**ATOC7500 – Application Lab #2**

**Regression, Autocorrelation, Red Noise Timeseries**

**in class Feb. 10/15, 2022**

**Notebook #1 – Autocorrelation and Effective Sample Size using Fort Collins, Colorado weather observations**

**ATOC5860\_applicationlab2\_AR1\_Nstar.ipynb**

**LEARNING GOALS:**

1) Calculate the autocorrelation at a range of lags using two methods available in python (np.correlate, dot products)

2) Estimate the effective sample size (N\*) using the lag-1 autocorrelation

3) Evaluate the influence of changing the sampling frequency and the specified weather variable on the memory/redness of the data as quantified by the autocorrelation and N\*.

**DATA and UNDERLYING SCIENCE:**

In this notebook, you will analyze the memory (red noise) in weather observations from Fort Colins, Colorado at Christman Field. The observations are from one year, but are sampled hourly. The default settings for the notebook analyze the air temperature in degrees F sampled once daily (every midnight). But other standard weather variables and sampling frequencies can also be easily analyzed. The file containing the data is called christman\_2016.csv and it is a comma-delimited text file.

**Non-exhaustive Questions to guide your analysis of Notebook #1:**

1. Start with the default settings in the code. In other words – Read in the data and find the air temperature every 24 hours (every midnight) over the entire year. Calculate the lag-1 autocorrelation using np.correlate and the direct method using dot products. Compare the python syntax for calculating the autocorrelation with the formulas in Barnes. Equation numbers are provided to refer you back to the Barnes Notes. What is the lag-1 autocorrelation?

Lag-1 autocorrelation gives a metric of the “memory” of data. For each data point, we compare it to the previous data point in index. Then we do a standard correlation metric on that data. This data has a lag-1 autocorrelation of 0.846, indicating that it has a relatively high amount of memory.

1. Calculate the autocorrelation at a range of lags using np.correlate and the direct method using dot products. Compare the python syntax for calculating the autocorrelation with the formulas in Barnes. Equation numbers are provided to refer you back to the Barnes Notes. How does the autocorrelation change as you vary the lag from -40 days to +40 days?

Autocorrelation generally decreases as the lag time goes further away. This indicates that there is no periodicity in the daily data. Since the peak at the center isn’t extremely sharp, there is some memory present.

1. Calculate the effective sample size (N\*) and compare it to your original sample size (N). Equation numbers are provided to refer you back to the Barnes Notes. How much memory is there in temperature sampled every midnight?

Calculating effective sample size gives us 31 independent samples. The dramatic reduction in sample size indicates that there is a lot of memory in the data. These data points are not really independent of eachother!

1. Now you are ready to tinker … i.e., make minor adjustments to the code with the parameters set in the code to see how your results change. *Suggestion: Make a copy of the notebook for your tinkering so that you can refer back to your original answers and the unmodified original code.* For example: Repeat steps 1-3) above with a different variable (e.g., relative humidity (RH), wind speed (wind\_mph)). Repeat steps 1-3) above with a different temporal sampling frequency (e.g., every 12 hours, every 6 hours, every 4 days). How do you answers change?

Comparing temperature data hourly

This data has less memory, giving a very high day-1 autocorrelation of 0.986. This is higher memory than daily temperature.

We see a much higher autocorrelation at a lag of 24 hours. This is a representation of the diurnal cycle showing up in temperature

The reduction in sample size is even more dramatic, going down to 733 samples from 8784.

Wind gust direction daily

There is no memory in this data, giving an autocorrelation of 0. There is a distinct spike at 0 days and basically nothing anywhere else in lag space. There is no reduction in sample size due to the complete lack of memory in the data.

**Notebook #2 – Red noise time series generation, Regression, and Statistical Significance Testing While Regressing**

**ATOC5860\_applicationlab2\_AR1\_regression\_AO.ipynb**

**LEARNING GOALS:**

1) Calculate and analyze the autocorrelation at a range of lags using output from an EOF analysis (the Arctic Oscillation Index).

2) Generate a red noise time series with equivalent memory as an observed time series (i.e., given lag-1 autocorrelation).

3) Correlate two time series and calculate the statistical significance.

4) Evaluate the statistical significance obtained in the context of the number of chances provided for success. What happens when you go “fishing” for correlations and give yourself lots of opportunity for success? Can you critically evaluate the chances that your regression is statistically different than 0 just by chance?

**DATA and UNDERLYING SCIENCE:**

In this notebook, you will analyze the monthly Arctic Oscillation (AO) timeseries from January 1950 to present. The AO timeseries comes from an Empirical Orthogonal Function (EOF) analysis. We will implement EOFs in the next application lab so in this lab we are actually using multiple analysis methods introduced in this class, some that you have learned and some that you are still yet to learn ☺.

How do you find the AO value each month? To identify the atmospheric circulation patterns that explain the most variance, NOAA regularly applies EOF analysis to the monthly mean 1000-hPa height anomalies poleward of 20° latitude for the Northern Hemisphere. The AO spatial pattern (Figure 1 below) emerges as the first EOF (explaining the most variance, 19%). The AO timeseries we will analyze is a measure of the amplitude of the pattern in Figure 1 in a given month. In other words – the AO timeseries is the first principal component (a timeseries) associated with the first EOF (a spatial structure). More information on the EOF analysis here:

http://www.cpc.ncep.noaa.gov/products/precip/CWlink/daily\_ao\_index/history/method.shtml



Figure 1. The loading pattern of the Arctic Oscillation (AO), i.e., the structure explaining the most variance of monthly mean 1000mb height during 1979-2000 period. In other words – this is the first EOF.

The data are available and regularly updated here:

<http://www.cpc.ncep.noaa.gov/products/precip/CWlink/pna/norm.nao.monthly.b5001.current.ascii>

You can work with the data directly on the web (assuming you have an internet connection). I have also downloaded the data and made them available – The name of the data file is “monthly.ao.index.b50.current.ascii”.

**Questions to guide your analysis of Notebook #2:**

1. Start with the default settings in the code. First read in the Arctic Oscillation (AO) data. Look at your data!! Plot it as a timeseries. Save the timeseries plot as a postscript file and put it in this document.

Chart, line chart

Description automatically generated

1. Calculate the lag-one autocorrelation (AR1) of the AO data and record it here. Use two methods (np.correlate, dot products). Check that they give you the same result. Interpret the value. How much memory (red noise) is there in the AO from month to month?

I found an autocorrelation of 0.30855. This indicates that there is some memory in AO, but not much.

1. Calculate and plot the autocorrelation of the AO data at all lags. Describe your results. How red are the data at lags other than lag=1? Is there any interesting behavior of the autocorrelation as a function of lag? What would you expect for red noise timeseries with an AR1=value reported in 2)?

The autocorrelation plot shows a significant spike at 0 months, but does have memory. In particular, there looks to be a spike in memory around 6 months, but it is slight. I would expect a timeseries to of this red noise to look pretty random, but with some trends. For example, the timeseries data from question 1 has some periods with higher data, and some period with lower data that are not isolated.

1. Generate a synthetic red noise time series with the same lag-1 autocorrelation as the AO data. Your synthetic dataset should have different time evolution but the same memory as the AO. Plot the AO timeseries and the synthetic red noise time series. Put the plot below.

A picture containing text, lined, day, line

Description automatically generated

1. Do you expect to find any correlation between the two datasets, i.e., the synthetic red noise and the actual AO data? What is the correlation between the synthetic red noise and the actual AO data? Calculate a regression coefficient and other associated regression statistics.

I do not expect any correlation between these data sets. While both have memory, there are completely independent of each other. I found an r squared value of 0.002, indicating very little correlation.

1. Next -- Have some fun and go “fishing for correlations”. What happens if you try correlating subsets of the two datasets many times? When you try 200 times -- what is the maximum correlation/variance explained you can obtain between the synthetic red noise and the actual data? *Note: you are effectively searching for a high correlation with no a priori reason to do so.... THIS IS NOT good practice for science but we are doing it here because it is instructive to see what happens :)*

I was able to find a dataset that correlated very well with our data, r^2=0.44. This could be deceiving if I didn’t know that these were randomly generated data sets and not other variables.

1. Calculate the correlation statistics for the highest correlation obtained in question 6). Two methods are provided - they should give you the same answers. Place a confidence interval on your correlation. Because you have found a correlation that is not equal to 0, use the Fisher-Z Transformation. Did your "fishing" for a statistically significant correlation work? Is your highest correlation statistically significant (i.e., can you reject the null hypothesis that the correlation is zero)? Write out the steps for hypothesis testing and use the values you calculate to formally assess.

1. We are using a significance level of alpha = 0.05

2. Our null hypothesis (H0) is that there is no correlation between these data sets, our alternative hypothesis (H1) is that there is a correlation between these data sets.

3. We will reject the null hypothesis if t > 1.96. Since we have no prior reasoning to assume the correlation will be higher or lower, we are using a two-sided t test. This is also a test for correlation coefficients, so we must use the Fisher-Z statistic. A t test is being used since the sample size is less than 30.

4. The critical region is t < 1.96. If this occurs, we will not reject the null hypothesis. I found a correlation value of -0.67, which translates to a critical region of [-0.87, -0.28].

5. Since our expected correlation was 0, we can reject the null hypothesis. These data are correlated.

1. You went searching for correlations, you searched long and hard (200 times!) You should have been concerned that the largest correlation you found would be a false positive. Do you think you found a false positive? Explain what you found and potentially why you think it is important statistically but not physically. What lessons did you learn by “fishing for correlations”?

This is a false positive. We generated a bunch of random datasets with the same type of memory as our data. Of course, we did find a dataset that was correlated due to chance. This shows statistically that datasets will happen to be correlated regardless of if there is any connection between them. This is a reminder of why rigor in statistics is important, you must be very careful about what assumptions your tests make and cautious about what you can conclude via statistical tests.

FOR FUN: Check out - <https://www.tylervigen.com/spurious-correlations>