

Data Management: Data Standardization

Angelika Heil
Martin Schupfner
Fabian Wachsmann

*Data Management: Data Standardization by Heil, A;
Schupfner, M; Wachsmann, F. . Slides presented at the
DKRZ User Workshop, 13. October 2022, Hamburg.
This work is licensed under [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/).*

Agenda Data Standardization Workshop: October 13, 2022

11:45 – 12:05	Overview: CF conventions, NetCDF formats, FAIR data publication, CMORization, Quality Assurance	Angelika Martin
12:05 – 12:25	Exercise: Simple data standardization	Angelika
12:25 – 13:00	Exercise: CMORization	Martin
13:00 – 14:00	Lunch Break	

We prepared various Jupyter notebooks for this training...

see our Jupyter book at

<https://dkrz-user-workshop-dm-training1.gitlab-pages.dkrz.de/data-standardization/intro.html>

or our gitlab repository

<https://gitlab.dkrz.de/dkrz-user-workshop-dm-training1/data-standardization>

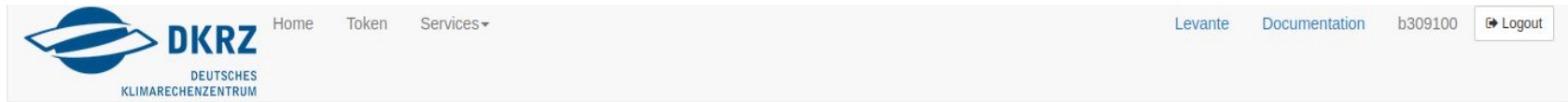
Please try them out on DKRZ's JupyterHub: <https://jupyterhub.dkrz.de/>

Set up jupyterhub

Find the instructions also here (to copy paste commands if necessary):

<https://dkrz-user-workshop-dm-training1.gitlab-pages.dkrz.de/data-standardization/intro.html>

First, visit <https://jupyterhub.dkrz.de> and click on Preset **start**



Spawner Options



Set up jupyterhub

Select the **job profile** from below and **enter** one of your **accounts** that allows you to launch parallel jobs on levante. Reservation and QoS can be left empty. Then click on **Start**.

Server Options

Reset options

Select a job profile:

20 GB memory, interactive, 12:00h

Account (--account):

bk1261

Reservation (--reservation):

QoS (--qos):

Start

If you do not know any of your accounts:

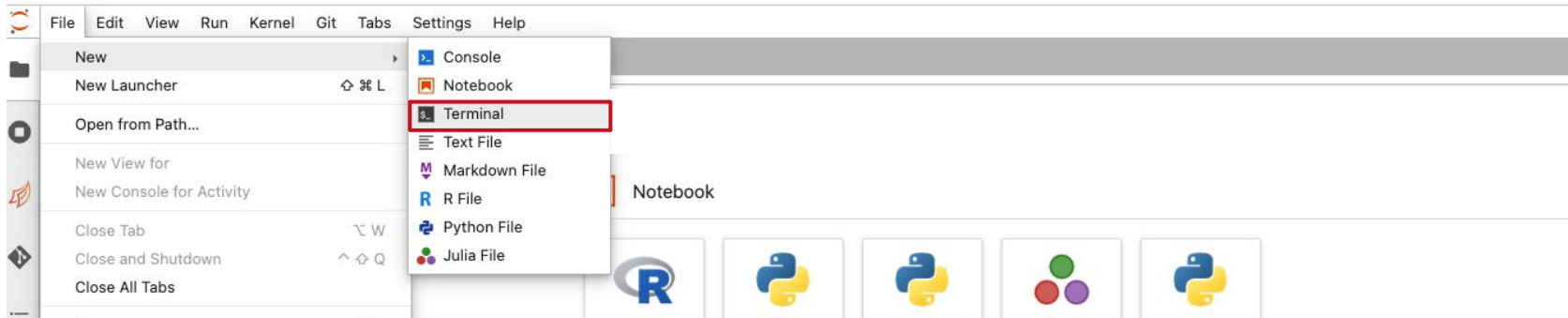
1. Open a terminal
2. ssh-login to levante
3. Execute the *groups* command to list all projects, and try one of the 6-digit group names

```
~$ ssh -X b309100@levante.dkrz.de  
[b309100@levante5 ~]$ groups  
id0853 b b309 bk1261
```

Now wait until your server is starting up...you will be redirected automatically to the JupyterLab Interface

Set up jupyterhub

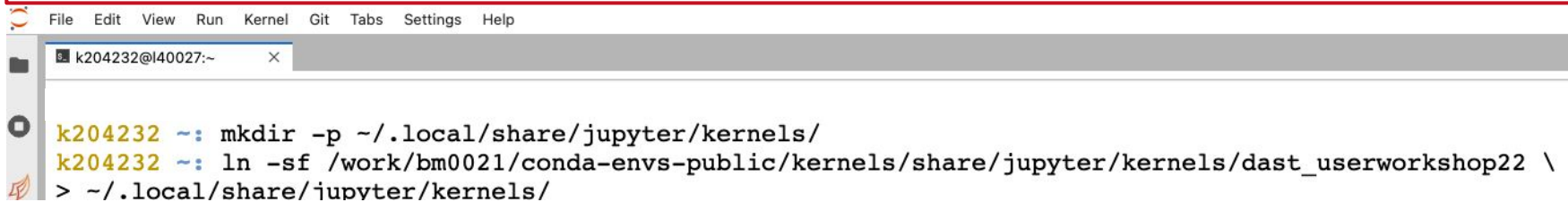
To be able to use the pre-installed jupyter kernel, open a terminal with jupyterhub.



Alternative: open a terminal on your local machine and ssh-login to levante ...

Set up jupyterhub

Then link the pre-installed kernel to your user directory.



```
File Edit View Run Kernel Git Tabs Settings Help
k204232@l40027:~
k204232 ~: mkdir -p ~/.local/share/jupyter/kernels/
k204232 ~: ln -sf /work/bm0021/conda-envs-public/kernels/share/jupyter/kernels/dast_userworkshop22 \
> ~/.local/share/jupyter/kernels/
```

Then clone the git repository to a directory of your choice (eg. your HOME or SCRATCH directory).

```
k204232 ~: cd ~
k204232 ~: git clone https://gitlab.dkrz.de/dkrz-user-workshop-dm-training1/data-standardization.git
Cloning into 'data-standardization'...
```

The commands for copy/paste:

configure kernel (required for https://jupyterhub.dkrz.de)

```
mkdir -p ~/.local/share/jupyter/kernels/
```

```
ln -sf /work/bm0021/conda-envs-public/kernels/share/jupyter/kernels/dast_userworkshop22 ~/.local/share/jupyter/kernels/
```

clone git repo

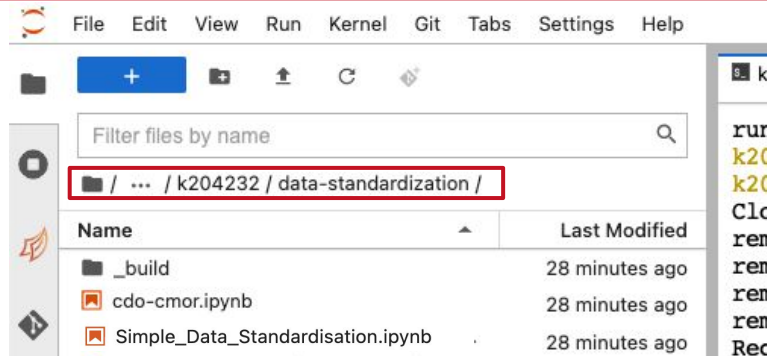
```
cd ~ # HOME directory; alternative your SCRATCH: cd /scratch/b/${USER}
```

```
module load git
```

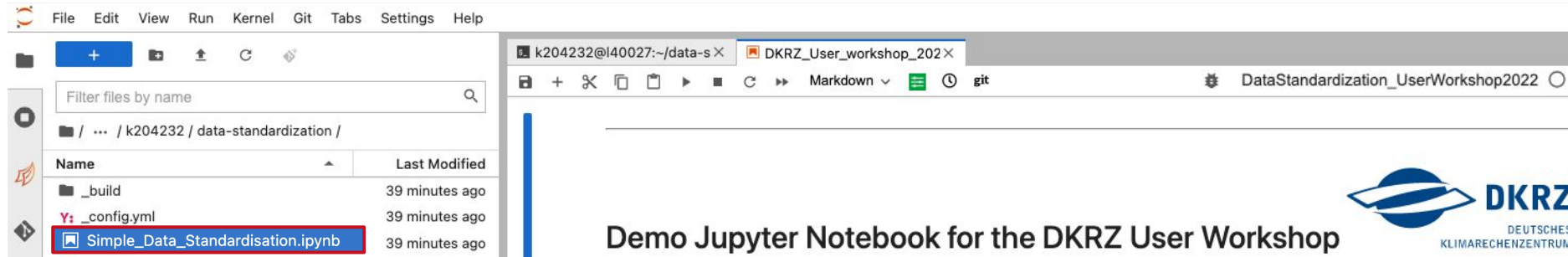
```
git clone https://gitlab.dkrz.de/dkrz-user-workshop-dm-training1/data-standardization.git
```

Set up jupyterhub

Now open the data-standardization directory in the jupyter file browser on the left.



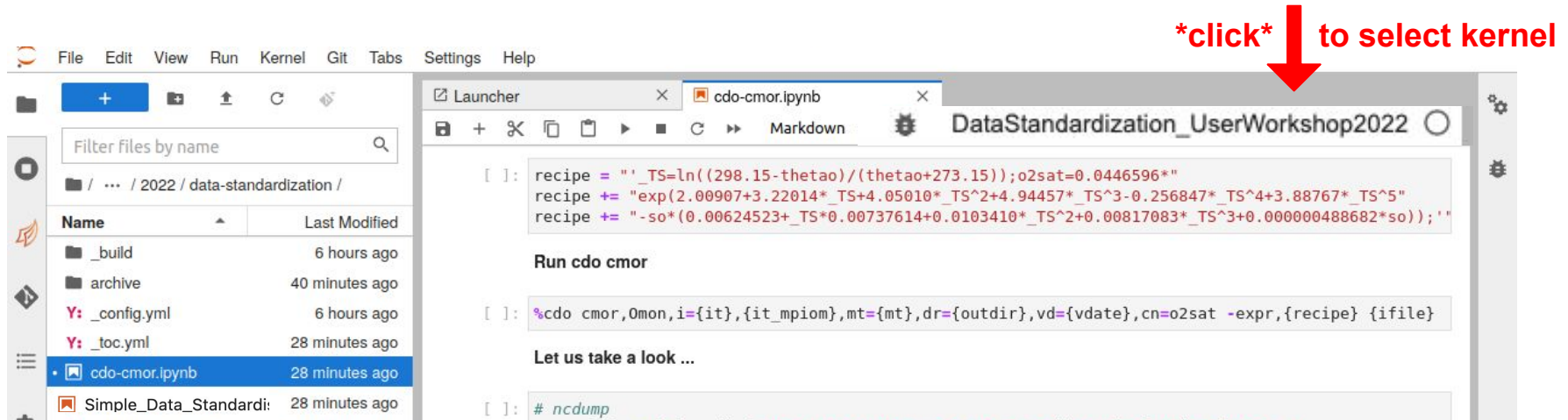
Then open the Simple_Data_Standardisation.ipynb jupyter notebook.



Set up jupyterhub

When you open any of the workshop notebooks (*.ipynb files) in the data-standardization folder, please check that the fitting kernel “DataStandardization_UserWorkshop22” is selected

click ↓ **to select kernel**



The screenshot displays the JupyterLab interface. On the left, the file browser shows the directory structure: / ... / 2022 / data-standardization /. The file 'cdo-cmor.ipynb' is selected, showing its last modified time as 28 minutes ago. On the right, the notebook editor shows the 'cdo-cmor.ipynb' notebook. The kernel 'DataStandardization_UserWorkshop22' is selected in the top right corner. The notebook content includes a code cell with a recipe definition, a text cell 'Run cdo cmor', another code cell with a command to run 'cdo cmor', a text cell 'Let us take a look ...', and a final code cell with the command '# ncdump'.

```
[ ]: recipe = "'_TS=ln((298.15-thetao)/(thetao+273.15));o2sat=0.0446596*"
recipe += "exp(2.00907+3.22014*_TS+4.05010*_TS^2+4.94457*_TS^3-0.256847*_TS^4+3.88767*_TS^5"
recipe += "-so*(0.00624523+_TS*0.00737614+0.0103410*_TS^2+0.00817083*_TS^3+0.000000488682*so));'"

Run cdo cmor

[ ]: %cdo cmor,0mon,i={it},{it_mpiom},mt={mt},dr={outdir},vd={vdate},cn=o2sat -expr,{recipe} {ifile}

Let us take a look ...

[ ]: # ncdump
```


Why Do We Need Data Standardization?

Data standardization means, e.g. to rewrite data

- in a common file format (e.g. netCDF)
- with similar data structures
- with sufficient & consistent metadata

Data standardization

- is vital when you want to reuse your data on the longer term
- is vital when sharing data with others
- accelerates scientific progress, particularly in the era of Big Data

Why Do We Need Data Standardization?

Data standardization means, e.g., to rewrite data....

- in a common file format (e.g. netCDF)
- with similar data structures
- with sufficient & consistent metadata

Data standardization

- is vital when you want to reuse your data on the longer term
- is vital when sharing data with others
- accelerates scientific progress, particularly in the era of Big Data
- is a precondition for a **successful data publication**

Announcement: FAIR data in Earth science **nature**



<https://www.nature.com/articles/d41586-019-00075-3>

Announcement: FAIR data in Earth science

Nature - Nature backs the Enabling FAIR Data initiative and requires authors to deposit data in community repositories.



Data Standardization – Make Your Data FAIR!

The **FAIR principles*** are the most widely adopted guiding principles for **scientific data management**.



Findable

Data are described by rich, **meaningful metadata**.

Data and metadata are assigned unique **persistent identifiers** (PIDs).

Data and metadata are indexed in a searchable Data Catalogue.



Accessible

Data and metadata can be **accessed** through **standard data protocols**.

Metadata remain accessible even if the data are no longer available.



Interoperable

Data are **readable for machines**.

Metadata are defined in **controlled vocabularies** (CVs; “standardised terms”).

Metadata include cross-references to related metadata (meaningful links).



Reusable

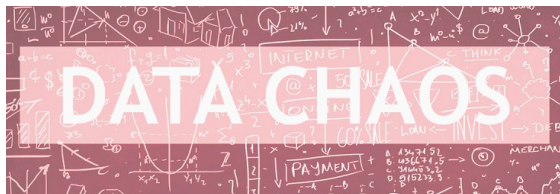
Data are clearly described and contain detailed **provenance** information.

Data are released with a clear **data licensing** status.

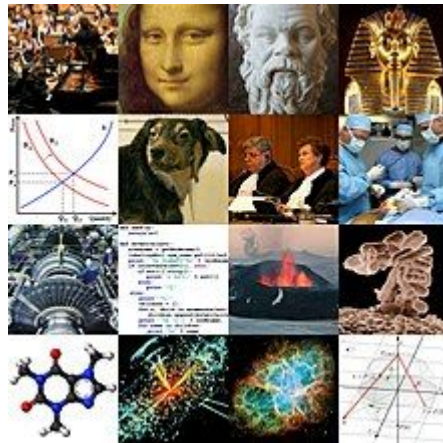
Data and metadata meet **relevant domain standards** (e.g. file formats).

Why Do We Need Data Standardization?

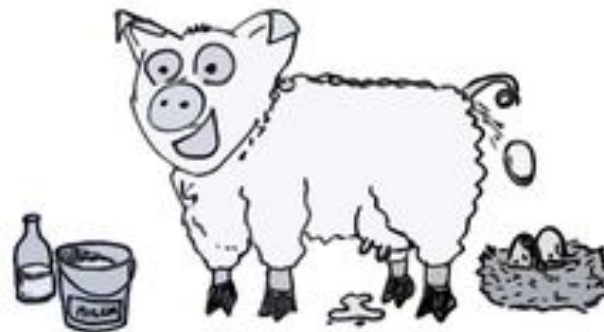
No Data Standard



Discipline/Project-specific Standard



One Standard for All Data



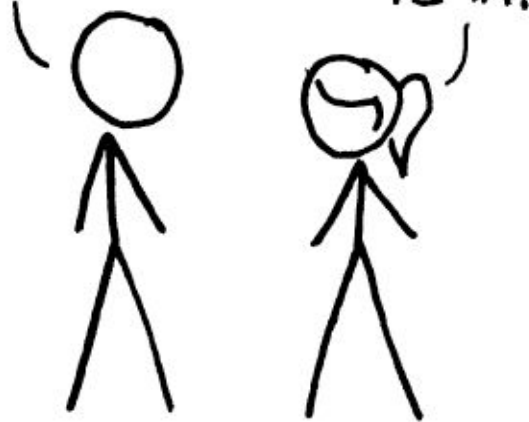
Data Standardization Scale

HOW STANDARDS PROLIFERATE:

(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)

SITUATION:
THERE ARE
14 COMPETING
STANDARDS.

14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.



SOON:

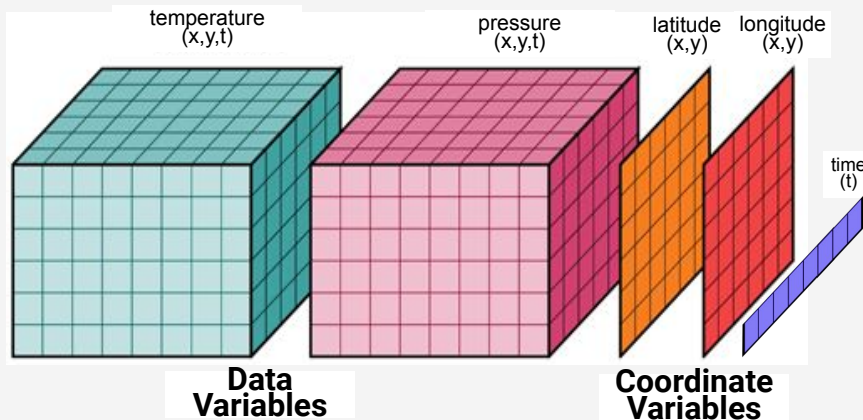
SITUATION:
THERE ARE
15 COMPETING
STANDARDS.

(Meta)data Standards in Earth System Sciences (ESS)

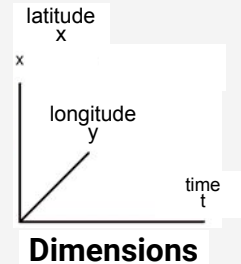
- Data Formats
 - netCDF(3,4), HDF5, GeoTIFF, GRIB...(self-describing)
 - General (Meta)data
 - CF (Climate and Forecast) metadata conventions
CF standard names: controlled vocabulary (CV)
 - COARDS conventions (predecessor of CF)
 - Discipline/Project-Specific (Meta)data
 - CMIP5/6
 - ESA CCI; Copernicus; ATMODAT;...
- } building upon CF

Climate and Forecast (CF) Metadata Conventions

- most widely used *community standard in ESS* for encoding data in netCDF data format (“CF-netCDF”).
- CF provides guidance on how to provide a definitive description of what the data in each variable represents, the spatial and temporal properties of the data, and the **placement of metadata within the netCDF file**.



A netCDF file has dimensions, variables, and attributes.



The attributes are

- variable metadata
- global metadata.

The netCDF Header

ncdump -h filename

- 1) Dimensions
- 2) Variables
incl.
Variable Attributes
- 3) Global attributes
- 4) Data

```
netcdf filename {
dimensions:
    lat = 3 ;
    lon = 4 ;
    time = UNLIMITED ; // (2 currently)

variables:
    float lat(lat) ;
        lat:long_name = "Latitude" ;
        lat:units = "degrees_north" ;
    float lon(lon) ;
        lon:long_name = "Longitude" ;
        lon:units = "degrees_east" ;
    int time(time) ;
        time:long_name = "Time" ;
        time:units = "days since 1895-01-01" ;
        time:calendar = "gregorian" ;
    float rainfall(time, lat, lon) ;
        rainfall:long_name = "Precipitation" ;
        rainfall:units = "mm yr-1" ;
        rainfall:missing_value = -9999.f ;

// global attributes:
    :title = "Historical Climate Scenarios" ;
    :Conventions = "CF-1.0" ;

data:
    lat = 48.75, 48.25, 47.75;
    lon = -124.25, -123.75, -123.25, -122.75;
    time = 364, 730;
    rainfall =
        761, 1265, 2184, 1812, 1405, 688, 366, 269, 328, 455, 524, 877,
        1019, 714, 865, 697, 927, 926, 1452, 626, 275, 221, 196, 223;
}
```

Coordinate
variable

Variable
attribute

Global
attribute

header

data

CF Standard Names

- are a controlled vocabulary (CV) CF standard name \neq netCDF variable name
- used to label the geophysical variables within CF compliant data
- have a precise definition and an prescribed “canonical” units (\Rightarrow Udunits2)

<http://cfconventions.org/Data/cf-standard-names/current/build/cf-standard-name-table.html>

Standard Name	Canonical Units
▼ air_temperature Air temperature is the bulk temperature of the air, not the surface (skin) temperature.	K

What are canonical units?

Every CF standard name must be specified together its canonical units, except for unitless quantities.

Instead of the canonical units, any Udunits2-compatible units can be specified.

Udunits-2 is the Unidata units library <https://github.com/Unidata/UDUNITS-2>

CF Standard Names

- are a controlled vocabulary (CV)
- used to label the geophysical variables within CF compliant data
- have a precise definition and an prescribed “canonical” unit (=> Udunits2)
- are used as value for the `standard_name` variable attribute, e.g.

```
float tas(time, rlat, rlon);
```

```
tas:standard_name = "air_temperature";
```

```
tas:units = "K";
```

```
tas:coordinates = "lon lat height";
```

auxiliary coordinate variables with height = 2 m

```
tas:long_name = "Near-Surface Air Temperature";
```

Software that “understands” CF-netCDF

python tools implementing an abstracted “CF data model”

- *cfdm*
and the extensions *cf-python* and *cdf-plot/cf-view*
- *iris*

tools that support parts of the CF conventions

- *CDO, NCO*
- *xarray, cf-xarray*

ATMODAT standard for simple FAIR data publications

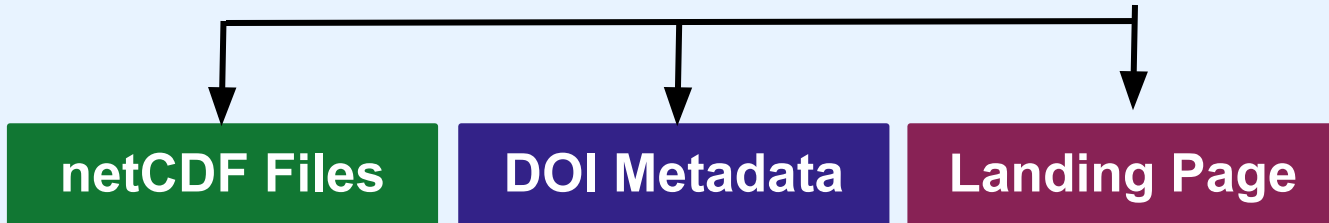
Ganske et al. (2021) https://doi.org/10.35095/WDCC/atmodat_standard_en_v3_0

- ❑ **CF-netCDF** data files
- ❑ a set of **mandatory, recommended** and **optional metadata**
- ❑ **atmodat checker** can check compliance
- ❑ assumes a open license **data publication** with a **DataCite DOI**
- ❑ highlighted in **WDCC** with



*a new quality seal for FAIR &
open Earth System Science data*

ATMODAT standardization



Requirements for the netCDF files

Always NetCDF

CF Conventions/
Controlled Vocabulary

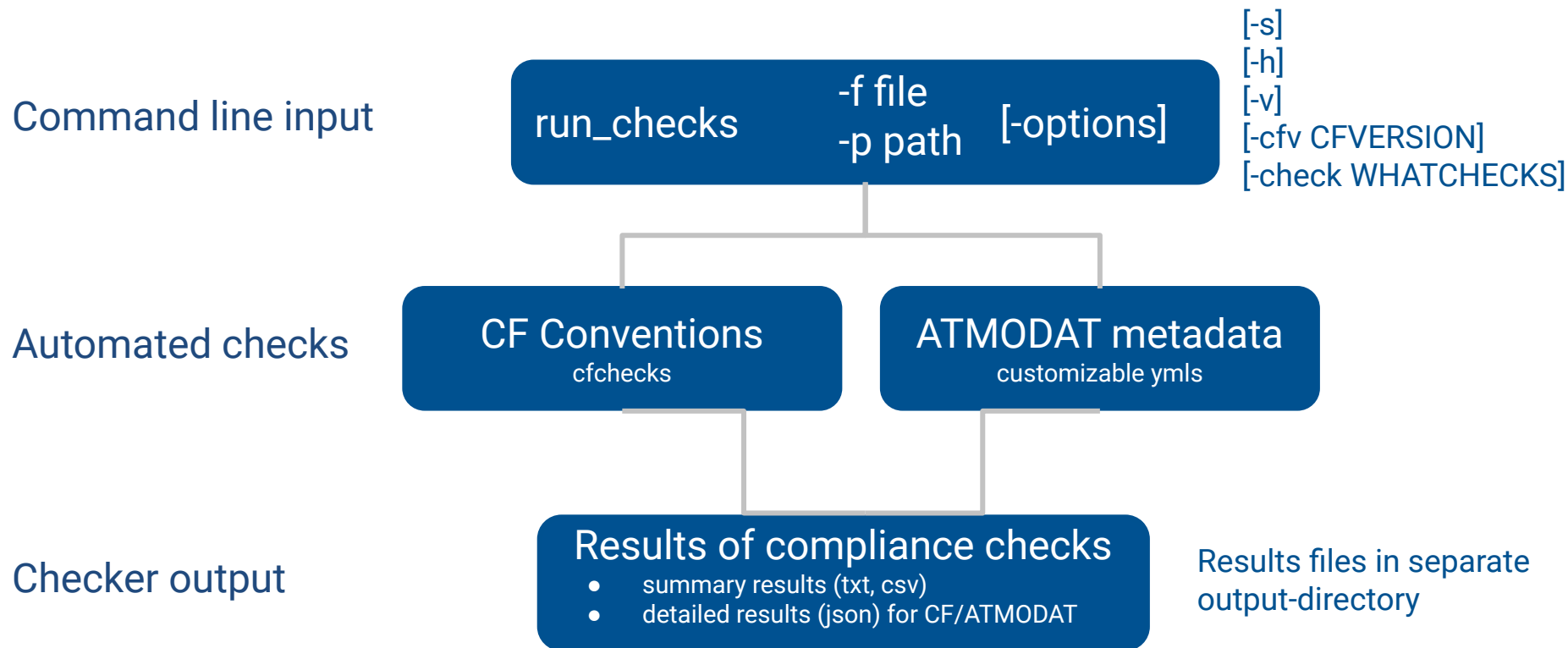
Global Attributes to
make the file
self-describing

```
netcdf CD24_base_2008_dec_1_1915785082846561030 {
    dimensions:
        ...
    variables:
        float gas_so2(time, lat, lon) ;
        gas_so2:standard_name = "mass_concentration_of_sulfur_dioxide_in_air" ;
        gas_so2:units = "kg m-3" ;
        ...

    //global attributes:
        :Conventions = "CF-1.6 ATMODAT-3.0" ;
        :institution = "Helmholtz-Zentrum Geesthacht, ....." ;
        :license = "CC-BY-4.0"
        :source = "model: CMAQ v5.0.1 cb05tump ae5; ....." ;
        :summary = "Standard CMAQ Model run over Northwestern Europe [...]" ;
        :title = "Concentrations of gaseous pollutants [...] over [...] Europe [...] in 2008";
        :creation_date = "2015-04-02" ;
        :history = "Fri Oct 7 11:52:24 2022: ncatted -O -a standard_name,height,c,c,..\\n",
            "Fri Oct 7 11:52:24 2022: ncap2 -o ....\\n",
            .....
}
```

atmodat checker

https://github.com/AtMoDat/atmodat_data_checker



atmodat checker

https://github.com/AtMoDat/atmodat_data_checker

create summary

Example short summary.txt: `run_checks.py -s -p myfolder/`

Short summary of checks:

Checking against: atmodat_standard:3.0, CF table version: 77

Version of the AtMoDat checker: 1.1.0

Checked at: 2021-08-11T14:54:17.517485

Number of checked files: 3

Total checks passed: 4/31

Mandatory checks passed: 2/4

Recommended checks passed: 2/18

Optional checks passed: 0/9

CF checker errors: 1

Please try it out!

<https://gitlab.dkrz.de/dkrz-user-workshop-dm-training1/data-standardization>


Jupyter Notebook: Simple_Data_Standardisation.ipynb

gitlab.dkrz.de/dkrz-user-workshop-dm-training1/data-standardization/-/blob/main/Simple_Data_Standardisation.ipynb

Menu

Simple_Data_Standardisation.ipynb 37.43 KiB

Open in Web IDE Replace

Demo Jupyter Notebook for the DKRZ User Workshop  **DKRZ**
DEUTSCHES
KLIMARECHENZENTRUM

Data Standardization: Simple Tools for Achieving FAIR CF-compliant Data

Angelika Heil (heil@dkrz.de), 12 October 2022

README

In this Jupyter Notebook with a IPython kernel, we use both, python and BASH commands.

To execute BASH commands, put

- (i) %%bash is in first line of a cell --> entire cell is BASH
- (ii) !command --> individual line is BASH.
- (iii) {pyvar} to pass a python variable to BASH

Step 0: Load the required libraries

Project specific data standardization - CV & DRS

Project specific standardization - Example CMIP6

→ The CMIP6 metadata requirements build on CF

CMIP6 Data Reference Syntax ([DRS](#)) and **Controlled Vocabularies** ([CVs](#)) prescribe

- global attributes related to
 - experiment
 - variant / simulation
 - institution
 - licensing restrictions
 - etc
- components to compose filenames and directory structure
 - These components uniquely define each dataset (i.e. a single variable of a single simulation)
- variables and related attributes (“MIP tables” / “CMOR tables”)

Project specific data standardization - CV & DRS

DRS/CV characterize the data and are essential for organizing, managing and accessing large data collections by

- enabling the intercomparison of data from different sources
- helping to define and structure data catalogs (eg. intake catalogs)
- serving as search facets in data portals (eg. [ESGF](#) Data Portal)



Hosted by   Powered by  

Welcome, Guest | Login | Create Account

You are at the [ESGF-DATA.DKRZ.DE](#) node

Home **Technical Support**

WCRP CMIP6
World Climate Research Programme

WARNING: Not all models include a variant 'r11p1f1' and across models, identical values of variant_label do not imply identical variants! To learn which forcing datasets were used in each variant, please check modeling group publications and documentation provided through ES-DOC.

Enter Text: Display results per page [\[More Search Options \]](#)

☐ Show All Replicas ☐ Show All Versions ☐ Search Local Node Only (Including All Replicas)

Search Constraints: ☒ AWI-CM-1-1-MR ☒ historical ☒ cct

Total Number of Results: 5

Please login to add search results to your Data Cart

Expert Users: you may display the search URL and return results as XML or return results as JSON

1. CMIP6.CMIP.AWI-AWI-CM-1-1-MR.historical.r51p1f1.Amon.cct.gn
Data Node: [esgf3.dkrz.de](#)
Version: 20181218
Total Number of Files (for all variables): 165
Full Dataset Services: [\[Show Metadata \]](#) [\[List Files \]](#) [\[THREDDS Catalog \]](#) [\[WGET Script \]](#) [\[LAS Visualization \]](#) [\[Show Citation \]](#) [\[PID \]](#) [\[Globus Download \]](#) [\[Further Info \]](#)
2. CMIP6.CMIP.AWI-AWI-CM-1-1-MR.historical.r11p1f1.Amon.cct.gn
Data Node: [esgf3.dkrz.de](#)
Version: 20181218
Total Number of Files (for all variables): 165
Full Dataset Services: [\[Show Metadata \]](#) [\[List Files \]](#) [\[THREDDS Catalog \]](#) [\[WGET Script \]](#) [\[LAS Visualization \]](#) [\[Show Citation \]](#) [\[PID \]](#) [\[Globus Download \]](#) [\[Further Info \]](#)
3. CMIP6.CMIP.AWI-AWI-CM-1-1-MR.historical.r21p1f1.Amon.cct.gn
Data Node: [esgf3.dkrz.de](#)
Version: 20181218
Total Number of Files (for all variables): 165
Full Dataset Services: [\[Show Metadata \]](#) [\[List Files \]](#) [\[THREDDS Catalog \]](#) [\[WGET Script \]](#) [\[Show Citation \]](#) [\[PID \]](#) [\[Globus Download \]](#) [\[Further Info \]](#)
4. CMIP6.CMIP.AWI-AWI-CM-1-1-MR.historical.r31p1f1.Amon.cct.gn
Data Node: [esgf3.dkrz.de](#)
Version: 20181218

Project specific data standardization - CMOR

CMOR - Climate Model Output Rewriter

- library with functions to read, reformat and write climate data variables
- reads the official project standard configuration
→ no user side preparation of the project format description for community projects
- requires an extensive configuration
- its application produces output compliant with project metadata requirements!

→ Can (but does not have to) be replaced by

[xarray](#) / [NCO](#) in small scale projects

→ For large scale projects like CMIP6 (2000+ variables requested for 100+ experiments) there is no way around CMOR!

```
cmor_setup();  
cmor_dataset_json();  
cmor_load_table();  
cmor_set_table();  
cmor_axis();  
cmor_grid();  
cmor_set_grid_mapping();  
cmor_time_varying_grid_coordinate();  
cmor_zfactor();  
cmor_variable();  
cmor_set_deflate();  
cmor_set_variable_attribute();  
cmor_create_output_path();  
cmor_write();  
cmor_close();
```

Project specific data standardization - cdo cmor

Climate Data Operator (CDO) cmor

- CMOR has been integrated into the widely known and commonly appreciated CDOs
- **benefit of CDOs data interface** - allowing input in various file formats and structures
- **only one call** required - cdo cmor carries out the complex orchestration of the CMOR functions

Our advice:

Use synergies and profit of cdo CMORs capabilities. Avoid to repeat work by writing your own CMORization tool.

Links

- [cdo cmor wiki](#) (incl. User Guide, Hands-On, ...)
- CMIP6 Standardization with cdo cmor extensive [Tutorial / Workshop Material](#)
- soon an extensive example to be found [here](#)

```
cmor_setup();  
cmor_dataset_json();  
cmor_load_table();  
cmor_set_table();  
cmor_axis();  
cmor_grid();  
cmor_set_grid_mapping();  
cmor_time_varying_grid_coordinate();  
cmor_zfactor();  
cmor_variable();  
cmor_set_deflate();  
cmor_set_variable_attribute();  
cmor_create_output_path();  
cmor_write();  
cmor_close();
```

cdo cmor,Amon ifile.grb

Project specific data standardization - cdo cmor

CMIP6_Amon.json

contains parts of the data request in a CMOR-readable format

```
cdo cmor, CMIP6_Amon.json, \
    i=.cdocmorinfo, \
    mt=mapping_table.txt, \
    gi=grid_info.nc \
infile
```

grid_info.nc

contains a grid description including coordinates and bounds

```
variables:
    double lat(lat);
    double
lat_bnds(lat,bnds);
```

.cdocmorinfo

contains the CMOR configuration

```
activity_id="CMIP"
institution_id="MPI-M"
```

can be created with

<https://c6dreq.dkrz.de/cdocmorinfo/index.html>

mapping_table.txt

links model output variables with CMOR variables

```
&parameter pmt=Amon cmor_name=tasmax code=201
/
```

Project specific data standardization - Quality Control

[PrePARE](#) is distributed with CMOR and checks that the CV and DRS requirements of files are met

PrePARE is the minimal requirement to publish CMIP6 data via the ESGF. At DKRZ we perform additional checks with the Tool [QA-DKRZ](#) - a selection of possible checks is listed below:

- folder structure
- compliance with CF conventions
- consistency between different files of the same dataset
- gaps in the timeline (missing files or timesteps)
- outliers

The [Jupyter Notebook example](#) contains CMORization examples and a PrePARE check → **cdo-cmor.ipnb**

CMORization - Concluding remarks

- Application of CMOR ensures compliance with project metadata requirements
 - Application of NCO / xarray or custom CMOR-like tools ensures weeks of back and forth with the Data Quality Assurance team
 - (Setting up the) CMORization takes time and requires efforts and **planning**:
 - We are happy to provide support and to guide you through the CMORization process
 - Please get in touch with us early - preferably before starting the simulations:
 - certain output namelist settings might make CMORization much easier
 - for certain output namelists we might have CMORization scripts ready
 - cdo (cmor) might require updates to process your model data
- If properly planned, CMORization is a manageable task

You can reach us via esgf-publication@dm-rt.dkrz.de

Extra slides

CMIP and CMOR

Data standardization means, to rewrite data

- in a common format
- with files structured similarly
- with sufficient and uniformly stored metadata

This is done to enable

- an easy and fast acquisition or listing of the data (eg. in catalogs or for analysis) with common and widespread tools
- the possibility for intercomparison of data from different sources without requiring complex unit conversions or other post processing steps

Additionally a FAIR publication infrastructure

- quality checks the metadata
- adds identifiers for unique identification and persistent referencing of the datasets
- supports reproducibility of results by keeping track of errata information and retracted / replaced datasets

→ all only possible due to the standardized (meta)data format and structure

Project specific data standardization

```
ncdump -h test.nc
```

```
netcdf test {
dimensions:
    lon = 192 ;
    lat = 96 ;
    height = 1 ;
    time = UNLIMITED ; // (120 currently)
variables:
    double lon(lon) ;
        lon:standard_name = "longitude" ;
        lon:long_name = "longitude" ;
        lon:units = "degrees_east" ;
        lon:axis = "X" ;
    double lat(lat) ;
        lat:standard_name = "latitude" ;
        lat:long_name = "latitude" ;
        lat:units = "degrees_north" ;
        lat:axis = "Y" ;
    double height(height) ;
        height:standard_name = "height" ;
        height:long_name = "height" ;
        height:units = "m" ;
        height:positive = "up" ;
        height:axis = "Z" ;
    double time(time) ;
        time:standard_name = "time" ;
        time:units = "day as %Y%m%d.%f" ;
        time:calendar = "proleptic_gregorian" ;
        time:axis = "T" ;
    float var1(time, height, lat, lon) ;
        var1:table = 255 ;
        var1:grid_type = "gaussian" ;

// global attributes:
    :CDI = "Climate Data Interface version 1.7.0 (http" ;
    :Conventions = "CF-1.4" ;
    :history = "Thu Sep 07 14:27:25 2017: cdo -f nc ci" ;
    :CDO = "Climate Data Operators version 1.7.0 (http"
```

Restructuring

- Generating bounds
- Adding attributes

Renaming

```
ncdump -h ../CMIP6/CMIP/MPI-M/MPIESM-1-2-HR_
netcdf tas_Amon_MPIESM-1-2-HR_historical_r11p1f1_gm_200101-20
dimensions:
    time = UNLIMITED ; // (120 currently)
    lat = 96 ;
    lon = 192 ;
    bnds = 2 ;
variables:
    double time(time) ;
        time:bounds = "time_bnds" ;
        time:units = "days Since 1850-1-1 00:00:00" ;
        time:calendar = "proleptic_gregorian" ;
        time:axis = "T" ;
        time:long_name = "time" ;
        time:standard_name = "time" ;
    double time_bnds(time, bnds) ;
    double lat(lat) ;
        lat:bounds = "lat_bnds" ;
        lat:units = "degrees_north" ;
        lat:axis = "Y" ;
        lat:long_name = "latitude" ;
        lat:standard_name = "latitude" ;
    double lat_bnds(lat, bnds) ;
    double lon(lon) ;
        lon:bounds = "lon_bnds" ;
        lon:units = "degrees_east" ;
        lon:axis = "X" ;
        lon:long_name = "longitude" ;
        lon:standard_name = "longitude" ;
    double lon_bnds(lon, bnds) ;
    double height ;
        height:units = "m" ;
        height:axis = "Z" ;
        height:positive = "up" ;
        height:long_name = "height" ;
        height:standard_name = "height" ;
    float tas(time, lat, lon) ;
        tas:standard_name = "air temperature" ;
        tas:long_name = "Near-Surface Air Temperature" ;
        tas:comment = "near-surface (usually, 2 meter)" ;
        tas:units = "K" ;
        tas:cell_methods = "area: time: mean" ;
```

Benefits from the CMIP data infrastructure

Benefits from the CMIP data infrastructure

The CMIP6 data at DKRZ comprises about 4 PB. In order to tackle the challenges of data provision and dissemination for such a repository size, a [state-of-the-art data infrastructure](#) has been developed around that pool. In the following, we highlight three aspects of the data workflow.

Data quality

CMIP6 data is only available in a common and **reliable** [Data format](#)

- No adaptations needed for output of specific models
- Makes data **interoperable** and enables general cmip-analysing software products as ESMVal

CMIP6 data was **quality controlled** before published with [PrePARE](#)

- We can ensure e.g. correct coordinates and attribute configurations in the data

CMIP6 data is **transparent** about occurring errors

- Search the [errata](#) data base for origins of suspicious analysis results

Data publication

- **Find and access** data via the fail-safe [ESGF portal](#) due to the redundant network of ESGF nodes
- Extended **documentation** for simulation conducts provided in the [ES-Doc](#) data base
- **Persistent Identifier** (PIDs) ensure long-term webaccess to dataset information
- **Citation information** and DOIs for all published datasets easily retrievable

Data curation

The CMIP6 data pool is maintained by

- automatic **ingest** and **egest** which keeps the data updated
- **Back ups** of the data pool in the tape archive
- **Long term archival** for a range of datasets used in the IPCC report

⇒ High quality Analysis services only because of a high quality data supply

Keep *Acknowledging* and *requesting* high data quality!

Other jupyter notebooks

Notebooks der summer school:

netcdf basics

https://github.com/IS-ENES-Data/summer-school-2022/blob/main/notebooks/files_and_to_ols/NetCDF%20and%20CF%20-%20The%20Basics.ipynb

xarray basics:

https://github.com/IS-ENES-Data/summer-school-2022/blob/main/notebooks/files_and_to_ols/XArray%20and%20CF.ipynb

nco ncdump cdo basics:

https://github.com/IS-ENES-Data/summer-school-2022/blob/main/notebooks/files_and_to_ols/basic_netcdf_utils.ipynb

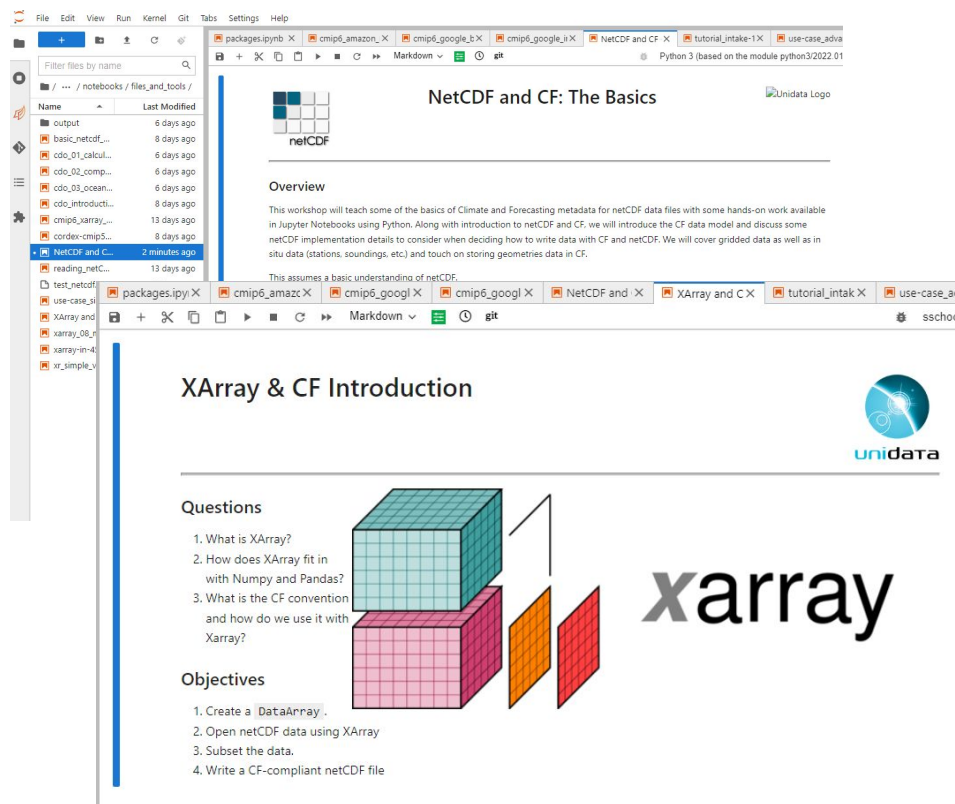
xarray (with cf-xarray part):

https://github.com/IS-ENES-Data/summer-school-2022/blob/main/notebooks/files_and_to_ols/xarray-in-45-min.ipynb

CMIP and CMOR

Hands on examples for NetCDF-CF:

- **NetCDF and CF - The Basics.ipynb**
 - shortend version of notebook from [Unidata training material](#) repo
 - uses netCDF4-python library



The screenshot shows a Jupyter Notebook environment with two open notebooks. The top notebook is titled "NetCDF and CF: The Basics" and features the Unidata logo. The bottom notebook is titled "XArray & CF Introduction" and includes a list of questions and objectives related to XArray.

NetCDF and CF: The Basics

Overview

This workshop will teach some of the basics of Climate and Forecasting metadata for netCDF data files with some hands-on work available in Jupyter Notebooks using Python. Along with introduction to netCDF and CF, we will introduce the CF data model and discuss some netCDF implementation details to consider when deciding how to write data with CF and netCDF. We will cover gridded data as well as in situ data (stations, soundings, etc) and touch on storing geometries data in CF.

XArray & CF Introduction

Questions

1. What is XArray?
2. How does XArray fit in with Numpy and Pandas?
3. What is the CF convention and how do we use it with Xarray?

Objectives

1. Create a DataArray.
2. Open netCDF data using XArray
3. Subset the data.
4. Write a CF-compliant netCDF file

- **XArray and CF.ipynb**
 - from [Unidata workshop training material](#) repo
- if you are not yet familiar with xarray please get back to this after the next session where xarray basics will be introduced ..

Elements of CF-netCDF

Element	Description
Data variable	Scientific data discretised within a domain
Dimension	Independent axis of the domain
Coordinate variable	Unique coordinates for a single axis
Auxiliary coordinate variable	Additional/alternative coordinates for any axes
Boundary variable	Cell vertices
Ancillary data variable	Metadata that depends on the domain
Formula terms attribute	Vertical coordinate system
Feature type attribute	Characteristics of discrete sampling geometry
Cell methods attribute	Description of variation within cells
...

