Time series analysis

DMPR: Lab activities for Week 13

Subhasis Ray

April 28, 2023

In this class we shall explore time series data from last practice class. Submit your code on LMS with your results embedded as comments in appropriate locations. Separate out sections with a series of comment chars like this:

```
########## 1. Read the data ############
```

```
#%% import modules
import os
import numpy as np
import pandas as pd
# scipy.signal this contains signal processing functions that
↪  are
# useful for timeseries analysis
from scipy import signal as sig
import matplotlib
import matplotlib.pyplot as plt
```

1 Read the data

In the last class we used netCDF4 module to read the netCDF files. There is another module, `xarray` which is more versatile and offer better syntax for multidimensional data. Getting started guide here: `https://docs.xarray.dev/en/stable/getting-started-guide`

`xarray` needs `dask` module read multifile dataset. Install them with `conda install dask`, `conda install netcdf4`, and `conda install xarray` (`pip install ...` can work, but is known to cause problems).

```
import xarray as xr
#%% load data

ddir = os.path.join('..', '..', 'data', 'precipitation')  #
↪  modify this for your case
pr = xr.open_mfdataset(f'{ddir}/*.nc')
print(pr)
print(pr.dims)   # while netCDF4 uses `dimensions`, xarray
↪  uses `dims`
print(pr.attrs)  # attribute dictionary
```

You may notice that `xarray` handled the time data elegantly, without you having to mess around with pandas `to_datetime()`.

Remember that you should close the dataset objects at the end of your code.

`xarray` has two major classes, `DataArray` which represents a multidimensional array, and `Dataset` which is a colletion of named `DataArrays`.

You can select the tempereature and precipitation data for every month from each file for a specific location: (31.25 N, 76.25 E). With `xarray` datastructures, you can use the `sel` method to select entries by coordinate values (instead of index).

Also try, `ttemp = temp.sel(lat=31.24, lon=76.25)` before closing the dataset. What output do you see?

`ttemp.to_array()` does not honor the name 'tasmax' name, and it shows up as the attribute `variable`. You can fix this using this `ttemp = ttemp.rename({'variable': 'temp'})`

2 Plot data

You can plot the data using matplotlib, or using xarray's builtin plot functions:

```
plt.close('all')  # close any existing plots
# DataArray.values returns the underlying numpy array
plt.plot(ttemp.time, ttemp.values)
plt.savefig('temperature_vs_time.png')
plt.show()
```

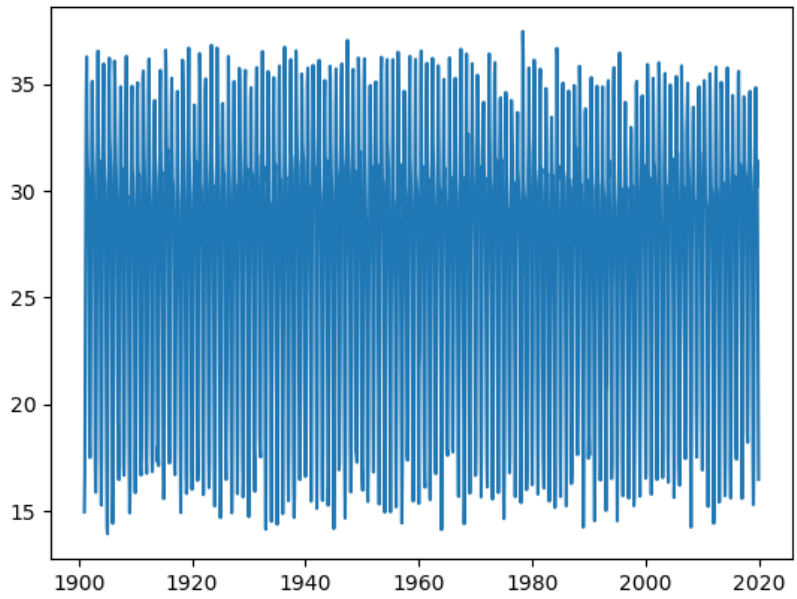

Figure 1: Temperature vs time

3 Fourier transform

You can look for oscillatory behavior in the DFT of the time series data

```
tmp_fft = np.fft.fft(ttemp.values)
# see
↪  https://numpy.org/doc/stable/reference/generated/numpy.fft.fftfreq.html
# to understand what the unit of frequency is
freq = np.fft.fftfreq(ttemp.shape[0], d=1)
plt.close('all')
plt.plot(freq, np.abs(tmp_fft))
plt.savefig('fft_temp.png')
plt.show()
```

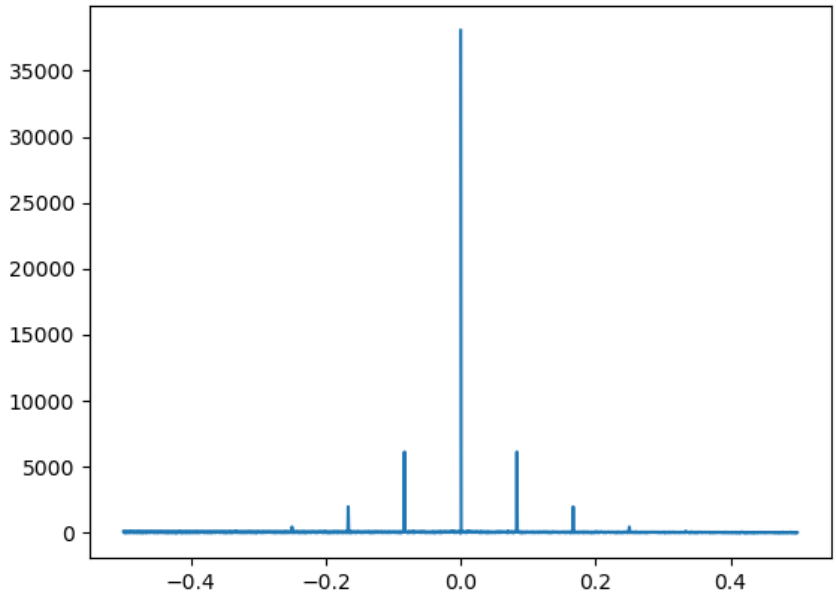

Figure 2: Fourier transform of temperature

What do the peaks mean? Why do they seem to be at multiples of one another?

4 Look at autocorrelation

Pandas provides a convenience function to plot autocorrelation `pd.plotting.autocorrelation_plot`

```
# see
↪  https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.butter.html
plt.close('all')
pd.plotting.autocorrelation_plot(ttemp.values)
plt.savefig('acorr_temp.png')
plt.show()
```

Zoom into this plot and find the period of the data.

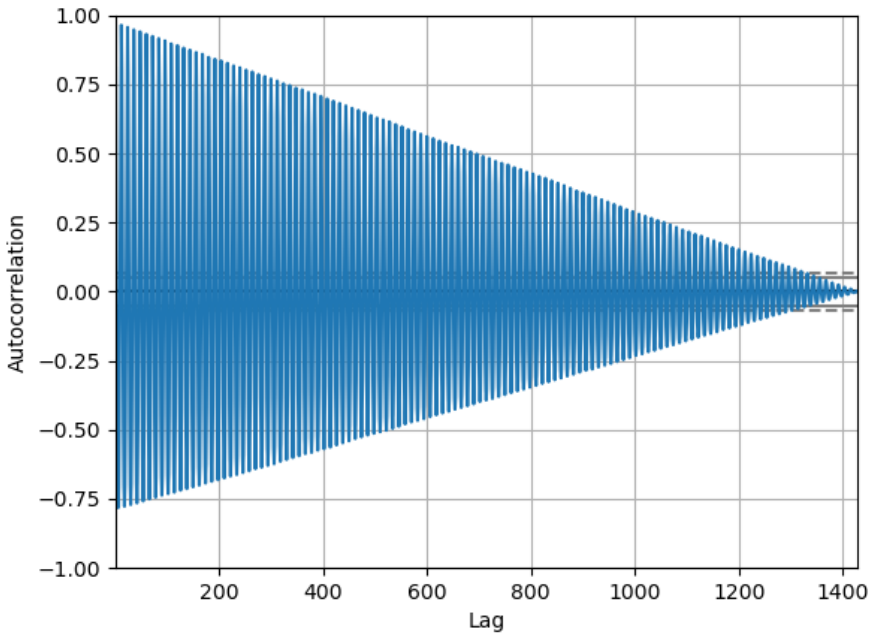

Figure 3: Temperature autocorrelation

5 Plot just January temperature each year5points

6 How would you get rid of this seasonality in the data? 10points

Provide the code for this. Explain the result in your final submission.

7 Try fitting an ARIMA model to the data using statsmodel. 15points

Show the summary of the model. It does not necessarily have to be ARIMA, you can set the appropriate order parameters and get AR, MA, ARMA or ARIMA.