NIWA

Taihoro Nukurangi

Climate, Freshwater & Ocean Science

Accessing and analysing climate data with the Python Scientific ecosystem

Dr. Nicolas Fauchereau

NIWA Hamilton

Climate, Freshwater & Ocean Science

NIWA
Taihoro Nukurangi

Who am I ?

- Nicolas Fauchereau
- Climate Scientist, NIWA (National Institute for Water and Atmospheric research Ltd.), Hamilton office
- Research interests: Scale interactions in the climate system, predictability, sub-seasonal to seasonal forecasting, Machine Learning, Complex Networks …
- Python user since the early 2000s (!)
- Ran several workshops on Python for climate data analysis and visualisation
- Contact: Nicolas.Fauchereau@niwa.co.nz
- Blog (infrequent posts): https://nicolasfauchereau.github.io/

Workshop structure

- A brief overview of **climate data sources**, useful resources, etc
- A short introduction to **Python** and the **Python scientific ecosystem**
- Creating / managing **Python environments**: conda / mamba
- Jupyter notebook / Jupyterlab [demo]
- (very) brief overview of **Pandas** and **xarray** [demo]
- Accessing and analysing **reanalysis data** (past climates, climate monitoring) [demo]
- Accessing and analysing **CMIP6 data** (climate change simulations) [demo]

Climate Data Sources

Main types of climate data

- In-situ **observations** / measurements
- Climate proxies (paleoclimates)
- Satellite remote sensing
- Weather, sub-seasonal, seasonal, decadal climate forecasts
- **Reanalyses**
- **Climate models simulations**

Climate Data Sources

In situ-observations

- Weather / climate stations (standard measurements and environment: World Meteorological Organisation [WMO])
- Low cost sensors
- Buoys (fixed, or drifting weather buoys, ARGO array)
- Ships of opportunity
- Aircrafts
- Radio-sondes
- Various formats, text, csv, tab-delimited, etc

Some resources

- Berkeley Earth Temperature dataset: http://berkeleyearth.lbl.gov/station-list/
- Global Historical Climatology Network (monthly): https://www.ncei.noaa.gov/products/land-based-station/global-historical-climatology-network-monthly
- GHCN (daily): https://www.ncei.noaa.gov/products/land-based-station/global-historical-climatology-network-daily
- Aotearoa New Zealand daily temperature and precipitation:
    - https://data.mfe.govt.nz/table/105056-daily-temperature-1909-2019/
    - https://data.mfe.govt.nz/table/105055-rainfall-1960-2019/

Climate Data Sources

Atmospheric reanalyses

- Data **assimilation** projects
- Uses a numerical weather model to ingest past, historical (and realtime) observations to produce physically consistent fields of multiple variables (multiple levels in the atmosphere, conservation of (some) quantities)
- **Gridded** (latitude, longitude, levels or surface) datasets (spatialisation, regionalisation, clipping using shapefiles, etc)
- Outputs generally available in **NetCDF** (Network Common Data Form: https://en.wikipedia.org/wiki/NetCDF)

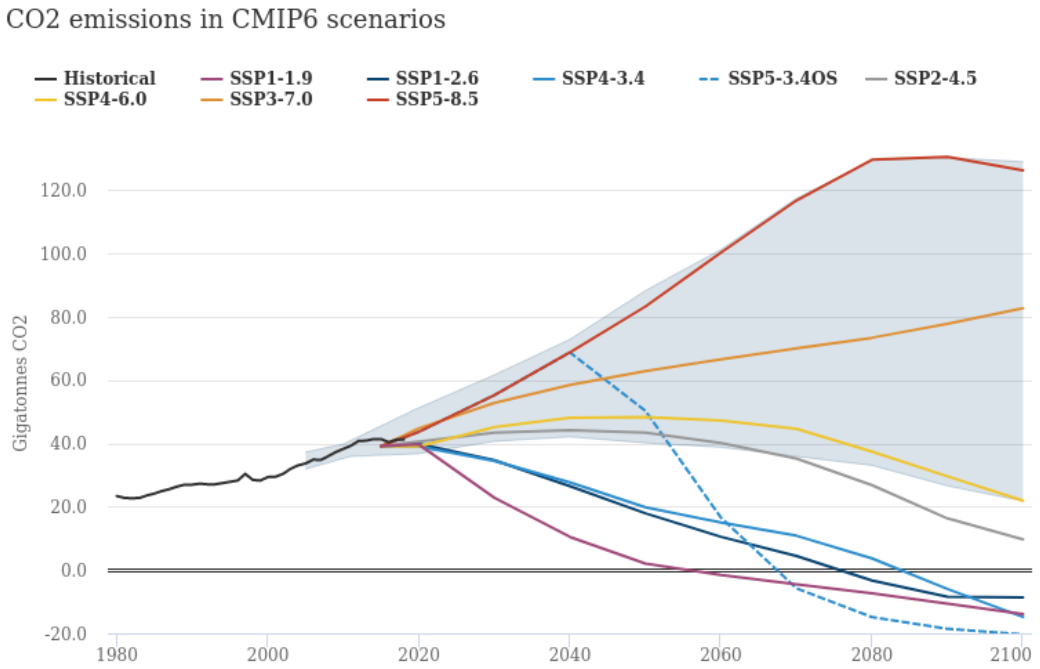
Some resources

- https://climatedataguide.ucar.edu/climate-data/atmospheric-reanalysis-overview-comparison-tables
- https://climatereanalyzer.org/
- **NCEP / NCAR (NCEP1, 1950 – now)**: https://psl.noaa.gov/data/reanalysis/reanalysis.shtml
- NCEP / DOE II (NCEP2, 1979 – now): https://psl.noaa.gov/data/gridded/data.ncep.reanalysis2.html
- **ERA5 (1979 – now, extended to 1950):** https://cds.climate.copernicus.eu/

Climate Data Sources

CMIP6 simulations

- Future climate projections (inform the IPCC WG1 reports)
- Different emissions scenarios (SSPs)
- 49 modelling groups, ~ 100 distinct climate models

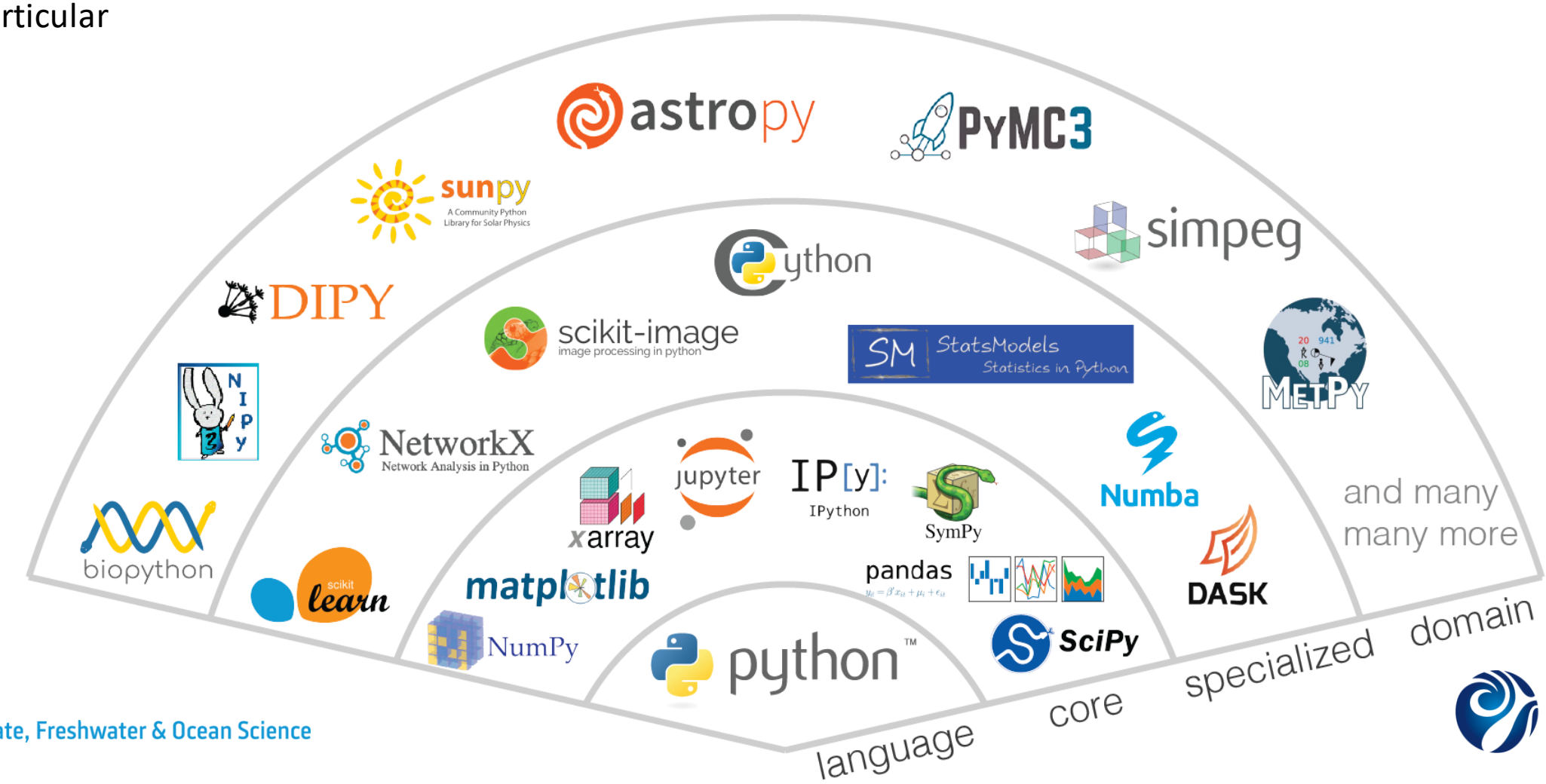


Some resources

- https://www.carbonbrief.org/cmip6-the-next-generation-of-climate-models-explained/
- https://www.carbonbrief.org/qa-how-do-climate-models-work/#cmip
- https://esgf-node.llnl.gov/projects/cmip6/ (Earth System Grid Federation)
- https://cloud.google.com/blog/products/data-analytics/new-climate-model-data-now-google-public-datasets (CMIP6 data on the Google Cloud)
- http://gallery.pangeo.io/repos/pangeo-gallery/cmip6/ PANGEO CMIP6 examples notebooks

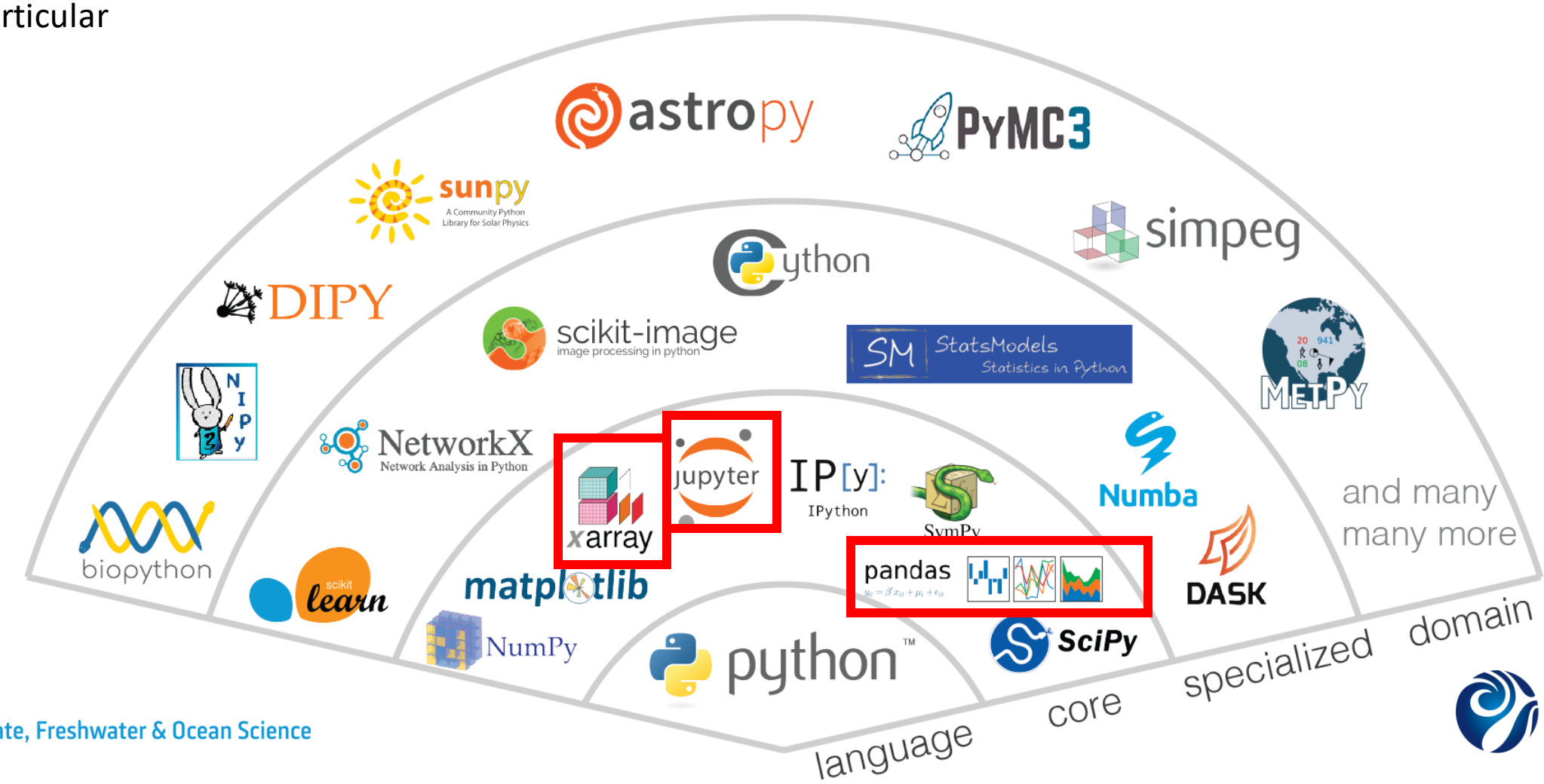
The Python scientific ecosystem

- Python is a **general purpose** scripting (as opposed to compiled) programming language
- Designed to be simple, easy to learn yet powerful
- Has been increasingly adopted in scientific computing communities in general, and ocean / atmosphere science in particular



The Python scientific ecosystem

- Python is a **general purpose** scripting (as opposed to compiled) programming language
- Designed to be simple, easy to learn yet powerful
- Has been increasingly adopted in scientific computing communities in general, and ocean / atmosphere science in particular



The Python scientific ecosystem

- **Pandas: https://pandas.pydata.org/**
  - Tabular (2D) data (e.g. csv, excel, etc)
  - labels along axes (columns names, row indexes)
  - loads of I/O functions (read from and write to variety of formats)
  - data structures: Series and DataFrames (collections of Series)
  - time-series (resampling, rolling windows operations, etc)
  - powerful groupby (split / apply / combine) operations
  - easy plotting functions
  - basic summary statistics functions (quantiles, etc)

- **xarray: https://xarray.pydata.org/**
  - Multi-dimensional labelled arrays (time / latitude / longitude / levels or depth, etc)
  - API very similar to pandas (function and method names, etc)
  - handles netcdf files easily, multiple files datasets, network protocols, zarr, etc
  - uses dask (https://dask.org/) to handles datasets too large to fit in memory
  - loads of specialized libraries built on top of xarray, see https://docs.xarray.dev/en/stable/ecosystem.html

See also: www.pangeo.io: community dedicated to build / organise tools for "big data" geoscience

Python environments: conda and mamba

- Creating a Python scientific development environment 'from scratch' can be challenging
- Multiple packages, non-python dependencies (C, C++, Fortran)
- What if you want to test the latest version of a package but have some 'operational' code that you don't want to modify ?
- What if you have conflicting dependencies ?

-> Python **'environments'**: separate, self contained environments with different Python executables and package set

-> different solutions, but in the scientific community: -> www.anaconda.com Python scientific **distribution**

https://docs.conda.io/en/latest/: **conda** environments and packages manager

Recently: https://mamba.readthedocs.io/en/latest/

See: https://github.com/conda-forge/miniforge for downloading https://github.com/conda-forge/miniforge#mambaforge

See https://github.com/nicolasfauchereau/climate_data_analytics/blob/main/environment.yml

```
$ conda env create -f environment.yml  or
$ mamba env create -f environment.yml
```