xgriddedaxis Documentation

Release 0.0.post43

xgriddedaxis developers

Apr 23, 2020

Documentation Contents

xgriddedaxis is a Python package for managing/working with one-dimensional axes with their respective cell boundaries information. xgriddedaxis consumes and produces xarray data structures, which are coordinate and metadata-rich representations of multidimensional array data.

xgriddedaxis was motivated by the fact that xarray is not aware of cell boundary variables when performing operations such as **resampling**. The main objective of xgriddedaxis is to provide a set of utilities that enables fluid translation between data at different intervals while being aware of the cell boundary variables.

The fundamental concept in xgriddedaxis is a *Remapper* object. *Remapper*'s role includes:

- Creating a source axis, i.e. the axis that your original data is on,
- Creating a destination axis, i.e. the axis that you want to convert your data to,
- Creating a *Remapper* object by passing the source and destination axis you created previously,
- Finally, converting your data from the source axis to the destination axis, using the *Remapper* object you created in previous step.

For more information, read the full xgriddedaxis documentation.

CHAPTER 1

Installation

xgriddedaxis can be installed from PyPI with pip:

```
python -m pip install xgriddedaxis
```

It is also available from *conda-forge* for conda installations:

```
conda install -c conda-forge xgriddedaxis
```

CHAPTER 2

Feedback

If you encounter any errors or problems with **xgriddedaxis**, please open an issue at the GitHub main repository.

2.1 Installation

xgriddedaxis can be installed from PyPI with pip:

```
python -m pip install xgriddedaxis
```

It is also available from *conda-forge* for conda installations:

```
conda install -c conda-forge xgriddedaxis
```

2.2 Tutorial

[1]:
```
from xgriddedaxis import Remapper
from xgriddedaxis.testing import create_dataset
import xarray as xr
xr.set_options(display_style="html")
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-1-d8ee320b5cf4> in <module>
      1 from xgriddedaxis import Remapper
----> 2 from xgriddedaxis.testing import create_dataset
      3 import xarray as xr
      4 xr.set_options(display_style="html")

ModuleNotFoundError: No module named 'xgriddedaxis.testing'
```

2.2.1 Input Data

For demonstration purposes, we are going to use the `create_dataset()` function for generating test data.

[2]:
```
ds = create_dataset(start='2000-01-01', end='2002-01-01', freq='D', nlats=90,␣
↪nlons=180)
ds
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-07ad2c7318d6> in <module>
----> 1 ds = create_dataset(start='2000-01-01', end='2002-01-01', freq='D', nlats=90,␣
↪nlons=180)
      2 ds

NameError: name 'create_dataset' is not defined
```

Our input data set consists of two variables `tmin`, and `tmax` plus the `time_bounds` variable. The data was generated at a daily frequency for two years.

[3]:
```
ds.tmin.isel(time=0).plot(robust=True)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-3-21f5d7dc9c47> in <module>
----> 1 ds.tmin.isel(time=0).plot(robust=True)

NameError: name 'ds' is not defined
```

[4]:
```
m = ds.mean(dim=['lat', 'lon'])
m.tmin.plot()
m.tmax.plot()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-4-ef7652c33a7a> in <module>
----> 1 m = ds.mean(dim=['lat', 'lon'])
      2 m.tmin.plot()
      3 m.tmax.plot()

NameError: name 'ds' is not defined
```

2.2.2 Remapper Object

Say we want to downsample the input data from `daily` to `monthly` frequency. To achieve this, we create a remapper object, and pass in:

- An xarray Dataset containing the time, time boundary information of the incoming time axis.
- An outgoing frequency. For e.g 'M', '2D', 'H', or '3T' For full specification of available frequencies, please see here

[5]:
```
remapper = Remapper(ds, freq='M')
remapper
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-5-dd8fccf6b83e> in <module>
```

(continues on next page)

(continued from previous page)

```
----> 1 remapper = Remapper(ds, freq='M')
      2 remapper

NameError: name 'ds' is not defined
```

During the `Remapper` object creation, `xgriddedaxis` uses the incoming time axis information in conjunction with the specified frequency to construct an outgoing time axis information. This information is stored as an xarray Dataset in the `.info` attribute of the remapper object:

[6]:
```
remapper.info
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-6-9abd55daf06e> in <module>
----> 1 remapper.info

NameError: name 'remapper' is not defined
```

The remapper is telling us that it can remap data from a daily time frequency `with 731 incoming timesteps (731 days)` to monthly time frequency with `24 outgoing timesteps (24 months)`.

The remapping weights are stored as a sparse matrix (following the Coordinate List (COO) layout) in the `weights` variable:

[7]:
```
remapper.info.weights.data
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-7-b2872189b48a> in <module>
----> 1 remapper.info.weights.data

NameError: name 'remapper' is not defined
```

[8]:
```
remapper.info.weights.data.todense()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-8-609d9ea0fe7a> in <module>
----> 1 remapper.info.weights.data.todense()

NameError: name 'remapper' is not defined
```

The outgoing time bounds are stored in the `outgoing_time_bounds` variable:

[9]:
```
remapper.info.outgoing_time_bounds
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-9-a25bd9ad4eb6> in <module>
----> 1 remapper.info.outgoing_time_bounds

NameError: name 'remapper' is not defined
```

More information about the incoming and outgoing time axes is stored in the `attrs` section:

[10]:
```
remapper.info.attrs
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-10-e5503ddef281> in <module>
----> 1 remapper.info.attrs

NameError: name 'remapper' is not defined
```

2.2.3 Performing remapping (resampling)

Now that we have an instance of the `Remapper` object, we can tell `xgriddedaxis` to convert data from the incoming time axis to the outgoing (destination) axis.

[11]:
```
tmin_out = remapper.average(ds.tmin)
tmin_out
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-11-d9de4c008527> in <module>
----> 1 tmin_out = remapper.average(ds.tmin)
      2 tmin_out

NameError: name 'remapper' is not defined
```

2.2.4 Check results

[12]:
```
tmin_out.isel(time=0).plot(robust=True)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-12-6ba65521348e> in <module>
----> 1 tmin_out.isel(time=0).plot(robust=True)

NameError: name 'tmin_out' is not defined
```

2.2.5 Check broadcasting over extra dimensions

The remapping should affect the time dimension **only**. We can check that `xgriddedaxis` tracks coordinate values over extra dimensions

[13]:
```
ds.lat
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-13-21970646b2de> in <module>
----> 1 ds.lat

NameError: name 'ds' is not defined
```

[14]:
```
# Passes if the output is exactly the same as the input
xr.testing.assert_identical(ds.lat, tmin_out.lat)
xr.testing.assert_identical(ds.lon, tmin_out.lon)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-14-22e9fd95de44> in <module>
      1 # Passes if the output is exactly the same as the input
----> 2 xr.testing.assert_identical(ds.lat, tmin_out.lat)
      3 xr.testing.assert_identical(ds.lon, tmin_out.lon)

NameError: name 'xr' is not defined
```

We can plot the time series at a specific location, to make sure the broadcasting is correct:

[15]:
```
ds.tmin.sel(lat=-90, lon=-180).plot()
tmin_out.sel(lat=-90, lon=-180).plot(color='red')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-15-ebffd2cf2e5d> in <module>
----> 1 ds.tmin.sel(lat=-90, lon=-180).plot()
      2 tmin_out.sel(lat=-90, lon=-180).plot(color='red')

NameError: name 'ds' is not defined
```

[16]:
```
%load_ext watermark
%watermark -v -m -g -p xarray,xgriddedaxis,cftime,pandas
```

```
CPython 3.7.3
IPython 7.13.0

xarray 0.15.1
xgriddedaxis 0.0.post43
cftime 1.1.2
pandas 1.0.3

compiler   : GCC 7.4.0
system     : Linux
release    : 5.0.0-1032-azure
machine    : x86_64
processor  : x86_64
CPU cores  : 1
interpreter: 64bit
Git hash   : b4e6bd9215269111a1dd602831c9d6ecec6f768d
```

2.3 API Reference

This page provides an auto-generated summary of xgriddedaxis's API. For more details and examples, refer to the relevant chapters in the main part of the documentation.

2.3.1 Axis

xgriddedaxis.Axis

2.3.2 Remapper

xgriddedaxis.Remapper()

class xgriddedaxis.**Remapper**

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Index

R