

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 Collins, 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?

The lag-1 autocorrelation is 0.846. Both methods for calculating the lag-1 autocorrelation produce identical results and are merely different ways of performing the same operation.

- 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?

As expected the autocorrelation peaks at 1 with a lag of zero days and then decays the farther way from zero you go. For example the lag-1 autocorrelation is around 0.40 at lag -40 days and 0.4 at lag 40 days. Indicating that memory decreases with increasing lag.

The python syntax should produce identical results as the equation in Barne's notes.

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?

366 samples, 31 independent samples. $366/31$ is roughly 12, so you can say that every 12 days you get an independent temperature. Thus there is memory of the proceeding day's temperature for about 12 days.

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 your answers change?

Wind Speed every 24 hours:

Lag-1 autocorrelation = -0.045

The autocorrelation remains very weak (magnitude < 0.2) at all lags, but does indeed appear to have a weak decay in autocorrelation with increasing lag time.

We have 366 samples and 335 independent samples. This makes sense given the small lag-1 autocorrelation and indicates that there is very little memory in wind speed, about 1 day.

Temperature every 5 days:

Lag-1 autocorrelation = 0.728

Again the autocorrelation decays with increasing lag but does not appear to be symmetric. So I am quite confused how to interpret these results. Interestingly the autocorrelation becomes negative on lags > 20 (which I believe corresponds to $20 \times 5 = 100$ days). I guess this can be explained by seasonal negative correlation.

Originally we had 74 samples, and calculate that we have 12 independent samples. $74/12$ is roughly 6, indicating that that we only get an independent sample every $6 \times 5 = 30$ days. Or about 1 independent sample every month. Note that this is in contrast to what we found (12 days) with daily sampling.

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

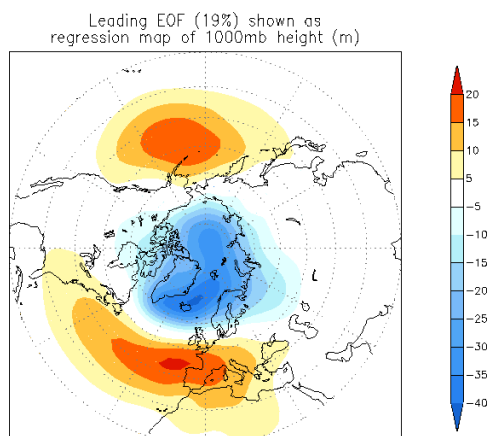


Figure 1. The loading pattern of the Arctic Oscillation (AO), i.e., the structure explaining the most variance of monthly mean 1000mb height during 1979-2000 period. In other words – this is the first EOF.

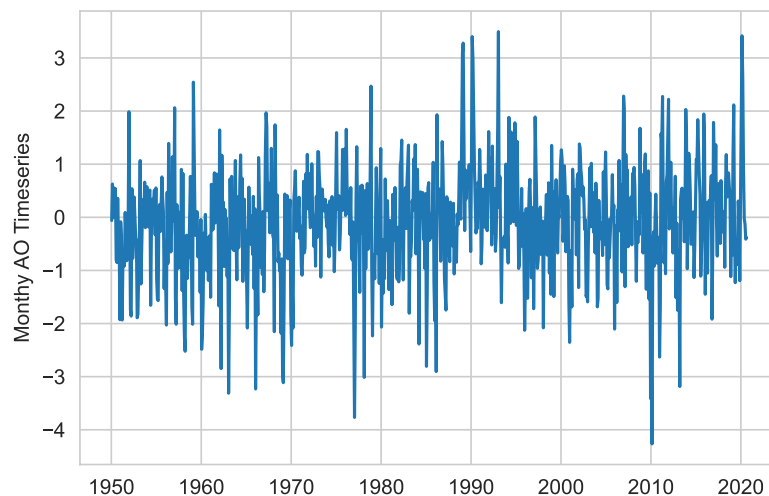
The data are available and regularly updated here:

<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.



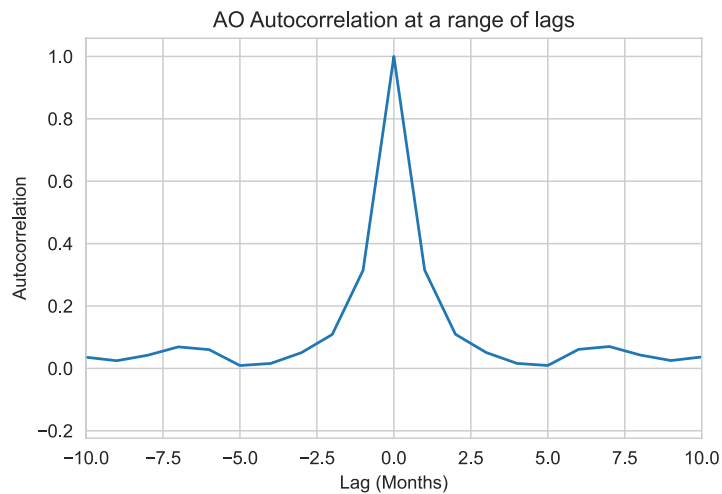
2) Calculate the lag-one autocorrelation (AR1) of the AO data and record it here. Use two methods (np.correlate, dot products). Check that they give you the same result. Interpret the value. How much memory (red noise) is there in the AO from month to month?

AR1 = 0.315 for both methods. The memory in the AO from month to month can qualitatively be described as “modest” and quantitatively described by the lag-one autocorrelation, which is equal to 0.315.

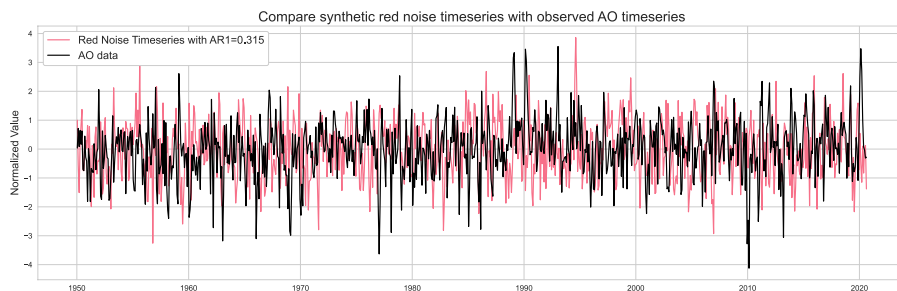
3) Calculate and plot the autocorrelation of the AO data at all lags. Describe your results. How red are the data at lags other than lag=1? Is there any interesting behavior of the autocorrelation as a function of lag? What would you expect for red noise timeseries with an AR1=value reported in 2)?

As expected, the autocorrelation peaks at 1 with a lag of zero, and then generally decays with increasing lag, while maintaining symmetry about lag = 0 months. Autocorrelation quickly decreases to approximately 0.07 at +/- 2.5 months and nearly zero at +/- 5 months. Interestingly, the autocorrelation begins to pick up at lags greater than 5 months, and then begins to decrease again at approximately 7 months. Perhaps, this can be explained by long term teleconnections? Or it could simply be due to chance?

A red noise timeseries with $AR1 = 0.315$ should see its autocorrelation decay exponentially with an e-folding time-scale of $T_e = -1 / \ln(0.315) = 0.87$ months.



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.



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.

I expect to find some small amount of correlation between the data sets strictly by chance, however because the synthetic dataset is not physically linked to the AO time series, I do not expect any correlation to be meaningful.

The correlation between the two-time series is given by $r = -0.034$. The regression coefficient (slope) is -0.034 . In addition 0.11 % of the variance in the AO can be explained by the synthetic data set with the same AR1. Notably, the regression statistics are small but not zero!

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 :)*

Maximum correlation: $r = -0.61$

Maximum variance explained: $r^2 = .3773$

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.

Correlation 95% confidence intervals: -0.84 to -0.2

1. 95% confidence level
2. Null Hypothesis: The correlation is zero
Alternative: The correlation is not zero
3. We will use the Fisher-Z transformation
4. To reject the null hypothesis we must have the 95% confidence interval not contain zero.
5. The 95% confidence interval does not contain zero. Thus we reject the null hypothesis.

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”?

Yes, I did find a false positive! I rejected the null-hypothesis and therefore found a non-zero correlation. However, there is not physical reason these time series should be correlated. I calculated the 95% confidence intervals, thus for each of the 200 samples there is a 5% chance that I make a type 1 error, i.e. find a false positive. Thus, the probability of me correctly rejecting all null-hypothesis is only 0.0035% ($0.95^{200} \cdot 100$)!

There is an important lesson here. Namely that if I give my statistic many chances to succeed, eventually it will. In fact, in this case it is very unlikely that I will not reject the null-hypothesis at least once.