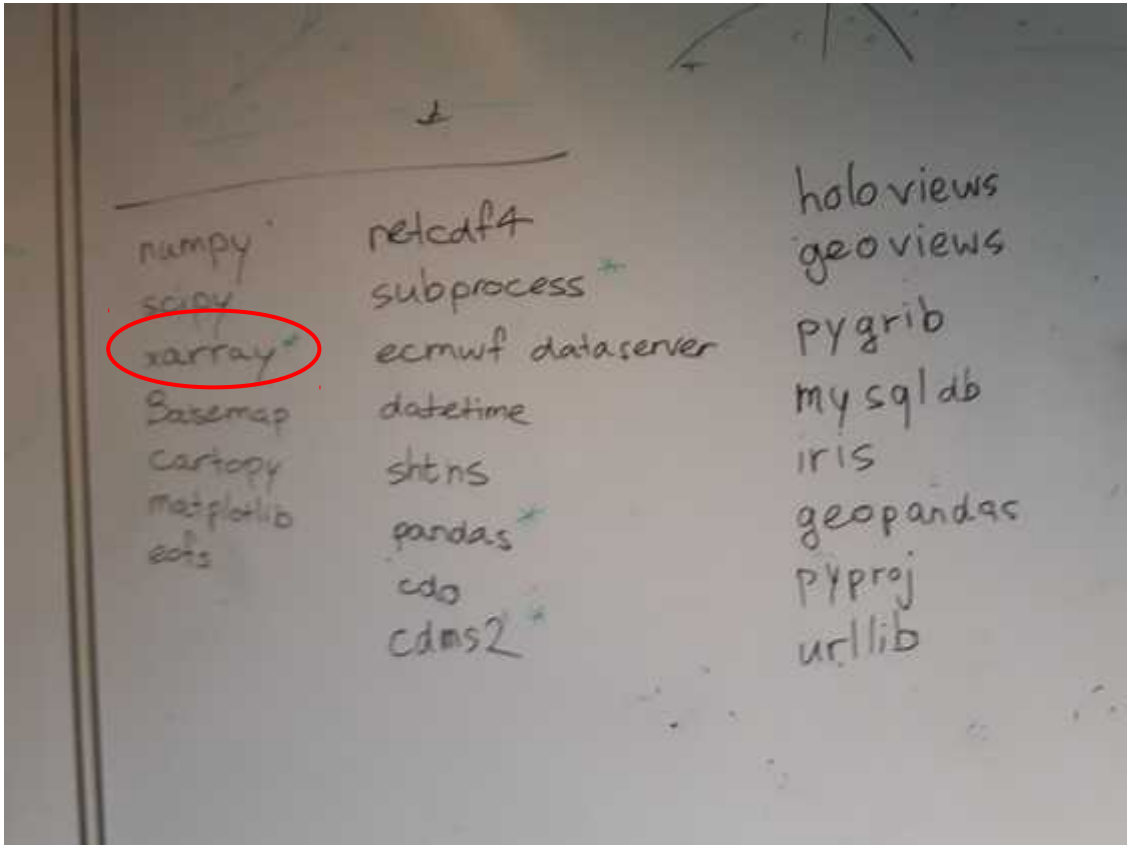
Python Working Group (PWG)



Topics today

- Python 2 > Python 3
- Introduction to Xarray
- Any ideas on topics for PWG
- What’s / who’s / when’s next
- AOB

Python 2 > Python 3

Xarray drops support for 2.7 in end of 2018

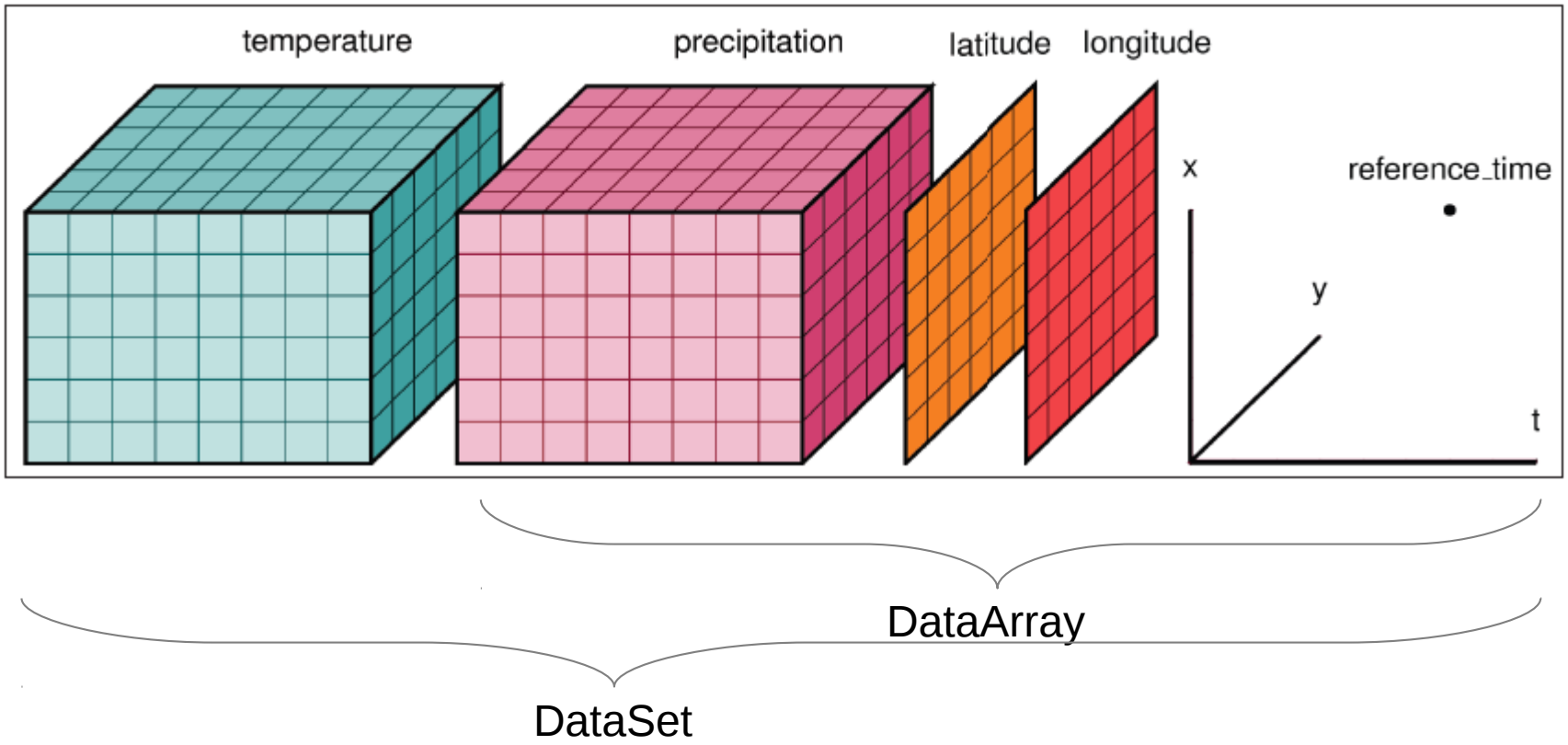
How to switch to 3.6?

- 2to3 (alias 2to3='/lib64/python3.6/Tools/scripts/2to3')
- reindent.py (reindent='/lib64/.. .. /scripts/reindent.py')
- ‘2to3 -w script.py’ rewrite script.py in python3 and make a backup file (‘script.py.bak’)
- Run (i)python by typing (i)python3
- Locally installed packages by pip have to be reinstalled

Xarray | Key features

- Open source project and python package
- Labelled data analysis on N-dimensional data
- Xarray.Dataset is an in-memory representation of a netcdf file
- Built on top of Pandas, so lots of functions from pandas available in xarray
- Main use in weather and climate research, but also other fields

Xarray | Key features



Xarray | open / inspect / select / indexing

- Load netcdf file

```
In [5]: ds = xr.tutorial.load_dataset('air_temperature')

In [6]: ds

Out[6]: <xarray.Dataset>
        Dimensions:  (lat: 25, lon: 53, time: 2920)
        Coordinates:
          * lat      (lat) float32 75.0 72.5 70.0 67.5 65.0 62.5 60.0 57.5 ...
          * lon      (lon) float32 200.0 202.5 205.0 207.5 210.0 212.5 ...
          * time     (time) datetime64[ns] 2013-01-01 2013-01-01T06:00:00 ...
        Data variables:
            air      (time, lat, lon) float64 241.2 242.5 243.5 244.0 ...
        Attributes:
            Conventions:  COARDS
            title:        4x daily NMC reanalysis (1948)
            description:  Data is from NMC initialized reanalysis\n(4x/day)...
            platform:     Model
            references:   http://www.esrl.noaa.gov/psd/data/gridded/data.nc...
```

xr.load_dataset()
xr.to_netcdf()

Xarray | open / inspect / select / indexing

- Open netcdf file
  - xr.open_dataset(‘dir/data.nc’) > local file
  - xr.open_dataset(‘http://somewhere/data.nc’) > opendap
  - xr.to_netcdf(‘dir/data_new.nc’) > save to netcdf
  - xr.open_datset(‘dir/data.nc’,chunck={‘time’:10})
    - If too large for memory, load data in chunks
- Open multiple netcdf files
  - xr.open_mfdataset(‘dir/data*.nc’) > load multiple files
    - Data automatically chunked per file

Xarray | open / inspect / select / indexing

- Inspecting dataset / DataArray
  - ds.air > dataArray of air
  - ds.air.dims > dimensions of ‘air’ [‘time’,’lat’,’lon’]
  - ds.lat > dataArray of latitude
  - ds.air.values > numpy array of air
  - ds.air.attrs > attributes of air

Xarray | open / inspect / select

- Selecting / indexing
  - Selecting by label:
    - ds.air.sel(time=’2007-10-01’)
    - ds.air.sel(lat=40.,lon=40.,method=’nearest’)
    - ds.air.loc[‘2007-10-01’:’2010-10-01’,45.5,60.]
  - Selecting by index: ds.air.isel(time=-1)
    - ds.air.isel(time=-1,lat=10,lon=3)
    - ds.air.isel(time=slice(None,10)), equivalent to numpy [:10]
    - ds.air[:10,5,5:10], similar to numpy
  - All returned objects are dataArrays

Xarray | operations and computations

- new_ds = ds – 3
- All of numpy scipy universl functions (ufunc)
  - np.abs(ds), np.sin(ds), etc etc
- Aggregation functions
  - ds.mean(dim='time')
  - ds.sum(dim='lat')
- Rolling window operations (e.g. running mean)
  - ds.rolling(time=3,min_periods=2,dropna=True).mean(dim='time')

Xarray | Groupby and split-apply-combine

- Groupby
  - Climatology of monthly data
    - clim_mon = ds.air.groupby('time.month').mean('time')
  - Seasonal climatology
    - clim_seas = ds.air.groupby('time.season').mean('time')
  - Monthly anomalies
    - anom = ds.air.groupby('time.month') - clim_mon
  - groupby_bins (easy binning of data)
    - ds.groupby_bins('lat',[-90,-45,0,45,90]).sum()

Xarray | Groupby and split-apply-combine

- Groupby and Apply
  - Standardize data
    - def standardize(x):

      return (x-x.mean()) / x.std()

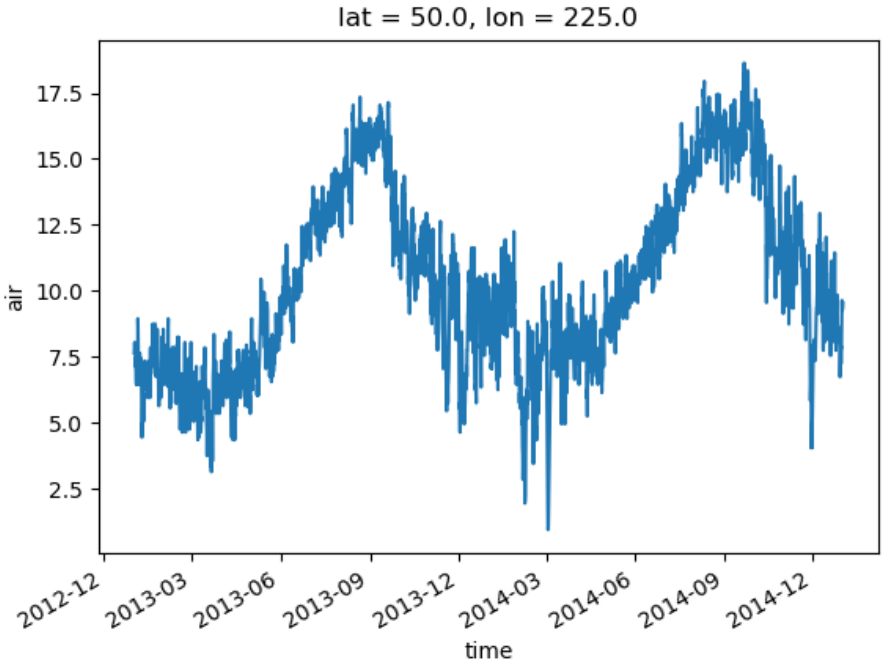
      ds.groupby('time.month').apply(standardize)
  - Apply in combination with lambda, e.g. get anomalies relative to longitudal mean
    - ds.groupby('lon').apply(lambda x: x-x.mean(),shortcut = False)

Xarray | Resampling / Regridding

- Change time frequency of data
  - Daily to monthly
    - ds.resample(‘1M’,dim=’time’)
  - Many more examples
  - Mainly time dimension resampling
- Spatial regridding
  - Pangeo? Near future..

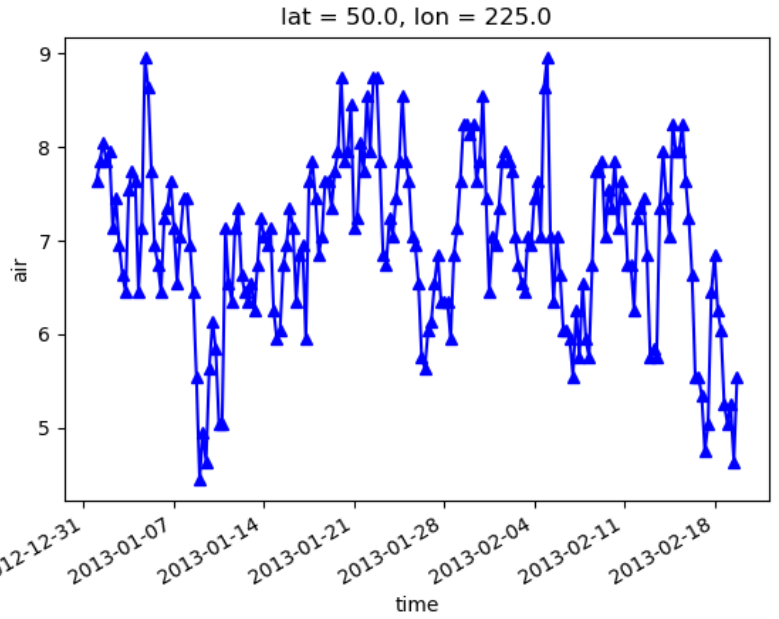
Xarray | Graphics / plotting

- Easy plotting, wrapper for matplotlib



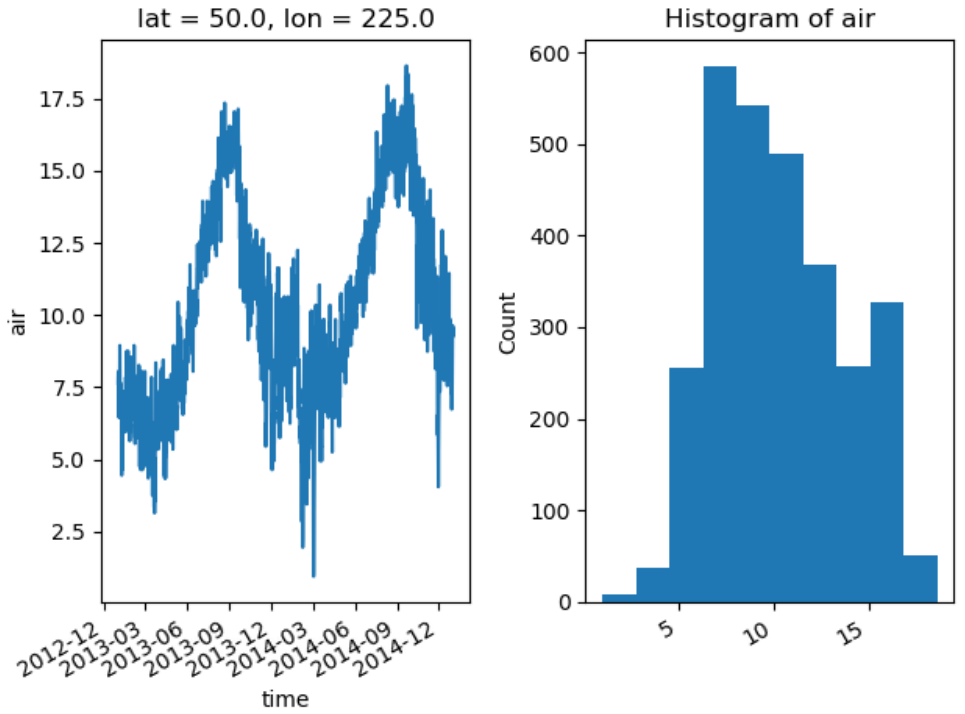

```
air1d = air.isel(lat=10, lon=10)
air1d.plot()
```

Xarray | Graphics / plotting



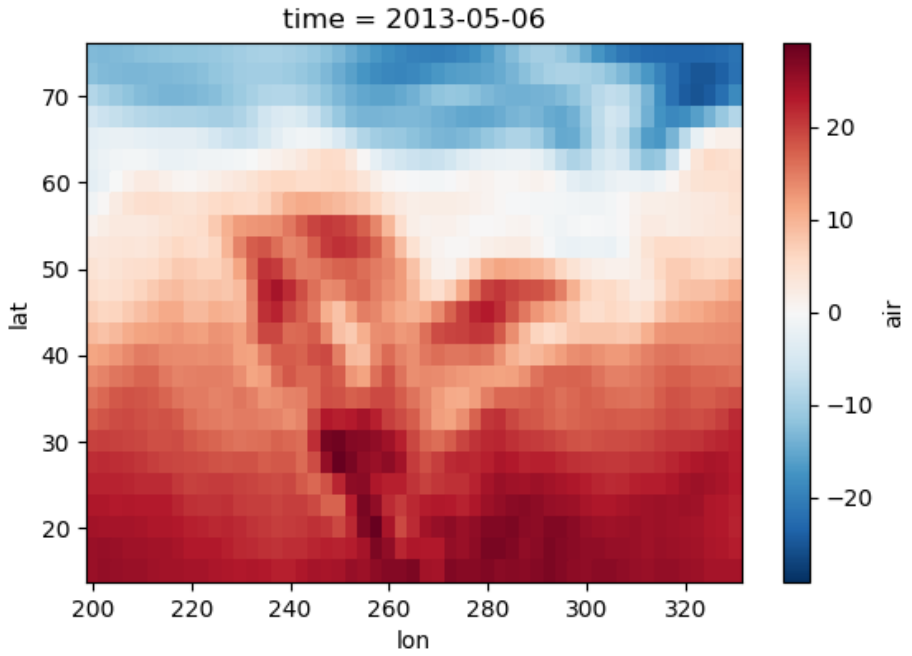

```
air1d[:200].plot.line('b-^')
```

Xarray | Graphics / plotting



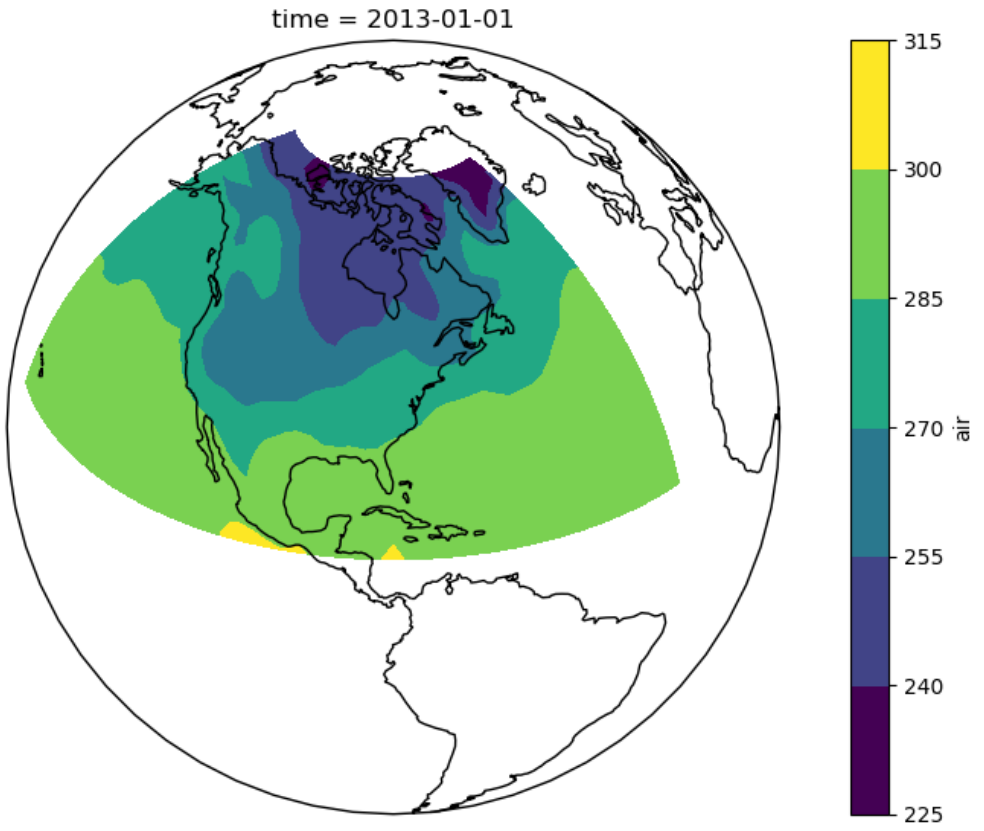

```
fig, axes = plt.subplots(ncols=2)
air1d.plot(ax=axes[0])
air1d.plot.hist(ax=axes[1])
plt.tight_layout()
plt.show()
```

Xarray | Graphics / plotting



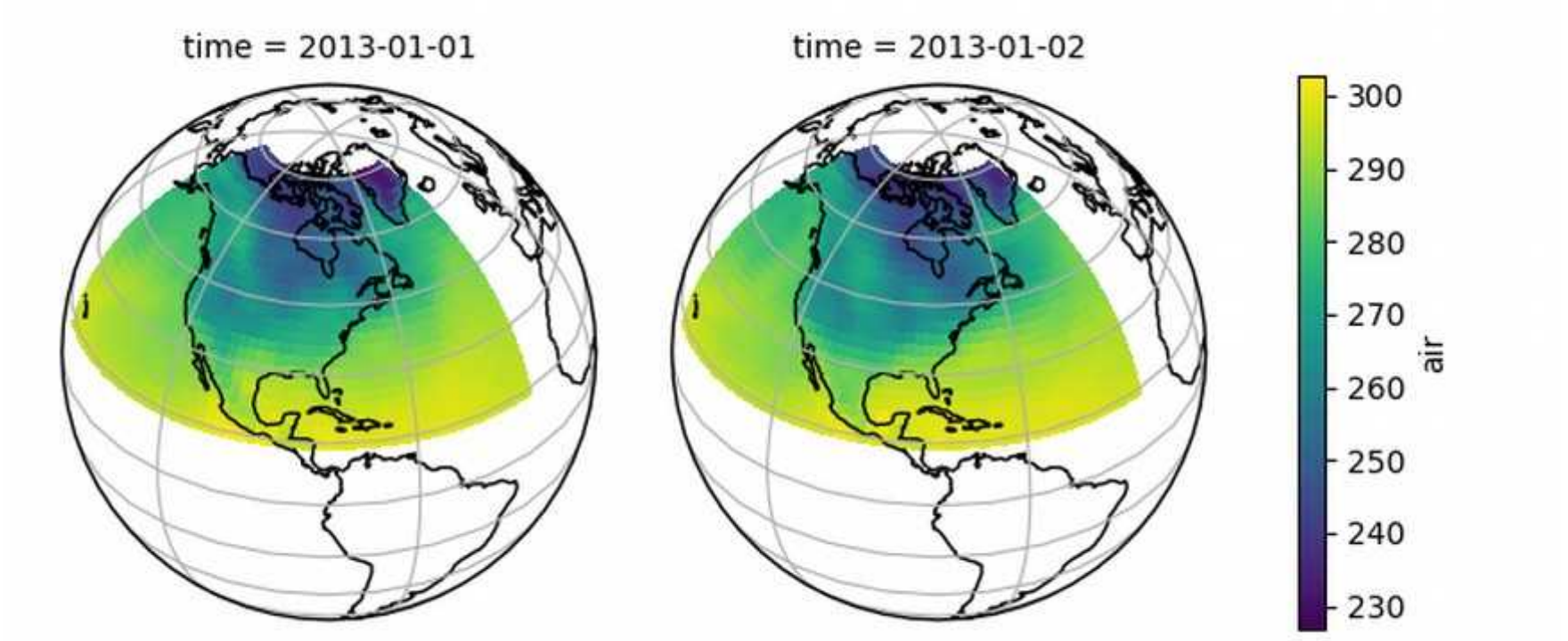

```
ds.air.isel(time=500).plot()
plt.show()
```

Xarray | Graphics / plotting




```
import cartopy.crs as ccrs

air = xr.tutorial.load_dataset(
    'air_temperature').air

ax = plt.axes(projection=
    ccrs.Orthographic(-80, 35))

air.isel(time=0).plot.contourf(ax=ax,
    transform=ccrs.PlateCarree())

ax.set_global(); ax.coastlines()
```

Xarray | Graphics / plotting



Xarray | Compatibility

- Eofs (eof analysis), Iris and CDAT (other climate data tools)
- Pangeo (climate data tools suitable for ‘big data’)
  - Work in progress
- Xgcm (easily change from staggered / non-staggered grid)
- Seaborn (statistics)
- Cartopy (plotting)
- Many more, quickly growing

AOB’s?

Any other business?

- How to setup python at your workstation
  > Laurens
- …..

Next meeting

- When?
- What is on the agenda?
- Who?