

# CO2\_SeaIceExtend\_Solution

September 30, 2022

```
[ ]: import xarray as xr
import numpy as np
import matplotlib as plt

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

```
[ ]: cdo.version()
```

## 1 Just taking a look at the raw data from one model

```
[ ]: import os
datadir='{}/src/pycourse/material/data/'.format(os.environ['HOME'])

!ls ~/src/pycourse/material/data

# A firest look into some sample data files
ds = xr.open_dataset(datadir+'CMIP.HAMMOZ-Consortium.MPI-ESM-1-2-HAM.historical.
↳Simon.gn.nc')
ds2 = xr.open_dataset(datadir+'CMIP.HAMMOZ-Consortium.MPI-ESM-1-2-HAM.
↳historical.Amon.gn.nc')
```

```
[ ]: ds2.co2mass.plot()
```

```
[ ]: ds.siextentn.plot()
```

## 2 Combining sea ice extent and CO2 mass dataset together in order to make a scatter plot

### 2.1 1 Focus on a single model output

```
[ ]: # use variables for the filenames corresponding to a single model
icefile=datadir+'CMIP.HAMMOZ-Consortium.MPI-ESM-1-2-HAM.historical.SImon.gn.nc'
co2file=datadir+'CMIP.HAMMOZ-Consortium.MPI-ESM-1-2-HAM.historical.Amon.gn.nc'
```

```
# check variable names
for ifile in [icefile, co2file]:
    print(f"Filename:{ifile}| Variable name: {cdo.showname(input=ifile)[0]}")
```

## 2.2 2 Check what is need on the time axis for the monthly input

```
[ ]: iceVarname = 'siextentn'
co2Varname = 'co2mass'

# Compute yearly mean values for CO2
co2ymean = cdo.yearmean(input=co2file)

# alternative:
# co2mass = cdo.yearmean(input=co2file,returnXArray=co2Varname)
# co2mass.plot()
# or
# co2mass = cdo.selmon(9,input=icefile,returnXArray=co2Varname) ?

# select September
siceex = cdo.selmon(9,input=icefile)
# create running mean over 30 years
siceex30 = cdo.runmean(30,input=f" -selmon,9 {icefile}")
```

## 2.3 3 Smoothing: compare runmean with plain sea ice extern data

```
[ ]: xr.open_dataset(siceex).plot.scatter('time',iceVarname)
xr.open_dataset(siceex30).plot.scatter('time',iceVarname)
```

## 2.4 4 Merge both inputs to get close to the scatter plot

```
[ ]: #field=xr.merge([siceex,co2],compat='override') #does not work because time
      ↪ stamps do not match
cdo.debug = True
field=cdo.merge(input=[co2ymean,siceex],returnXDataset=True)
cdo.debug = False
# print(cdo.sinfov(input=field))
```

```
[ ]: field.co2mass
#field[varname2]

# access modes for XDataSet
print(all(field.co2mass == field[co2Varname]))

# first scatter plot - yearly values for ice extend (not 30-year-runmean)
field.plot.scatter('co2mass','siextentn')
```

## 2.5 Now check the quality of the runmean input

```
[ ]: yearlist=cdo.showyear(input=siceex30)
      print(yearlist)

[ ]: # list is a string, use "split" to break them down into a list of years
      yearlist=cdo.showyear(input=siceex30)[0]
      print(yearlist)

      yearlist=cdo.showyear(input=siceex30)[0].split()
      print(yearlist)

      # use THIS list to select data from the CO2 input

      co2_30=cdo.selyear(*yearlist,input=co2ymean)
      #asterisk: converts array of years (use can, but don't have to)
      #selyear does not accept arrays, rather to provide all elements in the array,
      ↪ this functions as giving the whole array in one go

      #co2_30=cdo.selyear(yearlist,input=co2) #does not work!

      field30=cdo.merge(input=[co2_30,siceex30],returnXDataset=True)

[ ]: field30.plot.scatter('co2mass','siextentn')
      #Using sea ice extent that has a 30-year running mean
      # Compare with the original
      #field.plot.scatter('co2mass','siextentn')
```

## 3 5 Pull in from multiple models and make scatter plot

```
[ ]: # Plan: Write a function with 2 files as input

def computeCorrelation(afile,sfile,startyear=0):
    co2      = cdo.yearmean(input=afile)
    siceex   = cdo.selmon(9,input=sfile) #,returnXArray=varname2[0])
    field    = cdo.merge(input=[co2,siceex],returnXDataset=True)

    siceex30 = cdo.runmean(30,input=' -selmon,9 '+sfile)
    ↪ #,returnXArray=varname2[0])
    yearlist = cdo.showyear(input=siceex30)[0].split()
    co2_30    = cdo.selyear(*yearlist,input=co2)
    field30   = cdo.merge(input=[co2_30,siceex30],returnXDataset=True)
    return [field, field30]

# testing call
#f, f30 = computeCorrelation(afile,sfile)
```

```
#print(f)
#print(f30)
```

## 4 6 How to Loop over all files

```
[ ]: print(datadir)
import glob
files = glob.glob(datadir+'CMIP.*nc')
files.sort()
print(files)

#select files with corresponding Amon/Simon fields - tag those with their
↳model ID
data = {}
for file in files:
    print('Processing ... '+file)
    key = '-'.join(file.split('/')[1].split('.')[1:3])
    #DEBUG print('    KEY = '+key)
    if key in data:
        data[key].append(file)
    else:
        data[key] = [file]
print(data)

# from matplotlib import pylab as pl
# import xarray as xr
# cdo.debug = False
plotData = []
for key,value in data.items():
    model = key
    co2File, seaIceExtFile = value[0], value[1]
    corrField=computeCorrelation(co2File, seaIceExtFile)
    plotData.append(corrField[1])
    corrField[1].plot.scatter('co2mass','siextentn')
```