**ATOC7500 – Application Lab #2**

**Regression, Autocorrelation, Red Noise Timeseries**

**in class Monday/Wednesday September 21/23, 2020**

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

**ATOC7500\_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 = 0.845870. This is the same number using numpy or the direct calculations.**

2) 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 decreases slowly with increasing lag time, this is symmetric for positive and negative values. As autocorrelation remains elevated for large values of lag we can call this time series ‘red’.**

3) 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?

**# samples, N: 366**

**lag-1 autocorrelation: 0.8459**

**#independent samples, N\*: 30.561 = 31 (needs to be rounded as discrete quantity)**

**From Barnes Ch 2 (90) so 0.8441**

**As is a measure of memory in the system, 0.8441 is a high value .**

4) 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?

**RH is much less autocorrelated with only a small lag it quickly drops to near zero. This suggests little memory for this variable. N\* = 198 with N=366 so many independent samples!**

A close up of text on a white background

Description automatically generated

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

**ATOC7500\_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.

A picture containing outdoor, water, sitting, beach

Description automatically generated

2) 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?

**AR1 = 0.31526373511169. Exactly the same by using np.correlate and equation 67. This is effectively an r value between the time series and the lagged time series. If we square this value we obtain r2=0.0993 or 9.9% of the variance in the time series is due to the timestep before. This means there is little memory in the time series.**

3) 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)?

A close up of a tiled wall

Description automatically generated

**At lags of 1,2, and 3 months there is a reasonably high autocorrelation, but this quickly dies away suggesting that the datasets is not very red as a whole. At 6 and 7 and 12 and 13 months lag there appears more autocorrelation, which suggests persistence based on previous season or year. Fourier series analysis would have to be done to properly assess this. A time series which is entirely white noise would have near-zero autocorrelation for all lags.**

4) 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 sitting, long, lined, standing

Description automatically generated

5) 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.

**There should be no correlation between the two datasets, r=0.00824876 and r2=0.006804%.**

6) 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 :)*

**Highest r=-0.50025, so highest r2=25.02479%.**

7) 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.

**Linear regression: y=mx+c, y = -0.34009x -0.44417. The fishing didn’t quite work, as outlined below:**

1. **Confidence interval: 95%**
2. **Null Hypothesis: The regression coefficient is not statistically different from zero, i.e. there is more than a 5% chance that it could be found by random chance.**
3. **Statistic Used: Fisher-T transformation with t-test to find confidence limits.**
4. **Critical Region: The confidence limits on the value of is -0.37533 to 0.55722.**
5. **Reject null hypothesis: The highest r value was -0.50025 this exceeds the confidence limit of -0.37533 so we can say that it is less likely than 5% that this strong of a regression coefficient happened by chance.**

8) 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”?

**The probability of correctly rejecting all is only 0.003505%. If this was used instead of the 5% confidence interval to calculate the this would yield a much smaller critical region and thus support the null hypothesis.**