

Common practical numerics: Data analysis essentials with Python

January 20th 2023 - Rémy Lapere



Message of the utmost importance

Tomorrow we are going on a **snowshoe hike**.

It is **vital** (literally) that you bring:

- **Warm clothes, gloves and a cap**
- **Waterproof shoes/boots** (if possible, sturdy ankle boots)
- **Sunglasses** (it should be sunny)
- A backpack
- A bottle of water and your own picnic



What we will do this afternoon

- Python, python & python, using **Jupyter Notebook**
- Use *pangeo*, *xarray*, *pandas*, *cartopy*... all the cool stuff!
- Work with climate data from **CMIP6** and **Copernicus**

Objective: Provide you with an essential toolkit for starting data analysis with Python

Jupyter notebook



“web-based interactive development environment for notebooks, code, and data”

Main advantages

- You can write both text and code
- Data remains in cache
- Works with any programming language
- ...

The screenshot shows a Jupyter notebook titled "ERCA_intro_python" with a "Last Checkpoint: an hour ago (autosaved)" status. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The notebook content is divided into sections:

- SECTION 1: load climate data from Pangeo**
 - Pangeo** is a community platform for Big Data geoscience (<https://pangeo.io/>) which, among other things, provides access to data catalogs such as **CMIP6** output. Using the associated python libraries is a powerful tool for the analysis of climate data without ever having to download and store them.
 - The workflow I recommend for this use is:
 1. Assess the existence of the **CMIP6** data you are interested in through the **ESGF** platform (<https://esgf-node.lpsl.upmc.fr/search/cmip6-ipsi/>). I recommend using the **CMIP6** data catalog (<https://clips-services.ceda.ac.uk/dreq/mipVars.html>) for better understanding of the variable names and contents
 2. Once you have identified the data sets you want on **ESGF**, apply the associated filters in the scripts below when searching the **Pangeo** catalog
 - Connect to the pangeo storage catalog using intake**

```
In [3]: cat_url = "https://storage.googleapis.com/cmip6/pangeo-cmip6.json"
col = intake.open_esm_datastore(cat_url)
```
 - Search through the catalog**

In this example, we want to look at monthly ('Amon') surface temperature ('tas') in one CMIP6 model ('NorESM2-MM'), for the 'historical' experiment and the variant 'r1i1p1f1'. We do a search through the pangeo catalog using the appropriate keys.

```
In [4]: cat = col.search(variable_id=['tas'],
                        member_id=['r1i1p1f1'],
                        source_id=['NorESM2-MM'],
                        experiment_id=['historical'],
                        table_id=['Amon'])
```
 - NB:** you can do a search for multiple values in each field, e.g. you can try loading precipitation as well by replacing, ['tas'] with ['tas','pr']. This applies for any field in the search

A cloud solution to access geoscience data

Major interest: you can analyze climate data (e.g. with Python)
without ever having to download them

Among other things, **Copernicus** provides free access to products such as the famous **ERA5 reanalysis** through their **Climate Data Store**
(<https://cds.climate.copernicus.eu/cdsapp#!/home>)

Today we will discuss where/how you can get this data.

pandas, xarray



pandas: manipulate 2D data tables

xarray: manipulate >2D data in a pandas fashion

Structure of the tutorial

Section 1: load climate data from Pangeo

Section 2: select, resample

Section 3: plot nice maps

Section 4: interpolate and compare to observations

Section 5: Copernicus data

Section 6: pandas essentials

if you are very new to Python and/or work mostly with
1D data, better focus on this section

Section 7: interactive plots with hvplot

Bonus section: turn your data into music

Let's start

The tutorial is located at

https://github.com/rlapere/ERCA2023_tutorial_v2

You have several options to run this tutorial

1. Run online using Binder
2. Run locally on your computer
3. Run from the UGA Jupyterhub

1. Run online using Binder

Go to https://github.com/rlapere/ERCA2023_tutorial_v2

The screenshot displays the GitHub interface for the repository `rlapere/ERCA2023_tutorial_v2`. The repository is public and has 1 branch and 0 tags. The main branch is `main`. The repository structure shows the following files and their last update times:

File	Update	Time
<code>data</code>	Add files via upload	48 minutes ago
<code>ERCA_intro_python.ipynb</code>	Add files via upload	48 minutes ago
<code>README.md</code>	Update README.md	28 minutes ago
<code>environment.yml</code>	Add files via upload	48 minutes ago

The README.md content is as follows:

ERCA2023 tutorial

Welcome to this tutorial on environmental data analysis with Python. This notebook was created by Rémy Lapere for the 2023 European Research Course on Atmospheres.

A `launch binder` button is highlighted with a pink circle.

2. Run locally

Open a terminal

`cd` to the location where you want to do the tutorial

```
git clone https://github.com/rlapere/ERCA2023_tutorial_v2.git
```

```
cd ERCA2023_tutorial_v2
```

```
conda env create -f environment.yml
```

 ← This can take a while

```
conda activate erca
```

```
conda install -c anaconda ipykernel
```

```
python -m ipykernel install --user --name=erca
```

```
jupyter notebook ERCA_intro_python.ipynb
```

3. Run from the UGA Jupyterhub

Connect to: <https://jupyterhub.u-ga.fr/> (username & password same as your UGA account)

```
cd notebooks
```

```
git clone https://github.com/rlapere/ERCA2023_tutorial_v2.git
```

```
cd ERCA2023_tutorial_v2
```

```
conda env create -f environment.yml
```

 This can take a while

```
conda activate erca
```

```
conda install -c anaconda ipykernel
```

```
python -m ipykernel install --user --name=erca
```

Go back to the notebook tab, open the file `ERCA_intro_python.ipynb` in the `ERCA2023_tutorial` folder and you are ready to work!

In both cases

After opening the notebook and before launching it:

Kernel > Change kernel > erca