National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

The cf-python software library

5th December 2014

Data, metadata and the CF conventions

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

data

```
3.56, 6.78, ..., -0.32, 1.86
```

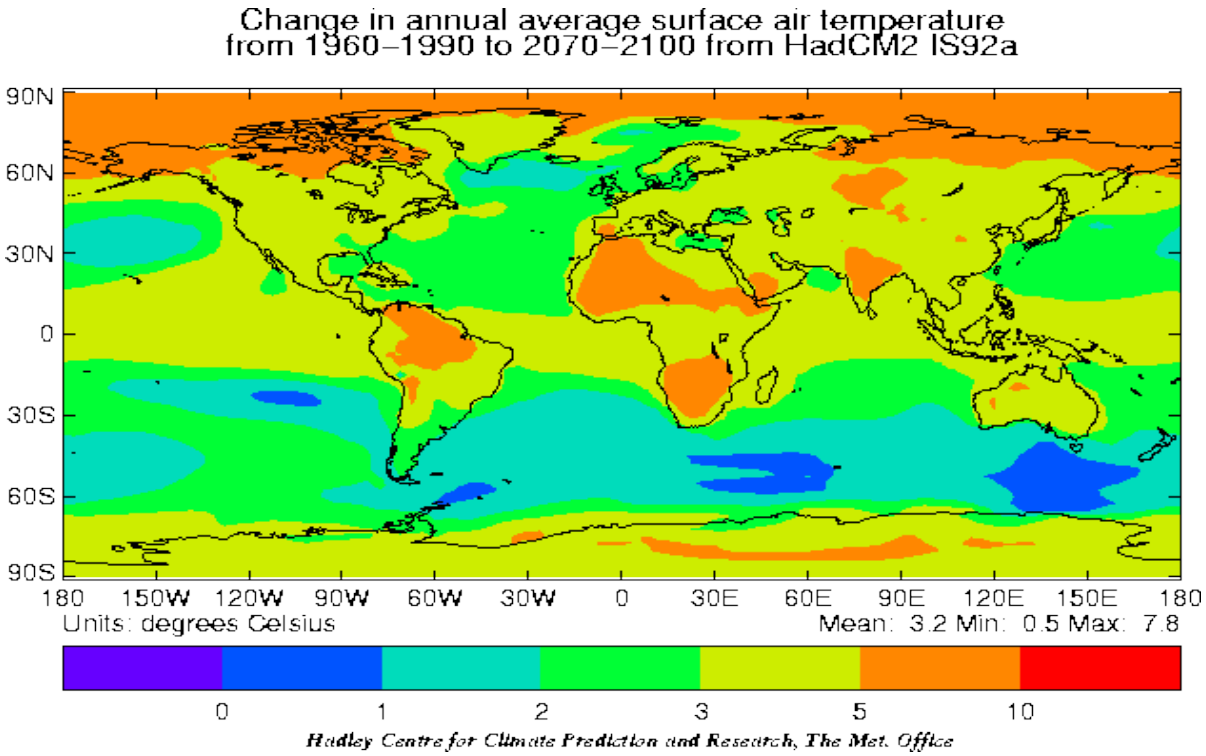
Data, metadata and the CF conventions

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

data

```
3.56, 6.78, ..., -0.32, 1.86
```

data + metadata



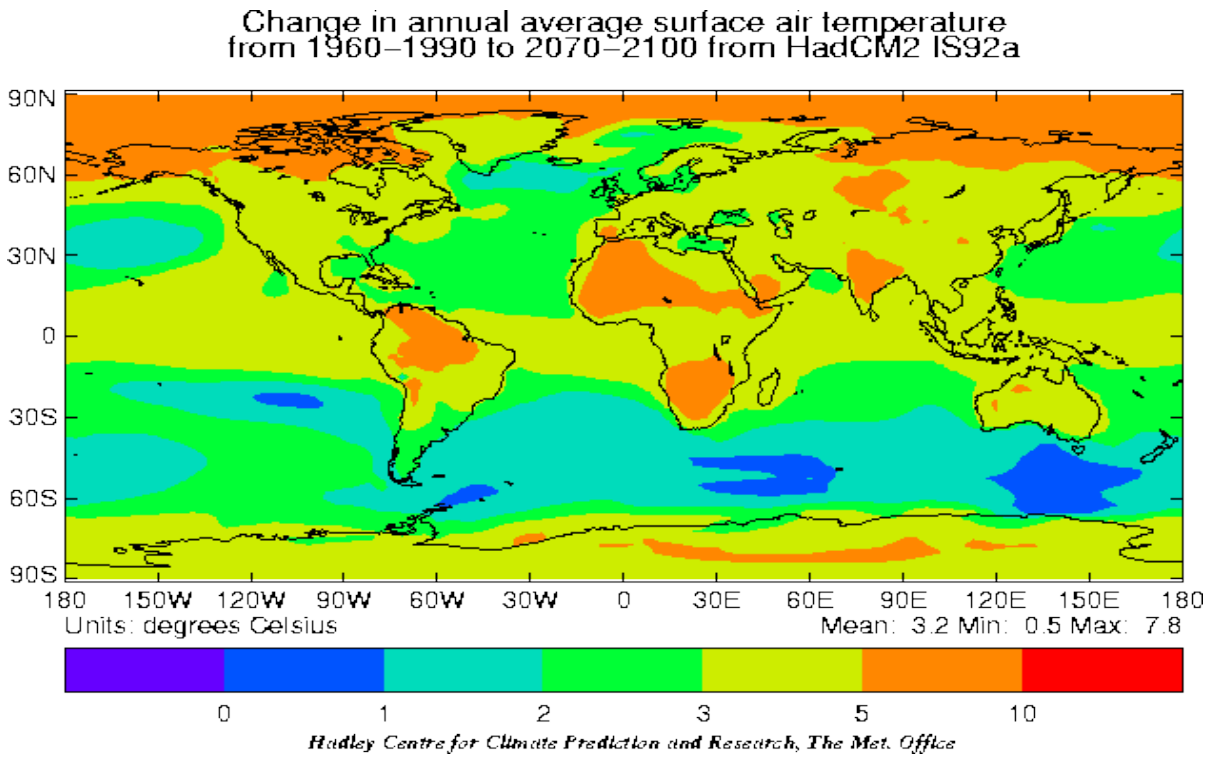
Data, metadata and the CF conventions

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

data

```
3.56, 6.78, ..., -0.32, 1.86
```

data + metadata



- CF conventions: A recognized standard for encoding **C**limate and weather **F**orecast metadata to support interoperability for earth science data from different sources
- Particularly suited to climate model outputs

The cf-python library

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

- An open source Python software package developed by NCAS to fully utilize the CF conventions

- Reads, manipulates and writes datasets from PP and netCDF files
  - and UM output fields files by early 2015

cfpython.bitbucket.org

The cf-python library

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

Read netCDF, netCDF and PP (and soon UM files) format files.

Create fields.

Write fields to netCDF files on disk.

Aggregate collections of fields into a single field.

Create, delete and modify a field's data and metadata.

Select fields according to their metadata.

Subspace a field's data to create a new field.

Perform broadcastable, metadata-aware arithmetic, comparison and trigonometric operations with fields.

Collapse fields by statistical operations.

Sensibly deal with date-time data.

Visualization with **cfplot**

Copes with larger-than-memory datasets

An example with big data

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

On a machine with a maximum of **32G** of memory …

Manipulate a larger-than-memory 4D field stored across multiple PP files and save it as a single CF-netCDF file:

```
$ du -h xgdriaa.pd*.pp
40G     xgdriaa.pdk9498.pp
40G     xgdriaa.pdk949a.pp
40G     xgdriaa.pdk949c.pp

$ python
>>> import cf
>>> x = cf.read('xgdriaa.pd*.pp', match='mass_fraction_of_cloud_ice_in_air')
>>> x
<CF Field: mass_fraction_of_cloud_ice_in_air(model_level_number(70), time(6), latitude(2890), longitude(10222))>
>>> y = cf.collapse(x, 'max', 'time')
>>> y
<CF Field: mass_fraction_of_cloud_ice_in_air(model_level_number(70), time(1), latitude(2890), longitude(10222))>
>>> cf.write(x, 'mass_fraction_of_cloud_ice_in_air.nc')
>>> cf.write(y, 'MAX_mass_fraction_of_cloud_ice_in_air.nc')

$ du -h *mass_fraction_of_cloud_ice_in_air.nc
53G     mass_fraction_of_cloud_ice_in_air.nc
9G      MAX_mass_fraction_of_cloud_ice_in_air.nc
```

- cf-python is the library behind the **cfa** and **cfdump** tools for converting UM output to netCDF format and inspecting file contents

- The cf-python software is a useful python analysis tool