Gina Jozef
2/15/22

# 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?

np.correlate syntax

```
lagNauto_np=np.correlate(t1_m,t2_m,mode='valid')/(n-lag)/(sigma**2)
```

direct method using dot products syntax

```
lagNauto=np.dot(t1_m,t2_m)/(n-lag)/sigma**2
```

Where t1_m and t2_m are the time series of temperature, offset by one index (24 hours), n is the number of temperature measurements, lag is 1, and sigma is the standard deviation of the temperature time series.

The direct method uses Barnes Eq. 68 in Chapter 2, divided by the variance. This is shown below:

Gina Jozef
2/15/22

Autocovariance function: $\gamma(\tau) = \overline{x'(t)x'(t+\tau)} = \frac{x(t)\cdot x(t+\tau)}{N-\tau}$ (68)

Variance: $\gamma(0) = \overline{x'(t)x'(t)} = \overline{x'^2} = \sigma^2$

$\sigma = standard\ deviation$

Autocorrelation: $\rho(\tau) = \frac{\gamma(\tau)}{\gamma(0)} = \frac{\overline{x'(t)x'(t+\tau)}}{\overline{x'^2}} = \frac{x(t)\cdot x(t+\tau)}{\frac{N-\tau}{\sigma^2}}$

Both of these methods give a lag-1 autocorrelation of 0.846.

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?

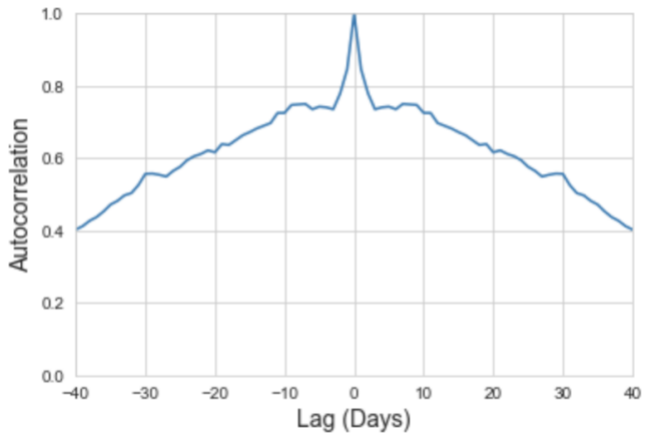


Figure 1: Autocorrelation at a range of lags for temperature at midnight using np.correlate method.

The direct method using dot products does not work to calculate autocorrelation at a range of lags, since it only calculates a single value for the dot product of the time series with itself, rather than an array of autocorrelation values which relate to different lag indices.

According to figure 1, the autocorrelation peaks to 1 at 0 lag, since the time series is being compared to itself at 0 lag. When you vary the lag from -40 to +40 days, the autocorrelation is symmetric about the 0 lag. This is by definition how autocorrelation works. The autocorrelation does not abruptly decrease to near 0 outside of 0 lag, or as it approaches -40 and +40 days, which indicates that there is still some memory at this time scale.

Gina Jozef
2/15/22

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?

The Barnes equation used to determine N* from N is:

$$\frac{N^*}{N} \cong \frac{1-\rho(\Delta t)}{1+\rho(\Delta t)}$$

Where N (total # of samples) is 366, and $\rho$ (the lag-1 autocorrelation) is 0.85. This gives the following conclusion for N*, for a lag of 1:

$$N^* \cong \frac{1-0.85(1)}{1+0.85(1)} 366 = 31$$

Therefore, the effective sample size is 31 independent samples. This gives a memory of about 11.8 days, as effectively we have an independent sample every 11.8 days (366/31=11.8)

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?

Different variable: testing out wind speed (wind_mph).

1. Lag-1 autocorrelation = -0.045
2.

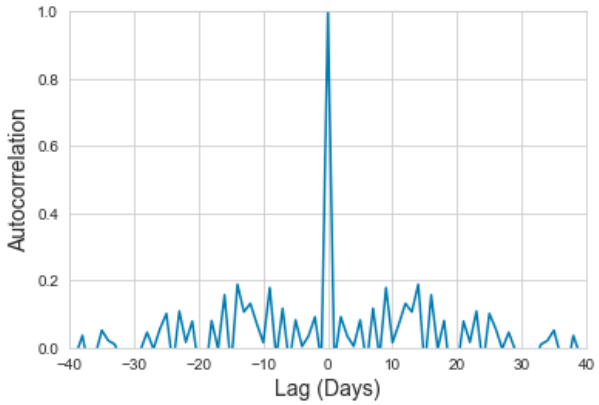


Figure 2: Autocorrelation at a range of lags for wind speed at midnight using np.correlate method.

According to figure 2, as for temperature, the autocorrelation peaks to 1 at 0 lag, since the time series is being compared to itself at 0 lag. However, unlike for temperature, the autocorrelation abruptly decrease to near 0 outside of 0 lag, and gets smaller as it approaches -40 and +40 days. This indicates that there is little some memory in the wind speed variable over time.

3. Finding effective sample size:

$$N^* \cong \frac{1 - -0.045(1)}{1 \pm 0.045(1)} 366 = 335$$

Therefore, the effective sample size is 335 independent samples. This gives a memory of about 1.1 days, as effectively we have an independent sample every 1.1 days (366/335=1.09). This suggest that wind speed is more variable over time, and more difficult to predict given a current value, than is temperature.

Different variable: testing every 6 hours with temperature

1. Lag-1 autocorrelation = 0.762
2.

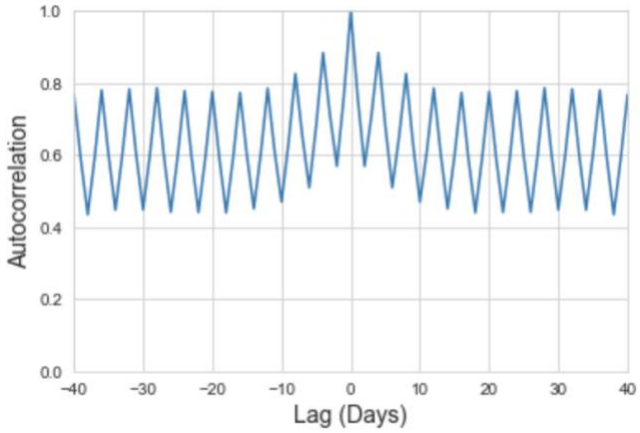


Figure 3: Autocorrelation at a range of lags for wind speed every six hours using np.correlate method.

According to figure 3, as for temperature at midnight, the autocorrelation peaks to 1 at 0 lag, since the time series is being compared to itself at 0 lag. However, unlike for temperature every 24 hours, the autocorrelation does not decrease as much outside of 0 lag, and has a periodic oscillation that remains around 0.4-0.8 even as the lag extends to -40 and +40, which is likely due to diurnal temperature patterns. This indicates that there is more memory in the temperature variable when sampled every 6 hours than every 24 hours.

3. Finding effective sample size:

Gina Jozef
2/15/22

$$N^* \cong \frac{1 - 0.762(1)}{1 \pm 0.762(1)} 366 = 198$$

Therefore, the effective sample size is 198 independent samples. This gives a memory of about 1.8 days, as effectively we have an independent sample every 1.8 days (366/198=1.85). It is a little surprising that the effective sample size would be higher for temperature sampled every 6 hours than every 24 hours, since the 6-hour measurements have a higher autocorrelation at every lag. However, this also makes sense since temperature every 6 hours will be more variable than that every 24 hours, so there would be more independent samples.

Gina Jozef
2/15/22

**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

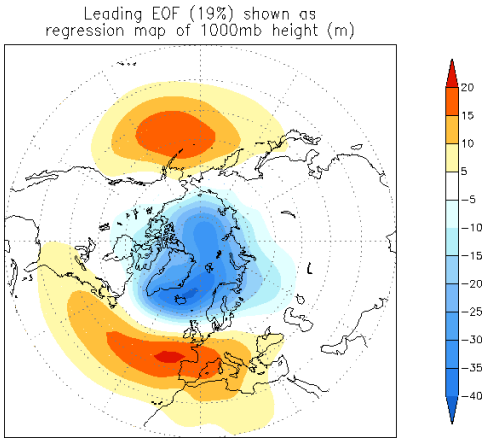


Figure 1. The loading pattern of the Arctic Oscillation (AO), i.e., the structure explaining the most variance of monthly mean 1000mb height during 1979-2000 period. In other words – this is the first EOF.

Gina Jozef
2/15/22

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.

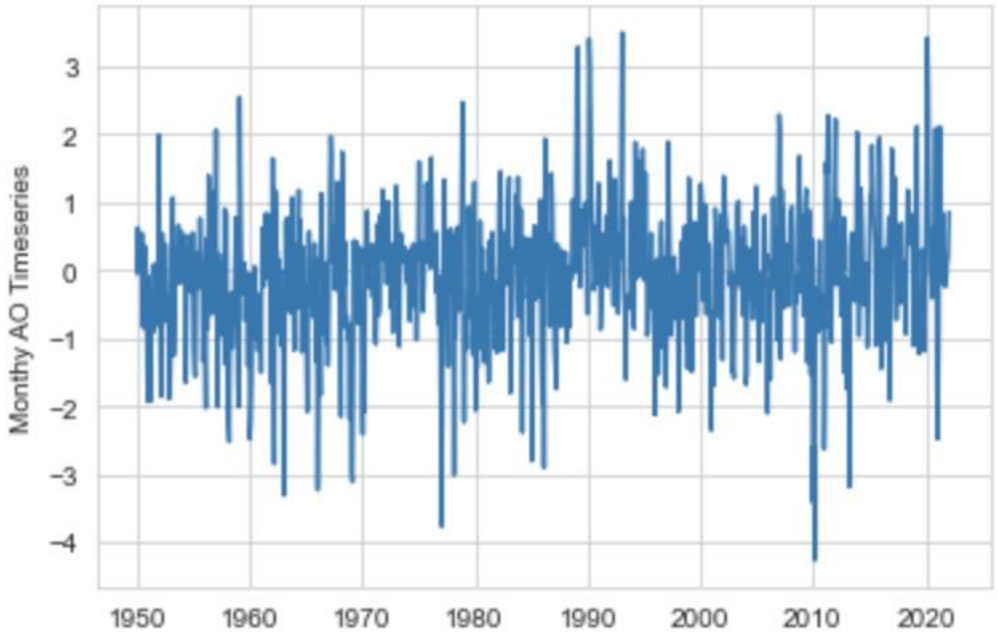


Figure 4: Time series of monthly AO.

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?

Both methods give a lag-1 autocorrelation of 0.30855. This value squared (0.095) tells us how much of the variation between the original dataset and the dataset at one lag. Therefore, in this case, only 9.5% of the variation is explained by autocorrelation. This tells us that there is not a lot of memory (red noise) in the AO from month to month.

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)?

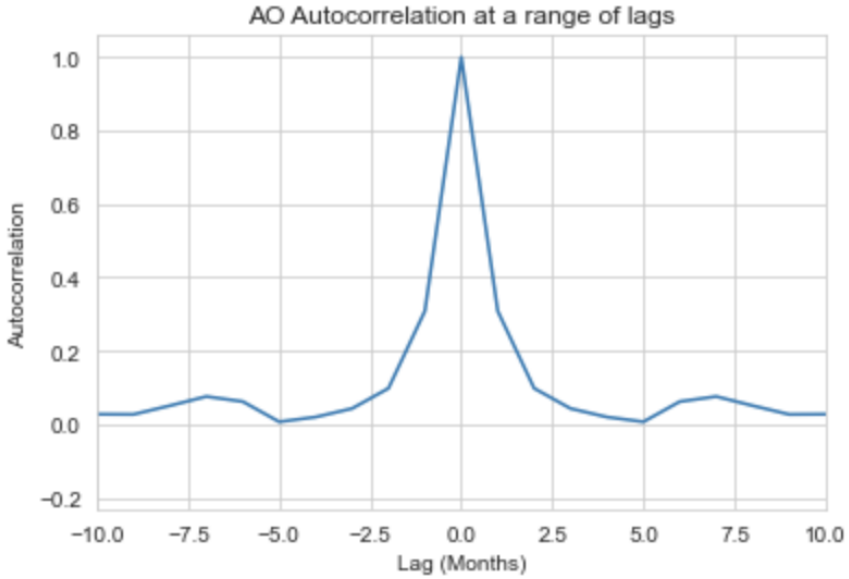


Figure 5: Autocorrelation of AO at a range of lags.

There is not a very high autocorrelation outside of lag 1; it decreases to near zero outside lag 1. This suggests that there is not much memory in the time series. However, there is an interesting feature that the autocorrelation increases around lag 6, which suggests that there is some pattern in AO with a period of 6 months. This is about what is expected for a time series with an autocorrelation at lag 1 that is fairly low, as we found in question 2.

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.

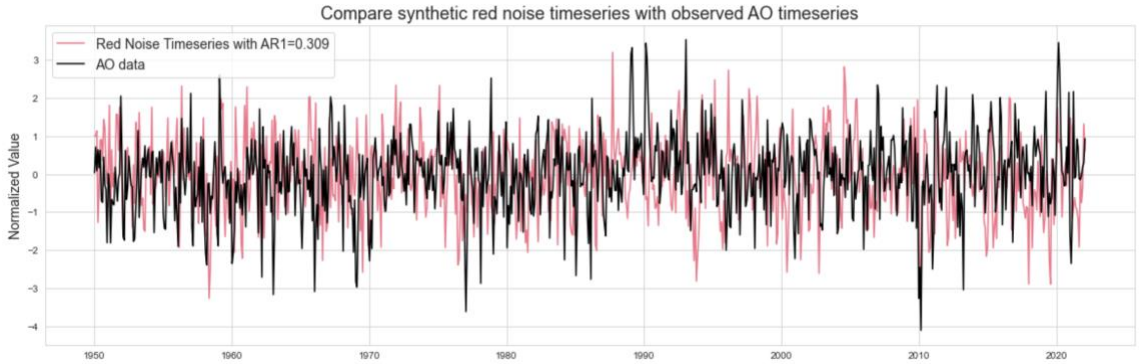


Figure 6: Time series of synthetic red noise with the same lag-1 autocorrelation as the AO data (red) and the original AO data (black).

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 do not expect to find any correlation between the two datasets since the synthetic red noise dataset is completely random and has no physical relationship to the AO data. The correlation between the synthetic red noise and actual AO data (percent of variance in the AO data explained by the synthetic red noise data) is 0.5246%.

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 :)*

The maximum variance explained when correlating subsets of the two datasets 200 time is 31.64%. This is a much higher variance explained than when you compare the entirety of the two datasets once.

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 statistics for the highest correlation obtained.

Slope: -0.454
y-intercept: 0.376
r value: -0.562

Screenshot of the results from both methods

```
scipy.stats.linregress slope:  -0.454
scipy.stats.linregress intercept:  0.376
scipy.stats.linregress r_value:  -0.562
direct method slope_fast:  -0.454
direct method intercept_fast:  0.376
direct method rvalue_fast:  -0.562
```

Gina Jozef
2/15/22

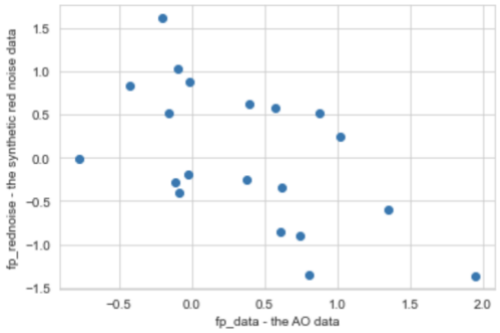


Figure 7: The subset of data which found the highest correlation obtained.

Confidence intervals on correlation using Fisher-Z

1. 95% confidence ($\alpha = 0.05$)

2. $H_0$ : higher AO data correlates to a lower value of synthetic red noise
   $H_1$ : there is no correlation between AO and synthetic red noise

3. We will use the Fisher-Z test because the true correlation appears to not be zero. The assumption is that the underlying distribution is not symmetric.

4. We reject the null hypothesis if the confidence bounds for the true correlation include 0.

$$Z - t_c\sigma_Z \leq \mu_Z \leq Z + t_c\sigma_Z$$

   For $\alpha = 0.05$ and df = 17 (N-3; N=20), $t_c = 2.11$.

5. Evaluate:

N = 20 (number of datapoints)

r = - 0.562

$$Fisher - Z = \frac{1}{2}\ln\left(\frac{1+r}{1-r}\right) = \frac{1}{2}\ln\left(\frac{1+-0.562}{1--0.562}\right) = -0.64$$

$$\sigma_Z = \frac{1}{\sqrt{N-3}} = \frac{1}{\sqrt{20-3}} = 0.24$$

$$\mu_Z = \frac{1}{2}\ln\left(\frac{1+\rho}{1-\rho}\right) \qquad \text{therefore} \qquad \rho = \frac{e^{2\mu_Z}-1}{e^{2\mu_Z}+1} = \tanh(\mu_z)$$

   Putting confidence limits on the true correlation:

$$Z - t_c\sigma_Z \leq \mu_Z \leq Z + t_c\sigma_Z$$
$$-0.64 - 2.11 * 0.24 \leq \mu_Z \leq -0.64 + 2.11 * 0.24$$

Gina Jozef
2/15/22

$$-1.15 \leq \mu_Z \leq -0.12$$

$$\rho = \tanh(\mu_z)$$
$$-0.82 \leq \rho \leq -0.12$$

Since the confidence interval on the true correlation does not include 0, we cannot reject the null hypothesis that higher AO data correlates to a lower value of synthetic red noise. This suggests that the correlation coefficient between these two variables is statistically significant at the 95% significance level.

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"?

I think this correlation is a false positive since there is no reason for AO to be correlated to a random synthetic red noise time series. The probability of rejecting the null hypothesis (meaning determining that there is no correlation based on how we defined the null hypothesis above) for all 200 correlations is 0.0035%. Therefore, it makes sense that we would fail to reject the null hypothesis in at least one case, just by chance. From this, I have learned that if you search hard enough, you will likely find a false positive saying that a relationship is significant, when really it isn't. I would then be wary of such an exercise in my research.

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