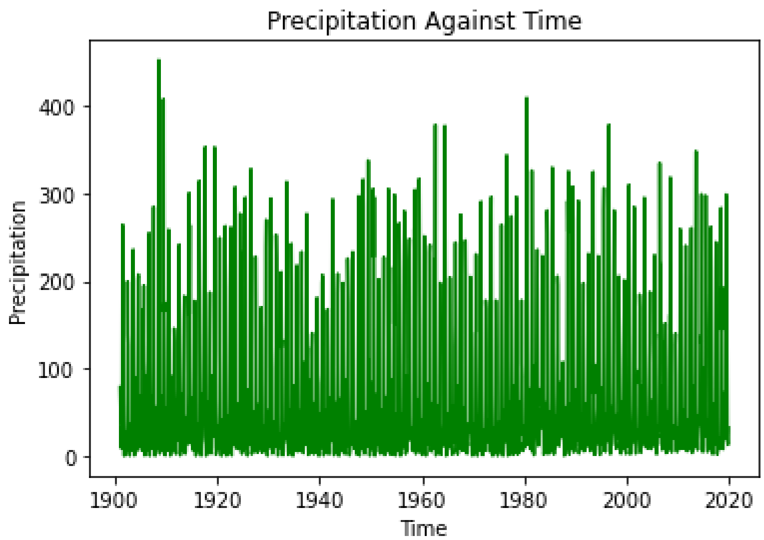
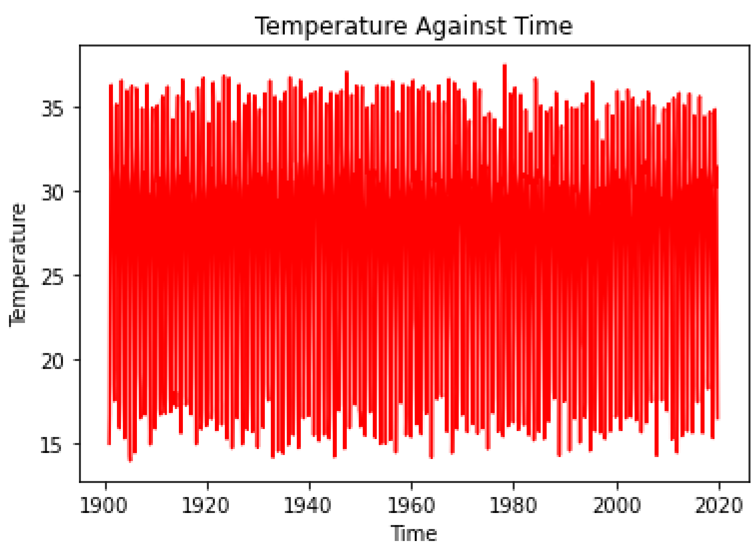
1. Unit of time in these files: **Days**
    a. Date for Time Value of 2953: **1909-02-01 00:00:00** as the start date is 1901-01-01 00:00:00
2. ***pr*** variable: this represents the **total monthly precipitation rate (lwe_precipitation_rate) data** collected at a particular time in a specified location given by the longitude and latitude.

    tasmax variable: this represents the **maximum monthly near-surface temperature (air_temperature)** data collected at a particular time period in a specified location given by the longitude and latitude.

3. **Line plot for precipitation against time:**



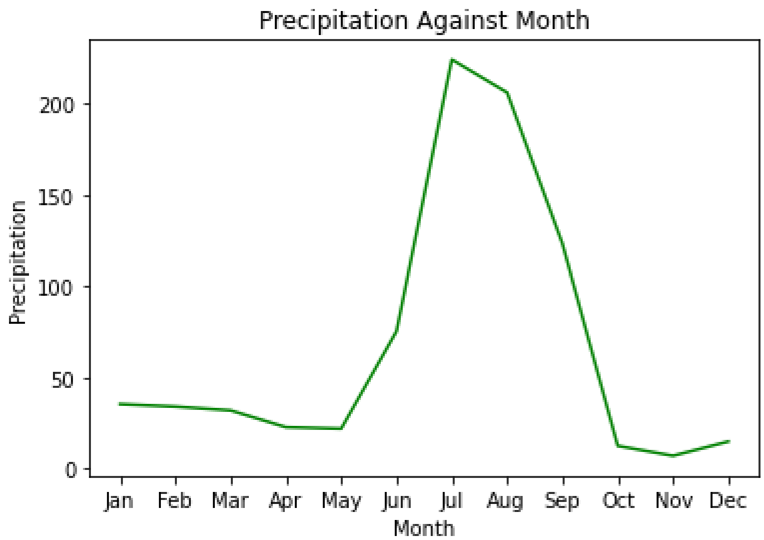
Line plot for temperature against time:



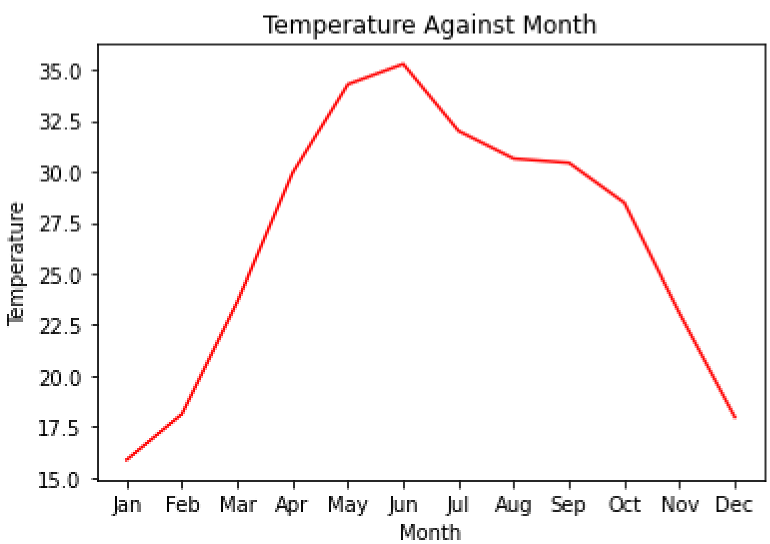
Trends:

Precipitation:

Computing the average of the precipitation by month, we realise precipitation is highest around the end of June till the end of August (raining season).

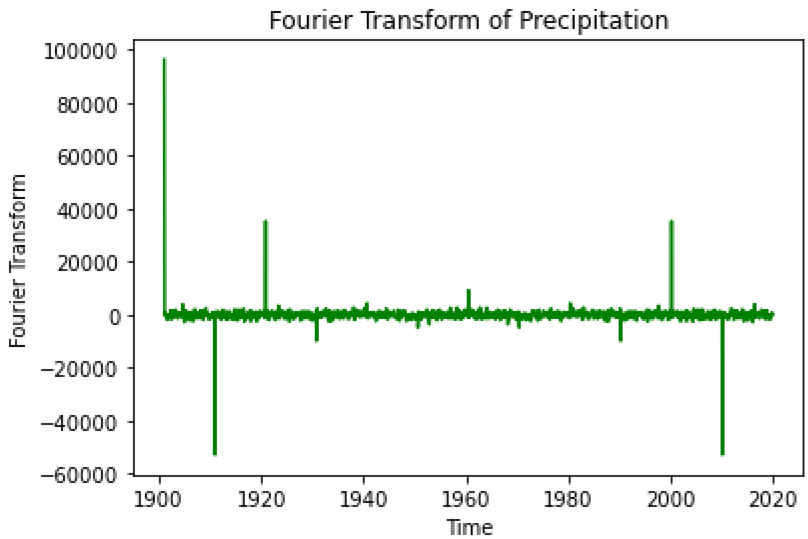


Doing the same for temperature, we notice that temperature is highest around summer as one would expect.

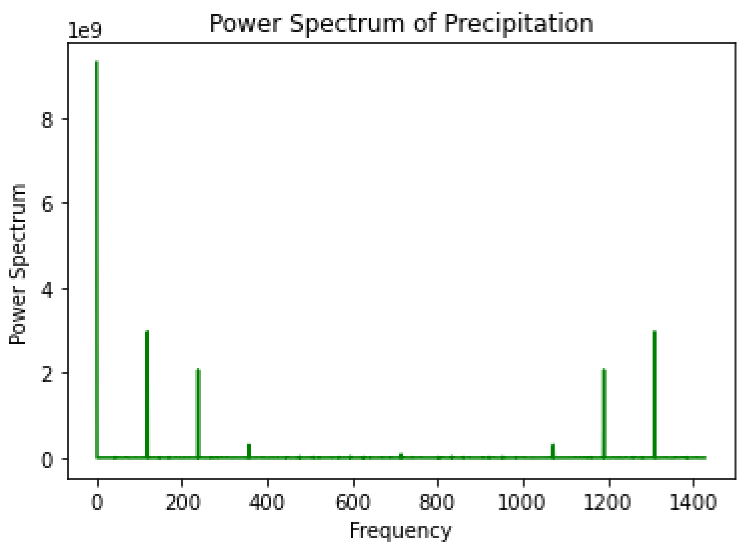


4. **Filtering the data**

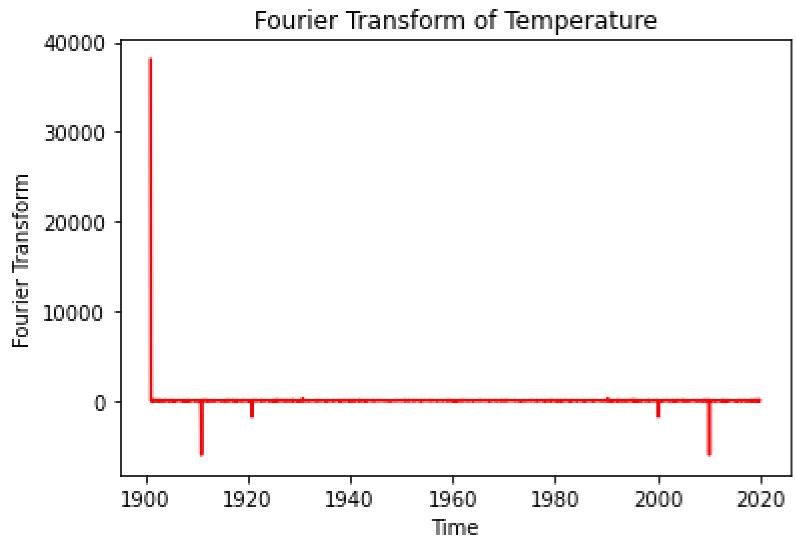
   Getting the dominant data using Fourier transforms:



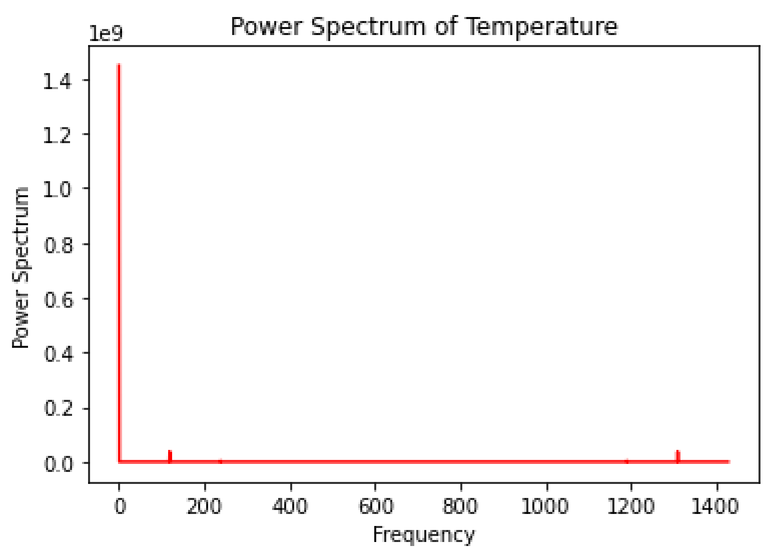
Corresponding power spectrum plot:



Fourier Transform of Temperature:

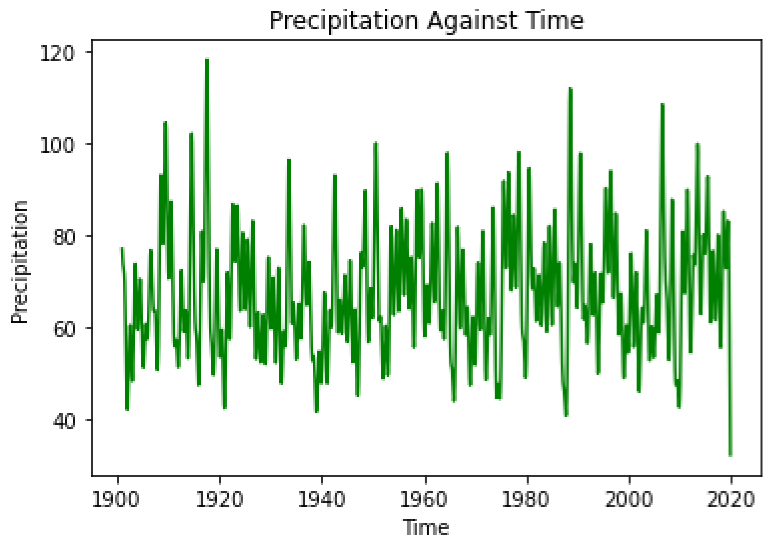


Corresponding Power Spectrum Plot:

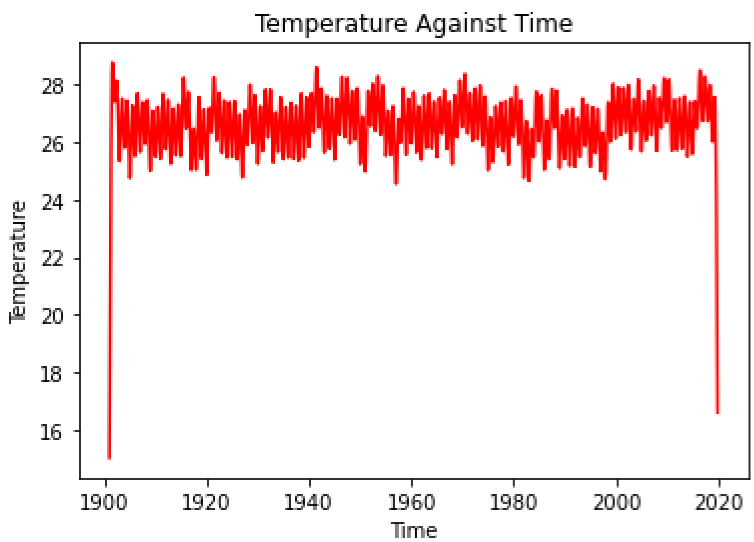


5. **Designing a filter with scipy.signal.butter:**

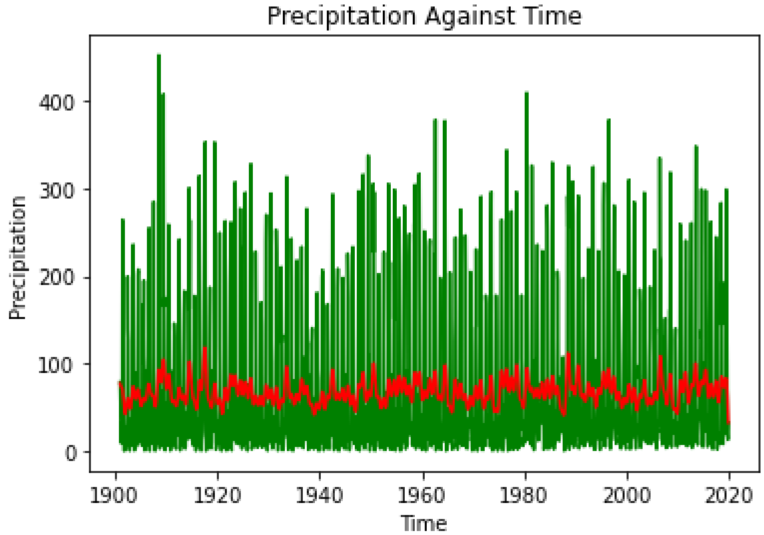
Filtered Signals for Precipitation:

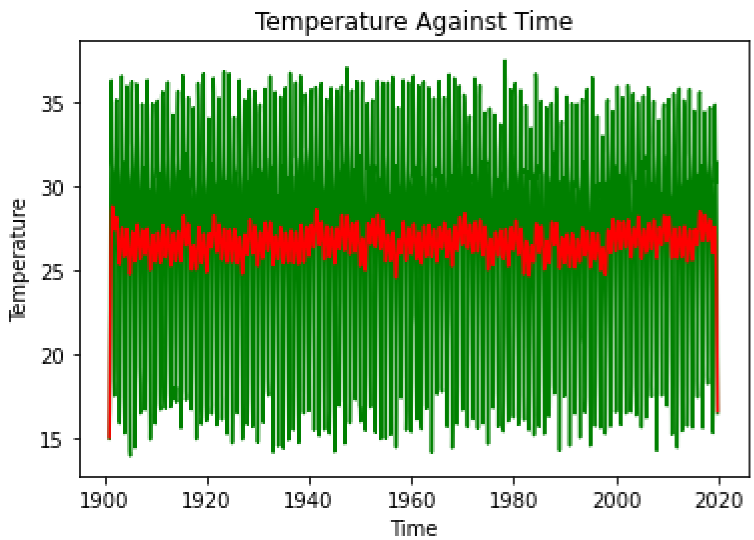


Filtered Signal for Temperature:



Overlaying the filtered plots on the original plots:



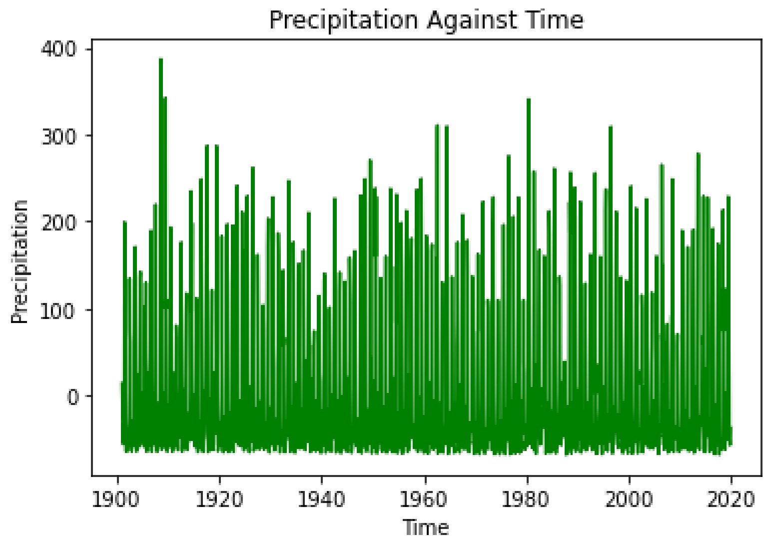


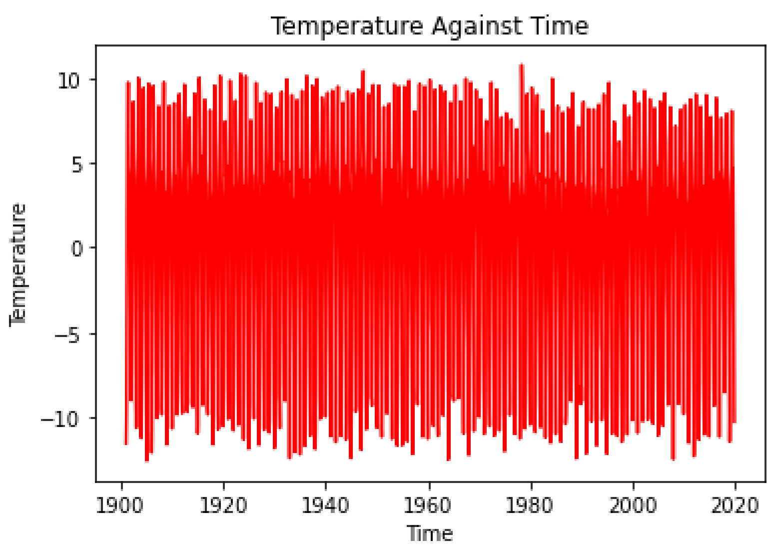
For deciding the cutoff frequency:

I have used a cut-off frequency of (**0.1** times the Nyquist frequency) beyond which other frequencies are filtered out. This is so as to preserve low-frequency variations while removing high-frequency noise and fluctuations.

6. Removing Linear Trend with scipy.signal.detrend:

On original plots:





Detrending on filtered data: