

Python-Anwendung_Ozean_und_Klima

June 9, 2022

1 Python-Kurs: Anwendung Ozean- und Klimaforschung

von Markus Reinert, 2022-06-07

Dieses Notebook zeigt, wie man in Python mit Daten aus der Ozean- und Klimaforschung arbeitet. Wir verwenden dafür einen Datensatz vom europäischen Copernicus-Projekt, der von <https://cds.climate.copernicus.eu/> heruntergeladen wurde.

1.1 Lade die benötigte Bibliothek

XArray ist eine Python-Bibliothek, mit der man u.a. Datensätze im NetCDF-Format laden und mit diesen arbeiten kann.

```
[1]: import xarray as xr
```

1.2 Lade den Datensatz

SST steht für *sea surface temperature*, also die Oberflächentemperatur des Meeres.

```
[3]: ds_SST = xr.load_dataset("../data/dataset_small.nc")
ds_SST
```

```
[3]: <xarray.Dataset>
Dimensions:                (time: 1, lon: 720, lat: 360)
Coordinates:
  * time                    (time) datetime64[ns] 2017-01-01T12:00:00
  * lon                     (lon) float64 0.0 0.5 1.0 1.5 ... 358.5 359.0 359.5
  * lat                     (lat) float64 -89.75 -89.25 -88.75 ... 89.25 89.75
Data variables:
  analysed_sst              (time, lat, lon) float32 nan nan nan ... 271.4 271.4
  analysis_uncertainty      (time, lat, lon) float32 nan nan nan ... 1.81 1.81
  sea_ice_fraction          (time, lat, lon) float32 nan nan nan ... 0.95 0.95
  mask                      (time, lat, lon) float32 2.0 2.0 2.0 ... 9.0 9.0 9.0
Attributes: (12/62)
  CDI:                      Climate Data Interface version 1.9.9rc1 ...
  history:                  Mon Jun 06 23:10:39 2022: cdo remapbil,r...
  source:                   AVHRR19_G-C3S-L3U-ICDR-v2.0 AVHRRMTA_G-C...
  institution:              C3S
  Conventions:              CF-1.5, Unidata Observation Dataset v1.0
```

```

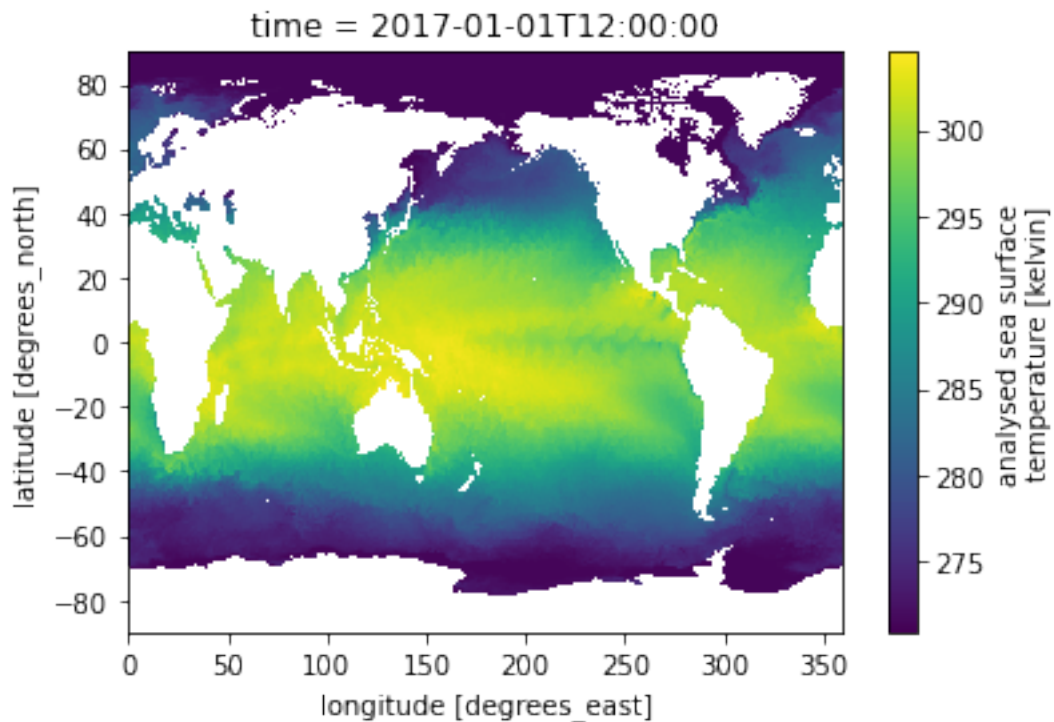
title:                C3S SST L4 product
...
publisher_email:     copernicus-support@ecmwf.int
processing_level:    L4
cdm_data_type:       grid
product_specification_version: SST_CCI-PSD-UKMO-201-Issue-H
contact:             http://copernicus-support.ecmwf.int
CDO:                Climate Data Operators version 1.9.9rc1 ...

```

1.3 Stelle die Daten grafisch dar

```
[4]: ds_SST.analysed_sst.plot()
```

```
[4]: <matplotlib.collections.QuadMesh at 0x7f2c2f8229d0>
```



1.4 Analysiere die Daten

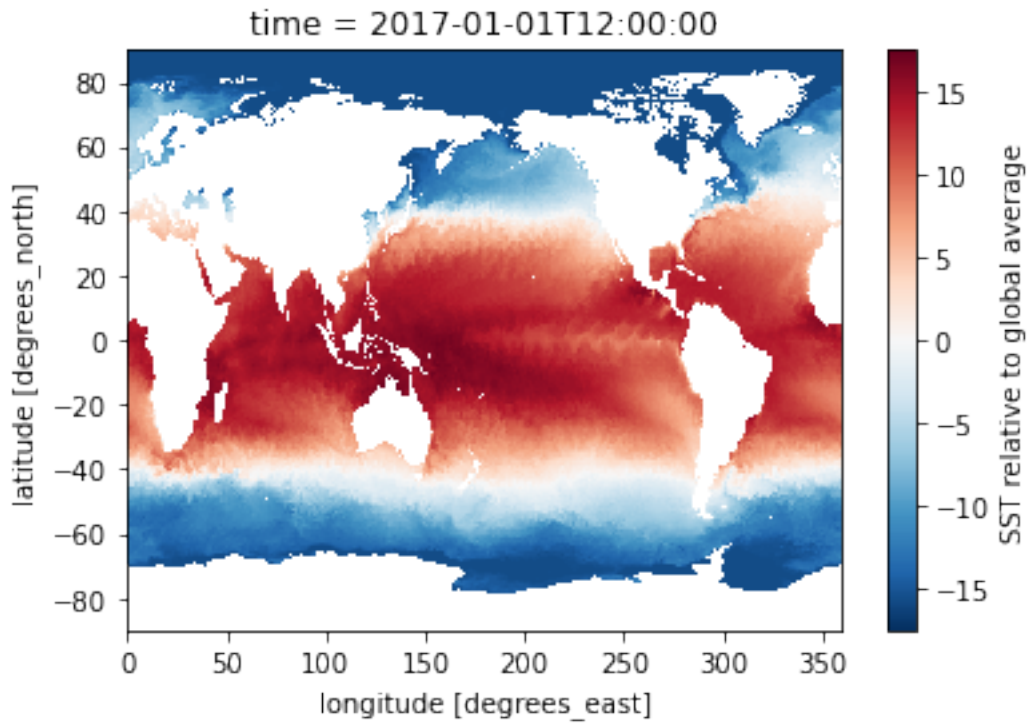
```
[5]: mean_temperature = ds_SST.analysed_sst.mean()
print(f"The global mean SST was {mean_temperature.values - 273.15:.2f} °C on_{
↳{ds_SST.time[0].values}.")
```

The global mean SST was 13.84 °C on 2017-01-01T12:00:00.000000000.

```
[6]: T_anom = ds_SST.analysed_sst - mean_temperature
T_anom.name = "SST relative to global average"

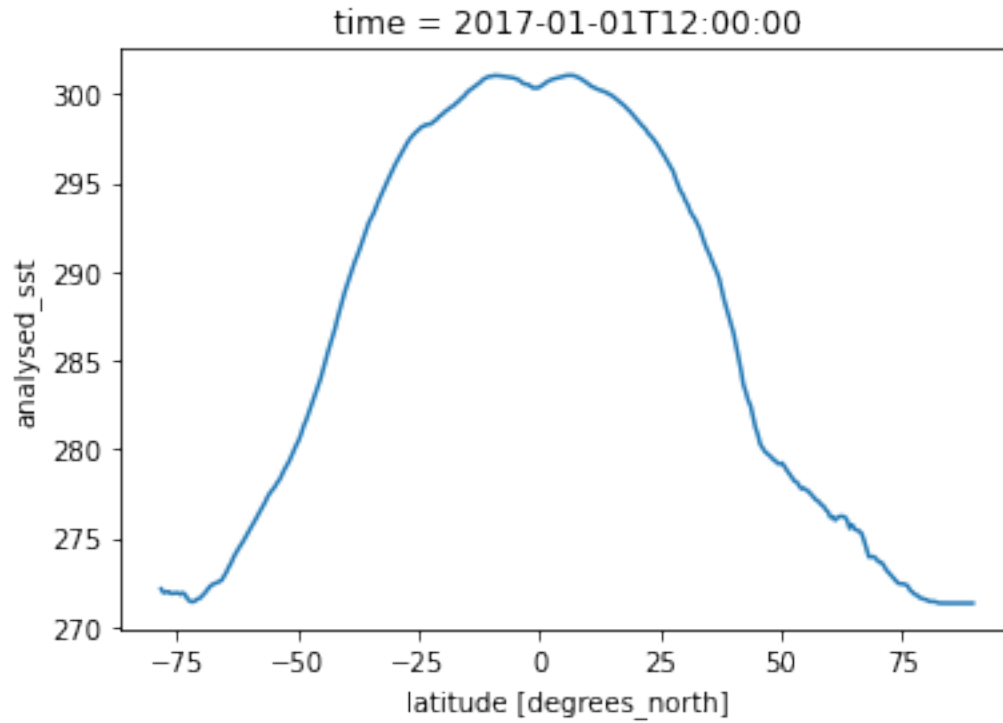
T_anom.plot()
```

```
[6]: <matplotlib.collections.QuadMesh at 0x7f2c2d5ee3d0>
```



```
[7]: ds_SST.analysed_sst.mean("lon").plot()
print("Average sea surface temperature per latitude circle:")
```

Average sea surface temperature per latitude circle:



1.5 Wähle Daten um Nordamerika und im Nordwest-Atlantik aus

```
[8]: ds_SST.analysed_sst.sel(lat=slice(20, 60), lon=slice(230, 330)).plot()
```

```
[8]: <matplotlib.collections.QuadMesh at 0x7f2c2d485b50>
```

