Python data analysis

3 core Python modules for climate data analysis

|  |  |
| --- | --- |
| Numpy | Allows you to create and manipulate sequences of numbers, strings, datetimes, i.e. objects. It is famous for handling multidimensional arrays (e.g. 2D or 3D matrices) efficiently.  Range of values:    You can easily select certain numbers based on a criteria:    Select numbers where criteria == True    Select data when two criteria are both True: |
| Pandas | Is a wrapper you can construct around a numpy array. Allows you to easily select certain dates for examples, of to create subgroups (e.g. select all Decembers). Pandas is restricted to 2D matrices (simple excel like column and rows). |
| Xarray | Xarray is an extension of pandas, that allows to go beyond 2 dimensions and is specifically developed for analyzing climate-like data. One can now not only select dates, but also select specific altitude levels, or latitudes, or longitudes.  How a Dataset looks like:    By typing T10hpa.to\_array(), we get the DataArray object:    Common manipulations are accessible once you have created an xr.DataArray() or xr.Dataset(). Type ‘*T10hpa.’* in your console and then press *tab* to see its functions.  Common functionality is for example:  ﻿*T2m.mean(dim='longitude')*  *T2m﻿.sel(latitude=70)*  *T2m﻿.sel(latitude=slice(90,10))*  On Xarray you can easily differentiate, i.e. calculate a gradient  *T2m.diff(‘latitude’)* Datetime With Datetime objects you can access information related to the .. date, e.g. access the year of the date.    You can get the dates from your netCDF into the datetime format by using pandas:  ﻿*all\_dates = pd.to\_datetime(T2m.time.values)*  You can also select based on criteria similar to numpy  If you have a mask for SSW dates, with False for non-SSW and True for SSWs:    Where the mask==False, the xarray all\_dates object return nans, by typing .dropna(), we are only kept with the SSW dates. |