In [1]:

```python
from pathlib import Path
import xarray as xr
import rasterio
import rioxarray
from pyproj import Proj, CRS
import numpy as np
```

Setup for testing

In [2]:

```python
#set the global keep_attrs to True, to avoid losing longitude attributes during computation
xr.set_options(keep_attrs=True)
```

Out[2]:

```
<xarray.core.options.set_options at 0x142df3500>
```

In [3]:

```python
#open an indicator file for testing
fp = Path('/Users/joshpaul/Desktop/SNAP/CMIP6/rx1day_CESM2-WACCM_ssp126_indicator.nc')
ds = xr.open_dataset(fp)
```

In [4]:

```python
#make a copy of the encoding (this is not persisted thru computation)
lon_enc = ds['lon'].encoding

#subtract from 0-360 lon coords to get -180 to 180 lon coords, and reapply encoding
ds['lon'] = ds['lon'] - 180
ds['lon'].encoding = lon_enc

#sort and verify
ds = ds.sortby(ds.lon, ascending=True)
ds
```

Out[4]:

```
xarray.Dataset

▶ Dimensions:        (lat: 43, lon: 288, year: 96, scenario: 1, model: 1)

▼ Coordinates:
    lat              (lat)                          float64  50.42 51.36 52.3 ... 89.06 90.0
    lon              (lon)                          float64  -180.0 -178.8 ... 177.5 178.8
    year             (year)                           int64  2015 2016 2017 ... 2108 2109 2110
    scenario         (scenario)                         <U6  'ssp126'
    model            (model)                           <U11  'CESM2-WACCM'

▼ Data variables:
    rx1day           (scenario, model, year, lat, lon)  float64  ...

▶ Indexes:   (5)

▶ Attributes:   (0)
```

In [5]:

```python
#make a copy for testing against rioxarray
ds_ = ds.copy()
```

Manual population of encodings / attributes for CF-compliant CRS

In [6]:

```python
#get CF-compliant crs attribute dict
cf_crs = CRS.from_epsg(4326).to_cf()
cf_crs
```

Out[6]:

```
{'crs_wkt': 'GEOGCRS["WGS 84",ENSEMBLE["World Geodetic System 1984 ensemble",MEMBER["World Geodetic System
1984 (Transit)"],MEMBER["World Geodetic System 1984 (G730)"],MEMBER["World Geodetic System 1984 (G873)"],M
EMBER["World Geodetic System 1984 (G1150)"],MEMBER["World Geodetic System 1984 (G1674)"],MEMBER["World Geo
detic System 1984 (G1762)"],MEMBER["World Geodetic System 1984 (G2139)"],ELLIPSOID["WGS 84",6378137,298.25
7223563,LENGTHUNIT["metre",1]],ENSEMBLEACCURACY[2.0]],PRIMEM["Greenwich",0,ANGLEUNIT["degree",0.0174532925
199433]],CS[ellipsoidal,2],AXIS["geodetic latitude (Lat)",north,ORDER[1],ANGLEUNIT["degree",0.017453292519
9433]],AXIS["geodetic longitude (Lon)",east,ORDER[2],ANGLEUNIT["degree",0.0174532925199433]],USAGE[SCOPE
["Horizontal component of 3D system."],AREA["World."],BBOX[-90,-180,90,180]],ID["EPSG",4326]]',
 'semi_major_axis': 6378137.0,
 'semi_minor_axis': 6356752.314245179,
 'inverse_flattening': 298.257223563,
 'reference_ellipsoid_name': 'WGS 84',
 'longitude_of_prime_meridian': 0.0,
 'prime_meridian_name': 'Greenwich',
 'geographic_crs_name': 'WGS 84',
 'horizontal_datum_name': 'World Geodetic System 1984 ensemble',
 'grid_mapping_name': 'latitude_longitude'}
```

In [7]:

```python
#create a spatial_ref coordinate, which is an empty array but has the CF-compliant crs attribute dict
ds = ds.assign_coords({
    "spatial_ref": ([],np.array(0), cf_crs)
                 })
#add a second attribute "spatial_ref" identical to "crs_wkt" (to match rioxarray output)
ds["spatial_ref"].attrs["spatial_ref"] = cf_crs['crs_wkt']
```

In [8]:

```python
#verify
ds
```

```
Out[8]: xarray.Dataset

▶ Dimensions:      (lat: 43, lon: 288, year: 96, scenario: 1, model: 1)

▼ Coordinates:
    lat            (lat)                              float64  50.42 51.36 52.3 ... 89.06 90.0
    lon            (lon)                              float64  -180.0 -178.8 ... 177.5 178.8
    year           (year)                               int64  2015 2016 2017 ... 2108 2109 2110
    scenario       (scenario)                            <U6  'ssp126'
    model          (model)                              <U11  'CESM2-WACCM'
    spatial_ref    ()                                   int64  0

▼ Data variables:
    rx1day         (scenario, model, year, lat, lon)  float64  ...

▶ Indexes:    (5)

▶ Attributes:    (0)
```

```
In [9]:  #manually add attributes / encodings to the data variable
         ds["rx1day"].encoding["grid_mapping"] = "spatial_ref"
         #ds["rx1day"].encoding["coordinates"] = "spatial_ref lon lat"
```

```
In [10]: #verify encodings (especially dtypes) are there
         for c in list(ds.coords):
             print(ds[c].encoding)

         print(ds.rx1day.encoding)
```

```
{'dtype': dtype('float64'), 'zlib': True, 'szip': False, 'zstd': False, 'bzip2': False, 'blosc': False, 'sh
uffle': True, 'complevel': 2, 'fletcher32': False, 'contiguous': False, 'chunksizes': (43,), 'preferred_chu
nks': {'lat': 43}, 'source': '/Users/joshpaul/Desktop/SNAP/CMIP6/rx1day_CESM2-WACCM_ssp126_indicator.nc',
'original_shape': (43,), '_FillValue': nan}
{'dtype': dtype('float64'), 'zlib': True, 'szip': False, 'zstd': False, 'bzip2': False, 'blosc': False, 'sh
uffle': True, 'complevel': 2, 'fletcher32': False, 'contiguous': False, 'chunksizes': (288,), 'preferred_ch
unks': {'lon': 288}, 'source': '/Users/joshpaul/Desktop/SNAP/CMIP6/rx1day_CESM2-WACCM_ssp126_indicator.nc',
'original_shape': (288,), '_FillValue': nan}
{'dtype': dtype('int64'), 'zlib': False, 'szip': False, 'zstd': False, 'bzip2': False, 'blosc': False, 'shu
ffle': False, 'complevel': 0, 'fletcher32': False, 'contiguous': True, 'chunksizes': None, 'source': '/User
s/joshpaul/Desktop/SNAP/CMIP6/rx1day_CESM2-WACCM_ssp126_indicator.nc', 'original_shape': (96,)}
{'dtype': dtype('<U6')}
{'dtype': dtype('<U11')}
{}
{'dtype': dtype('float64'), 'zlib': False, 'szip': False, 'zstd': False, 'bzip2': False, 'blosc': False, 's
huffle': False, 'complevel': 0, 'fletcher32': False, 'contiguous': True, 'chunksizes': None, 'source': '/Us
ers/joshpaul/Desktop/SNAP/CMIP6/rx1day_CESM2-WACCM_ssp126_indicator.nc', 'original_shape': (1, 1, 96, 43, 2
88), '_FillValue': nan, 'grid_mapping': 'spatial_ref'}
```

In [11]:

```
#verify
print(ds.rio.crs)
ds
```

```
EPSG:4326
```

Out [11]:

```
xarray.Dataset

▶ Dimensions:      (lat: 43, lon: 288, year: 96, scenario: 1, model: 1)

▼ Coordinates:
   lat             (lat)                             float64   50.42 51.36 52.3 ... 89.06 90.0
   lon             (lon)                             float64   -180.0 -178.8 ... 177.5 178.8
   year            (year)                            int64     2015 2016 2017 ... 2108 2109 2110
   scenario        (scenario)                        <U6       'ssp126'
   model           (model)                           <U11      'CESM2-WACCM'
   spatial_ref     ()                                int64     0

▼ Data variables:
   rx1day          (scenario, model, year, lat, lon) float64   ...

▶ Indexes:   (5)

▶ Attributes:   (0)
```

In [12]:

```python
#save output for testing in QGIS / ArcPro
ds.to_netcdf('/Users/joshpaul/Desktop/SNAP/CMIP6/test_in_GIS/test_manual.nc')
```

Rioxarray population of encodings / attributes for CF-compliant CRS

In [13]:

```python
#write the basic coordinate system "in a CF compliant manner" (https://corteva.github.io/rioxarray/stable/
ds_.rio.write_crs("epsg:4326", inplace=True)
#verify
ds_.rio.crs
```

Out [13]:

```
CRS.from_epsg(4326)
```

In [14]:

```python
for c in list(ds_.coords):
    print(ds_[c].encoding)
print(ds_.rx1day.encoding)
```

```
{'dtype': dtype('float64'), 'zlib': True, 'szip': False, 'zstd': False, 'bzip2': False, 'blosc': False, 'sh
uffle': True, 'complevel': 2, 'fletcher32': False, 'contiguous': False, 'chunksizes': (43,), 'preferred_chu
nks': {'lat': 43}, 'source': '/Users/joshpaul/Desktop/SNAP/CMIP6/rx1day_CESM2-WACCM_ssp126_indicator.nc',
'original_shape': (43,), '_FillValue': nan}
{'dtype': dtype('float64'), 'zlib': True, 'szip': False, 'zstd': False, 'bzip2': False, 'blosc': False, 'sh
uffle': True, 'complevel': 2, 'fletcher32': False, 'contiguous': False, 'chunksizes': (288,), 'preferred_ch
unks': {'lon': 288}, 'source': '/Users/joshpaul/Desktop/SNAP/CMIP6/rx1day_CESM2-WACCM_ssp126_indicator.nc',
'original_shape': (288,), '_FillValue': nan}
{'dtype': dtype('int64'), 'zlib': False, 'szip': False, 'zstd': False, 'bzip2': False, 'blosc': False, 'shu
ffle': False, 'complevel': 0, 'fletcher32': False, 'contiguous': True, 'chunksizes': None, 'source': '/User
s/joshpaul/Desktop/SNAP/CMIP6/rx1day_CESM2-WACCM_ssp126_indicator.nc', 'original_shape': (96,)}
{'dtype': dtype('<U6')}
{'dtype': dtype('<U11')}
{}
{'dtype': dtype('float64'), 'zlib': False, 'szip': False, 'zstd': False, 'bzip2': False, 'blosc': False, 's
huffle': False, 'complevel': 0, 'fletcher32': False, 'contiguous': True, 'chunksizes': None, 'source': '/Us
ers/joshpaul/Desktop/SNAP/CMIP6/rx1day_CESM2-WACCM_ssp126_indicator.nc', 'original_shape': (1, 1, 96, 43, 2
88), '_FillValue': nan, 'grid_mapping': 'spatial_ref'}
```

In [15]:

```
#save output for testing in QGIS / ArcPro
ds_.to_netcdf('/Users/joshpaul/Desktop/SNAP/CMIP6/test_in_GIS/test_rio.nc')
```

Final comparison of attributes

In [16]:

```
ds
```

Out[16]:

```
xarray.Dataset

▶ Dimensions:     (lat: 43, lon: 288, year: 96, scenario: 1, model: 1)

▼ Coordinates:
   lat           (lat)                              float64  50.42 51.36 52.3 ... 89.06 90.0
   lon           (lon)                              float64  -180.0 -178.8 ... 177.5 178.8
   year          (year)                               int64  2015 2016 2017 ... 2108 2109 2110
   scenario      (scenario)                             <U6  'ssp126'
   model         (model)                               <U11  'CESM2-WACCM'
   spatial_ref   ()                                   int64  0

▼ Data variables:
   rx1day        (scenario, model, year, lat, lon)  float64  ...

▶ Indexes:   (5)

▶ Attributes:   (0)
```

In [17]:

```
ds_
```

```
Out[17]: xarray.Dataset

▶ Dimensions:    (lat: 43, lon: 288, year: 96, scenario: 1, model: 1)
▼ Coordinates:
   lat           (lat)                               float64  50.42 51.36 52.3 ... 89.06 90.0
   lon           (lon)                               float64  -180.0 -178.8 ... 177.5 178.8
   year          (year)                                int64  2015 2016 2017 ... 2108 2109 2110
   scenario      (scenario)                             <U6  'ssp126'
   model         (model)                               <U11  'CESM2-WACCM'
   spatial_ref   ()                                    int64  0
▼ Data variables:
   rx1day        (scenario, model, year, lat, lon)  float64  ...
▶ Indexes:   (5)
▶ Attributes:   (0)
```

```
In [ ]:
```