Where next?

ISC summary

CONGRATULATIONS!

You did it!

We learnt about python basics

- Basics and control flow, booleans
- Lists, slicing and tuples
- Input/output
- Strings and text processing
- Functions, libraries and scripts
- Sets and dictionaries
- Errors and debugging
- OOP

And more advanced libraries

- Numpy: arrays and masked arrays
- Matplotlib: plotting
- NetCDF (& CF): reading and writing data

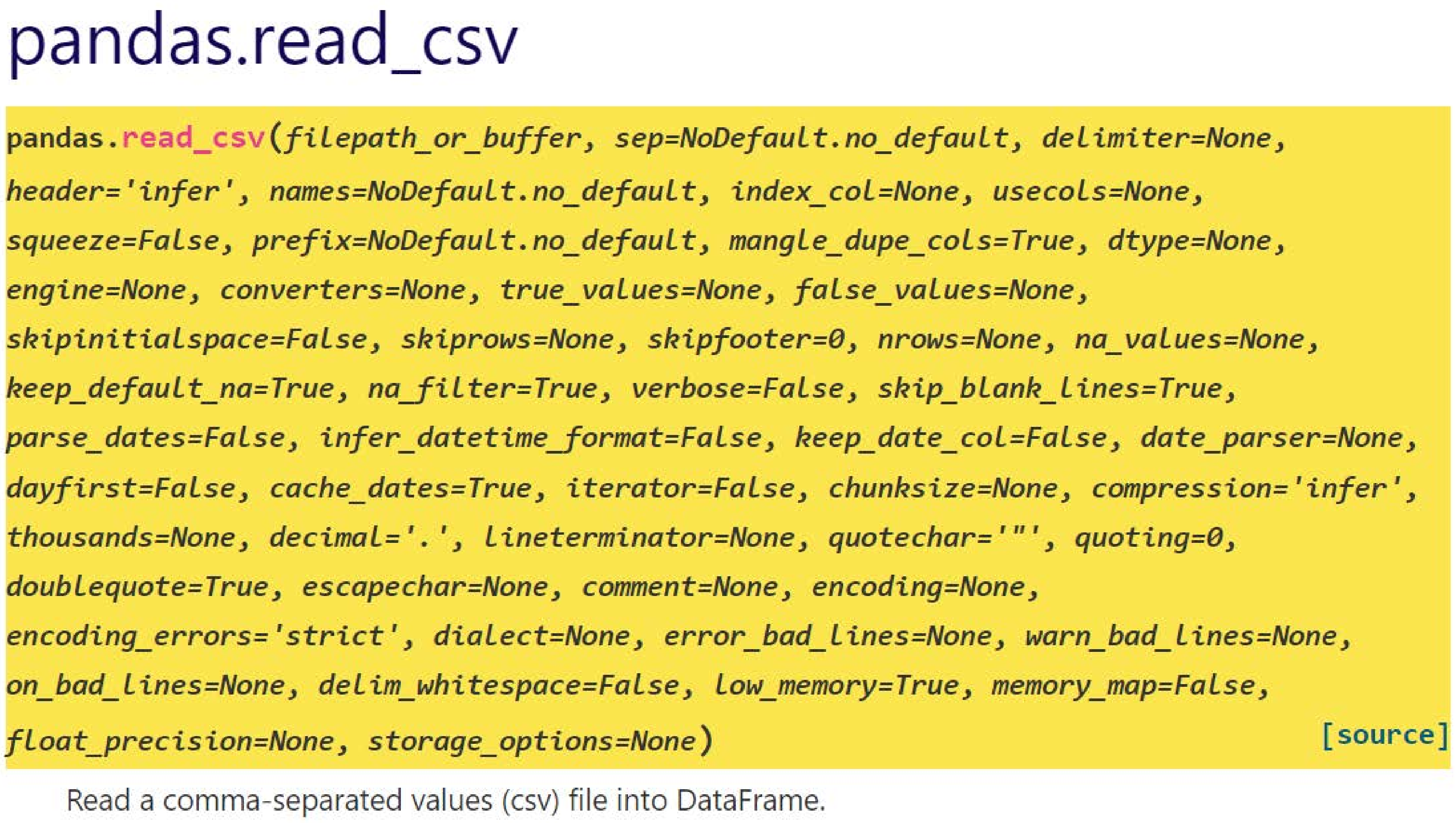
We didn't have time for…

https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html

- For tabulated data (e.g. Excel on steroids)
- Powered by Numpy
- Fast and efficient
- Can be integrated with Dask (for parallel / delayed / out-of-memory operations)

pandas for tabulated data

Sorting by an axis:

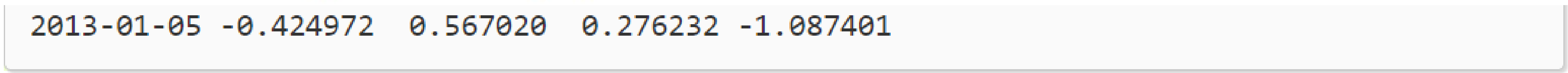


pandas.read_csv

```
pandas.read_csv(filepath_or_buffer, sep=NoDefault.no_default, delimiter=None,
header='infer', names=NoDefault.no_default, index_col=None, usecols=None,
squeeze=False, prefix=NoDefault.no_default, mangle_dupe_cols=True, dtype=None,
engine=None, converters=None, true_values=None, false_values=None,
skipinitialspace=False, skiprows=None, skipfooter=0, nrows=None, na_values=None,
keep_default_na=True, na_filter=True, verbose=False, skip_blank_lines=True,
parse_dates=False, infer_datetime_format=False, keep_date_col=False, date_parser=None,
dayfirst=False, cache_dates=True, iterator=False, chunksize=None, compression='infer',
thousands=None, decimal='.', lineterminator=None, quotechar='"', quoting=0,
doublequote=True, escapechar=None, comment=None, encoding=None,
encoding_errors='strict', dialect=None, error_bad_lines=None, warn_bad_lines=None,
on_bad_lines=None, delim_whitespace=False, low_memory=True, memory_map=False,
float_precision=None, storage_options=None)
```

[source]

Read a comma-separated values (csv) file into DataFrame.



```
2013-01-05 -0.424972  0.567020  0.276232 -1.087401
```

We didn't have time for...

cf python

Iris

xarray

- python library: handles NC, PP
- integrates with matplotlib/cartopy (*cfplot*)
- subsetting, averaging, regridding etc.

- python library: handles NC, PP, GRIB
- integrates with matplotlib/cartopy
- subsetting, averaging, regridding etc.

- python library: handles NC, Zarr
- integrates with matplotlib/cartopy
- subsetting, averaging, regridding etc.

https://ncas-cms.github.io/cf-python/

http://xarray.pydata.org/en/stable/

https://scitools-iris.readthedocs.io/en/latest/

cf python

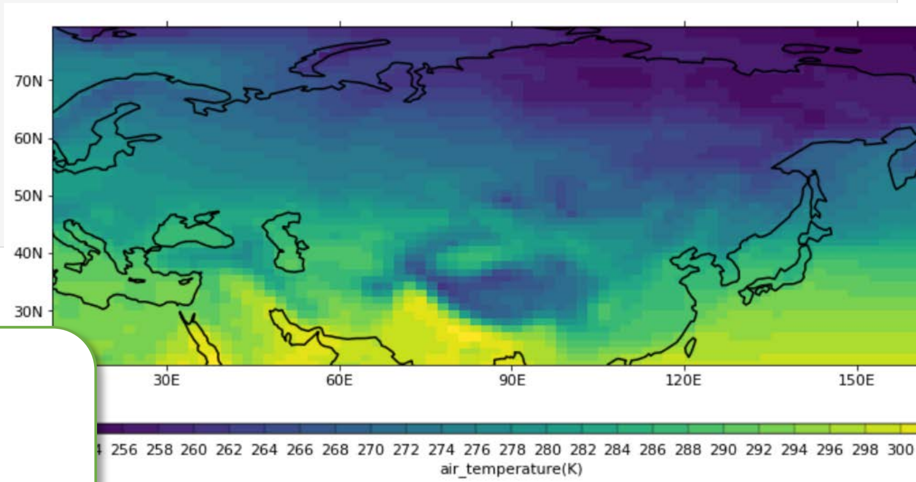
```python
# Import cf and cfplot
import cf
import cfplot as cfp

# Read the data from NetCDF files
f = cf.read(file_pattern)[0]

# Sub-select a spatial region
subset = f.subspace(longitude=cf.wi(10, 170, 'degrees'), latitude=cf.wi(20, 80, 'degrees'))

# Calculate an average over all time steps
average = subset.collapse('T: mean')

# Plot a map of the temporal average
cfp.con(average, blockfill=True, lines=False)
```



Look out for the next NCAS course: https://ncas.ac.uk/study-with-us/data-analysis-tools/

SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

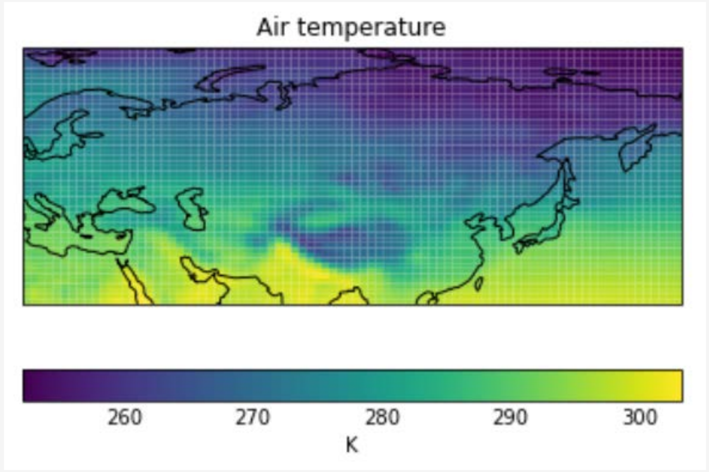
```python
# Import iris and required libraries
import iris
from iris.util import equalise_attributes
import matplotlib.pyplot as plt
import iris.quickplot as qplt

# Read the data from NetCDF files
f = iris.load(file_pattern)
equalise_attributes(f)
merged = f.concatenate_cube()

# Sub-select a spatial region
subset = merged.extract(iris.Constraint(longitude=lambda cell: 10 <= cell <= 170) &
                        iris.Constraint(latitude=lambda cell: 20 <= cell <= 80))

# Calculate an average over all time steps
average = subset.collapsed('time', iris.analysis.MEAN)

# Plot a map of the temporal average
plot = qplt.pcolor(average)
plt.gca().coastlines()
plt.show()
```



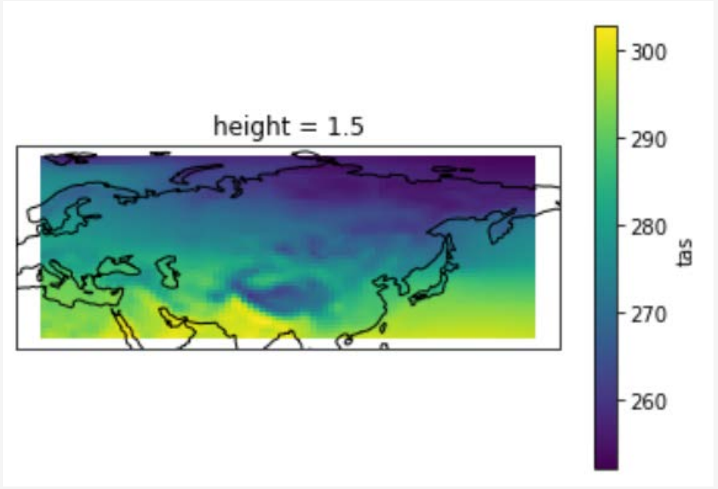
```python
# Import xarray and required libraries for plotting
import xarray as xr
import cartopy.crs as ccrs
import matplotlib.pyplot as plt

# Read the data from NetCDF files
f = xr.open_mfdataset(file_pattern)

# Sub-select a spatial region
subset = f.sel(lon=slice(10, 170), lat=slice(20, 80))

# Calculate an average over all time steps
average = subset.mean(dim="time")

# Plot a map of the temporal average
ax = plt.subplot(projection=ccrs.PlateCarree())
average['tas'].plot()
ax.coastlines()
```



Where to go next?

- The best way to learn is to play...
- Get python installed on your desktop/laptop (on Windows, MAC or Linux).
- Read/write files
- Move/copy files/folders using scripts
- Make some nice plots
- Start a small project:
  - Write down the steps – *think algorithmically!*
  - Convert the steps to pseudo-code – *test ideas/code as you go*
  - Convert to Python functions / classes / scripts / modules

Places to learn more/practice

- Codecademy site has great exercises:

  https://www.codecademy.com/learn/learn-python

- Free Code Camp: https://www.freecodecamp.org/learn/scientific-computing-with-python/

- Python website has documentation for all the standard library modules (sys, maths, etc.):

  https://docs.python.org/

Places to learn more/practice

- Python website also has tutorials:

  https://docs.python.org/3/tutorial/

- Software-Carpentry web site hosts videos and presentations and lots more:

  https://software-carpentry.org/lessons/

ISC course materials

Full version of the modules and exercises/solutions:

https://github.com/ncasuk/ncas-isc

Centre for Environmental Data Analysis
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

Good luck!