Deutscher Wetterdienst
Wetter und Klima aus einer Hand

pyku - handling climate data with ease*

Stephane Haussler, Christian Steger, Harald Rybka, Birgit Mannig, Igor Detring, et al.

Coming Soon!

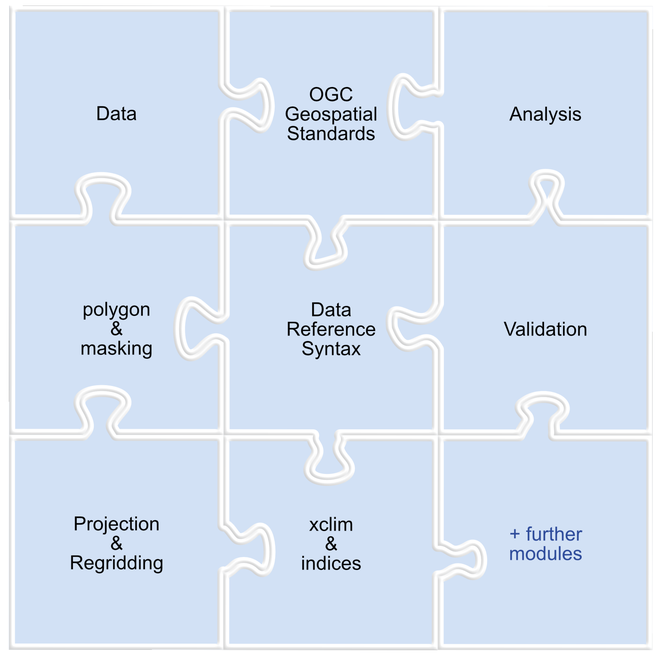
A python framework to work with standardized climate data

We are happy to announce the open-source publication of a python library to foster collaborative development on the use of climate data

pyku is an extension built on xarray for handling climate data comprehensively. Its primary aim is to facilitate the reading and manipulation of all climate-related data and metadata.

It features the handling of multiple **Data Reference Syntaxes** (DRS), **reprojection**, **regridding**, **polygonization**, **masking**, basic **analysis and validation, CMORization**, and providing **OGC-conform** data structures as product-ready output. Its modularity makes it easy to extend with scientific or structural modules.

*** pyku is the current working name of this project. The final package name may change in the future**



Data Reference Syntax

- **pyku** includes and allows to include individual Data Reference Syntaxes (**DRS**)
- It provides functions to **CMOR**ize data, automatically handling a significant number of attributes and unit conversions.
- However, global attributes must be manually set, as they cannot be inferred

code snippet

```
import xarray as xr
import pyku

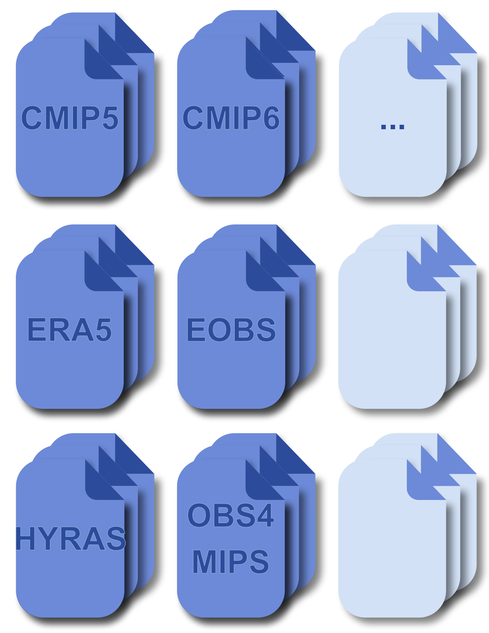
data = xr.open_dataset('example.nc')
cmorized =data.pyku.get_geodataset('tas')
cmorized.pyku.check_drs(standard='cordex') # valid
cmorized.pyku.check_drs(standard='cmip5') # invalid
```

check_drs(cordex)

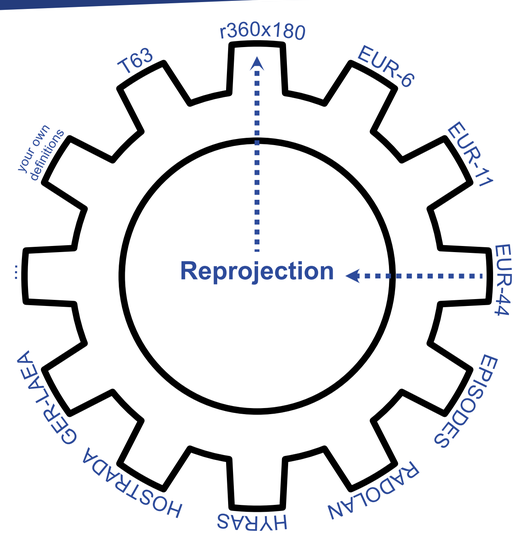
	key	value	issue
0	CORDEX_domain	EUR-44	None
1	driving_model_id	MPI-M-MPI-ESM-LR	None
2	driving_experiment_name	historical	None
3	driving_model_ensemble_member	r1i1p1	None
4	model_id	SMHI-RCA4	None
5	rcm_version_id	v1	None
6	frequency	day	None
7	product	output	None
8	institute_id	SMHI	None
9	experiment_id	historical	None

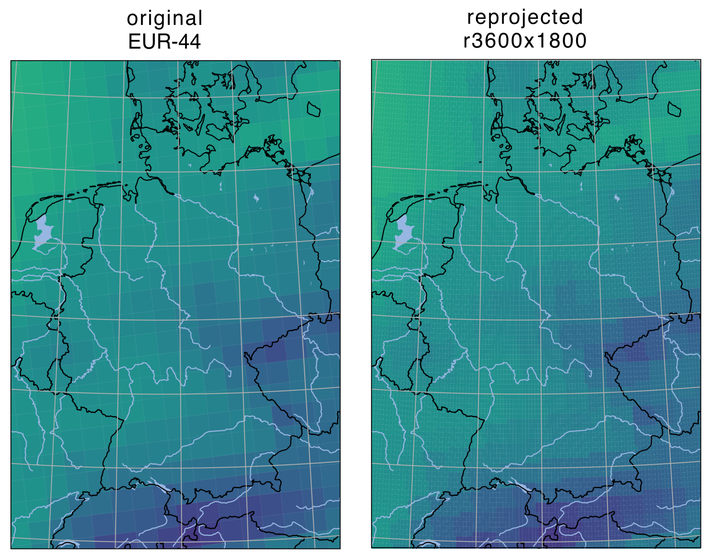
check_drs(cmip5)

	key	value	issue
0	MIP_table	None	missing value
1	model	None	missing value
2	experiment	historical	None
3	ensemble_member	None	missing value
4	activity	None	missing value
5	product	output	None
6	institute	None	missing value
7	frequency	day	None
8	modeling_realm	None	missing value



Reprojection & Regridding





Working with projected data is often tricky and elaborative.

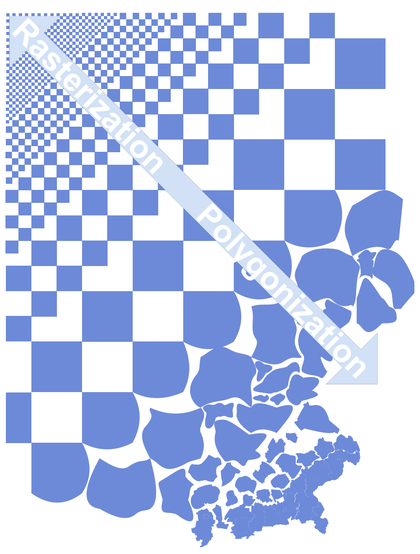
- **pyku** enables re-projection with ease.
- **pyku** gathers most common projections as well as DWD-product specific definitions.
- **pyku** furthermore integrates **xESMF** to allow complex regridding

code snippet

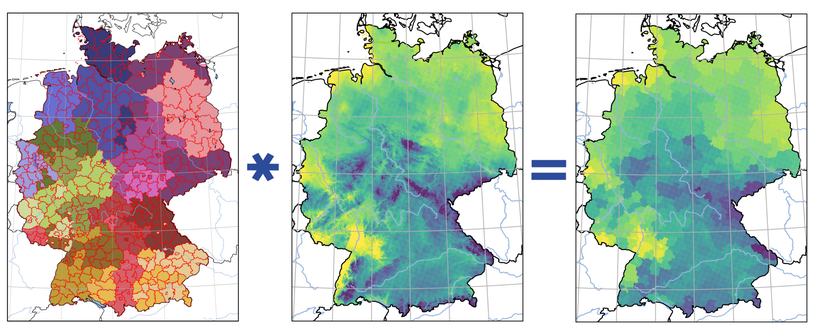
```
import xarray as xr
import pyku

data = xr.open_dataset('example.nc')
projected = data.pyku.project('world_3600x1800')
pyku.analyse.two_maps(data, projected, var='tas')
```

Rasterization & Polygonization



- **pyku** includes sohpisticated ways of handling rasterization and polygonization.
- An experimental downscaling method is integrated.
- Many standard resources, such as. shapes of naturalearth, are natively included. External sources can be easily integrated



Summary

- **pyku** is meant to be a common ground to jointly work on climate data.
- It will be published open-source within the next few months, including the invitation to contribute.
- Many additional modules will be added in the future, such as **bias-correction, unit conversion**, and more.
- While not reinventing the wheel, **pyku** also functions as a wrapper to well established libraries and packages.
- **pyku** development aims to follow principle standards of software development focusing not exclusively on **unittesting** and test-driven-development, or **documentation** and documentation-driven-development.

Contact: igor.detring@dwd.de