

Exercise: CMIP6 - Plot Time Series of annual means

In this task, the global annual means of the CMIP6 variable `tas` are to be calculated and plotted.

Since one time series alone would be quite boring we want to generate a total of 4 time series of the scenarios

```
SSP 1-2.6
SSP 2-4.5
SSP 3-7.0
SSP 5-8.5
```

For more information about the scenarios see

<https://www.dkrz.de/en/communication/climate-simulations/cmip6-en/the-ssp-scenarios>

The data are stored in several separate files and are to be merged with CDO and then their field mean average calculated.

In the exercise you will:

1. Learn to use the netCDF input files
2. Process the data with CDO or xarray
3. Visualize the processed data using matplotlib
4. Create a plot for a poster using seaborn

An example plot of the scenarios of some more ensembles and their mean values can be found on the DKRZ page for CMIP6

<https://www.dkrz.de/de/kommunikation/klimasimulationen/cmip6-de/cmip6-aktivitaeten-am-dkrz-ueberblick>

Input Data

Historical: /pool/data/CMIP6/data/CMIP/MPI-M/MPI-ESM1-2-LR/historical/r1i1p1f1/Amon/tas/gn/v20190710/

Scenario SSP 1-2.6: /pool/data/CMIP6/data/ScenarioMIP/MPI-M/MPI-ESM1-2-LR/ssp126/r1i1p1f1/Amon/tas/gn/v20190710/

Scenario SSP 2-4.5: /pool/data/CMIP6/data/ScenarioMIP/MPI-M/MPI-ESM1-2-LR/ssp245/r1i1p1f1/Amon/tas/gn/v20190710/

Scenario SSP 3-7.0: /pool/data/CMIP6/data/ScenarioMIP/MPI-M/MPI-ESM1-2-LR/ssp370/r1i1p1f1/Amon/tas/gn/v20190710/

Scenario SSP 5-8.5: /pool/data/CMIP6/data/ScenarioMIP/MPI-M/MPI-ESM1-2-LR/ssp585/r1i1p1f1/Amon/tas/gn/v20190710/

CDO hints:

1. Merge files along time (mergetime)
2. Compute the annual mean (yearmean)
3. Compute the field means (fldmean)
4. Take care of the shapes
5. Important: clean temporary files

Xarray hints:

1. Open multiple files at once
2. Compute the weights (http://xarray.pydata.org/en/stable/examples/area_weighted_temperature.html)
3. Compute the annual mean (resample)
4. Compute the field means (mean)

Solution

Import CDO module

```
from cdo import *  
  
cdo = Cdo()
```

Import Matplotlib module

```
from matplotlib import pyplot as plt  
  
%matplotlib inline
```

Let's see what is in the ssp126 data directory

There are two ways to do it:

```
%%bash  
ls /pool/data/CMIP6/data/ScenarioMIP/MPI-M/MPI-ESM1-2-LR/ssp126/r1i1p1f1/Amon/tas/gn/v20190710/
```

or

```
!ls /pool/data/CMIP6/data/ScenarioMIP/MPI-M/MPI-ESM1-2-  
LR/ssp126/r1i1p1f1/Amon/tas/gn/v20190710/
```

The result is the same.

```
%%bash  
ls /pool/data/CMIP6/data/ScenarioMIP/MPI-M/MPI-ESM1-2-  
LR/ssp126/r1i1p1f1/Amon/tas/gn/v20190710/
```

```
tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_201501-203412.nc  
tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_203501-205412.nc  
tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_205501-207412.nc  
tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_207501-209412.nc  
tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_209501-210012.nc
```

```
!ls /pool/data/CMIP6/data/ScenarioMIP/MPI-M/MPI-ESM1-2-  
LR/ssp126/r1i1p1f1/Amon/tas/gn/v20190710/
```

```
tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_201501-203412.nc  
tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_203501-205412.nc  
tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_205501-207412.nc  
tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_207501-209412.nc  
tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_209501-210012.nc
```

```
import os  
os.listdir('/pool/data/CMIP6/data/ScenarioMIP/MPI-M/MPI-ESM1-2-  
LR/ssp126/r1i1p1f1/Amon/tas/gn/v20190710/')
```

```
['tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_209501-210012.nc',  
 'tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_207501-209412.nc',  
 'tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_203501-205412.nc',  
 'tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_201501-203412.nc',  
 'tas_Amon_MPI-ESM1-2-LR_ssp126_r1i1p1f1_gn_205501-207412.nc']
```

Input files

For the sake of simplicity, we take the full path of the directories.

```
data_ssp126 = '/pool/data/CMIP6/data/ScenarioMIP/MPI-M/MPI-ESM1-2-
LR/ssp126/r1i1p1f1/Amon/tas/gn/v20190710/tas_*.nc'
data_ssp245 = '/pool/data/CMIP6/data/ScenarioMIP/MPI-M/MPI-ESM1-2-
LR/ssp245/r1i1p1f1/Amon/tas/gn/v20190710/tas_*.nc'
data_ssp370 = '/pool/data/CMIP6/data/ScenarioMIP/MPI-M/MPI-ESM1-2-
LR/ssp370/r1i1p1f1/Amon/tas/gn/v20190710/tas_*.nc'
data_ssp585 = '/pool/data/CMIP6/data/ScenarioMIP/MPI-M/MPI-ESM1-2-
LR/ssp585/r1i1p1f1/Amon/tas/gn/v20190710/tas_*.nc'
```

```
#cdo.sinfon(input=data_ssp126)
```

Compute the field mean of the yearly mean

Remember that CDO commands are written from right to left!

We want to do

1. merge all files of a scenario respecting the time dimension
2. compute the yearly mean
3. compute the field mean
4. assign the output time series to a variable of type xarray.DataArray

```
tas126_ymean = cdo.fldmean(input='-yearmean -mergetime ' + data_ssp126,
returnXArray='tas')
tas245_ymean = cdo.fldmean(input='-yearmean -mergetime ' + data_ssp245,
returnXArray='tas')
tas370_ymean = cdo.fldmean(input='-yearmean -mergetime ' + data_ssp370,
returnXArray='tas')
tas585_ymean = cdo.fldmean(input='-yearmean -mergetime ' + data_ssp585,
returnXArray='tas')
```

```
tas585_ymean.values.flatten()
```

```
array([287.70435, 287.73053, 287.76355, 287.89386, 287.72226, 287.56876,
      287.82382, 288.0481 , 288.0251 , 287.9403 , 288.0292 , 287.88144,
      288.01163, 288.13986, 287.9492 , 287.87506, 288.07248, 288.0994 ,
      288.04593, 288.06824, 288.01892, 288.06216, 288.22488, 288.3303 ,
      288.22784, 288.477 , 288.41583, 288.48923, 288.69855, 288.65036,
      288.45996, 288.60834, 288.58353, 288.55496, 288.6775 , 288.7628 ,
      288.67184, 288.78217, 288.8452 , 289.073 , 288.87112, 288.81064,
      288.90036, 289.01886, 289.08652, 289.15994, 289.29324, 289.26648,
```

```
289.18594, 289.27737, 289.44357, 289.47787, 289.32672, 289.378 ,
289.54367, 289.97443, 289.76486, 289.50275, 289.52002, 289.8608 ,
290.16742, 289.96774, 289.9263 , 290.06207, 290.0172 , 290.028 ,
290.08728, 290.10663, 290.37936, 290.37994, 290.40854, 290.49976,
290.50775, 290.63763, 290.59042, 290.55408, 290.6601 , 290.61914,
290.7049 , 290.7331 , 290.79883, 290.77448, 290.87228, 291.2188 ,
291.13248, 290.88776], dtype=float32)
```

Keep an eye on the data shapes

Printing the variables shape shows that we have 3 shapes which is a problem for matplotlib in our case.

```
print(tas126_ymean.shape)
print(tas245_ymean.shape)
print(tas370_ymean.shape)
print(tas585_ymean.shape)

(86, 1, 1)
(86, 1, 1)
(86, 1, 1)
(86, 1, 1)
```

We can remove the lat and lon dimensions with size 1 using array slicing.

```
data126 = tas126_ymean[:,0,0]
data245 = tas245_ymean[:,0,0]
data370 = tas370_ymean[:,0,0]
data585 = tas585_ymean[:,0,0]
```

To prevent this step use CDO's option `--reduce_dims` within the CDO command line. In addition we use explicitly the data output type float64 for the comparison with the output of the xarray computation.

For example

```
tas126_ymean = cdo.fldmean(options='-b F64 --reduce_dim',
                           input='-yearmean -mergetime ' + data_ssp126,
                           returnXArray='tas')
```

```
cdo.mergetime(input=data_ssp126, returnXArray='tas')
```

```
:root { --xr-font-color0: var(--jp-content-font-color0, rgba(0, 0, 0, 1)); --xr-font-color2: var(--jp-content-font-
color2, rgba(0, 0, 0, 0.54)); --xr-font-color3: var(--jp-content-font-color3, rgba(0, 0, 0, 0.38)); --xr-border-color:
var(--jp-border-color2, #e0e0e0); --xr-disabled-color: var(--jp-layout-color3, #bdbdbd); --xr-background-
```

```
color: var(--jp-layout-color0, white); --xr-background-color-row-even: var(--jp-layout-color1, white); --xr-
background-color-row-odd: var(--jp-layout-color2, #eeeeee); }

html[theme=dark], body.vscode-dark { --xr-font-color0: rgba(255, 255, 255, 1); --xr-font-color2: rgba(255, 255,
255, 0.54); --xr-font-color3: rgba(255, 255, 255, 0.38); --xr-border-color: #1F1F1F; --xr-disabled-color:
#515151; --xr-background-color: #111111; --xr-background-color-row-even: #111111; --xr-background-
color-row-odd: #313131; }

.xr-wrap { display: block !important; min-width: 300px; max-width: 700px; }

.xr-text-repr-fallback { /* fallback to plain text repr when CSS is not injected (untrusted notebook) */ display:
none; }

.xr-header { padding-top: 6px; padding-bottom: 6px; margin-bottom: 4px; border-bottom: solid 1px var(--xr-
border-color); }

.xr-header > div, .xr-header > ul { display: inline; margin-top: 0; margin-bottom: 0; }

.xr-obj-type, .xr-array-name { margin-left: 2px; margin-right: 10px; }

.xr-obj-type { color: var(--xr-font-color2); }

.xr-sections { padding-left: 0 !important; display: grid; grid-template-columns: 150px auto auto 1fr 20px 20px;
}

.xr-section-item { display: contents; }

.xr-section-item input { display: none; }

.xr-section-item input + label { color: var(--xr-disabled-color); }

.xr-section-item input:enabled + label { cursor: pointer; color: var(--xr-font-color2); }

.xr-section-item input:enabled + label:hover { color: var(--xr-font-color0); }

.xr-section-summary { grid-column: 1; color: var(--xr-font-color2); font-weight: 500; }

.xr-section-summary > span { display: inline-block; padding-left: 0.5em; }

.xr-section-summary-in:disabled + label { color: var(--xr-font-color2); }

.xr-section-summary-in + label:before { display: inline-block; content: '►'; font-size: 11px; width: 15px; text-
align: center; }

.xr-section-summary-in:disabled + label:before { color: var(--xr-disabled-color); }

.xr-section-summary-in:checked + label:before { content: '▼'; }

.xr-section-summary-in:checked + label > span { display: none; }

.xr-section-summary, .xr-section-inline-details { padding-top: 4px; padding-bottom: 4px; }

.xr-section-inline-details { grid-column: 2 / -1; }

.xr-section-details { display: none; grid-column: 1 / -1; margin-bottom: 5px; }
```

```
.xr-section-summary-in:checked ~ .xr-section-details { display: contents; }

.xr-array-wrap { grid-column: 1 / -1; display: grid; grid-template-columns: 20px auto; }

.xr-array-wrap > label { grid-column: 1; vertical-align: top; }

.xr-preview { color: var(--xr-font-color3); }

.xr-array-preview, .xr-array-data { padding: 0 5px !important; grid-column: 2; }

.xr-array-data, .xr-array-in:checked ~ .xr-array-preview { display: none; }

.xr-array-in:checked ~ .xr-array-data, .xr-array-preview { display: inline-block; }

.xr-dim-list { display: inline-block !important; list-style: none; padding: 0 !important; margin: 0; }

.xr-dim-list li { display: inline-block; padding: 0; margin: 0; }

.xr-dim-list:before { content: '('; }

.xr-dim-list:after { content: ')'; }

.xr-dim-list li:not(:last-child):after { content: ','; padding-right: 5px; }

.xr-has-index { font-weight: bold; }

.xr-var-list, .xr-var-item { display: contents; }

.xr-var-item > div, .xr-var-item label, .xr-var-item > .xr-var-name span { background-color: var(--xr-background-color-row-even); margin-bottom: 0; }

.xr-var-item > .xr-var-name:hover span { padding-right: 5px; }

.xr-var-list > li:nth-child(odd) > div, .xr-var-list > li:nth-child(odd) > label, .xr-var-list > li:nth-child(odd) > .xr-var-name span { background-color: var(--xr-background-color-row-odd); }

.xr-var-name { grid-column: 1; }

.xr-var-dims { grid-column: 2; }

.xr-var-dtype { grid-column: 3; text-align: right; color: var(--xr-font-color2); }

.xr-var-preview { grid-column: 4; }

.xr-var-name, .xr-var-dims, .xr-var-dtype, .xr-preview, .xr-attrs dt { white-space: nowrap; overflow: hidden; text-overflow: ellipsis; padding-right: 10px; }

.xr-var-name:hover, .xr-var-dims:hover, .xr-var-dtype:hover, .xr-attrs dt:hover { overflow: visible; width: auto; z-index: 1; }

.xr-var-attrs, .xr-var-data { display: none; background-color: var(--xr-background-color) !important; padding-bottom: 5px !important; }

.xr-var-attrs-in:checked ~ .xr-var-attrs, .xr-var-data-in:checked ~ .xr-var-data { display: block; }
```

```
.xr-var-data > table { float: right; }
```

```
.xr-var-name span, .xr-var-data, .xr-attrs { padding-left: 25px !important; }
```

```
.xr-attrs, .xr-var-attrs, .xr-var-data { grid-column: 1 / -1; }
```

```
dl.xr-attrs { padding: 0; margin: 0; display: grid; grid-template-columns: 125px auto; }
```

```
.xr-attrs dt, .xr-attrs dd { padding: 0; margin: 0; float: left; padding-right: 10px; width: auto; }
```

```
.xr-attrs dt { font-weight: normal; grid-column: 1; }
```

```
.xr-attrs dt:hover span { display: inline-block; background: var(--xr-background-color); padding-right: 10px; }
```

```
.xr-attrs dd { grid-column: 2; white-space: pre-wrap; word-break: break-all; }
```

```
.xr-icon-database, .xr-icon-file-text2 { display: inline-block; vertical-align: middle; width: 1em; height: 1.5em !important; stroke-width: 0; stroke: currentColor; fill: currentColor; }
```

```
<xarray.DataArray 'tas' (time: 1032, lat: 96, lon: 192)>
[19021824 values with dtype=float32]
Coordinates:
```

- time (time) datetime64[ns] 2015-01-16T12:00:00 ... 2100-12-16T12:00:00
- lon (lon) float64 0.0 1.875 3.75 5.625 7.5 ... 352.5 354.4 356.2 358.1
- lat (lat) float64 -88.57 -86.72 -84.86 -83.0 ... 83.0 84.86 86.72 88.57
- height float64 2.0

Attributes:

```
standard_name:    air_temperature
long_name:        Near-Surface Air Temperature
units:            K
CDI_grid_type:    gaussian
CDI_grid_num_LPE: 48
comment:          near-surface (usually, 2 meter) air temperature
cell_methods:     area: time: mean
cell_measures:     area: areacella
history:          2019-10-29T16:19:11Z altered by CMOR: Treated
scalar d...
```

```
data126 = cdo.fldmean(options='-b F64 --reduce_dim', input='-yearmean -mergetime '
+ data_ssp126, returnXArray='tas')
data245 = cdo.fldmean(options='-b F64 --reduce_dim', input='-yearmean -mergetime '
+ data_ssp245, returnXArray='tas')
```



```
data370 = cdo.fldmean(options='-b F64 --reduce_dim', input='-yearmean -mergetime '
+ data_ssp370, returnXArray='tas')
data585 = cdo.fldmean(options='-b F64 --reduce_dim', input='-yearmean -mergetime '
+ data_ssp585, returnXArray='tas')
```

```
data585.shape
```

```
(86,)
```

Create the time series plot of scenarios

In the next step

1. define the plotting figure and axis
2. draw all 4 scenario time series
3. draw grid lines
4. add a title string
5. define the axis titles
6. add a legend

Alternative without using the x-list

```
fig, ax = plt.subplots(figsize=(8,4))

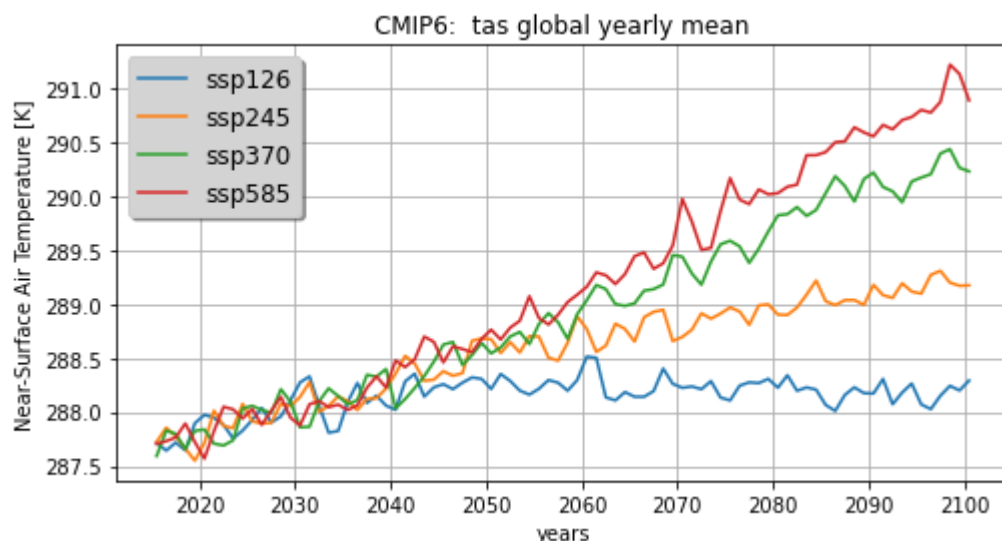
# draw scenario time series
ax.plot(data126.time, data126, label='ssp126')
ax.plot(data245.time, data245, label='ssp245')
ax.plot(data370.time, data370, label='ssp370')
ax.plot(data585.time, data585, label='ssp585')

# draw grid lines
plt.grid()

# add a title string
ax.set_title('CMIP6: tas global yearly mean')

# set axis labels
ax.set_xlabel('years')
ax.set_ylabel(data126.long_name+' [' +data126.units+']')

#add a legend
legend = ax.legend(loc='upper left', shadow=True, fontsize=12)
legend.get_frame().set_facecolor('lightgray')
```



Delete temporary files

```
cdo.cleanTempDir()
```

Xarray not CDO

And now let's see how much more effort we have to put in with xarray as opposed to CDO.

Load the packages xarray and numpy.

```
import xarray as xr
import numpy as np
```

Read the files for each scenario into an xarray.DataSet.

```
ds126 = xr.open_mfdataset(data_ssp126)
ds245 = xr.open_mfdataset(data_ssp245)
ds370 = xr.open_mfdataset(data_ssp370)
ds585 = xr.open_mfdataset(data_ssp585)
```

```
ds370
```

```
:root { --xr-font-color0: var(--jp-content-font-color0, rgba(0, 0, 0, 1)); --xr-font-color2: var(--jp-content-font-
color2, rgba(0, 0, 0, 0.54)); --xr-font-color3: var(--jp-content-font-color3, rgba(0, 0, 0, 0.38)); --xr-border-color:
var(--jp-border-color2, #e0e0e0); --xr-disabled-color: var(--jp-layout-color3, #bdbdbd); --xr-background-
color: var(--jp-layout-color0, white); --xr-background-color-row-even: var(--jp-layout-color1, white); --xr-
background-color-row-odd: var(--jp-layout-color2, #eeeeee); }
```

```
html[theme=dark], body.vscode-dark { --xr-font-color0: rgba(255, 255, 255, 1); --xr-font-color2: rgba(255, 255, 255, 0.54); --xr-font-color3: rgba(255, 255, 255, 0.38); --xr-border-color: #1F1F1F; --xr-disabled-color: #515151; --xr-background-color: #111111; --xr-background-color-row-even: #111111; --xr-background-color-row-odd: #313131; }

.xr-wrap { display: block !important; min-width: 300px; max-width: 700px; }

.xr-text-repr-fallback { /* fallback to plain text repr when CSS is not injected (untrusted notebook) */ display: none; }

.xr-header { padding-top: 6px; padding-bottom: 6px; margin-bottom: 4px; border-bottom: solid 1px var(--xr-border-color); }

.xr-header > div, .xr-header > ul { display: inline; margin-top: 0; margin-bottom: 0; }

.xr-obj-type, .xr-array-name { margin-left: 2px; margin-right: 10px; }

.xr-obj-type { color: var(--xr-font-color2); }

.xr-sections { padding-left: 0 !important; display: grid; grid-template-columns: 150px auto auto 1fr 20px 20px; }

.xr-section-item { display: contents; }

.xr-section-item input { display: none; }

.xr-section-item input + label { color: var(--xr-disabled-color); }

.xr-section-item input:enabled + label { cursor: pointer; color: var(--xr-font-color2); }

.xr-section-item input:enabled + label:hover { color: var(--xr-font-color0); }

.xr-section-summary { grid-column: 1; color: var(--xr-font-color2); font-weight: 500; }

.xr-section-summary > span { display: inline-block; padding-left: 0.5em; }

.xr-section-summary-in:disabled + label { color: var(--xr-font-color2); }

.xr-section-summary-in + label:before { display: inline-block; content: '►'; font-size: 11px; width: 15px; text-align: center; }

.xr-section-summary-in:disabled + label:before { color: var(--xr-disabled-color); }

.xr-section-summary-in:checked + label:before { content: '▼'; }

.xr-section-summary-in:checked + label > span { display: none; }

.xr-section-summary, .xr-section-inline-details { padding-top: 4px; padding-bottom: 4px; }

.xr-section-inline-details { grid-column: 2 / -1; }

.xr-section-details { display: none; grid-column: 1 / -1; margin-bottom: 5px; }

.xr-section-summary-in:checked ~ .xr-section-details { display: contents; }
```

```
.xr-array-wrap { grid-column: 1 / -1; display: grid; grid-template-columns: 20px auto; }

.xr-array-wrap > label { grid-column: 1; vertical-align: top; }

.xr-preview { color: var(--xr-font-color3); }

.xr-array-preview, .xr-array-data { padding: 0 5px !important; grid-column: 2; }

.xr-array-data, .xr-array-in:checked ~ .xr-array-preview { display: none; }

.xr-array-in:checked ~ .xr-array-data, .xr-array-preview { display: inline-block; }

.xr-dim-list { display: inline-block !important; list-style: none; padding: 0 !important; margin: 0; }

.xr-dim-list li { display: inline-block; padding: 0; margin: 0; }

.xr-dim-list:before { content: '('; }

.xr-dim-list:after { content: ')'; }

.xr-dim-list li:not(:last-child):after { content: ','; padding-right: 5px; }

.xr-has-index { font-weight: bold; }

.xr-var-list, .xr-var-item { display: contents; }

.xr-var-item > div, .xr-var-item label, .xr-var-item > .xr-var-name span { background-color: var(--xr-background-color-row-even); margin-bottom: 0; }

.xr-var-item > .xr-var-name:hover span { padding-right: 5px; }

.xr-var-list > li:nth-child(odd) > div, .xr-var-list > li:nth-child(odd) > label, .xr-var-list > li:nth-child(odd) > .xr-var-name span { background-color: var(--xr-background-color-row-odd); }

.xr-var-name { grid-column: 1; }

.xr-var-dims { grid-column: 2; }

.xr-var-dtype { grid-column: 3; text-align: right; color: var(--xr-font-color2); }

.xr-var-preview { grid-column: 4; }

.xr-var-name, .xr-var-dims, .xr-var-dtype, .xr-preview, .xr-attrs dt { white-space: nowrap; overflow: hidden; text-overflow: ellipsis; padding-right: 10px; }

.xr-var-name:hover, .xr-var-dims:hover, .xr-var-dtype:hover, .xr-attrs dt:hover { overflow: visible; width: auto; z-index: 1; }

.xr-var-attrs, .xr-var-data { display: none; background-color: var(--xr-background-color) !important; padding-bottom: 5px !important; }

.xr-var-attrs-in:checked ~ .xr-var-attrs, .xr-var-data-in:checked ~ .xr-var-data { display: block; }

.xr-var-data > table { float: right; }
```

```
.xr-var-name span, .xr-var-data, .xr-attrs { padding-left: 25px !important; }

.xr-attrs, .xr-var-attrs, .xr-var-data { grid-column: 1 / -1; }

dl.xr-attrs { padding: 0; margin: 0; display: grid; grid-template-columns: 125px auto; }

.xr-attrs dt, .xr-attrs dd { padding: 0; margin: 0; float: left; padding-right: 10px; width: auto; }

.xr-attrs dt { font-weight: normal; grid-column: 1; }

.xr-attrs dt:hover span { display: inline-block; background: var(--xr-background-color); padding-right: 10px; }

.xr-attrs dd { grid-column: 2; white-space: pre-wrap; word-break: break-all; }

.xr-icon-database, .xr-icon-file-text2 { display: inline-block; vertical-align: middle; width: 1em; height: 1.5em
!important; stroke-width: 0; stroke: currentColor; fill: currentColor; }
```

```
<xarray.Dataset>
Dimensions:      (time: 1032, bnds: 2, lat: 96, lon: 192)
Coordinates:

•
time            (time) datetime64[ns] 2015-01-16T12:00:00 ... 2100-12-
16T12:00:00

•
lat             (lat) float64 -88.57 -86.72 -84.86 -83.0 ... 84.86 86.72
88.57

•
lon             (lon) float64 0.0 1.875 3.75 5.625 ... 352.5 354.4 356.2
358.1
height          float64 2.0
Dimensions without coordinates: bnds
Data variables:
time_bnds       (time, bnds) datetime64[ns] dask.array<chunksize=(240, 2),
meta=np.ndarray>
lat_bnds        (time, lat, bnds) float64 dask.array<chunksize=(240, 96,
2), meta=np.ndarray>
lon_bnds        (time, lon, bnds) float64 dask.array<chunksize=(240, 192,
2), meta=np.ndarray>
tas             (time, lat, lon) float32 dask.array<chunksize=(240, 96,
192), meta=np.ndarray>
Attributes: (12/47)
Conventions:    CF-1.7 CMIP-6.2
```

```
activity_id:      ScenarioMIP AerChemMIP
branch_method:    standard
branch_time_in_child: 60265.0
branch_time_in_parent: 60265.0
contact:          cmip6-mpi-esm@dkrz.de
...              ...
title:            MPI-ESM1-2-LR output prepared for CMIP6
variable_id:      tas
variant_label:    r1i1p1f1
license:          CMIP6 model data produced by MPI-M is licensed
un...
cmor_version:     3.5.0
tracking_id:      hdl:21.14100/2d98a4ae-be7f-4b30-869f-
5a3dbc8d7bab
```