NAME: Matthew Thompson
DATE: 05/09/22
Final Project: Technical Review
STAT/CS 387

Introduction

The construction of my project was split into the 6 following files (in chronological order of date created):

1. Data_Cleaning.ipynb
2. Preliminary_Data_Analysis.ipynb
3. Regional_Streamflow_Exploration.ipynb
4. Regional_Streamflow_Teleconnection_Analysis.ipynb
5. Wavelet_Coherence_Testing.R
6. Wavelet_Coherence_Analysis.R

This project's direction was guided by my contact at the United States Geological Survey (USGS), which is also where I received these datasets from.

Code Description

Data_Cleaning.ipynb

The goal of this file was to clean each datasets that will be used for this project and save them as new CSV files for easy future use. First, the streamflow gauge metadata file needed to be accessed in order to acquire a list of the unregulated gauges. Unregulated, in this case, means that the flow is natural. Streamflow values can be "unnatural" if there is a dam upstream because the dam operations may not accurately represent the environmental systems at play. Next, a method for cleaning on gauge dataset is defined. The steps taken were as follows:

1. Remove null values
2. Drop irrelevant columns
3. Check for variable correlation
4. Check for outliers via box plot

Null values were luckily only near the beginning and end of the datasets, and because having complete time range data is so important to the analysis for this project, each dataset was shrunk to the middle section that contained no missing data. This still left me with daily data from 1980-2020. Once defined, it was assumed that each gauge dataset followed the same structure, so the first 2 cleaning steps were repeated for all 70 unregulated streamflow datasets. All these cleaned gauge datasets were then exported as CSVs.

Next, the Atlantic Multidecadal Oscillation (AMO), El Niño–Southern Oscillation (ENSO), Pacific Decadal Oscillation (PDO), and Pacific-North America (PNA) teleconnection pattern datasets were cleaned. Each one did not include traditional "NaN" values to represent missing data, and instead used different, but all very low negative values (ie. -99.90, -99.99, or -9999). These were treated like null values and those observations were only at the end of the chronologically ordered datasets, so those rows were easily removed. Each of these datasets were then exported as CSV files.

Finally, the Sun Spot dataset was cleaned. Messy column names from the original data were renamed and null values were investigated. It was found that the original sun spot counts variables divided into the

northern and southern hemispheres were the only columns without missing data, however, they were also the only columns necessary for this analysis, so each other feature was dropped. Outliers were quickly checked and then the dataset was exported as a CSV.

No severe challenges were encountered during the data cleaning process. There was a small hiccup importing the metadata file because it was stored using an unexpected file encoding, but that was easily solved using the "detect()" function within the Chardet package.

Preliminary_Data_Analysis.ipynb

The goal of this file was to run a preliminary data analysis on each of the 5 teleconnection pattern datasets, as well as on one chosen streamflow gauge dataset. The analysis process taken was as follows:

1. Plot line chart (Matplotlib): To visualize and understand the structure of the raw time series data before analysis. Used Matplotlib because it is easy to generate quick plots and I have years of experience with it.

2. Plot autocorrelation and partial autocorrelation for short term (couple months) and long term (1-5 years) lag values (both from Statsmodels package): To see the way the data correlates with itself at different lengths between observations. Used Statsmodels package because it the functions were very accessible.

3. Check if the data is stationary using the Augmented Dickey-Fuller (ADF) Test (Statsmodels package), if not stationary then transform until stationary: To prepare the data for future analysis. I used this test because it is an improved version of the original Dickey-Fuller test that supports the modeling of time trends, and I have used it in the past.

4. Decompose the time series using STL at different period lengths (Statsmodels package): To visualize the separated trend and seasonality components of the data. I used the Statsmodels method for this because it was easy to implement and made plotting the results extremely straightforward when compared to other packages.

The streamflow dataset contains data from 425 gauges across the Colorado River Basin. Gauge 09081600 was chosen for this analysis because it is within Colorado, which is the primary state of focus for this USGS data, and because its drainage area in square kilometers was very near the mean for every unregulated gauge (mean = 415.53, 09081600 = 432.90). This gauge is on Crystal River near Redstone, CO. The data is daily and extends from 1980-04-07 to 2020-03-31, resulting in 14,604 observations. This dataset contains the following variables:

- value = Daily streamflow discharge values in cubic feet per second.
- jd = The Julian Date for this day, which is essentially what number day into the year this day is.
- weibull_site = Steamflow percentile data, current day and 3-days before and 3-days after were combined into a 7-day rolling mean, then that value was compared to the past year of values to define its percentile using the Weibull plotting position function.
- weibull_jd = Steamflow percentile data, current day and 3-days before and 3-days after were combined into a 7-day rolling mean, then that value was compared to the past values from that same day of the year for all previous years to define its percentile using the Weibull plotting position function.

The AMO dataset contains monthly index values for the AMO index from 1950-01-01 to 2021-09-01. This leaves us with 861 entries to work with. During preliminary analysis it was found that the data was not stationary after failing the ADF test. The dataset was differenced using the "diff()" method within Pandas in order to make it stationary, with this approach chosen because of its easy access and solid results. After this transformation, the dataset now starts at 1950-02-01 meaning there are actually 860 entries.

The ENSO dataset contains monthly index values for the El Nino-Southern Oscillation (ENSO) from 1950-01-01 to 2021-10-01. This translates to 862 index observations.

This dataset contains monthly index data on the Pacific Decadal Oscillation (PDO) from 1950-01-01 to 2021-10-01, meaning there are 862 values.

The PNA dataset contains monthly index data for the Pacific-North American (PNA) pattern from 1950-01-01 to 2021-10-01. This translates to 862 data points.

The sun spot dataset holds daily data on the number of sun spots for each of the northern and southern hemispheres. This data is from 1978-01-01 to 2020-10-31, containing 15,645 entries.

No significant challenges appeared during the creation of this file outside of researching how to accomplish each analysis step. There was a small process of finding where the time range of each dataset overlapped as preparation for the coming analysis, but that was relatively straight-forward.

Regional_Streamflow_Exploration.ipynb

The goal of this file was to clean and run preliminary analysis on each of the 6 streamflow gauges that my contact at USGS selected for me to investigate. Each gauge is unregulated, within about the same drainage size, and represents a different region. The regions explored are as follows:

- Upper Missouri
- Arkansas-White-Red
- Rio Grande
- Upper Colorado
- Lower Colorado
- Pacific Northwest

In order to not repeat too much code, the data cleaning for these files was done using the same script highlighted with the Data_Cleaning.ipynb file, but not included within this file. For the most part this code contains the same preliminary data analysis approach as in the Preliminary_Data_Analysis.ipynb, but includes plots that directly compare the results of each step of the analysis for each region.

During the data cleaning process for these 6 gauge datasets, it was discovered that the datasets for the Arkansas-White-Red and Lower Colorado regions included strange, inconsistent null values starting in 2017. Luckily, this was near the end of the time range I hoped to explore, so the option that made the most sense was to drop the last few years of each dataset. The time range for the daily streamflow gauge data was now from 1980-05-01 to 2016-12-31. No other challenges were encountered within this file.

Regional_Streamflow_Teleconnection_Analysis.ipynb

The goal of this file was to run the correlation and wavelet analyses on the relationships between daily streamflow percentile data and teleconnection pattern data. First, each dataset was sliced to the time range defined in the Regional_Streamflow_Exploration.ipynb file and each streamflow gauge dataset was downsampled to monthly averaged observations to match the monthly teleconnection pattern data. Then, the following analysis approach for each region was taken:

1. Multicollinearity for the "weibull_site" and "weibull_jd" variables between each region was checked: This was done by plotting scatter matrices and correlation heatmaps using the respective Pandas functions. Spearman's correlation coefficient was used for this and all following correlation heatmaps because it is capable of recognizing monotonic relationships between variables, which are relationships I expect to see due to the nonlinearity of environmental interaction. This analysis confirmed my suspicion that "weibull_site" was not worth spending time investigating because it is not significantly different from region to region like "weibull_jd" is.

2. Correlation between streamflow percentile data and teleconnection pattern data for each region: This analysis used the Pandas scatter matrix and correlation heatmap functions to measure the strength of the direct relationships between these datasets

3. Statistical significance of Spearman's correlation coefficient checked for each teleconnection pattern variable's relationship to "weibull_jd" for each region: This was accomplished by using the "spearmanr()" function from the SciPy package. The correlation coefficients were relatively low, so I wanted to double check which relationships were statistically significant to support my understanding of the most important variable interactions.

4. Cross correlation between streamflow percentile and teleconnection patterns datasets at lags 1 through 12 for each region: This analysis was explored to see if teleconnection pattern data changes correlated with daily streamflow percentiles months in the future. This was accomplished using the "shift()" function within Pandas to realign the gauge data.

5. Statistical significance of Spearman's correlation coefficient checked for each teleconnection pattern variable's relationship to "weibull_jd" for each lag for each region: The same approach as the correlation analysis above was used for this exploration of cross correlation statistical significance.

6. Continuous Wavelet Transformation (CWT) was plotted for each variable for each region: This analysis was done to begin gathering an understanding of how the frequencies within the data of each time series variable vary in time. This is similar to a Fourier Transformation, except because this technique is built off of wavelets rather sine waves, it allows you to analyze the signal within a time space as well. The data was first standardized by subtracting the mean and dividing by the standard deviation. Then they were plugged into the "CWT()" function from the Scaleogram python package specifying a logged scale for the y-axis based on 220 months (half the dataset), as well as Morlet for the wavelet family. The scale of 220 months was a rather arbitrary selection, I just wanted at least 15 years in order to see past the 12 year patterns that were recognized during early analysis. The decision for the Morlet wavelet was based on the success of this wavelet in previous literature for recognizing patterns within the relationships between teleconnection pattern and streamflow datasets [1, 2].

The challenges associated with this file have to do with learning how to run wavelet analysis within Python. The ".cwt()" function within SciPy, PyWavelets, and Scaleogram were all explored for CWT. SciPy was the first package I looked into because it is the one I am most familiar with, however, the visualization it created was odd in comparison to the CWT plots I had seen and I could not find examples of how to create plots like that. Then, I tried PyWavelets, but it also created a strange visualization and also suffered from lacking examples. I believe that if I had a better understanding of wavelet analysis that I could have gotten these packages to work, however, that that level of understanding is outside the scope of a semester project. Scaleogram is a package built off of PyWavelets with the sole purpose of easily producing industry-standard CWT visualizations. My implementation worked well and after some parameter tinkering I was able to display the CWT output the way I wanted to.

Moving on to WTC is where this project got super challenging. After researching, I identified PyCWT and PIWavelet as being the only Python packages available for accessible WTC. However, PyCWT was similar to SciPy and PyWavelets in the sense that it provides the tools needed to create visualizations, but without a deep understanding of wavelet analysis it was extremely difficult to do so. After playing with it for some time I was finally able to create a visualization, but it looked absolutely nothing like what I had expected. So, I moved on to PIWavelets. In order to install PIWavelets, you also need to install GNU Octave and a package called Oct2Py. From my understanding, GNU Octave is an open-source alternative shell to MATLAB, and what I learned during my research is that the industry standard language for wavelet analysis is MATLAB. What PIWavelets does is translate Python functions to MATLAB and run them within the GNU Octave shell by using Oct2Py as a medium. Unfortunately, getting these packages and applications to work nicely together on my computer proved to be an impossible task. I spent many hours altering lines within my PATH environment, installing this packages locally vs. within my Anaconda environment, sorting

through strange download errors, and even installing them on a friends computer, all to no avail. Eventually, I made the decision to give up on Python, export my current datasets as CSV files, and to try using R instead.

Wavelet_Coherence_Testing.R

The goal of this file was to see if I could get any of the R packages for Wavelet Coherence (WTC) to work. Those packages were WaveletComp and biwavelet. WaveletComp was tested first and right out of the box it created a beautiful and informative plot. The only issue was that I could not get it to display a period larger than 130 months, or 11 years. The period is the scale, or size, of the wavelet as it analyzed the specified signal relationship. In a way, this shows the frequency of two signals moving together. It is really important that I analyze the time space around a period of at least 12 years due to the patterns I saw during my preliminary analysis. I toyed with this for some time before deciding to try biwavelet. Just like WaveletComp, I easily followed the well written documentation and examples and was able to create a gorgeous WTC plot, and this time there was an easy parameter for adjusting maximum period. A very clear takeaway from this experience is that Python is severely behind when it comes to well documented and accessible packages for wavelet analysis.

Wavelet_Coherence_Analysis.R

The goal of this file was to run WTC analysis for each variable relationship to "weibull_jd" for each region. For each combination I loaded the CSV files from the previous Python script, ran wavelet coherence, and plotted it using the packages specific plotting function the takes in wavelet analysis outputs. This process also runs a number of Monte Carlo randomizations for determining statistical significance. I set mine to 1000 randomizations at a significance level of 0.95.

Scope Reflection

The original goal of this project was to explore the relationships between teleconnection patterns and daily streamflow percentile data so that they could be incorporated into the Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN) combined model I am working on for CS 354: Deep Learning. This goal was adjusted to be something to pursue given extra time because the analysis and relationship extraction problem turned out to be much larger than expected. After speaking with my contact at USGS, it also seemed like a significantly more interesting project to look into regional differences rather than focusing on outputting relationships for the potential of a small improvement to forecasting. I also stated that I would use techniques like continuous wavelet transform (CWT), fourier spectrum analysis (FSA), wavelet coherence (WTC), and cross wavelet (XWT) analysis, as well as run an ARIMA model, and explore different regions of streamflow gauges. I did end up exploring different regions of streamflow gauges, but I did not use fourier spectrum analysis (FSA), cross wavelet (XWT) analysis, or ARIMA for this analysis. All decisions to not use specific tools also included the consideration of the time constraint that this project had. Fourier spectrum analysis (FSA) was not used because continuous wavelet transform (CWT) has supporting literature of better performance, on top of the fact that its lacking ability to understand frequency alignment within specific time periods [2]. Cross wavelet (XWT) analysis was proven to be outperformed by wavelet coherence when attempting to identify correlation within local time and frequency spaces [3]. ARIMA was not used strictly because I did not have the time. I am very happy with the way that this project evolved because I think the results I found are not only more interesting than trying to plug them into a model, but also more novel in the sense that I directly compared how so many different sub-basins relate differently to a large collection of teleconnection patterns.

Scientific Computing Best Practices

I think that I did a good job of following the best practices of scientific computing within this project. I have identified the following as practices that I successfully followed:

- Consistent variable names and code formatting: imported packages at the top; separated notebook cells for readability and ease of showing related lines of code; organized file structure for code and data for ease of importing and exporting
- Documenting code design and purpose: clear code commenting; included thought process/explanations where necessary; included URLs of websites I used code from if they were not documentation
- Make incremental changes

After looking through my code and reflecting, I realized that I definitely could have made my code more modular in spots. A lot of the time I would start with analysis or plotting on just a single example and would build from there, however, rather than constructing a function to call multiple times I got sucked into the trap of copy and pasting code. Some of the time this was justified for ease of seeing the outputs for individual parts of the analysis, but especially for the CWT code I could have used a function instead. I lost sight of proper coding best practices in the process of exploring some totally new subjects and tools. This is a very valuable lesson for me.

Failures

The primary failure of this project was getting Wavelet Coherence to run within a Python script as explained within the Regional_Streamflow_Teleconnection_Analysis.ipynb section. The only other experience that could be considered a failure is using 1 gauge for each region to represent to entire region, instead of my original plan to find a way to standardize and average each gauge within each region. Unfortunately, due to how many strange null values and different time ranges there were across all the gauges, this option became unfeasible for the time constraints of this project because each of the 425 gauges would have had to have been cleaned and converted to a uniform time range.

References

1. Canchala, T., Loaiza Cer on, W., Franc es, F., Carvajal-Escobar, Y., Andreoli, R.V., Kayano, M.T., Alfonso-Morales, W., Caicedo-Bravo, E., Ferreira de Souza, R.A. (2020). Streamflow Variability in Colombian Pacific Basins and Their Teleconnections with Climate Indices. Water, 12, 526.
2. Fu, C., et al. (2012). Analyzing the combined influence of solar activity and El Ni no on streamflow across southern Canada. Water Resources Research, 48
3. Grinsted, A., Moore, J., Jevrejeva, S. (2004). Application of Cross Wavelet Transform and Wavelet Coherence to Geophysical Time Series. Nonlinear Processes in Geophysics, 11.