**Goal: Drought indicator data based on meteorological data**

File will be stored as an HDF5-file: File type used to store large datasets and read/write fast using those.

A HDF5-file works a bit like a directory on your computer: It contains “groups”, which are equivalent to “sub directories”, and “datasets” which correspond to “files”.

The file itself will be called: **[COUNTRY]\_meteorological\_drought\_indicators.hdf5**

And contains groups/datasets as: **[Admin level]/[indicator]**.

Hence: **file[adm1/DMP]** has inside stored a DataFrame containing all the time-series information for both DMP and normalized DMP arranged by admin level 1. The first rows of the dataset correspond to the time series of the first admin boundary, the next set of rows to the second boundary etc.

The reason for storing every indicator in a separate table is to prevent any issues with different quantities being measured on different days. Also, this enables us to update each indicator one-by-one, which is allot easier to program (instead of thinking how to properly combine the rows in the final dataset). Finally, this reserves the option to only update 1 indicator in stead of updating all of them.

A HDF5-file has another slight advantage: It can contain “metadata”. This allows us to store information like the source of each dataset and the day the last update has been made.

During the update, the user will download the shapefile from Geonode server and the raster files from Copernicus service. Both can be deleted after the final dataset has been created.

Given the zonal statistics (using the shapefile on the raster file to get one average value per Polygon) most naturally loops over all the shapes using a single satellite image, the data we gather is actually sorted by date, not by admin boundary. Hence, an intermediate dataset is create prior to making the one containing the actual time series.

All temporary files will be stored inside a folder called “/temp” and the user has the option to delete after everything is done.

Finally, we will have to run the code/update once to get the entire historical data as this will require allot of computer time and storage space for all the satellite images.

The final user will just retrieve data taken later than the latest entry into the HDF5-dataset.