y
```

```
Out[5]: <xarray.Dataset>
Dimensions:    (latitude: 214, longitude: 302, time: 153)
```

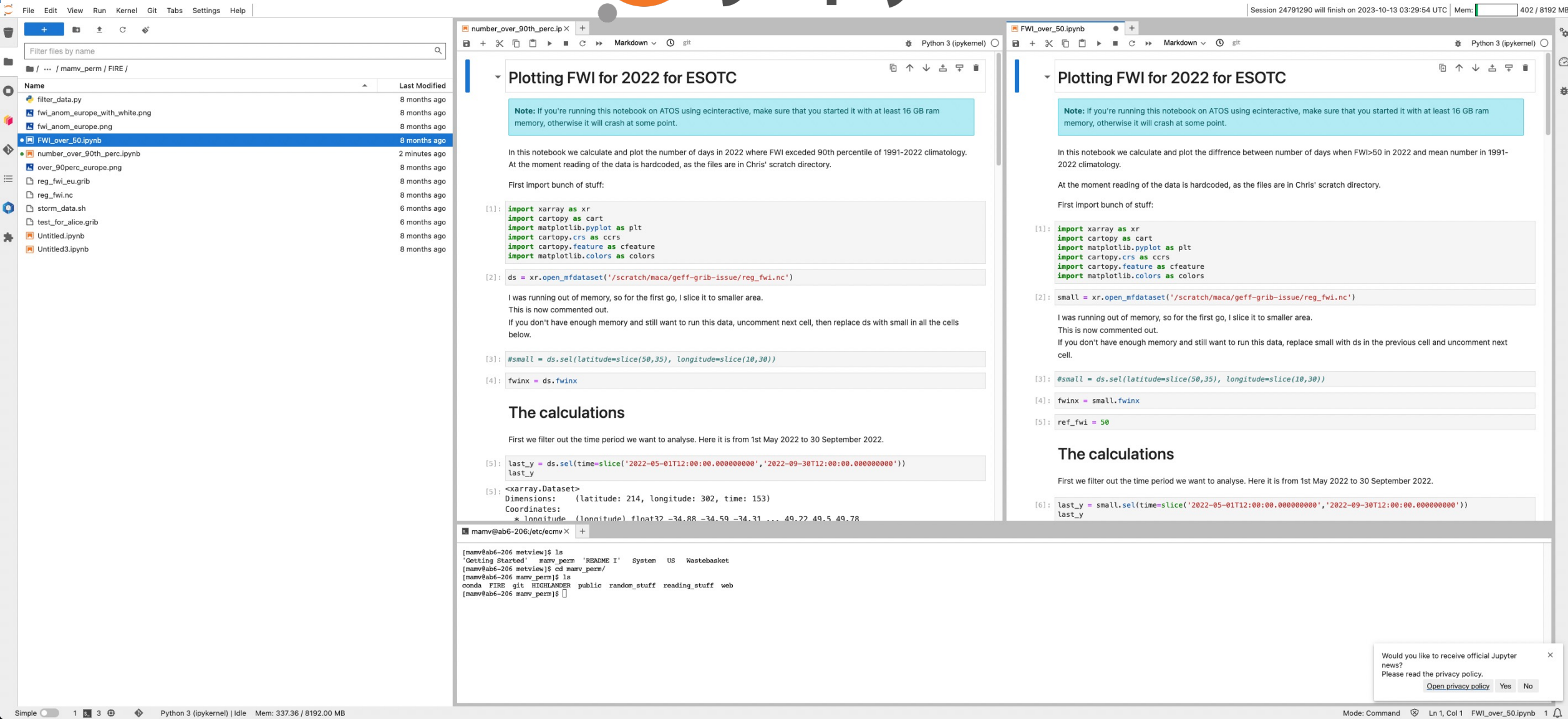
Markdown cells

Code cells

TERMINOLOGY



Documentation: <https://jupyter-notebook.readthedocs.io/>



The screenshot displays the JupyterLab interface with two notebooks open: 'number_over_90th_perc.ipynb' and 'FWL_over_50.ipynb'. Both notebooks contain the same text and code blocks, but the 'FWL_over_50.ipynb' notebook includes an additional code cell (cell 5) that sets a reference value for FWI.

number_over_90th_perc.ipynb

Plotting FWI for 2022 for ESOTC

Note: If you're running this notebook on ATOS using ecinteractive, make sure that you started it with at least 16 GB ram memory, otherwise it will crash at some point.

In this notebook we calculate and plot the number of days in 2022 where FWI exceeded 90th percentile of 1991-2022 climatology. At the moment reading of the data is hardcoded, as the files are in Chris' scratch directory.

First import bunch of stuff:

```
[1]: import xarray as xr
import cartopy as cart
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import matplotlib.colors as colors
```

```
[2]: ds = xr.open_mfdataset('/scratch/maca/geff-grib-issue/reg_fwi.nc')
```

I was running out of memory, so for the first go, I slice it to smaller area.
This is now commented out.
If you don't have enough memory and still want to run this data, uncomment next cell, then replace ds with small in all the cells below.

```
[3]: #small = ds.sel(latitude=slice(50,35), longitude=slice(10,30))
```

```
[4]: fwinx = ds.fwinx
```

The calculations

First we filter out the time period we want to analyse. Here it is from 1st May 2022 to 30 September 2022.

```
[5]: last_y = ds.sel(time=slice('2022-05-01T12:00:00.000000000', '2022-09-30T12:00:00.000000000'))
last_y
```

```
[5]: <xarray.Dataset>
Dimensions: (latitude: 214, longitude: 302, time: 153)
Coordinates:
  * longitude (longitude) float32 -34.88 -34.59 -34.31 ... 49.22 49.5 49.78
```

FWL_over_50.ipynb

Plotting FWI for 2022 for ESOTC

Note: If you're running this notebook on ATOS using ecinteractive, make sure that you started it with at least 16 GB ram memory, otherwise it will crash at some point.

In this notebook we calculate and plot the difference between number of days when FWI>50 in 2022 and mean number in 1991-2022 climatology.

At the moment reading of the data is hardcoded, as the files are in Chris' scratch directory.

First import bunch of stuff:

```
[1]: import xarray as xr
import cartopy as cart
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import matplotlib.colors as colors
```

```
[2]: small = xr.open_mfdataset('/scratch/maca/geff-grib-issue/reg_fwi.nc')
```

I was running out of memory, so for the first go, I slice it to smaller area.
This is now commented out.
If you don't have enough memory and still want to run this data, replace small with ds in the previous cell and uncomment next cell.

```
[3]: #small = ds.sel(latitude=slice(50,35), longitude=slice(10,30))
```

```
[4]: fwinx = small.fwinx
```

```
[5]: ref_fwi = 50
```

The calculations

First we filter out the time period we want to analyse. Here it is from 1st May 2022 to 30 September 2022.

```
[6]: last_y = small.sel(time=slice('2022-05-01T12:00:00.000000000', '2022-09-30T12:00:00.000000000'))
last_y
```

Terminal

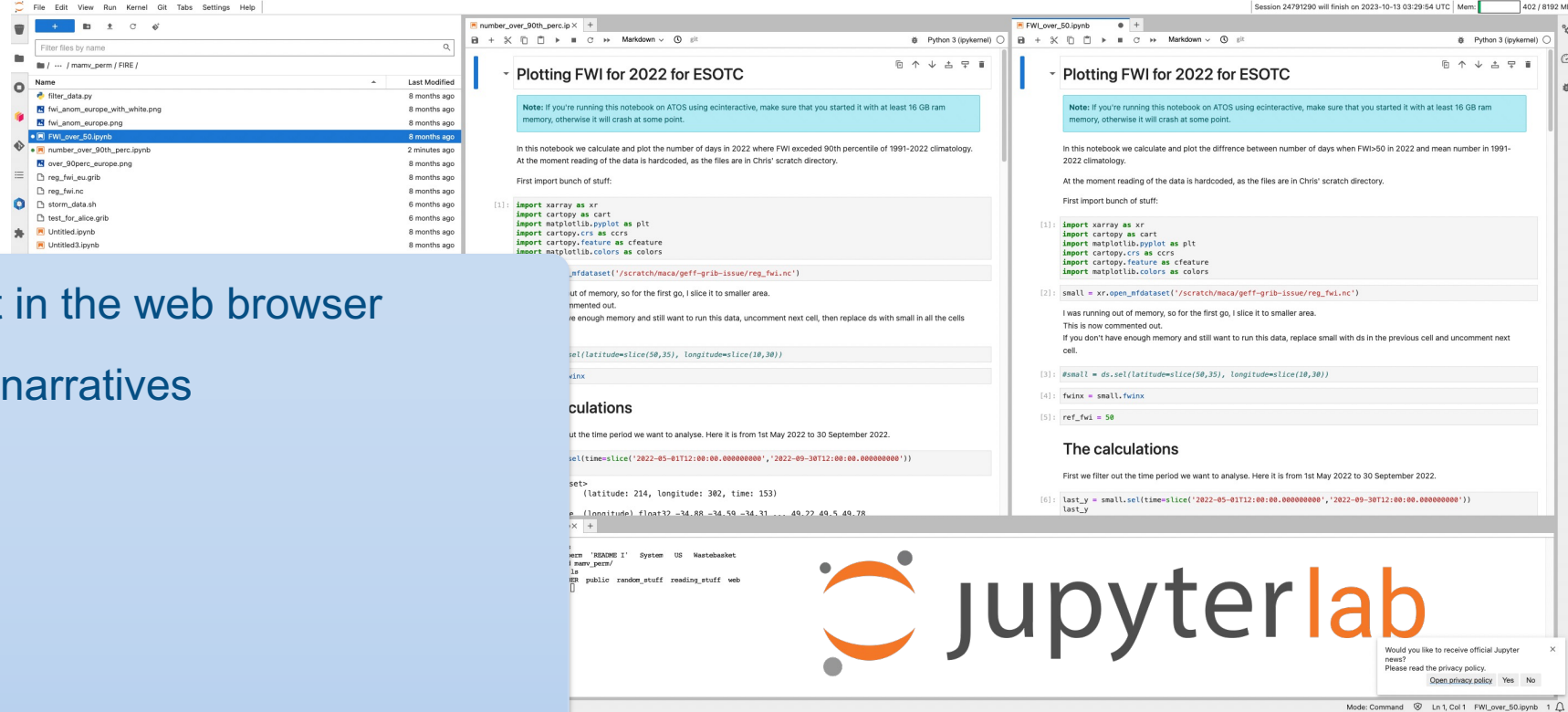
```
mamv@ab6-206/etc/ecmv X +
[mamv@ab6-206 metview]$ ls
'Getting Started' mamv_perm 'README I' System US Wastebasket
[mamv@ab6-206 metview]$ cd mamv_perm/
[mamv@ab6-206 mamv_perm]$ ls
conda FIRE git HIGHLANDER public random_stuff reading_stuff web
[mamv@ab6-206 mamv_perm]$ []
```

TERMINOLOGY

- **Interactive** working environment in the web browser
- For the creation of reproducible narratives
- Jupyter = Julia + Python + R

Multi-purpose working environment

- Language agnostic
- Supports execution environments (“kernels”)
 - For dozens of languages: Python, R, Julia, C++, ...
- Extensible software design („extensions”)
 - Many server/client plug-ins available
 - Eg. in-browser-terminal, git, file-browsing, modules



TERMINOLOGY

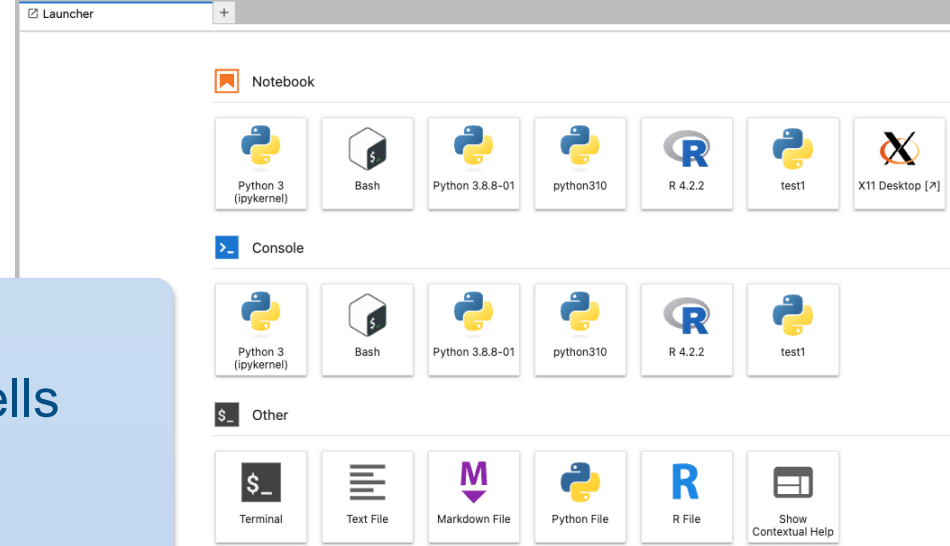
What is a Jupyter Kernel?

Jupyter Kernel

A “kernel” refers to the separate process which executes code cells within a Jupyter notebook.

Jupyter Kernel

- **run code** in different programming languages and environments
- can **be connected** to a notebook (one at a time)
- communicates via ZeroMQ with the JupyterLab
- Multiple **preinstalled** Jupyter Kernels can be found on our clusters (Python, R, Bash)
- You can **create your own kernel** which for example runs your specialized virtual Python environment.



TERMINOLOGY

What is a JupyterLab Extension?

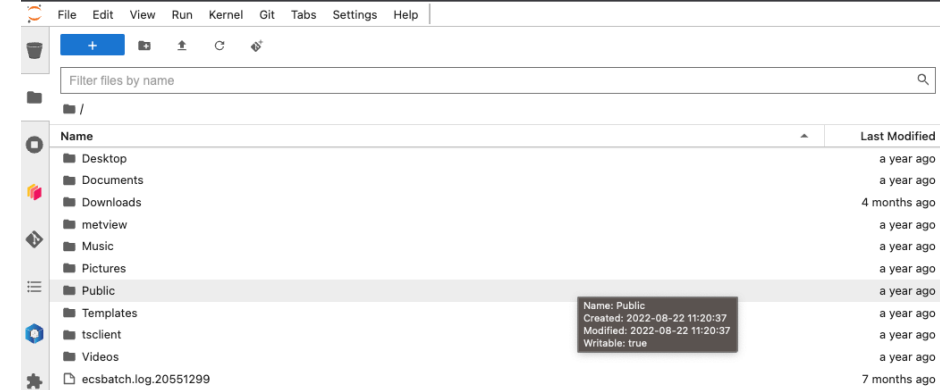
JupyterLab Extension

JupyterLab extensions can customize or enhance any part of JupyterLab.

JupyterLab Extensions

- provide new file viewers, editors, themes
- provide renderers for rich outputs in notebooks
- add items to the menu or command palette
- add keyboard shortcuts
- add settings in the settings system
- Extensions can even provide an API for other extensions to use and can depend on other extensions.

The whole JupyterLab itself is simply a collection of extensions that are no more powerful or privileged than any custom extension.



TERMINOLOGY

JupyterLab Extensions at ECMWF

The screenshot shows the JupyterLab file browser interface. At the top, there is a menu bar with options: File, Edit, View, Run, Kernel, Git, Tabs, Settings, and Help. Below the menu bar is a toolbar with icons for home, new (+), refresh, and delete. A search bar labeled "Filter files by name" is present. The main area displays a file browser view with a sidebar on the left showing a tree structure of folders: Desktop, Documents, Downloads, metview, Music, Pictures, Public (highlighted), Templates, tsclient, and Videos. The main pane shows a table of files and folders with columns for Name and Last Modified. A tooltip is visible over the 'Public' folder, displaying its metadata.

Name	Last Modified
Desktop	a year ago
Documents	a year ago
Downloads	4 months ago
metview	a year ago
Music	a year ago
Pictures	a year ago
Public	a year ago
Templates	a year ago
tsclient	a year ago
Videos	a year ago
ecsbatch.log.20551299	7 months ago

Tooltip for 'Public' folder:

- Name: Public
- Created: 2022-08-22 11:20:37
- Modified: 2022-08-22 11:20:37
- Writable: true

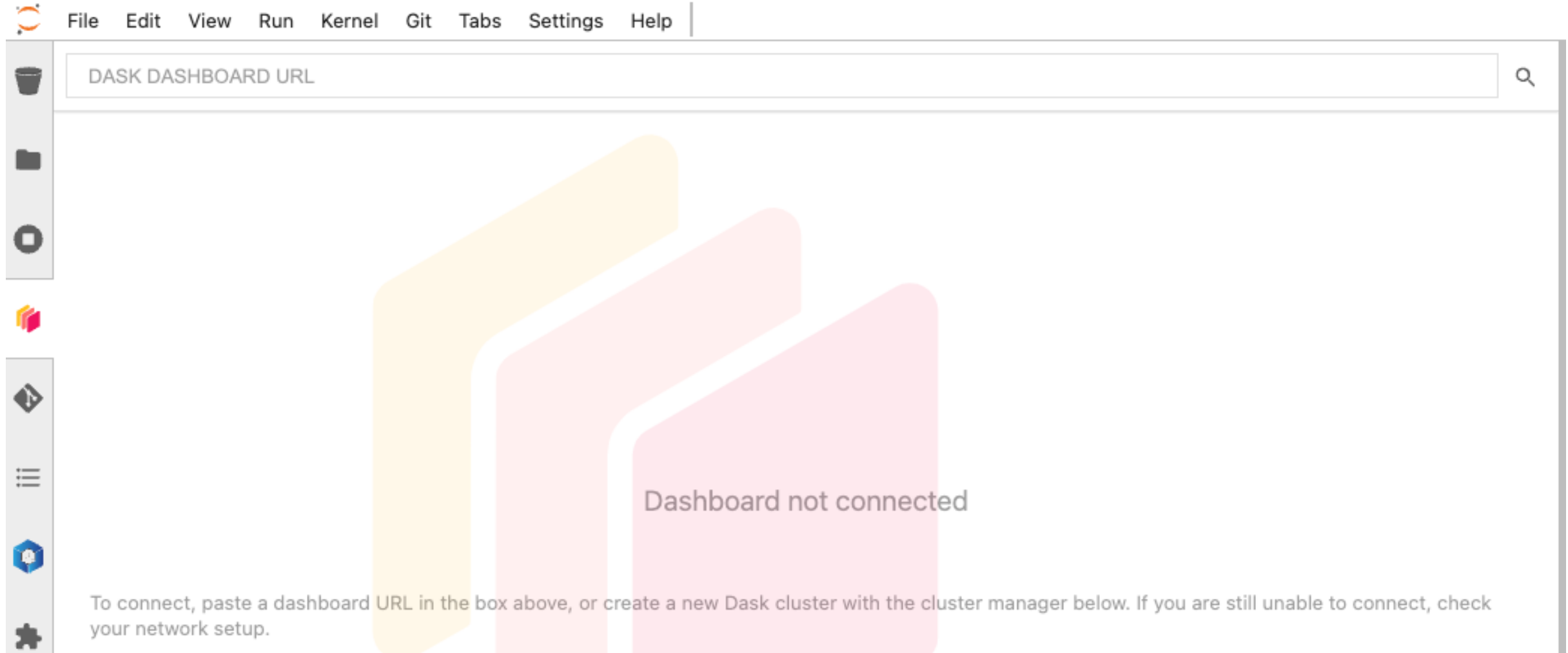
TERMINOLOGY

JupyterLab Extensions at ECMWF

The screenshot displays the JupyterLab interface. At the top, a horizontal menu bar contains the following items: File, Edit, View, Run, Kernel, Git, Tabs, Settings, and Help. Below the menu bar is a vertical sidebar on the left side of the interface. The sidebar is divided into three main sections: **OPEN TABS**, **KERNELS**, and **TERMINALS**. The **OPEN TABS** section shows a trash can icon at the top left and a refresh icon at the top right. It lists two items: 'Launcher' with a folder icon and 'FWI_over_50.ipynb' with a notebook icon. The **KERNELS** section shows a square icon with a play symbol at the top left and lists one item: 'FWI_over_50.ipynb' with a notebook icon. The **TERMINALS** section shows a terminal icon at the top left. At the bottom of the sidebar, there are several other icons: a diamond with a question mark, a hamburger menu, a gear, and a puzzle piece. On the right side of the sidebar, there are three orange buttons: 'Close All' next to the Open Tabs section, 'Shut Down All' next to the Kernels section, and 'Shut Down All' next to the Terminals section.

TERMINOLOGY

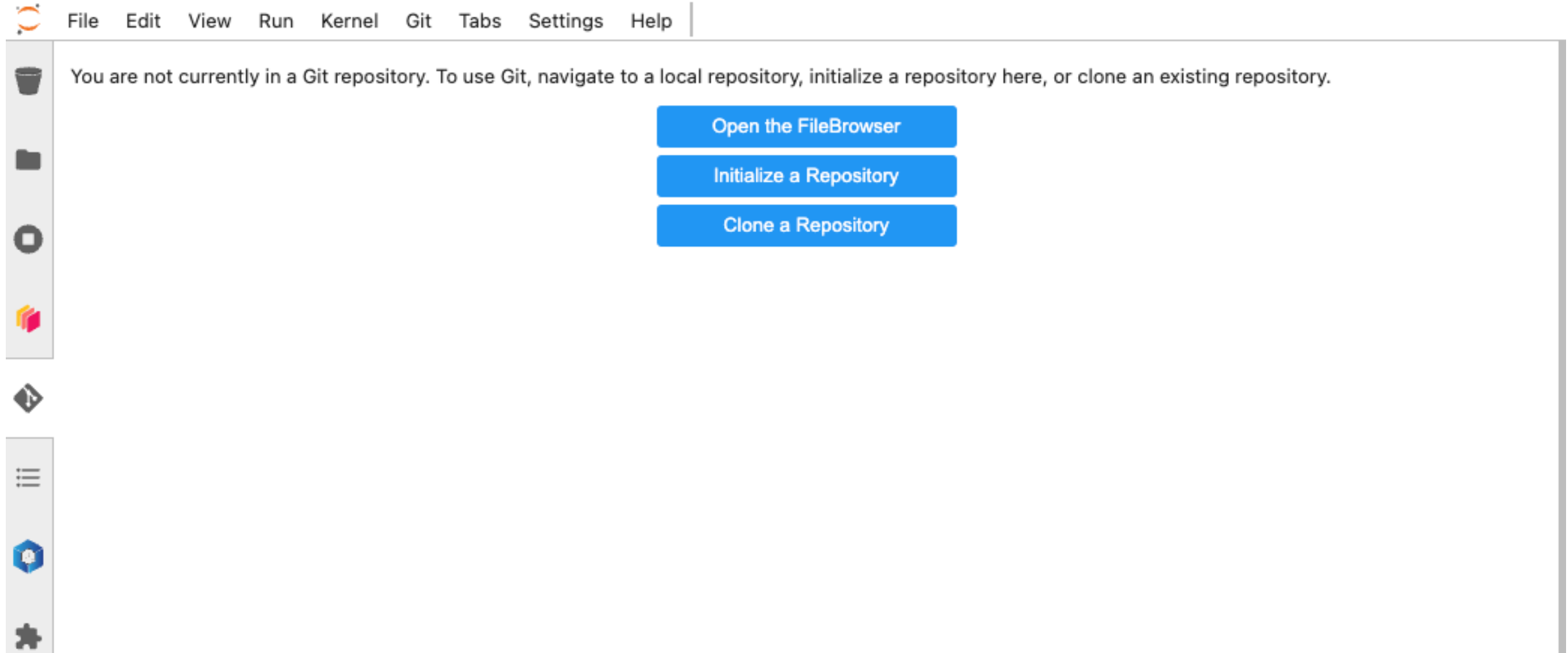
JupyterLab Extensions at ECMWF



The screenshot shows the JupyterLab interface. At the top, there is a menu bar with options: File, Edit, View, Run, Kernel, Git, Tabs, Settings, and Help. Below the menu bar is a search bar containing the text "DASK DASHBOARD URL". On the left side, there is a vertical toolbar with icons for trash, home, terminal, and other functions. The main content area displays a message: "Dashboard not connected". Below this message, there is a paragraph of text: "To connect, paste a dashboard URL in the box above, or create a new Dask cluster with the cluster manager below. If you are still unable to connect, check your network setup."

TERMINOLOGY

JupyterLab Extensions at ECMWF



TERMINOLOGY

JupyterLab Extensions at ECMWF

The screenshot displays the JupyterLab interface. At the top, there is a menu bar with the following items: File, Edit, View, Run, Kernel, Git, Tabs, Settings, and Help. Below the menu bar, the notebook title 'FWI_OVER_50.IPYNB' is visible. The notebook content area shows a code cell with the following text:

```
<> M ≡
```

Plotting FWI for 2022 for ESOTC

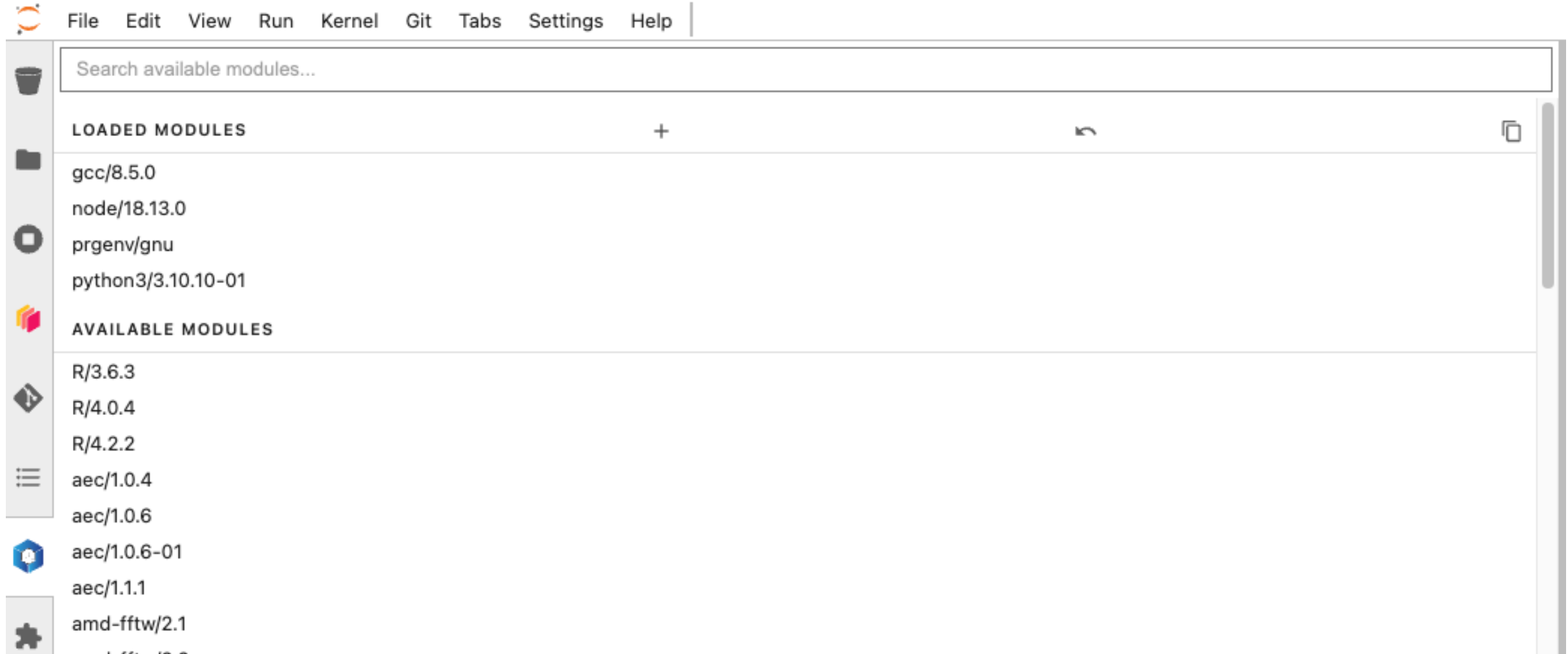
The calculations

Plotting

The left sidebar contains several icons: a trash can, a folder, a square, a cube, a diamond, a list, a gear, and a puzzle piece.

TERMINOLOGY

JupyterLab Extensions at ECMWF



The screenshot displays the JupyterLab environment manager sidebar. At the top, there is a search bar labeled "Search available modules...". Below this, the "LOADED MODULES" section is expanded, showing a list of currently active modules: gcc/8.5.0, node/18.13.0, prgenv/gnu, and python3/3.10.10-01. The "AVAILABLE MODULES" section is collapsed. The sidebar includes various icons for trash, folders, environment, and a search icon.

File Edit View Run Kernel Git Tabs Settings Help

Search available modules...

LOADED MODULES + 5

- gcc/8.5.0
- node/18.13.0
- prgenv/gnu
- python3/3.10.10-01

AVAILABLE MODULES

- R/3.6.3
- R/4.0.4
- R/4.2.2
- aec/1.0.4
- aec/1.0.6
- aec/1.0.6-01
- aec/1.1.1
- amd-fftw/2.1

TERMINOLOGY

JupyterLab Extensions at ECMWF

The screenshot displays the JupyterLab interface with a menu bar at the top containing 'File', 'Edit', 'View', 'Run', 'Kernel', 'Git', 'Tabs', 'Settings', and 'Help'. Below the menu is a search bar with the text 'SEARCH'. The main area shows a list of installed extensions under the heading 'INSTALLED'. The extensions listed are:

- @jupyter-widgets/jupyterlab-manager**: The JupyterLab extension providing Jupyter widgets. Includes an 'About' link.
- @jupyterlab/git**: A JupyterLab extension for version control using git. Includes an 'About' link.
- @jupyterlab/server-proxy**: Jupyter server extension to supervise and proxy web services. Includes an 'About' link.
- jupyterlab_pygments**: Pygments theme using JupyterLab CSS variables.

TERMINOLOGY

JupyterLab Extensions at ECMWF

File Edit View Run Kernel Git Tabs Settings Help



S3 Object Storage Browser

This extension allows you to browse S3-compatible object storage instances, such as AWS S3 and IBM Cloud Object Storage.

Endpoint URL

Access Key ID

Secret Access Key

(Optional) Session Token

Connect

Useful links

- ECMWF ecinteractive documentation:
<https://confluence.ecmwf.int/display/UDOC/HPC2020%3A+Persistent+interactive+job+with+ecinteractive>
- eccodes grib notebooks:
https://git.ecmwf.int/projects/USS/repos/eccodes_notebooks/browse
notebook to start: **eccodes_api_grib_ex1.ipynb**

Jupyter Documentation:

- Jupyter notebook: <https://jupyter-notebook.readthedocs.io/>
- JupyterLab interface: <https://jupyterlab.readthedocs.io/en/stable/user/interface.html>
- Files and outputs in JupyterLab: https://jupyterlab.readthedocs.io/en/stable/user/file_formats.html#