

Tiered of downloading Terabytes out of Petabytes of climate model data ?

Large European climate data centers offer the possibility to directly exploit locally available large climate data pools (e.g. CMIP6 data)

Two types of service:

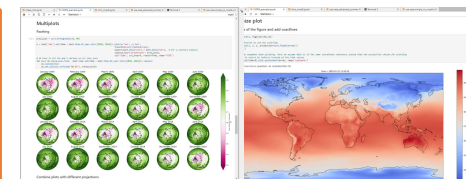
- “Jump start service”:
 - + minimal application procedure
 - limited compute resources
- Analysis platform service:
 - short project proposal required
 - + guaranteed resource allocation

The offering from DKRZ, IPSL-CNRS, UKRI-CEDA, CMCC:

- Access to large European climate model data pools (multi-PByte data collections including CMIP6, CORDEX, ..)
- Access to associated HPC compute resources
- Access to interaktive analysis environments (including jupyter-hub installations at DKRZ, CMCC and STFC)
 - support for e.g. pangeo sw stack (xarray, dask), cdo, ESMValTool and user tailored environments..

Interested? Further information:

- **Climate Analytics service (ECAS):**
<https://portal.enes.org/data/data-metadata-service/climate-analytics-service>
- **Analysis platforms application:**
<https://portal.enes.org/data/data-metadata-service/analysis-platforms>
- **Demos, use-cases, example jupyter notebooks:**
<https://github.com/IS-ENES-Data/Climate-data-analysis-service>



Next
deadline:
31.05.2021



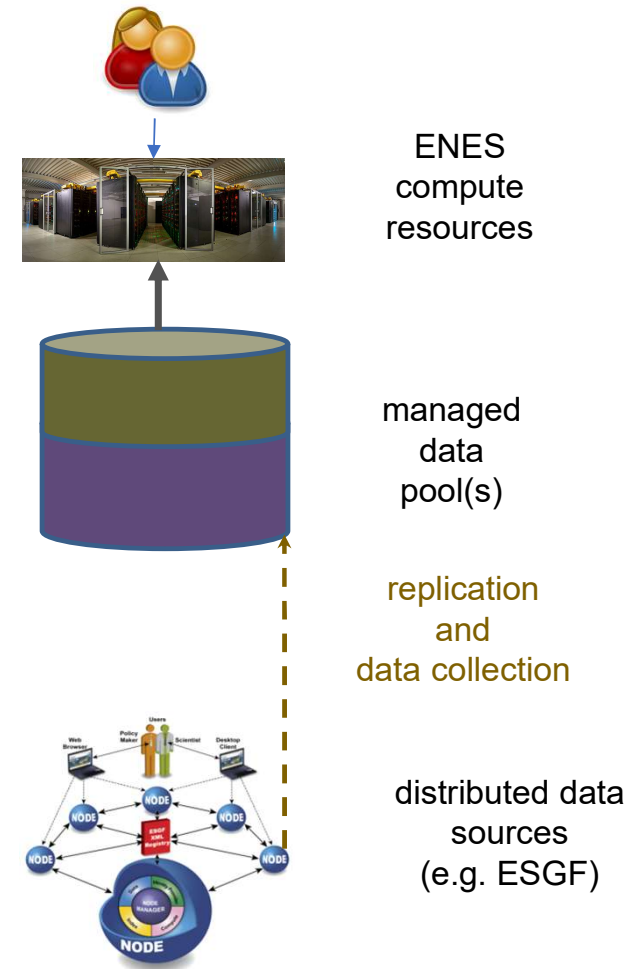
The IS-ENES3 project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 824084

IS-ENES Climate Data Pools

The IS-ENES climate data centers in Germany, France, England and Italy (DKRZ, IPSL-CNRS, UKRI-STFC and CMCC) provide managed data pools supporting climate model data analysis activities. Important data collections include CMIP5/6, CORDEX and ERA5.

Each center provides specific analysis platforms which are described at <https://portal.enes.org/data/data-metadata-service/analysis-platforms>

To be able to highlight specific features of the IS-ENES3 data analysis service offerings, we pick one service provider (DKRZ) in the following and illustrate features provided there ...



Climate Data Pool: Processing



- The data pools are directly associated to compute resources (e.g. the DKRZ HPC system) and can be accessed interactively as well as job based (e.g. using the SLURM batch system)
- Also modern interactive environments are supported e.g. jupyter notebooks
- [DKRZ Jupyter-hub](#) installation example:



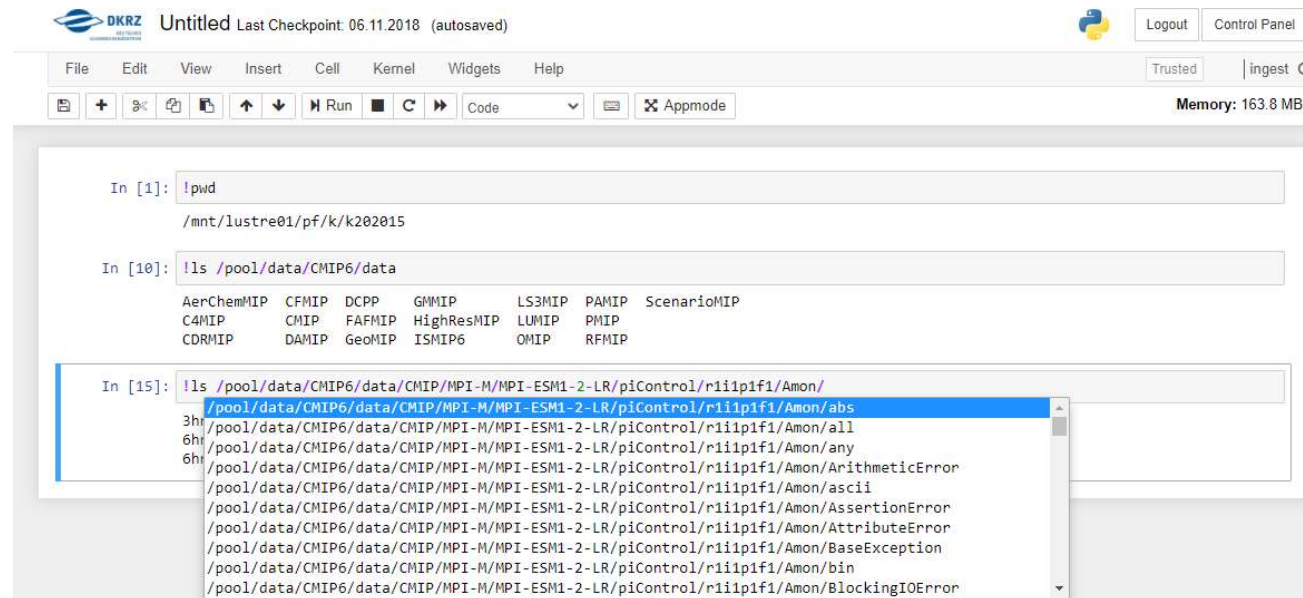
The Jupyter hub installation at DKRZ is supporting:

- specific resource profiles
- Predefined and user-defined kernels
- Intake catalogs (CMIP5/6, CORDEX, ERA5 ..)
- Parallel processing: e.g. dask

```
% mkdir $HOME/kernels
% conda create --prefix $HOME/kernels/tensorflow ipykernel python=3.x
% source activate $HOME/kernels/tensorflow
% python -m ipykernel install --user --name tensorflow --display-name="ten
% conda deactivate
```

Example: Accessing the data pool at DKRZ

- The data is part of the global HPC file system accessible locally in `/pool/data/CMIP6`



The screenshot shows a JupyterLab interface with a terminal window. The terminal output is as follows:

```
DKRZ Untitled Last Checkpoint: 06.11.2018 (autosaved) Logout Control Panel
File Edit View Insert Cell Kernel Widgets Help Trusted ingest
+ %< > Run Code Appmode Memory: 163.8 MB

In [1]: !pwd
/mnt/lustre01/pf/k/k202015

In [10]: !ls /pool/data/CMIP6/data
AerChemMIP  CFMIP  DCPD  GMMIP  LS3MIP  PAMIP  ScenarioMIP
C4MIP      CMIP   FAFMIP HighResMIP LUMIP  PMIP
CDRMIP     DAMIP  GeoMIP  ISMIP6  OMIP    RFHIP

In [15]: !ls /pool/data/CMIP6/data/CMIP/MPI-M/MPI-ESM1-2-LR/piControl/r1i1p1f1/Amon/
/pool/data/CMIP6/data/CMIP/MPI-M/MPI-ESM1-2-LR/piControl/r1i1p1f1/Amon/abs
3h /pool/data/CMIP6/data/CMIP/MPI-M/MPI-ESM1-2-LR/piControl/r1i1p1f1/Amon/all
6h /pool/data/CMIP6/data/CMIP/MPI-M/MPI-ESM1-2-LR/piControl/r1i1p1f1/Amon/any
6h /pool/data/CMIP6/data/CMIP/MPI-M/MPI-ESM1-2-LR/piControl/r1i1p1f1/Amon/ArithmeticError
/pool/data/CMIP6/data/CMIP/MPI-M/MPI-ESM1-2-LR/piControl/r1i1p1f1/Amon/ascii
/pool/data/CMIP6/data/CMIP/MPI-M/MPI-ESM1-2-LR/piControl/r1i1p1f1/Amon/AssertionError
/pool/data/CMIP6/data/CMIP/MPI-M/MPI-ESM1-2-LR/piControl/r1i1p1f1/Amon/AttributeError
/pool/data/CMIP6/data/CMIP/MPI-M/MPI-ESM1-2-LR/piControl/r1i1p1f1/Amon/BaseException
/pool/data/CMIP6/data/CMIP/MPI-M/MPI-ESM1-2-LR/piControl/r1i1p1f1/Amon/bin
/pool/data/CMIP6/data/CMIP/MPI-M/MPI-ESM1-2-LR/piControl/r1i1p1f1/Amon/BlockingIOError
```

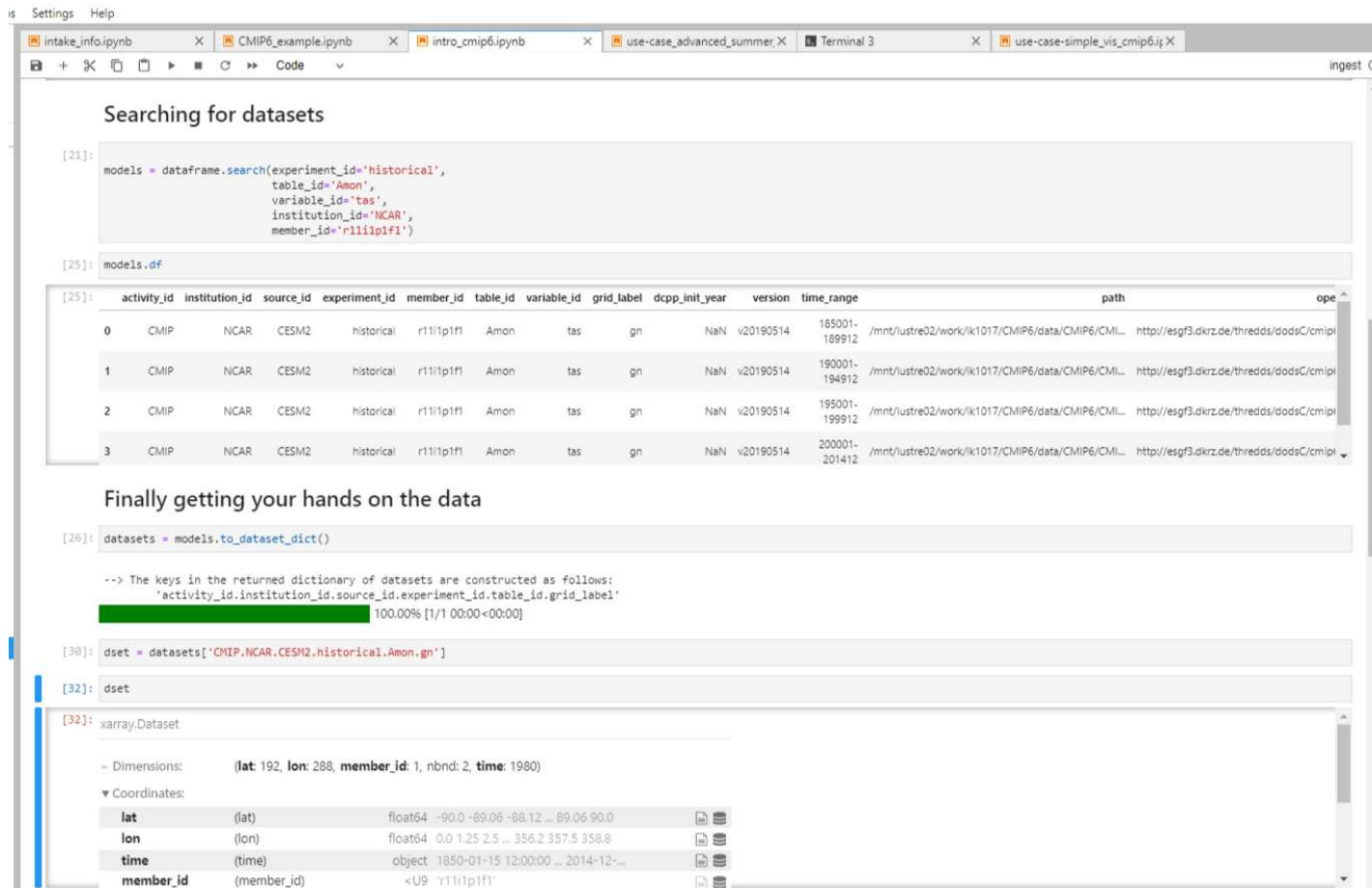
- Intake catalogs support the discovery process as well as the processing based on xarray, etc.
- Additional processing environments are available e.g. supporting cdo or evaluation activities based on the ESMValTool

From catalog search to the data ..

You can open a set of files at once in one xarray data cube

Data is always loaded lazily from netCDF files.

You can manipulate, slice and subset Dataset and DataArray objects, and no array values are loaded into memory until you try to perform some sort of actual computation



```
15 Settings Help
intake_info.ipynb x CMIP6_example.ipynb x intro_cmip6.ipynb x use-case_advanced_summer,X Terminal 3 x use-case-simple_vis_cmip6.ipynb X ingest O
Code
Searching for datasets
[21]: models = dataframe.search(experiment_id='historical',
                               table_id='Amon',
                               variable_id='tas',
                               institution_id='NCAR',
                               member_id='r111p1f1')
[25]: models.df
[25]:
```

	activity_id	institution_id	source_id	experiment_id	member_id	table_id	variable_id	grid_label	dcppl_init_year	version	time_range	path	ope
0	CMIP	NCAR	CESM2	historical	r111p1f1	Amon	tas	gn	NaN	v20190514	185001-189912	/mnt/lustre02/work/lk1017/CMIP6/data/CMIP6/CMI... http://esgf3.dkrz.de/thredds/dods/CMIP6/...	
1	CMIP	NCAR	CESM2	historical	r111p1f1	Amon	tas	gn	NaN	v20190514	190001-194912	/mnt/lustre02/work/lk1017/CMIP6/data/CMIP6/CMI... http://esgf3.dkrz.de/thredds/dods/CMIP6/...	
2	CMIP	NCAR	CESM2	historical	r111p1f1	Amon	tas	gn	NaN	v20190514	195001-199912	/mnt/lustre02/work/lk1017/CMIP6/data/CMIP6/CMI... http://esgf3.dkrz.de/thredds/dods/CMIP6/...	
3	CMIP	NCAR	CESM2	historical	r111p1f1	Amon	tas	gn	NaN	v20190514	200001-201412	/mnt/lustre02/work/lk1017/CMIP6/data/CMIP6/CMI... http://esgf3.dkrz.de/thredds/dods/CMIP6/...	

```
Finally getting your hands on the data
[26]: datasets = models.to_dataset_dict()
--> The keys in the returned dictionary of datasets are constructed as follows:
'activity_id.institution_id.source_id.experiment_id.table_id.grid_label'
100.00% [1/1 00:00<00:00]
[30]: dset = datasets['CMIP.NCAR.CESM2.historical.Amon.gn']
[32]: dset
[32]: xarray.Dataset
- Dimensions: (lat: 192, lon: 288, member_id: 1, nbnd: 2, time: 1980)
- Coordinates:
  lat (lat) float64 -90.0 -89.06 -88.12 ... 89.06 90.0
  lon (lon) float64 0.0 1.25 2.5 ... 356.2 357.5 358.8
  time (time) object 1850-01-15 12:00:00 ... 2014-12-...
  member_id (member_id) <U9 'r111p1f1'
```

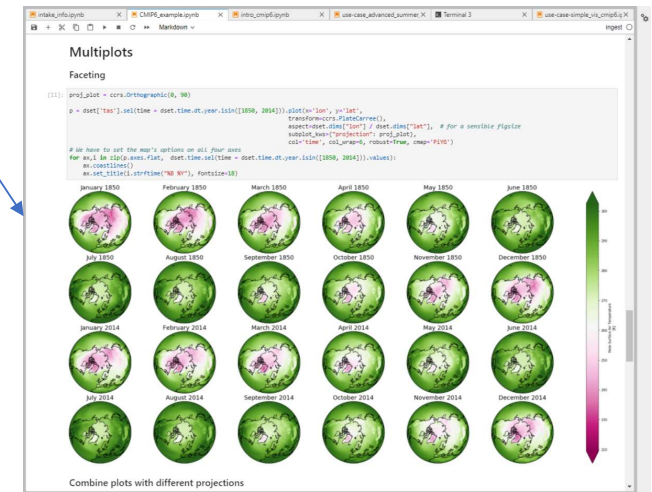
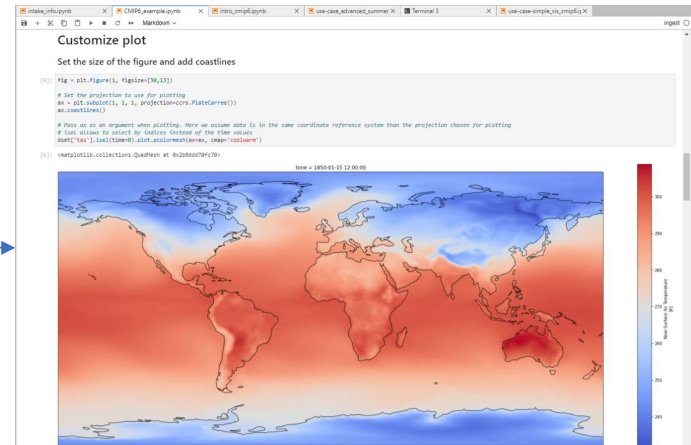
From the data to visualizations

Many useful basic visualization capabilities are already available as part of standard libraries (e.g. matplotlib)

Example notebooks also show more advanced interactive visualizations (e.g. using hvplot)

```
[3]: dset = xr.open_dataset(filename, decode_times=True, use_cftime=True)
dset
/home/dkr7k202015/miniconda3/envs/ingest/lib/python3.8/site-packages/xarray/conventions.py:512: SerializationWarning: variable 'tas' has multiple fill values (1e+20, 1e+20), decoding all values to NaN.
  new_vars[k] = decode_cf_variable(
[3]: xarray.Dataset
-----
Dimensions:      (lat: 192, lon: 288, nbind: 2, time: 1980)
Coordinates:
  lat            (lat) float64 -90.0 -89.06 -88.12 ... -89.06 90.0
  lon            (lon) float64  0.0 1.25 2.5 ... 356.2 357.5 358.8
  time           (time) object 1850-01-15 12:00:00 ... 2014-12-15
Data variables:
  tas            (time, lat, lon) float32 ...
  time_bnds     (time, nbind) object ...
  lat_bnds      (lat, nbind) float32 ...
  lon_bnds      (lon, nbind) float32 ...
Attributes: (45)

Get metadata corresponding to near-surface air temperature (tas)
[4]: print(dset['tas'])
xarray.DataArray 'tas' (time: 1980, lat: 192, lon: 288)
[189486880 values with dtype=float32]
Coordinates:
  * lat      (lat) float64 -90.0 -89.06 -88.12 -87.17 ... -87.17 88.12 89.06 90.0
  * lon      (lon) float64 0.0 1.25 2.5 3.75 5.0 ... 355.0 356.2 357.5 358.8
  * time     (time) object 1850-01-15 12:00:00 ... 2014-12-15 12:00:00
  ... time  ...
```



Example Notebooks



Demo and tutorial notebooks are freely accessible in the github repo at <https://github.com/IS-ENES-Data/Climate-data-analysis-service>

IS-ENES-Data / Climate-data-analysis-service

main 1 branch 0 tags

Go to file Add file Code

pull request #1 from MarcoKuluve/main 2 days ago 65 comments

File	Commit	Time
notebooks	work on comments	2 days ago
environment.yml	save intake selection	23 days ago
LICENSE	initial commit	2 months ago
README.md	update README.md	3 days ago
environment.yml	added copy of readme	7 days ago
make_kernel.sh	added copy of readme	7 days ago
spec-list.txt	added copy of readme	7 days ago

Using Notebooks for Climate Data Analysis

Welcome to the IS-ENES tutorials and use cases repository for the ENES Climate Analytics Services (ECAS) at DKRZ

In the "notebooks" folder here you can find Jupyter notebooks with coding examples showing how to use Big Data and High-Performance Computing software. Find more information on how to apply for the service and get a DKRZ account at the ECAS website.

Notebooks

- Multimodel Comparison of CMIP6 models**
 - use Python to select data in data-pool
 - use python-cdo and Xarray for computation.
- Frost Days climate index with CMIP6 models**
 - use Intake to search data in data-pool
 - use Xarray for computation.
- Tropical Nights climate index with CMIP6 models**
 - use Intake and Xarray.
- Summer Days climate index with CMIP6 models**
 - This is an advanced notebook. It requires additional installations steps ("Your own Jupyter kernel").
 - use Folium for maps.
 - use hvPlot for plots.

The Jupyter notebooks are meant to run in the Jupyterhub server of the German Climate Computing Center DKRZ which is an ESGF repository that hosts 4 petabytes of CMIP6 model data (more info on the data pool here).

Do not try to run these notebooks in your premise, which is also known as client-side computing. It will fail because you will not have direct access to the data pool. Direct access to the data pool is one of the main benefits of the server-side data-near computing demonstrated in these tutorials and use cases.

Quick start

You will only need a browser to install and run the above notebooks.

- Open the DKRZ Jupyterhub in your browser.
- Login with your DKRZ account (if you do not have one account yet, follow steps 1 and 2 in the service ECAS website).

DKRZ CMIP6_multimodel_example

Trusted Python 3 (using the module python/3.5.2) C

Edit View Insert Cell Kernel Widgets Help

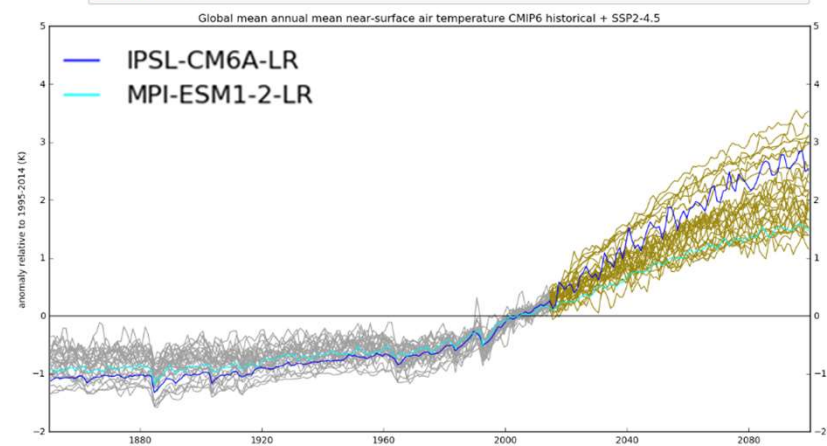
Run

```
In [1]: import xarray as xr
import intake
import cdo
```

Run a multimodel comparison in a supercomputer holding the data pool

We follow the original idea and code developed by Dr. Sebastian Milinski from the Max Planck Institute for Meteorology, MPI-M, a chapter scientist for Working Group 1 (chapter 4) of the IPCC AR6. The notebook was ported to jupyterhub.dkrz.de by Ralf Mueller, a developer at the German Climate Computing Center, DKRZ.

Import xarray, intake, and cdo

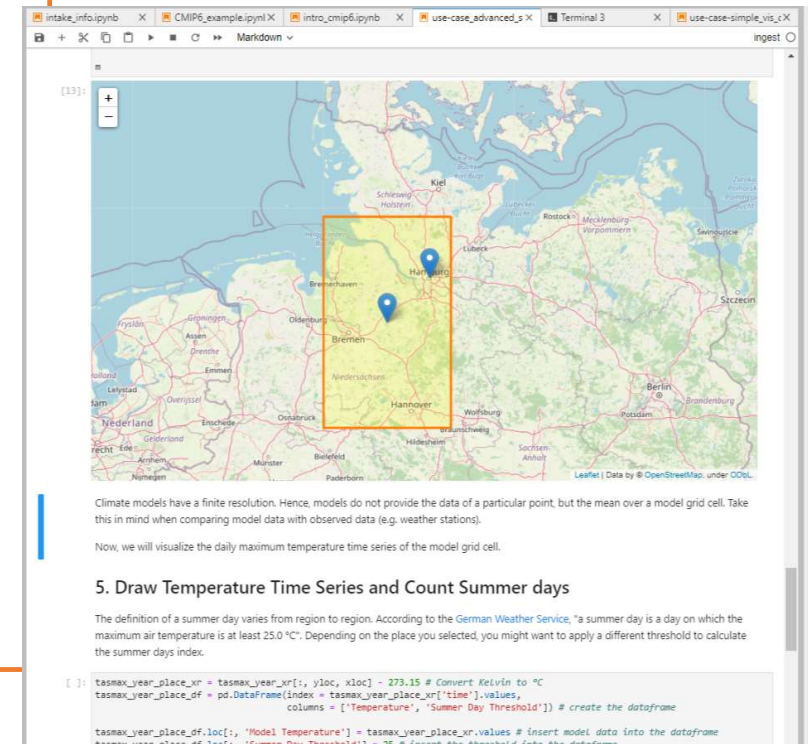


The IS-ENES climate analytics services:

Interested? Further information:

- **Climate Analytics service (ECAS):**
<https://portal.enes.org/data/data-metadata-service/climate-analytics-service>
- **Analysis platforms service application:**
<https://portal.enes.org/data/data-metadata-service/analysis-platforms>

- **Demos, use-cases, example jupyter notebooks:**
<https://github.com/IS-ENES-Data/Climate-data-analysis-service>



Climate models have a finite resolution. Hence, models do not provide the data of a particular point, but the mean over a model grid cell. Take this in mind when comparing model data with observed data (e.g. weather stations).

Now, we will visualize the daily maximum temperature time series of the model grid cell.

5. Draw Temperature Time Series and Count Summer days

The definition of a summer day varies from region to region. According to the [German Weather Service](#), "a summer day is a day on which the maximum air temperature is at least 25.0 °C". Depending on the place you selected, you might want to apply a different threshold to calculate the summer days index.

```
[ ]: tasmax_year_place_xr = tasmax_year_xr[:, yloc, xloc] - 273.15 # Convert Kelvin to °C
tasmax_year_place_of = pd.DataFrame(index = tasmax_year_place_xr['time'].values,
    columns = ['Temperature', 'Summer Day Threshold']) # create the dataframe

tasmax_year_place_of.loc[:, 'Model Temperature'] = tasmax_year_place_xr.values # insert model data into the dataframe
tasmax_year_place_of.loc[:, 'Summer Day Threshold'] = 25 # insert the threshold into the dataframe
```