Hackathon - Loading the data

Loading the data

A first approach

Let's first load the data. For that end, we will use the R package `ncdf4`.

```r
library(ncdf4)
```

Loading the data is completely handle by the package. We will focus on the data from the 1st exercice.

```r
nc_data <- nc_open(filename = "temp_anomaly_ex1.nc")
nc_data
```

```
File temp_anomaly_ex1.nc (NC_FORMAT_NETCDF4):

     2 variables (excluding dimension variables):
        double tempanomaly[lat,lon,time]   (Contiguous storage)
        double global[]   (Contiguous storage)
            history: Created for the Earth Day Event 2023 at Aalborg University
            source: GISTEMP Surface Temperature Analysis
            Missing data: Coded as 32767

     3 dimensions:
        time  Size:1420
            units: months from January 1880
        lon  Size:10
            units: degrees_east
        lat  Size:10
            units: degrees_north
```

We see that the data has one variable `tempanomaly` which has 3 dimensions: latitude, longitude and time. The rest is information on the data. This data format is not the most convenient to work with so we will see how to convert it into various formats.

Loading the longitute, latitude and time

We load the variables as follow.

```
lon <- ncvar_get(nc_data, "lon")
lat <- ncvar_get(nc_data, "lat")
time <- ncvar_get(nc_data, "time")
```

As you get a 10 by 10 grid cells, `lon` at `lat` are two vectors of length 10.

```
lon
```

```
 [1] 133 135 137 139 141 143 145 147 149 151
```

```
lat
```

```
 [1] -31 -29 -27 -25 -23 -21 -19 -17 -15 -13
```

However this will be different for `time` as you get measurments for everymonths since 1880.

```
time[1:10]
```

```
 [1] 300 301 302 303 304 305 306 307 308 309
```

Notice that, as written on the netcdf object, the time is given in months from January 1880 (counting from 1) so that 300 represents the 300th month after January 1880.

Loading the temperatures

The temperatures will not be a vector but an array.

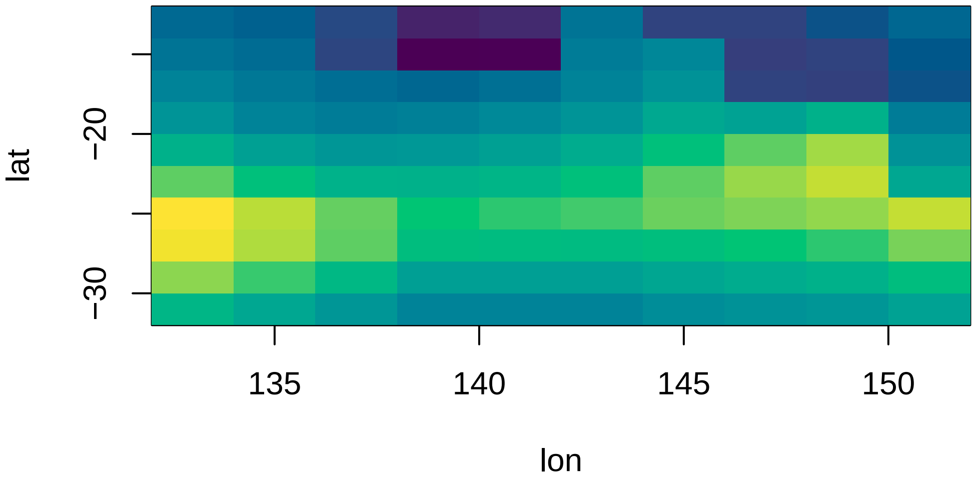
```
temp.array <- ncvar_get(nc_data, "tempanomaly")
dim(temp.array)
```

```
[1]   10   10 1420
```

The dimension is first the one of longitude, then latitude and finally the time.

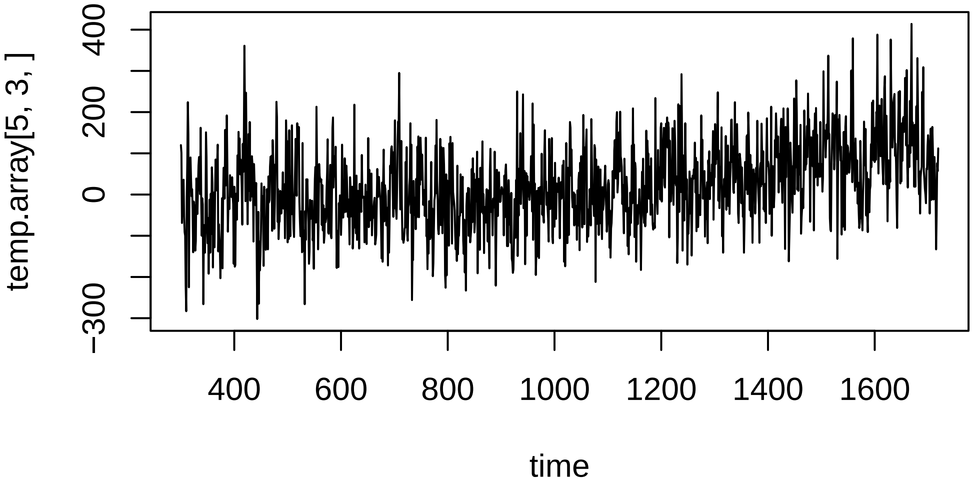
We can for example plot the data for the months 167 as follow.

```
image(lon, lat,temp.array[,,167],  col=hcl.colors(256, palette = "viridis"))
```



If you want to plot the temperatures at a given location with respect to the months, you can do as follow. I arbitrary choose the location on the cell $(5, 3)$ which is longitude 141 and latitude -27.

```
# Note that the variable "time" has been loaded
# with ncvar_get(nc_data, "time") in the code above.
plot(time, temp.array[5,3,], type="l")
```



You may then work with this array format. An alternative for the people liking the tidyverse is to look at the next section.

An tidy version, the `metR` package

Let's load the package

```
library(metR) # requires netcdf4 and PCICt
```

We may look at the file with the following function.

```
GlanceNetCDF(file = "./temp_anomaly_ex1.nc")
```

```
----- Variables -----
tempanomaly:
    tempanomaly
    Dimensions: lat by lon by time
global:
    global
    Dimensions:


----- Dimensions -----
  time: 1420 values from 300 to 1719 months from January 1880
  lon: 10 values from 133 to 151 degrees_east
  lat: 10 values from -31 to -13 degrees_north
```

To actually load the data, we use the function `ReadNetCDF` and the data are already in a dataframe in the a *tidy* format.

```
tidytemp <- ReadNetCDF(file = "./temp_anomaly_ex1.nc",
                        vars="tempanomaly")
tidytemp
```

```
        time lon lat tempanomaly
     1:  300 133 -31         117
     2:  300 133 -29         118
     3:  300 133 -27         119
     4:  300 133 -25         123
     5:  300 133 -23         122
    ---                         
141996: 1719 151 -21          52
141997: 1719 151 -19          88
141998: 1719 151 -17          90
141999: 1719 151 -15          92
142000: 1719 151 -13          93
```