Lecture2_Numpy_Xarray_Solutions

July 19, 2022

1 Exercises

[ ]:
```
import xarray as xr
import numpy as np
```

You are given the output of a climate model in the form of a csv file. The the data is supposed to contain 3 days of daily precipitation data for a 10*10 pixel area over Bern. First we read the data:

[166]:
```
# This is an example of how you can read a csv file
import csv
input_list = []
with open("cra2022_shared/ex2_data.csv", newline="") as file:
    reader = csv.reader(file, delimiter = ',')
    for row in reader:
        input_list.append(row)

input_list = [element[0] for element in input_list] # The csv reader gives us a␣
 ↪list of strings which we convert to floats here
```

Now we have the model data in the form of a one-dimensional list which is rather inconvenient. It would be easier to interpret if we had it in the form of a 3d-array. Can you create an appropriate array from the list? (Hint: the data is ordered with the values of each row appended and then the values of each day appended)

[ ]:
```
# your code here
precipitation_array = np.array(input_list).reshape(3,10,10)
```

Now that we have a nice 3d-array lets look at the data. First let's check what datatype the values have:

[ ]:
```
# your code here
precipitation_array.dtype
```

If we want to do any calculations this datatype will not work. Can you convert it into a more useful one? (Hint: we'd like to have a "float64" datatype)

[ ]:
```
# your code here
precipitation_array =  precipitation_array.astype("float64")
```

Ok, now that we have numeric values let's see what the maximum measured precipitation was:

[ ]:
```
# your code here
precipitation_array.max()
```

Now this value looks a bit weird for precipitation. The unit is actually $kg * m^{-2} * s^{-1}$ averaged for the whole day. Let's convert the data into a more easily interpretable unit. Can you calculate the number $mm * m^{-2}$ which fell for each pixel?

[ ]:
```
# your code here
precip_array_mm = precipitation_array * 24 * 60 * 60
precip_array_mm
```

Now let's find the maximum value again. Can you figure out on which day the highest precipitation sum was measured? (Hint: look at the "where" function in numpy.)

[ ]:
```
# your code here
np.where(precip_array_mm==precip_array_mm.max()) #we have to give a condition␣
 ↪to where function the == compares the two values and returns True/False

#The Answer is the 2nd timestep (index 1 of the first dimension)
```

While the daily maxima are interesting, we are actually even more interested in the 3 day precipitation mean for that pixel. Can you calculate it?

[ ]:
```
# your code here
np.mean(precip_array_mm, axis = 0)[0,1]
```

Now finally, can you figure out the total precipitation (in m^3) which fell in this pixel and all it's adjacent pixels over the 3 days? Let's assume that each pixel has an area of 1 km^2.

(Hint: Since the pixel is at the edge (as we can see in the previous exercise) of our $10 * 10$ grid we should select a $2 * 3$ area for all 3 days.)

[ ]:
```
precip_array_mm[:,0:2,0:3].sum()*1000
```

Now playing around with this small model output was fun but we'd like to look at a bit more data. You can find the full model output in this NetCDF file: '/scratch3/climriskdata/EUR-11S/ICHEC-EC-EARTH_CLMcom-CCLM4-8-17_v1/rcp85/pr/pr_EUR-11_ICHEC-EC-EARTH_rcp85_r12i1p1_CLMcom-CCLM4-8-17_v1_day_20210101-20211231_LL.nc'. Let's open it:

[ ]:
```
# your code here
ds = xr.open_dataset('/scratch3/climriskdata/EUR-11S/
 ↪ICHEC-EC-EARTH_CLMcom-CCLM4-8-17_v1/rcp85/pr/
 ↪pr_EUR-11_ICHEC-EC-EARTH_rcp85_r12i1p1_CLMcom-CCLM4-8-17_v1_day_20210101-20211231_LL.
 ↪nc')
```

Can you figure out what the dataset contains? Which timeframe do we have? When was it created? What's the title of the dataset?

[ ]:
```
# your code here
ds
# timeframe: 365 timesteps. Daily data for 2021
# When was it created:     2014-03-26 06:23:41
# Title: CLMcom-CCLM4-8-17 model output prepared for CORDEX RCP8.5
```

Again the units are in $kg * m^{-2} * s^{-1}$. Can you convert the dataset to total daily $mm * m^{-2}$ and also adapt its metadata?

[ ]:
```
# your code here
ds.pr.values = ds.pr.values*24*60**2
ds.pr.attrs['units'] = 'daily mm m-2'
# Be careful, if you run this cell multiple times it will multipy the values␣
 ↪each time!
```

Now can you find on what day the Bern had the maximum precipitation accumulation? (Bern is located at 46.9480° N, 7.4474° E) (Hint:You may need to google for the correct function.)

[ ]:
```
# your code here
bern_pixel = ds.pr.sel(lat=46.9480, lon=7.4474, method='nearest')
bern_pixel.idxmax()
```

Now let's see how the precipitation looked over switzerland on that day. Can you plot it?

[ ]:
```
# your code here
ds.pr.sel(time = '2021-08-04 12:00:00', lat = slice(45.5, 48), lon = slice(5.
 ↪5,10.7)).plot()
```

Can you also add the previous and the following day to your plot? Can you arrange them vertically, each plot above another?

[ ]:
```
ds.pr.sel(time = slice('2021-08-03 12:00:00', '2021-08-05 12:00:00'), lat =␣
 ↪slice(45.5, 48), lon = slice(5.5,10.7)).plot(col='time',col_wrap=1)
```

To finish up: can you properly close the dataset and delete all the xarray objects?

[ ]:
```
# your code here
ds.close()
del ds, bern_pixel
```