Python_course_5_oceanography

May 14, 2021

1 Python course 5: Introduction to tools for oceanography

by Markus Reinert, 2021-05-14

In this course, we use climate datasets from Copernicus, that can downloaded for free from their website at https://cds.climate.copernicus.eu/#!/home.

For datasets that are too big to fit into memory, you can use Dask: https://xarray.pydata.org/en/stable/user-guide/dask.html

1.1 Import the necessary modules

These modules, we already know:

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import cmocean
```

These modules are new today:

[2]:
```
import gsw

import xarray as xr

import cartopy.crs as ccrs
import cartopy.feature as cfeature
```

crs stands for Coordinate Reference System.

1.2 Configure the visualization

[3]:
```
plt.rcParams["font.size"] = 14
```

1.3 Example 1: sea surface temperature

1.3.1 Load the dataset

[4]:
```
ds_SST = xr.load_dataset("../Data/
 ↪dataset-satellite-sea-surface-temperature-f9354174-5df6-438e-913a-63ae8bc7fc02/
 ↪20170101120000-C3S-L4_GHRSST-SSTdepth-OSTIA-GLOB_ICDR2.0-v02.0-fv01.0.nc")
ds_SST
```

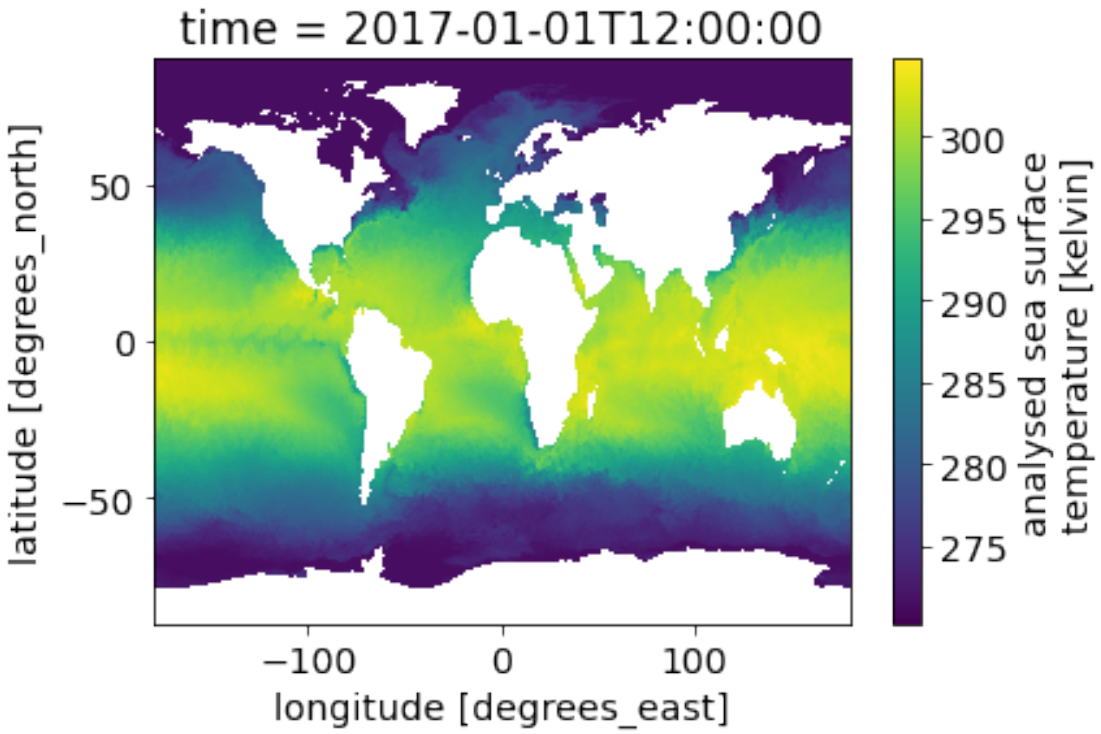
[4]:
```
<xarray.Dataset>
Dimensions:                 (lat: 3600, lon: 7200, time: 1)
Coordinates:
  * time                    (time) datetime64[ns] 2017-01-01T12:00:00
  * lat                     (lat) float32 -89.975 -89.925 … 89.925 89.975
  * lon                     (lon) float32 -179.975 -179.925 … 179.925 179.975
Data variables:
    analysed_sst            (time, lat, lon) float32 nan nan nan … 271.35 271.35
    analysis_uncertainty    (time, lat, lon) float32 nan nan … 0.72999996
    sea_ice_fraction        (time, lat, lon) float32 nan nan nan … 0.94 0.94
    mask                    (time, lat, lon) float32 2.0 2.0 2.0 … 9.0 9.0 9.0
Attributes:
    Conventions:                    CF-1.5, Unidata Observation Dataset v1.0
    title:                          C3S SST L4 product
    summary:                        OSTIA L4 product from the C3S project, p…
    references:                     http://www.esa-sst-cci.org
    institution:                    C3S
    history:                        Created using OSTIA reanalysis system v3.0
    comment:                        These data were produced by the Met Offi…
    license:                        Creative Commons Licence by attribution …
    id:                             OSTIA-C3S-L4-GLOB-v2.0
    naming_authority:               org.ghrsst
    product_version:                2.0
    uuid:                           7fdf2639-26e5-4d4f-a60e-0bcfc9744204
    tracking_id:                    7fdf2639-26e5-4d4f-a60e-0bcfc9744204
    gds_version_id:                 2.0
    netcdf_version_id:              4.2.1.1 of Oct 19 2012 14:25:16
    date_created:                   20180620T065809Z
    creation_date:                  2018-06-20T06:58:09Z
    file_quality_level:             3
    spatial_resolution:             0.05 degree
    start_time:                     20170101T000000Z
    time_coverage_start:            20170101T000000Z
    stop_time:                      20170102T000000Z
    time_coverage_end:              20170102T000000Z
    time_coverage_duration:         P1D
    time_coverage_resolution:       P1D
    northernmost_latitude:          90.0
    geospatial_lat_max:             90.0
```

```
southernmost_latitude:          -90.0
gepspatial_lat_min:             -90.0
easternmost_longitude:          180.00002
geospatial_lon_max:             180.00002
westernmost_longitude:          -180.0
geospatial_lon_min:             -180.0
geospatial_vertical_min:        -0.2
geospatial_vertical_max:        -0.2
source:                         AVHRR19_G-C3S-L3U-ICDR-v2.0 AVHRRMTA_G-C…
platform:                       NOAA-19, MetOpA, Sentinel-3A
sensor:                         AVHRR_GAC, SLSTR
Metadata_Conventions:           Unidata Dataset Discovery v1.0
metadata_link:                  http://www.esa-cci.org
keywords:                       Oceans > Ocean Temperature > Sea Surface…
keywords_vocabulary:            NASA Global Change Master Directory (GCM…
standard_name_vocabulary:       NetCDF Climate and Forecast (CF) Metadat…
geospatial_lat_units:           degrees_north
geospatial_lat_resolution:      0.05
geospatial_lon_units:           degrees_east
geospatial_lon_resolution:      0.05
acknowledgment:                 Funded by the Copernicus Climate Change …
creator_name:                   Copernicus Climate Change Service (C3S)
creator_email:                  science.leader@esa-sst-cci.org
creator_processing_institution: These data were produced on the JASMIN i…
creator_url:                    https://climate.copernicus.eu/
project:                        Copernicus Climate Change Service (C3S)
publisher_name:                 Copernicus Climate Data Store
publisher_url:                  https://climate.copernicus.eu/
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
```

1.3.2 Look at the data

[5]:
```
ds_SST.analysed_sst.plot()
```

[5]:
```
<matplotlib.collections.QuadMesh at 0x7f460bc75dd0>
```



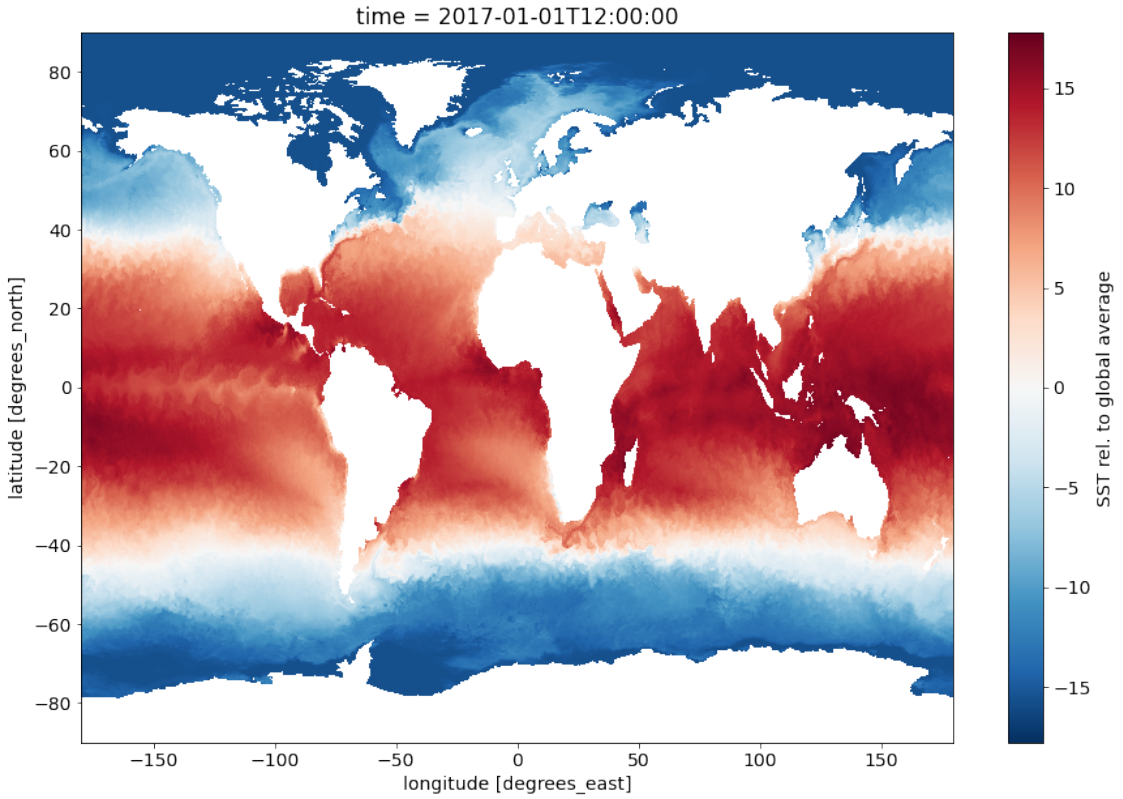
1.3.3 Analyse the data

[6]:
```
print(f"The global mean SST is {ds_SST.analysed_sst.mean().values - 273.15:.2f}␣
↪°C")
```

```
The global mean SST is 13.81 °C
```

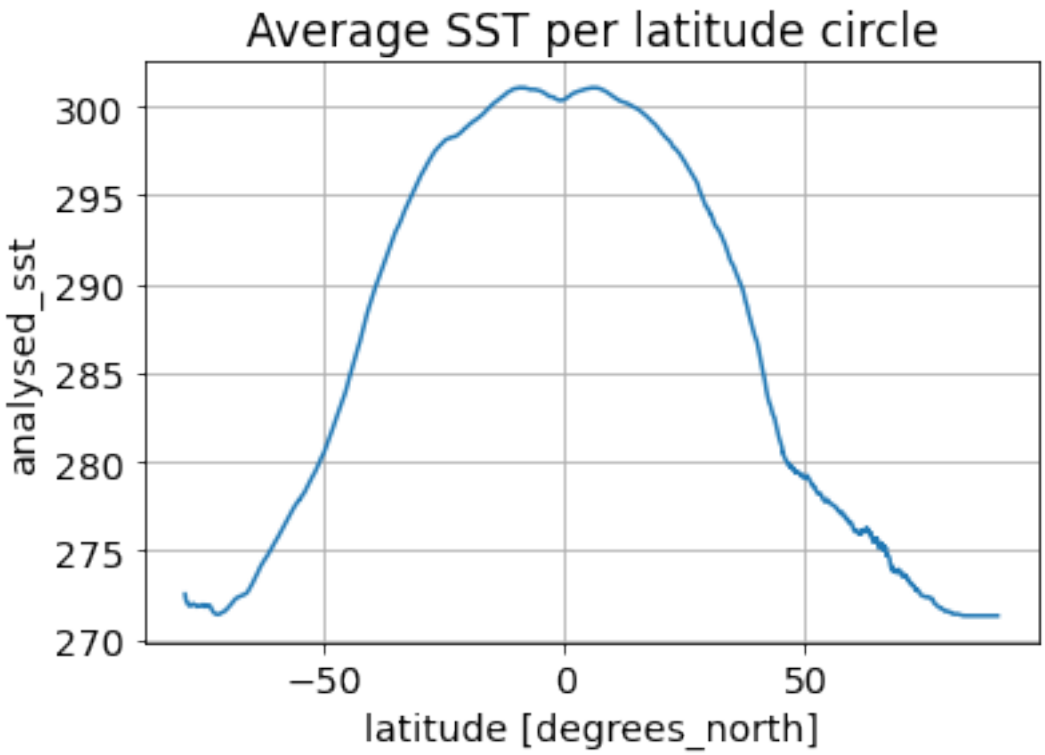
[7]:
```
T_anom = ds_SST.analysed_sst - ds_SST.analysed_sst.mean()
T_anom.name = "SST rel. to global average"

plt.figure(figsize=(15, 10))
T_anom.plot()
```

[7]:
```
<matplotlib.collections.QuadMesh at 0x7f45fd550dd0>
```



[8]:
```
ds_SST.analysed_sst.mean("lon").plot()
plt.grid()
plt.title("Average SST per latitude circle")
```

[8]: Text(0.5, 1.0, 'Average SST per latitude circle')

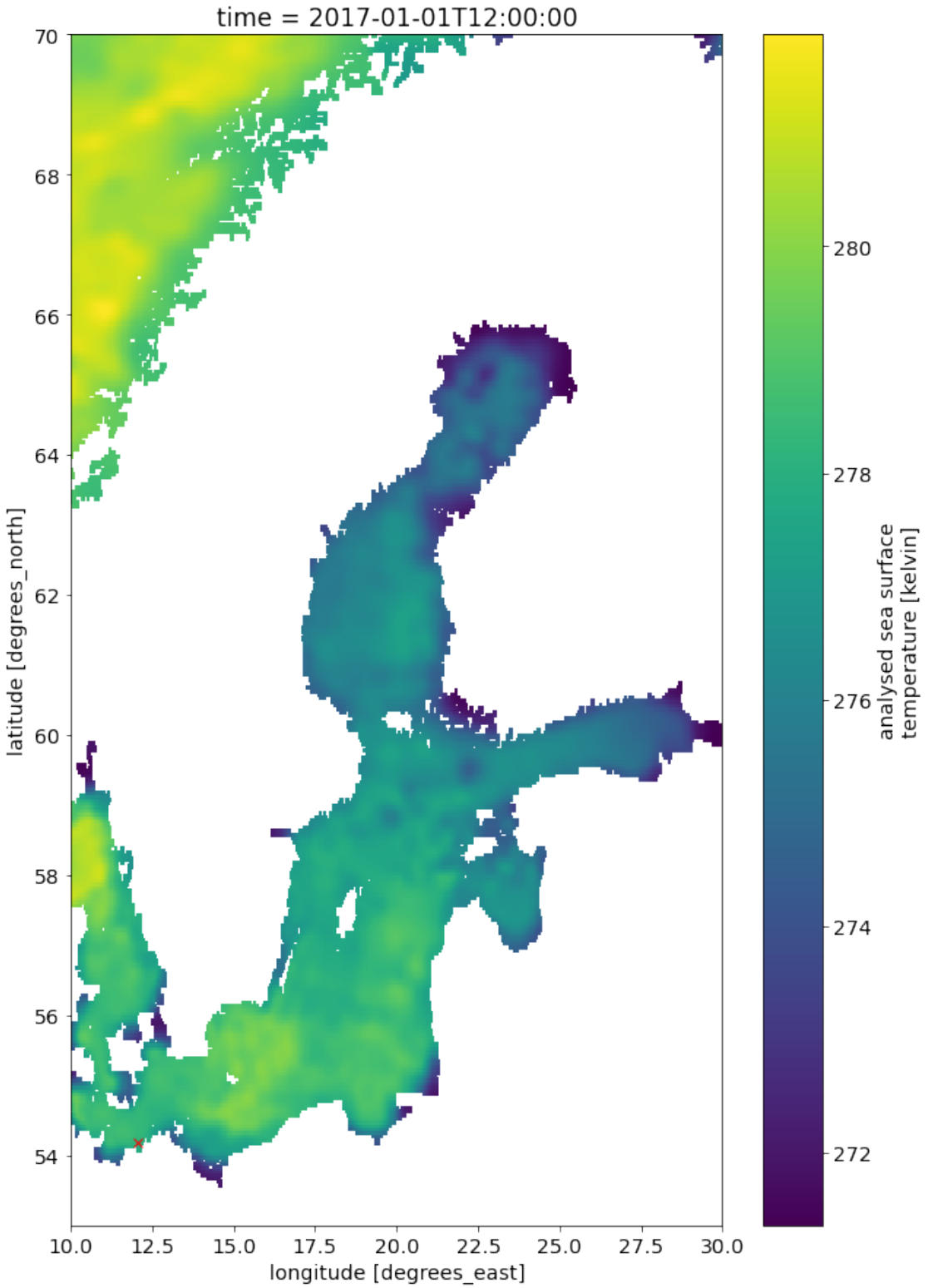


1.3.4 Select specific data

[9]:
```
plt.figure(figsize=(10, 15))
# Zoom to the Baltic Sea
ds_SST.analysed_sst.sel(lat=slice(53, 70), lon=slice(10, 30)).plot()
# Mark the mouth of the Warnow
lat_Warnemünde = 54.175
lon_Warnemünde = 12.075
plt.plot(lon_Warnemünde, lat_Warnemünde, "rx")
# Get the temperature in the mouth of the Warnow
ds_SST.analysed_sst.sel(lat=lat_Warnemünde, lon=lon_Warnemünde)
```

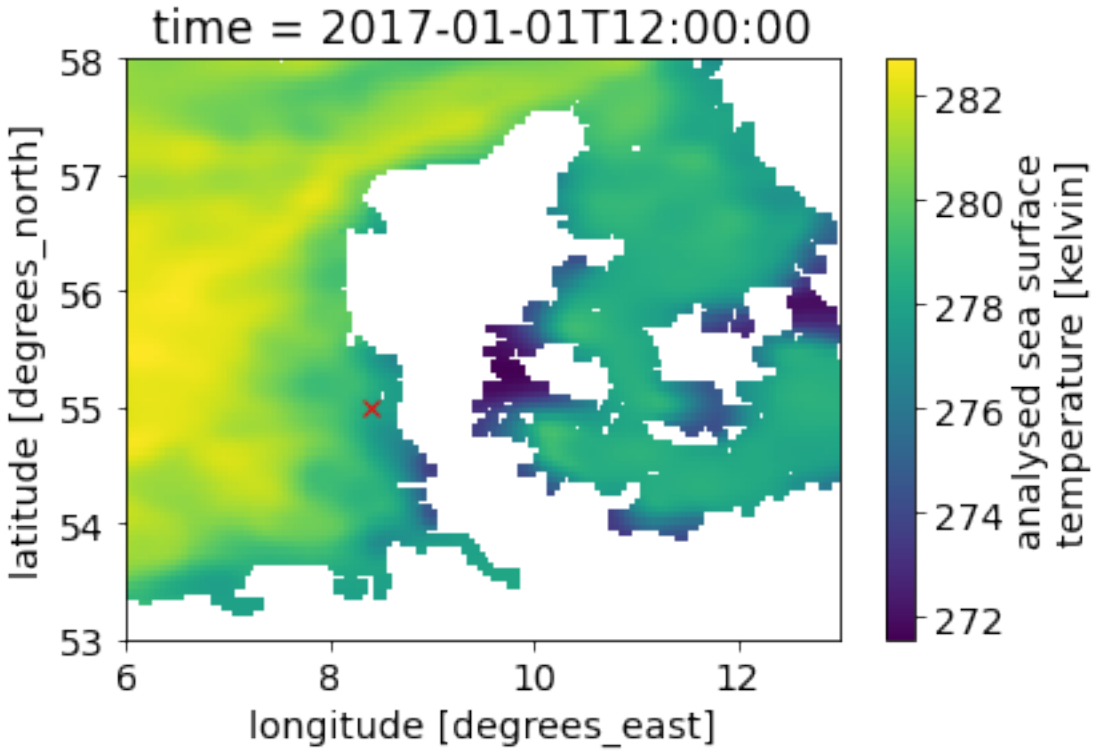
[9]:
```
<xarray.DataArray 'analysed_sst' (time: 1)>
array([278.04], dtype=float32)
Coordinates:
  * time     (time) datetime64[ns] 2017-01-01T12:00:00
    lat      float32 54.175
    lon      float32 12.075
Attributes:
    long_name:      analysed sea surface temperature
    standard_name:  sea_water_temperature
    units:          kelvin
```

```
valid_min:      -300
valid_max:      4500
source:         AVHRR19_G-C3S-L3U-ICDR-v2.0 AVHRRMTA_G-C3S-L3U-ICDR-v2.0 …
```



[10]:
```
# Zoom to the German and Danish coasts
ds_SST.analysed_sst.sel(lat=slice(53, 58), lon=slice(6, 13)).plot()
# Mark Sylt
plt.plot(8.4, 55, "rx")

# Get the temperature near Sylt
ds_SST.analysed_sst.sel(lat=55, lon=8.4, method="nearest")
```

[10]:
```
<xarray.DataArray 'analysed_sst' (time: 1)>
array([278.46], dtype=float32)
Coordinates:
  * time     (time) datetime64[ns] 2017-01-01T12:00:00
    lat      float32 55.025
    lon      float32 8.375
Attributes:
    long_name:      analysed sea surface temperature
    standard_name:  sea_water_temperature
    units:          kelvin
    valid_min:      -300
    valid_max:      4500
    source:         AVHRR19_G-C3S-L3U-ICDR-v2.0 AVHRRMTA_G-C3S-L3U-ICDR-v2.0 …
```



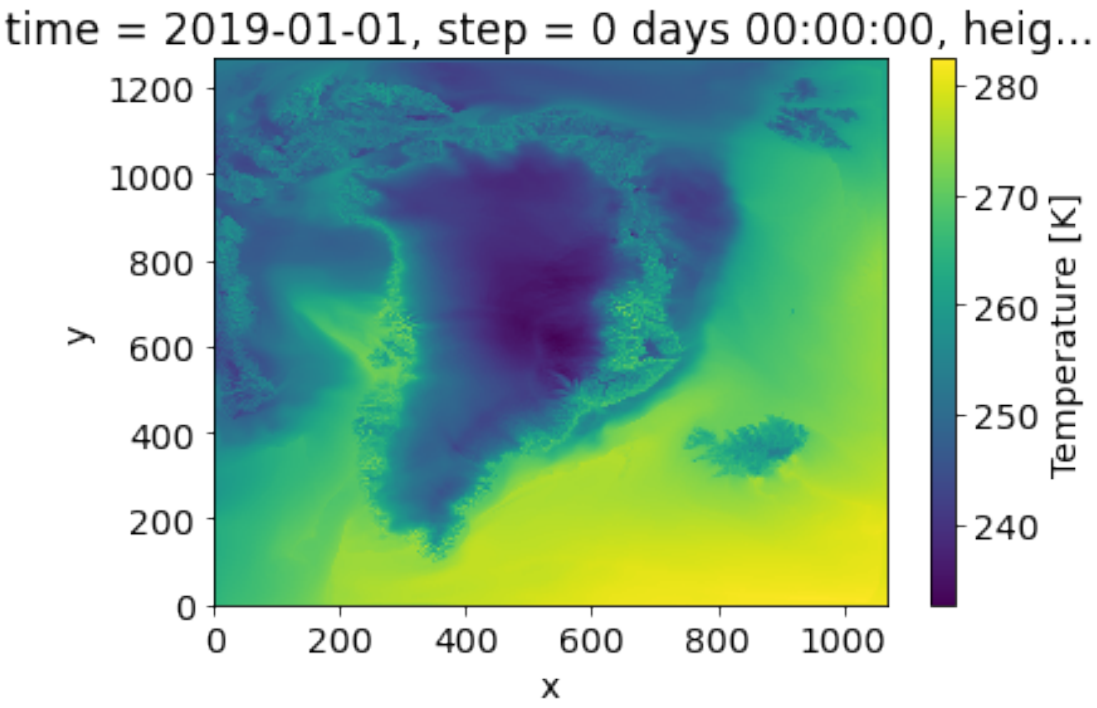
1.4 Example 2: unknown dataset

[4]:
```
ds = xr.load_dataset("../Data/adaptor.uerra_mars.external-1620771977.
↪1561432-27059-25-89c03805-2d54-4fc9-a4de-549d70380e0a.nc")
ds
```

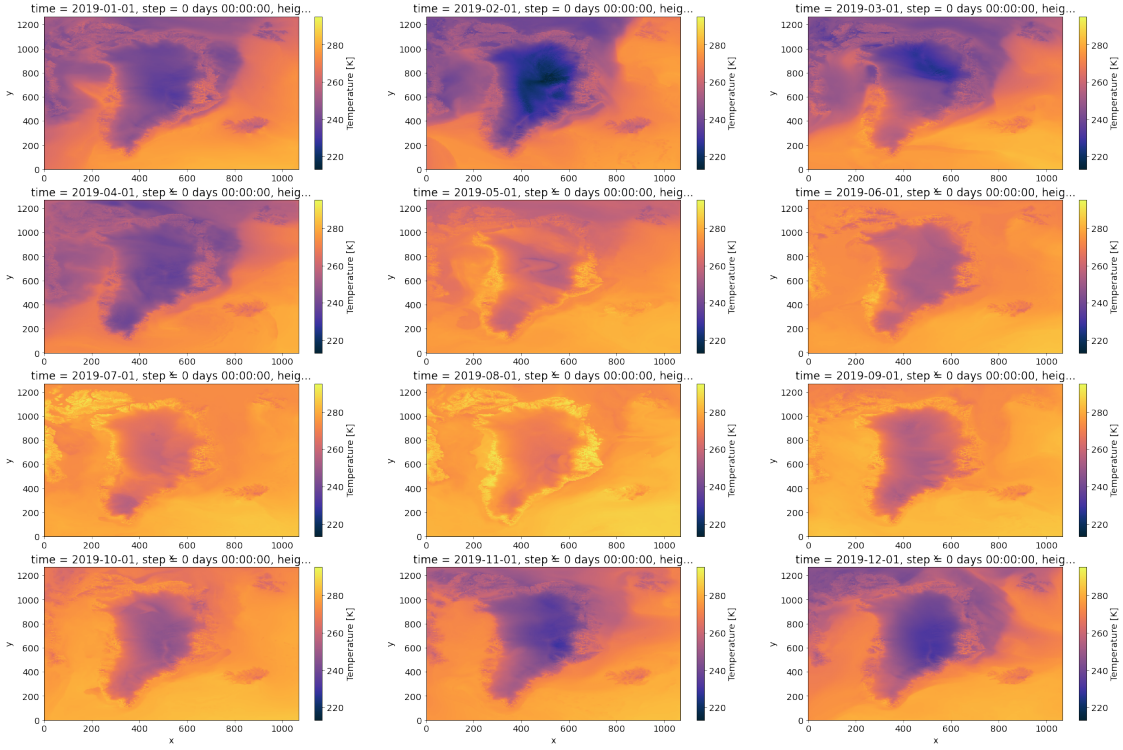
[4]:
```
<xarray.Dataset>
Dimensions:            (heightAboveGround: 11, time: 12, x: 1069, y: 1269)
Coordinates:
  * time               (time) datetime64[ns] 2019-01-01 … 2019-12-01
    step               timedelta64[ns] 00:00:00
  * heightAboveGround  (heightAboveGround) int64 15 30 50 75 … 250 300 400 500
    latitude           (y, x) float64 55.81 55.82 55.82 … 77.87 77.85 77.83
    longitude          (y, x) float64 302.9 302.9 303.0 … 37.56 37.6 37.63
    valid_time         (time) datetime64[ns] 2019-01-01 … 2019-12-01
Dimensions without coordinates: x, y
Data variables:
    pres               (time, heightAboveGround, y, x) float32 101643.31 … …
    t                  (time, heightAboveGround, y, x) float32 264.41602 … …
    wdir               (time, heightAboveGround, y, x) float32 320.44083 … …
    ws                 (time, heightAboveGround, y, x) float32 8.533268 … 2…
Attributes:
    GRIB_edition:            2
    GRIB_centre:             enmi
    GRIB_centreDescription:  Oslo
    GRIB_subCentre:          255
    Conventions:             CF-1.7
    institution:             Oslo
    history:                 2021-05-12T06:59:36 GRIB to CDM+CF via cfgrib-0…
```

[5]:
```
# Let us look at the dataset to get an idea for it
ds.t.isel(time=0, heightAboveGround=0).plot()
```

[5]:
```
<matplotlib.collections.QuadMesh at 0x7fd858f6b810>
```



[6]:
```
# Let us plot a time series of the near ground temperature
ground_temperature = ds.t.sel(heightAboveGround=0, method="nearest")

fig, axs = plt.subplots(ncols=3, nrows=4, figsize=(30, 20))
for i, ax in enumerate(axs.flatten()):
    ground_temperature.isel(time=i).plot(
        ax=ax, vmin=ground_temperature.min(), vmax=ground_temperature.max(),␣
 ↪cmap=cmocean.cm.thermal
    )
```



1.5 Example 3: Argo CTD profile

This dataset has been downloaded from Ifremer's FTP-server. For the link to the server and information on data access, see: http://www.argodatamgt.org/Access-to-data/Access-via-FTP-or-HTTPS-on-GDAC

For more information on the ARGO project, see: https://argo.ucsd.edu/

[7]:
```
ds_ARGO = xr.load_dataset("../Data/ARGO_20210512_prof.nc")
ds_ARGO = ds_ARGO.sel(N_PROF=136)  # select a single profile from the dataset
ds_ARGO
```

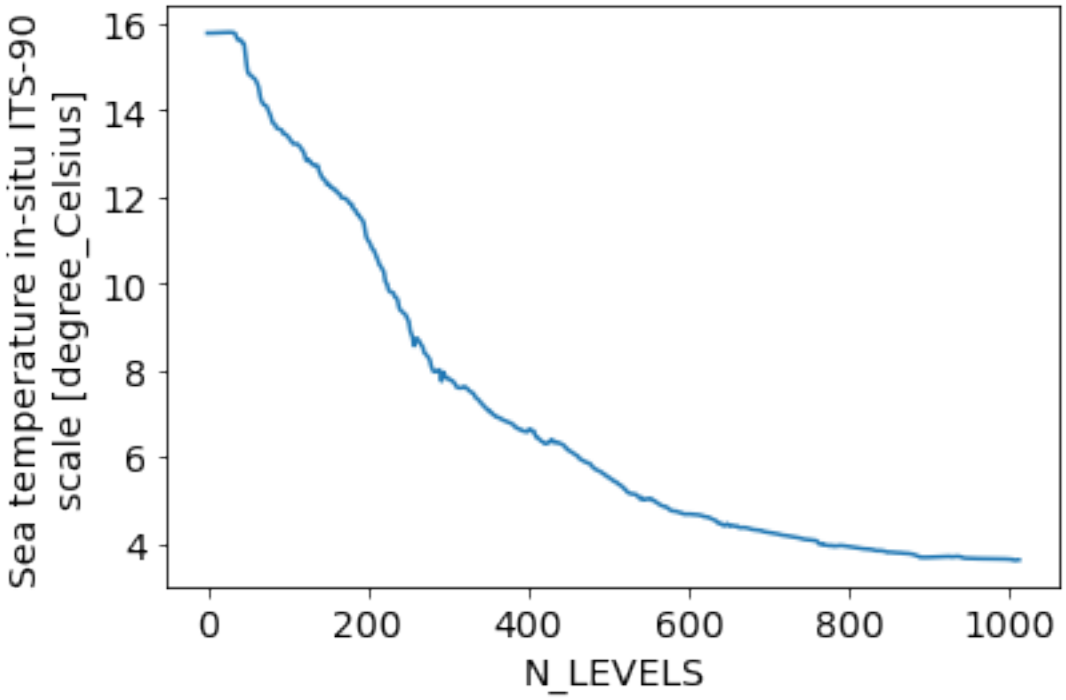
[7]:
```
<xarray.Dataset>
Dimensions:                          (N_CALIB: 2, N_HISTORY: 0, N_LEVELS: 2027,
N_PARAM: 3)
Dimensions without coordinates: N_CALIB, N_HISTORY, N_LEVELS, N_PARAM
Data variables:
    DATA_TYPE                        object b'Argo profile    '
    FORMAT_VERSION                   object b'3.1 '
    HANDBOOK_VERSION                 object b'1.2 '
    REFERENCE_DATE_TIME              object b'19500101000000'
    DATE_CREATION                    object b'20210513142752'
    DATE_UPDATE                      object b'20210513142752'
    PLATFORM_NUMBER                  object b'4902106 '
```

```
PROJECT_NAME                    object b'US ARGO PROJECT                    …
PI_NAME                         object b'BRECK OWENS, STEVEN JAYNE, P.E. RO…
STATION_PARAMETERS              (N_PARAM) object b'PRES            ' … b'…
CYCLE_NUMBER                    float64 207.0
DIRECTION                       object b'A'
DATA_CENTRE                     object b'AO'
DC_REFERENCE                    object b'6463                              '
DATA_STATE_INDICATOR            object b'2B  '
DATA_MODE                       object b'A'
PLATFORM_TYPE                   object b'S2A                               '
FLOAT_SERIAL_NO                 object b'7318                              '
FIRMWARE_VERSION                object b'SBE602 ARM_v2.0_xmsg_ve           '
WMO_INST_TYPE                   object b'854 '
JULD                            datetime64[ns] 2021-05-12T01:48:08
JULD_QC                         object b'1'
JULD_LOCATION                   datetime64[ns] 2021-05-12T01:52:00
LATITUDE                        float64 44.4
LONGITUDE                       float64 -34.1
POSITION_QC                     object b'1'
POSITIONING_SYSTEM              object b'GPS     '
PROFILE_PRES_QC                 object b'A'
PROFILE_TEMP_QC                 object b'A'
PROFILE_PSAL_QC                 object b'A'
VERTICAL_SAMPLING_SCHEME        object b'Primary sampling: averaged [nomina…
CONFIG_MISSION_NUMBER           float64 4.0
PRES                            (N_LEVELS) float32 1.2 2.0 3.04 … nan nan
PRES_QC                         (N_LEVELS) object b'1' b'1' b'1' … nan nan
PRES_ADJUSTED                   (N_LEVELS) float32 1.2 2.0 3.04 … nan nan
PRES_ADJUSTED_QC                (N_LEVELS) object b'1' b'1' b'1' … nan nan
PRES_ADJUSTED_ERROR             (N_LEVELS) float32 nan nan nan … nan nan nan
TEMP                            (N_LEVELS) float32 15.778 15.777 … nan nan
TEMP_QC                         (N_LEVELS) object b'1' b'1' b'1' … nan nan
TEMP_ADJUSTED                   (N_LEVELS) float32 15.778 15.777 … nan nan
TEMP_ADJUSTED_QC                (N_LEVELS) object b'1' b'1' b'1' … nan nan
TEMP_ADJUSTED_ERROR             (N_LEVELS) float32 nan nan nan … nan nan nan
PSAL                            (N_LEVELS) float32 36.096 36.096 … nan nan
PSAL_QC                         (N_LEVELS) object b'2' b'2' b'2' … nan nan
PSAL_ADJUSTED                   (N_LEVELS) float32 36.04031 36.04031 … nan
PSAL_ADJUSTED_QC                (N_LEVELS) object b'2' b'2' b'2' … nan nan
PSAL_ADJUSTED_ERROR             (N_LEVELS) float32 nan nan nan … nan nan nan
PARAMETER                       (N_CALIB, N_PARAM) object b'PRES           …
SCIENTIFIC_CALIB_EQUATION       (N_CALIB, N_PARAM) object b'none           …
SCIENTIFIC_CALIB_COEFFICIENT    (N_CALIB, N_PARAM) object b'               …
SCIENTIFIC_CALIB_COMMENT        (N_CALIB, N_PARAM) object b'               …
SCIENTIFIC_CALIB_DATE           (N_CALIB, N_PARAM) object b'              '…
HISTORY_INSTITUTION             (N_HISTORY) object
HISTORY_STEP                    (N_HISTORY) object
```

```
    HISTORY_SOFTWARE              (N_HISTORY) object
    HISTORY_SOFTWARE_RELEASE      (N_HISTORY) object
    HISTORY_REFERENCE             (N_HISTORY) object
    HISTORY_DATE                  (N_HISTORY) object
    HISTORY_ACTION                (N_HISTORY) object
    HISTORY_PARAMETER             (N_HISTORY) object
    HISTORY_START_PRES            (N_HISTORY) float32
    HISTORY_STOP_PRES             (N_HISTORY) float32
    HISTORY_PREVIOUS_VALUE        (N_HISTORY) float32
    HISTORY_QCTEST                (N_HISTORY) object
Attributes:
    title:                  Argo float vertical profile
    institution:            FR GDAC
    source:                 Argo float
    history:                2021-05-13T14:27:52Z creation
    references:             http://www.argodatamgt.org/Documentation
    user_manual_version:    3.1
    Conventions:            Argo-3.1 CF-1.6
    featureType:            trajectoryProfile
```

[8]:
```
ds_ARGO.TEMP.plot()
```

[8]:
```
[<matplotlib.lines.Line2D at 0x7fd8506fbad0>]
```

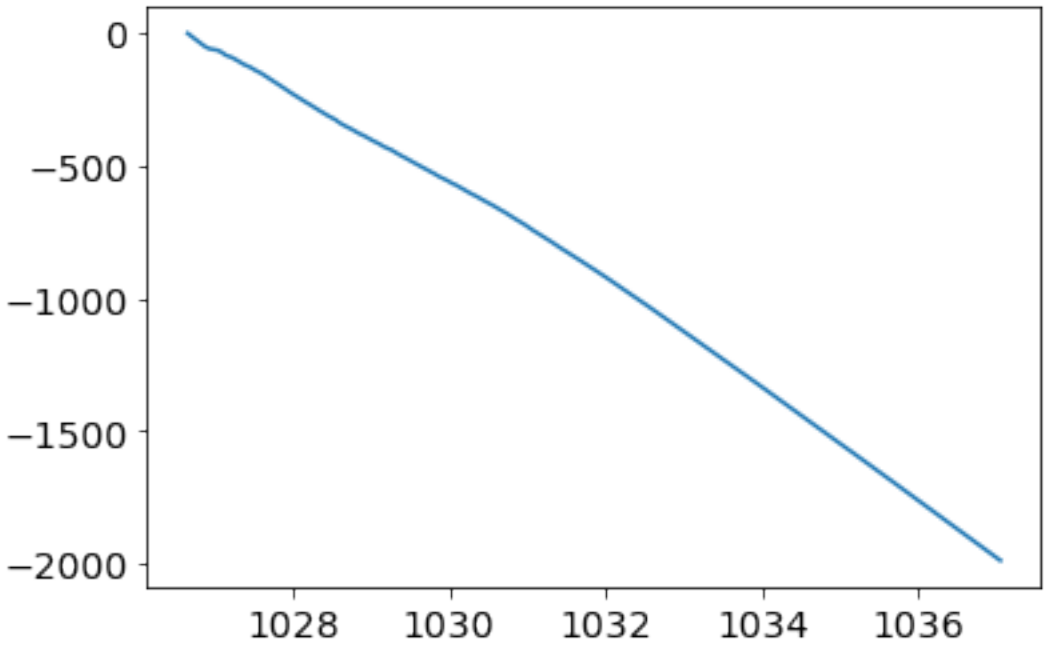


[9]:
```
# Convert practial to absolute salinity
SA = gsw.SA_from_SP(ds_ARGO.PSAL, ds_ARGO.PRES, ds_ARGO.LONGITUDE, ds_ARGO.
↪LATITUDE)
# Convert in-situ to conservative temperature
CT = gsw.CT_from_t(SA, ds_ARGO.TEMP, ds_ARGO.PRES)
# Calculate the density
rho = gsw.rho(SA, CT, ds_ARGO.PRES)
```

[10]:
```
# Calculate the vertical coordinate from pressure
z = gsw.z_from_p(ds_ARGO.PRES, ds_ARGO.LATITUDE)
```

[11]:
```
# Make a plot of a density profile
plt.plot(rho, z)
```

[11]:
```
[<matplotlib.lines.Line2D at 0x7fd8506eadd0>]
```



1.5.1 Show the position of the profile on a map

Check out https://scitools.org.uk/cartopy/docs/latest/crs/projections.html for a list of available map projections.

[12]:
```
ax = plt.axes(projection=ccrs.Mercator())
ax.gridlines()
ax.coastlines()
```

```
ax.set_global()

ax.plot(ds_ARGO.LONGITUDE, ds_ARGO.LATITUDE, "ro", transform=ccrs.PlateCarree())
```

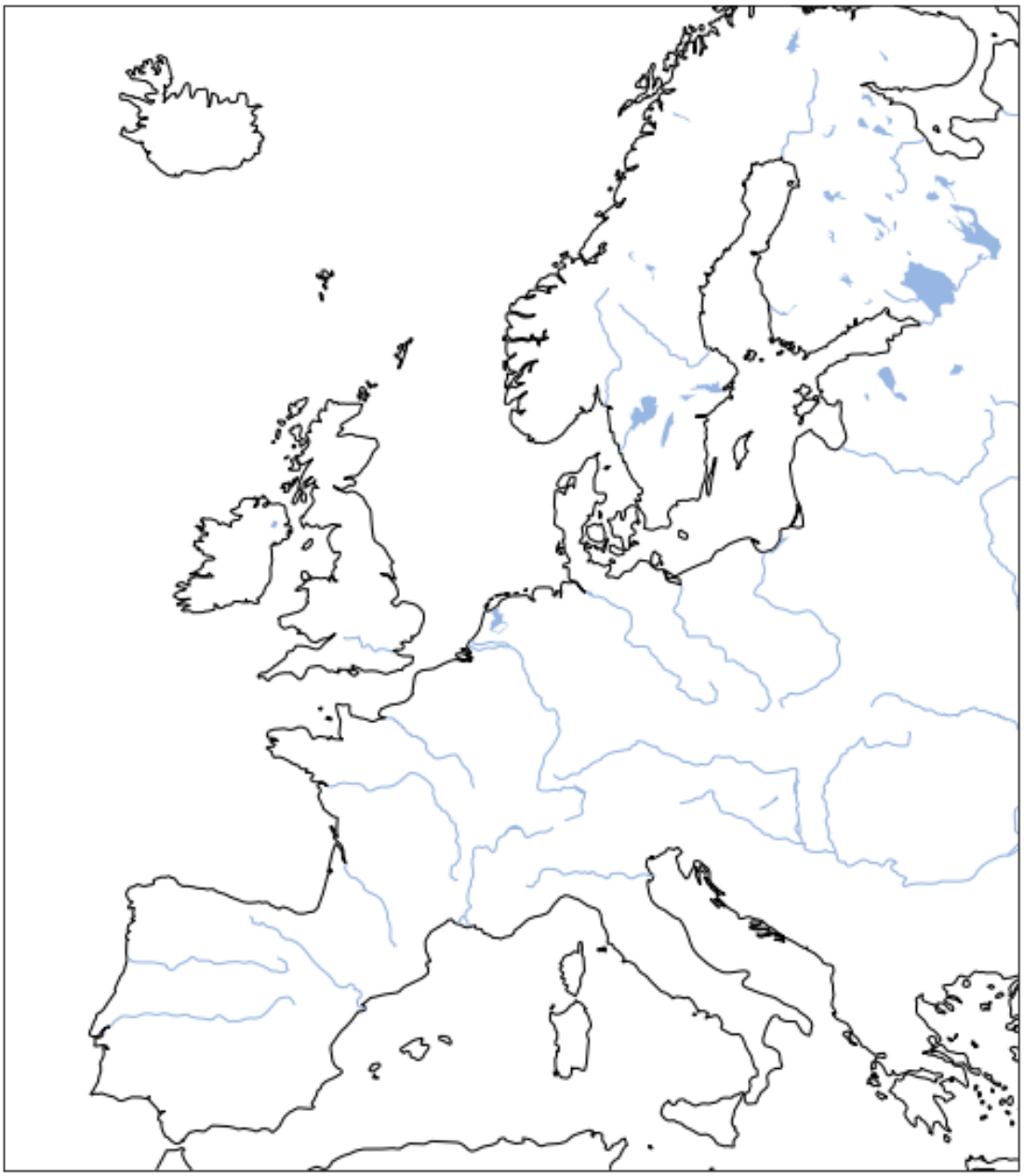
[12]: [<matplotlib.lines.Line2D at 0x7fd83d59ea10>]

1.5.2 Add features to the map

[13]:
```
fig = plt.figure(figsize=(10, 10))
ax = fig.add_subplot(projection=ccrs.EuroPP())

ax.coastlines(resolution='50m')
lakes = cfeature.NaturalEarthFeature(
    category='physical',
    name='lakes',
    scale='50m',
    edgecolor='none',
    facecolor=cfeature.COLORS['water'],
)
rivers = cfeature.NaturalEarthFeature(
    category='physical',
    name='rivers_lake_centerlines',
    scale='50m',  # can also be '10m' (more details) or '110m' (less details)
    edgecolor=cfeature.COLORS['water'],
    facecolor='none',
)
ax.add_feature(lakes)
```

```
ax.add_feature(rivers)
```

[13]: <cartopy.mpl.feature_artist.FeatureArtist at 0x7fd83d461250>



[ ]: