Extracting and visualising NASA's MERRA-2 data

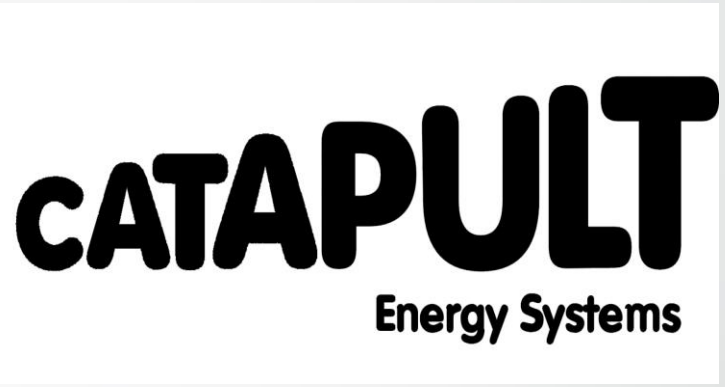
Angeliki (Kelly) Loukatou

30/04/2020

PyData Manchester

Some personal details

- Graduated as Power Systems Engineer from NTUA
- Completed PhD in the School of Mathematics of the UoM
- Working as Modelling Analyst in ESC

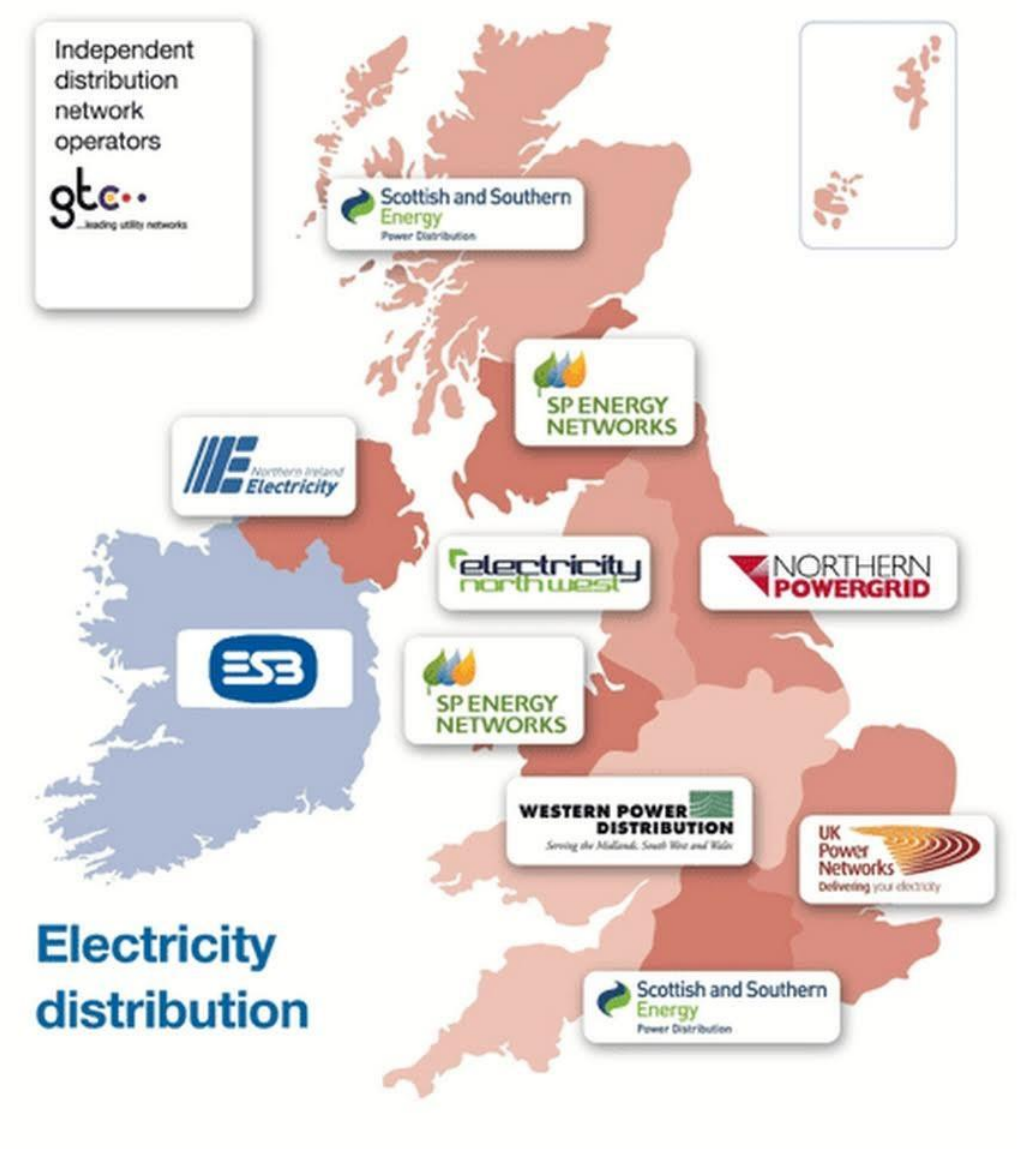


Before

2014-2018

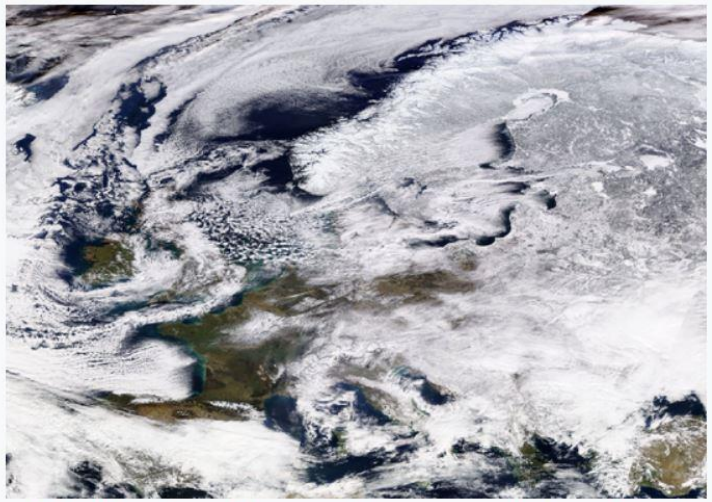
Now

Project details



Examine the impact of electric-vehicle and heat-pump penetration on power networks in future years.

Anticyclone Hartmut
Beast from the East

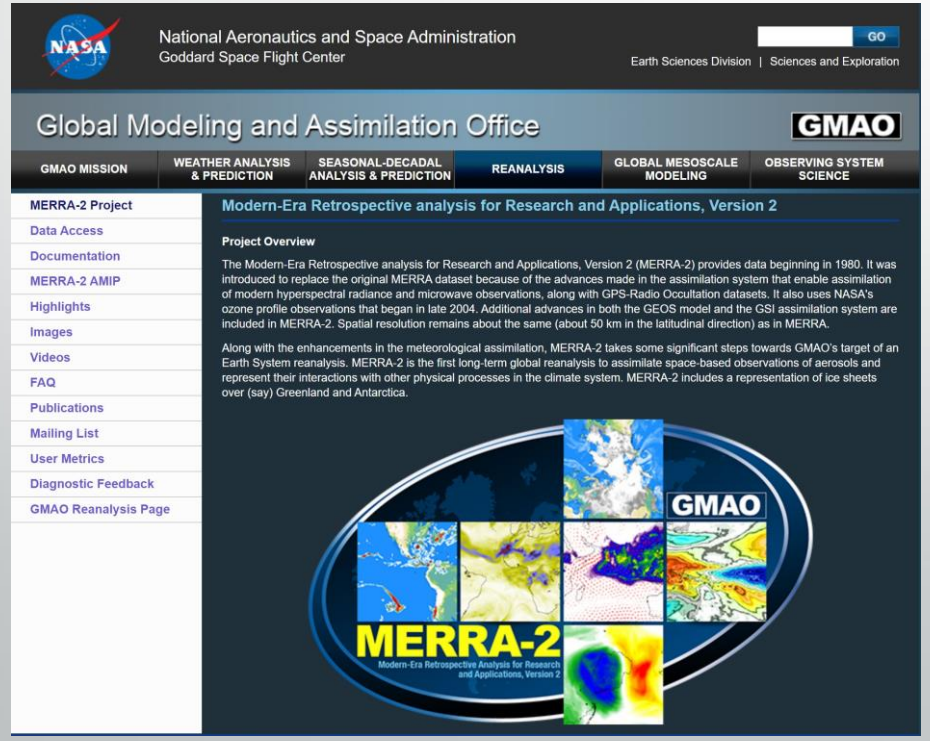


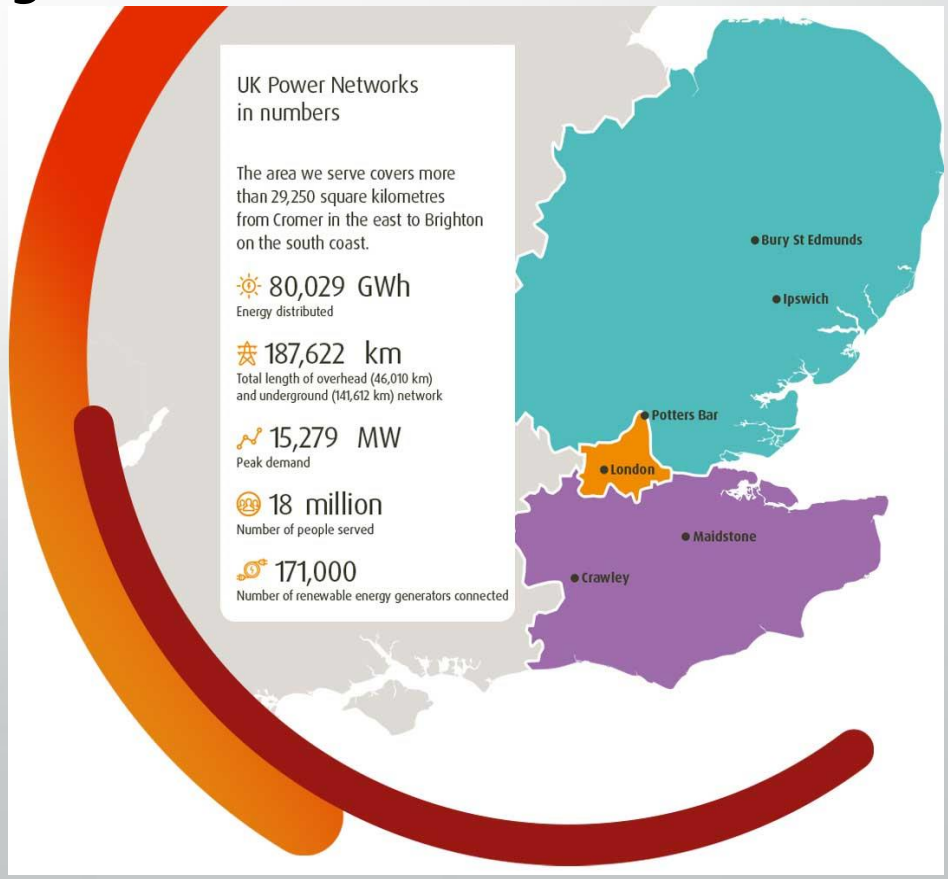
Satellite view showing Europe, including Great Britain and Ireland partially covered in snow under the lines of the strong cold wave on 27 February 2018

Temperature data

Various resources do exist but the project requirements were very specific

- **Access:** Open
- **Time resolution:** Hourly data for the period January-February 2018
- **Spatial resolution:** UK data around East Anglia





Extracting the data

Data are given in a less common format, i.e. **NetCDF-4** (Network Common Data Form, version 4) -> first version of this format was created back in 1980s and the purpose was to share data among atmospheric scientists

1. Data are managed by the NASA Goddard Earth Sciences (GES) **Data and Information Services Center (DISC)**, access them **here**
2. Check the documentation of MERRA-2 data **here** and find the (table) data of interest, i.e. **Table 'tavg1_2d_slv_Nx', Single Level Diagnostics**
3. Filter for hourly data (temporal resolution) and atmospheric temperature (subject) from **here**
4. Navigate into the **'Online Archive'** and select data for specific days, months, years from **here** or use this guidance to automatically subset and download data from here

Licence

- You need to create an **Earthdata account** **here**
- You need to authorise and associate the **'NASA GESDISCS DATA ARCHIVE'** repository to your account (**Applications -> Authorized Applications**)
- For questions, please use this e-mail: merra-questions@lists.nasa.gov

Retrieving temperature data from nc4 files

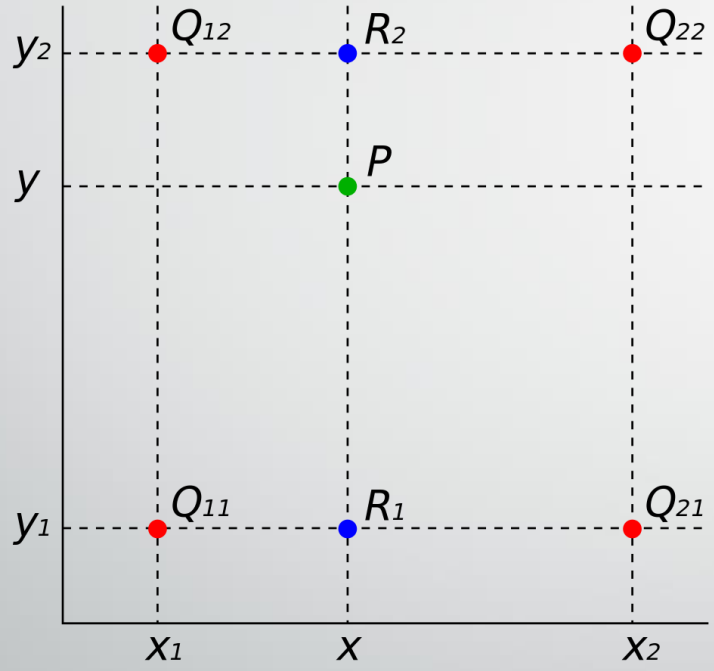
- Online conversion to excel format -> takes time, not free, not efficient
- netCDF4 module from Python
- Files are big, ~400MB per day of data!

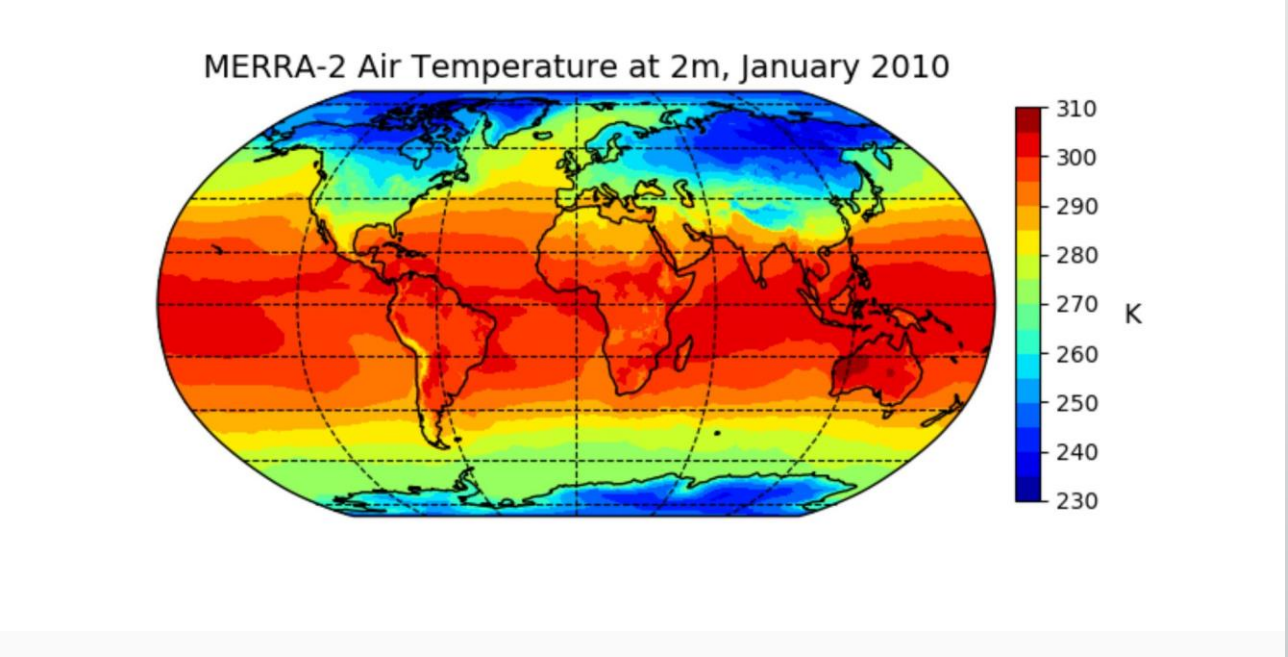
```
from netCDF4 import Dataset

# read the data
filename = 'MERRA2_400.tavg1_2d_slv_Nx.20180222.nc4'
data = Dataset(filename, "r", format="NETCDF4")
print(data)
Print(data.variables)

# extract longitude, latitude and temperature at 2 meters
lons = data.variables['lon'][:]
lats = data.variables['lat'][:]
T2M = data.variables['T2M'][:, :, :]
```

```
<class 'netCDF4._netCDF4.Dataset'>
root group (NETCDF4 data model, file format HDF5):
SouthernmostLatitude: -90.0  NorthernmostLatitude: 90.0
    WesternmostLongitude: -180.0  EasternmostLongitude: 179.375
    LatitudeResolution: 0.5  LongitudeResolution: 0.625
    DataResolution: 0.5 x 0.625
    Source: CVS tag: GEOSadas-5_12_4_p13_M2-OPS
     dimensions(sizes): lon(576), lat(361), time(24)

    variables(dimensions): float64 lon(lon), float64 lat(lat), int32 time(time), float32 CLDPRS(time,lat,lon), float32 CLDTMP(time,lat,lon), float32 DISPH(time,lat,lon), float32 H1000(time,lat,lon), float32 H250(time,lat,lon), float32 H500(time,lat,lon), float32 H850(time,lat,lon), float32 OMEGA500(time,lat,lon), float32 PBLTOP(time,lat,lon), float32 PS(time,lat,lon), float32 Q250(time,lat,lon), float32 Q500(time,lat,lon), float32 Q850(time,lat,lon), float32 QV10M(time,lat,lon), float32 QV2M(time,lat,lon), float32 SLP(time,lat,lon), float32 T10M(time,lat,lon), float32 T250(time,lat,lon), float32 T2M(time,lat,lon), float32 T2MDEW(time,lat,lon), float32 T2MWET(time,lat,lon), float32 T500(time,lat,lon), float32 T850(time,lat,lon), float32 TO3(time,lat,lon), float32 TOX(time,lat,lon), float32 TQI(time,lat,lon), float32 TQL(time,lat,lon), float32 TQV(time,lat,lon), float32 TROPPB(time,lat,lon), float32 TROPPT(time,lat,lon), float32 TROPPV(time,lat,lon), float32 TROPQ(time,lat,lon), float32 TROPT(time,lat,lon), float32 TS(time,lat,lon), float32 U10M(time,lat,lon), float32 U250(time,lat,lon), float32 U2M(time,lat,lon), float32 U500(time,lat,lon), float32 U50M(time,lat,lon), float32 U850(time,lat,lon), float32 V10M(time,lat,lon), float32 V250(time,lat,lon), float32 V2M(time,lat,lon), float32 V500(time,lat,lon), float32 V50M(time,lat,lon), float32 V850(time,lat,lon), float32 ZLCL(time,lat,lon)
```

Interpolating to get the right data

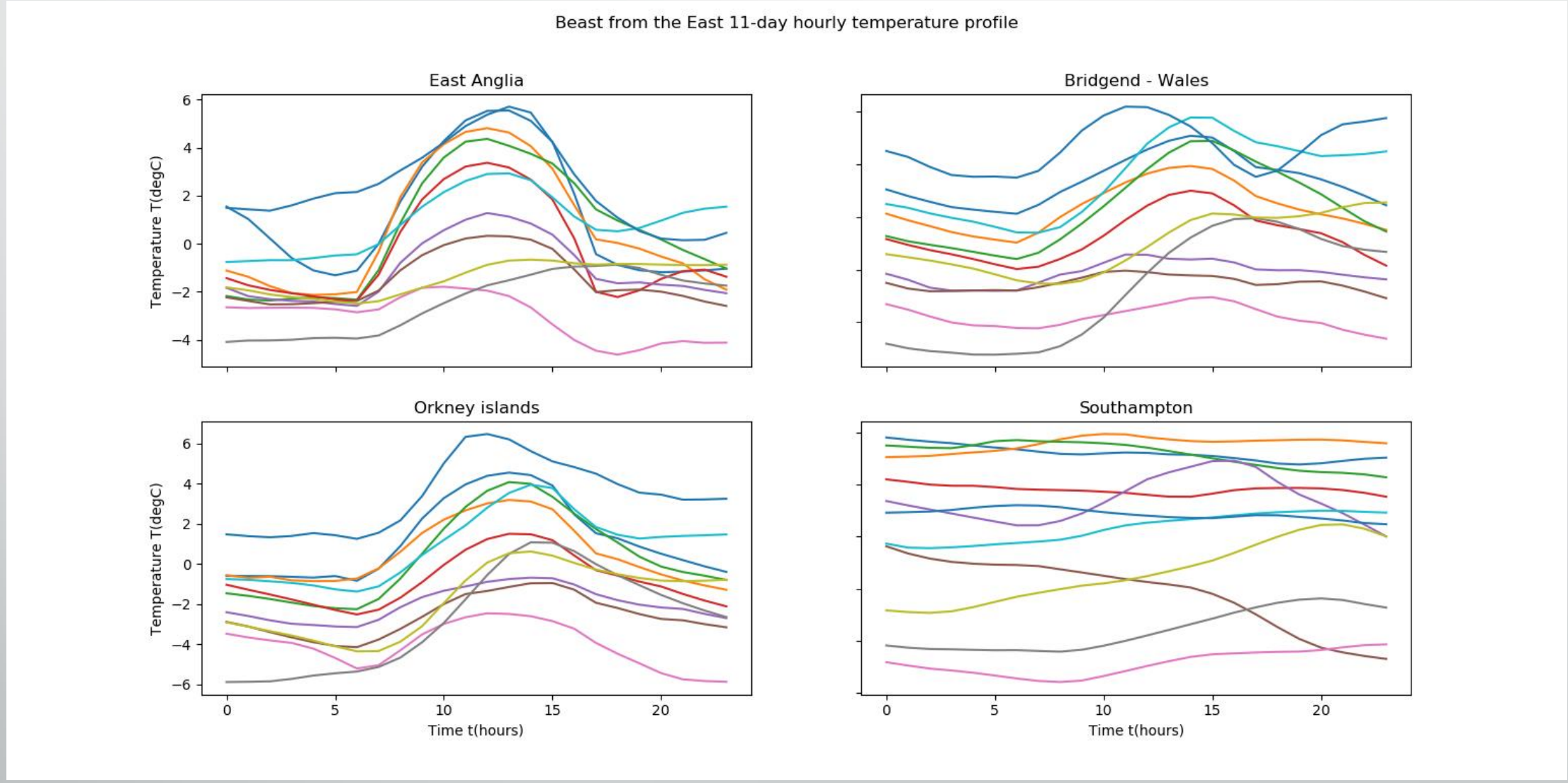
- Latitude has 0.5 degrees resolution whereas longitude 0.625!
- Bilinear interpolation (for each hour of the day) to get the temperature data at a **certain set of latitude, longitude**.





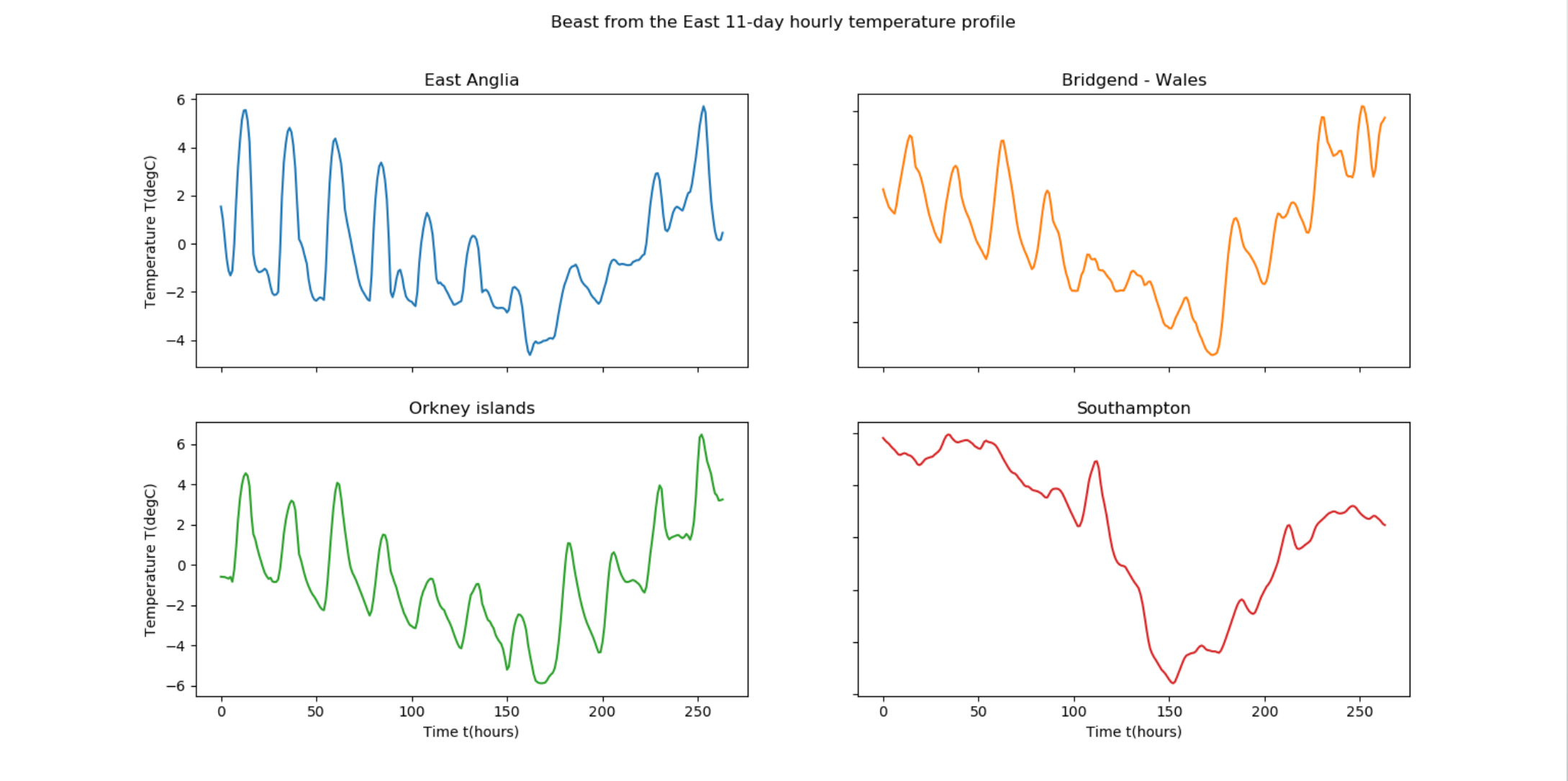
- More data manipulation -> Convert from Kelvin to degC

Visualising the data (1)



- More data manipulation -> Reconstruct to create time series of the temperature profiles

Visualising the data (2)



Thank you!

Any questions?

Angeliki Loukatou
Email: angeliki.loukatou@gmail.com
LinkedIn: https://www.linkedin.com/in/angeliki-l/
Twitter: @AngelikiLk