1 Suggested software for working with geoscientific data in python

1.1 Objectives

This document briefly describes what software and packages to setup for performing analysis steps and working with geoscientific data. You are likely going to use your own laptop and install the software locally. There are no heavy computing demands on the examples and exercises - every laptop should do it (I also ran these on my Raspberry Pi 3+ and it works just fine.

If you don't want to or can not install the software on your laptop, all steps can also be done through google colab (but you will be required to have a gmail account). You can view Python codes and Jupyter Labs without a google account, but you will not be able to run any of the exercises see here.

You can use Windows and most of the software will also be easily installed on a Mac. The easiest OS for handling data and python programming may be a Linux-based system such as Ubuntu or Debian, because they are the most flexible and allow to easily compile additional code. _

2 Software

2.1 Python Packages

We will rely on the following *python* packages and environments:

- Python 3.x (but we will use 3.12 in the examples - any version >3.9 should do)
- Jupyter Lab
- scipy
- numpy
- pandas
- matplotlib
- cartopy
- plotly
- scikit-learn
- scikit-image
- statsmodels
- seaborn
- GDAL
- xarray
- numba

2.1.1 *Windows Users*: Install command line tools and Python packages

One option is to install this via Anaconda and select the packages *gdal, numpy, pandas, scipy, cartopy, plotly, scikit-image, scikit-learn, statsmodels, jupyter-lab, seaborne, and matplotlib*.

If you know your terminal and shell environment, you likely would use the following options: You can also install the required packages via the `anaconda shell`. Depending on your installation, you may need to add the channel *conda-forge* to the search environment:

```
conda config --prepend channels conda-forge
```

I suggest to create a separate conda environment dedicated for the analysis of geoscience data (e.g. `Py3_geodata`):

```
conda config --prepend channels conda-forge
conda create -c conda-forge -y -n Py3_geodata python=3.12 pip scipy pandas ^
  numpy matplotlib scikit-image gdal ipython ^
  statsmodels jupyter jupyterlab pyproj ^
  h5py seaborn pytables seaborn ^
  scikit-learn jupyter-resource-usage numba xarray
conda activate Py3_geodata
```

You can additionally install `spyder` if you use it - otherwise it will take up too much space:

```
conda activate Py3_geodata
conda install spyder
```

If you have a GPU, you can also install `cupy` (but only if you know what you are doing)

```
conda activate Py3_geodata
conda install cupy
```

Make sure to add your conda environment to the Jupyter Notebook environment:

```
conda activate Py3_geodata
python -m ipykernel install --user --name=Py3_geodata
```

2.1.1.1 Alternative option Windows Users

Install Linux Subsystem on Windows and use miniconda (see next section). Installing the Linux subsystem (use Ubuntu 22.04 or 24.04) is generally a useful thing to do for Windows users (if your hardware space allows it), but is not required for this workshop.

2.1.2 *Ubuntu and Mac Users*: Install command line tools and Python packages

You can install Anaconda for Mac, but you may prefer the command line approach described below. Install miniconda3 and the packages via `conda install`. Download and install the required software via the command line/shell:

```
cd ~
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
sh ./Miniconda3-latest-Linux-x86_64.sh
```

You may have to include additional channels for installation:

```
conda config --prepend channels conda-forge
```

Install the conda packages (will take some time):

```
conda config --prepend channels conda-forge
conda create -c conda-forge -y -n Py3_geodata python=3.12 pip scipy pandas numpy  \
    matplotlib scikit-image gdal ipython statsmodels jupyter jupyterlab pyproj h5py  \
    seaborn pytables seaborn scikit-learn jupyter-resource-usage numba xarray
conda activate Py3_geodata
```

Make sure to add your conda environment to the Jupyter Notebook environment:

```
conda activate Py3_geodata
python -m ipykernel install --user --name=Py3_geodata
```

If you have a CUDA GPU, you may need to install the libraries to take full advantage of GPU processing:

```
conda activate Py3_geodata
conda install cuda-cudart cuda-version=12
```

3 Additional considerations

3.1 Editor

During this workshop we will only rely on Jupyter Lab (or the Google Colab alternative).

If you plan to do some coding, it may be useful to use an editor that is streamlined for your python experience. There exist many software packages that help you to do that and there are very strong opinions on what is the best available package. I am just suggesting that you may want to look into Spyder or VS Code. Spyder is included in the Windows Anaconda distribution and is installed via the command line above. If you intend to continue with data analysis methods and plan to deepen you python knowledge, you should look into these options. The serious user will use vim or neovim and lazyvim - there is no better editor, especially if you worn on remote servers and clusters.