Sara Boburka
INLS 792
Final Report

Project Motivation & Goals

I chose this particular domain and path of work because I'm interested in combining my data skills with environmental science. I had a connection with Dr. Sebastian and enjoyed her focus on computational skills within hydrology, so I felt this project would be a great opportunity to learn more about hydrology while getting to use the computational and quantitative skills I've learned during my time in the Certificate in Applied Data Science program.

My goals started quite lofty, looking to analyze general trends across all gages in North Carolina and create a model to predict floods based off that data. However, my final goals list ended up like this:

- Analyze trends across three gages in NC (Tar, Neuse, Cape Fear Rivers)
    - Variables of interest: water year, season, day of week, gage height, 7 day rolling average, 30 day rolling average.
    - Analyze discharge trends across months, seasons, and water year.
- Create a linear regression model to measure relationships between variables.
- Answer questions from Sarah Brannum's thesis (Streamflow & Design Floods with USGS).
    - Sarah was unable to fit gages that had years missing to the distribution.
        - Use other gage information/proximity data to fill in missing data.
        - Fill in the missing years using the average of the years around it.
        - Recalculate/regraph design flood at Tar, Cape Fear, and Neuse Rivers.
    - Examine how the inclusion or removal of events of a certain type (e.g. hurricanes) impacts the magnitude of the design flood.
        - Differentiate trends between different events, such as cyclones and hurricanes.
        - Determine what differentiates these events from each other.
        - 30 years is deemed "reasonably long" for a flood period – investigate minimum number of observations needed for accurate analysis.

Problem Domain

The data I used for this project comes from the USGS. Because my work builds off of some of the work discussed by Sarah Brannum in her thesis on design flood, I picked three gages in North Carolina to pull data from. Sarah had used these gages in her design flood models, and because I wanted to see the effect of data manipulation on those floods, I felt it was best to match the gages.

Hydrology and design flood play a large role in environmental planning, management, and urban planning. The southeastern United States is particularly vulnerable to floods, and the resulting damage from such events tends to be quite costly. Graphing and analyzing trends in design flood over time provides valuable resources to these communities and local practitioners so they can make efficient decisions regarding flood planning, preparation, and damage mitigation.

Methods & Results

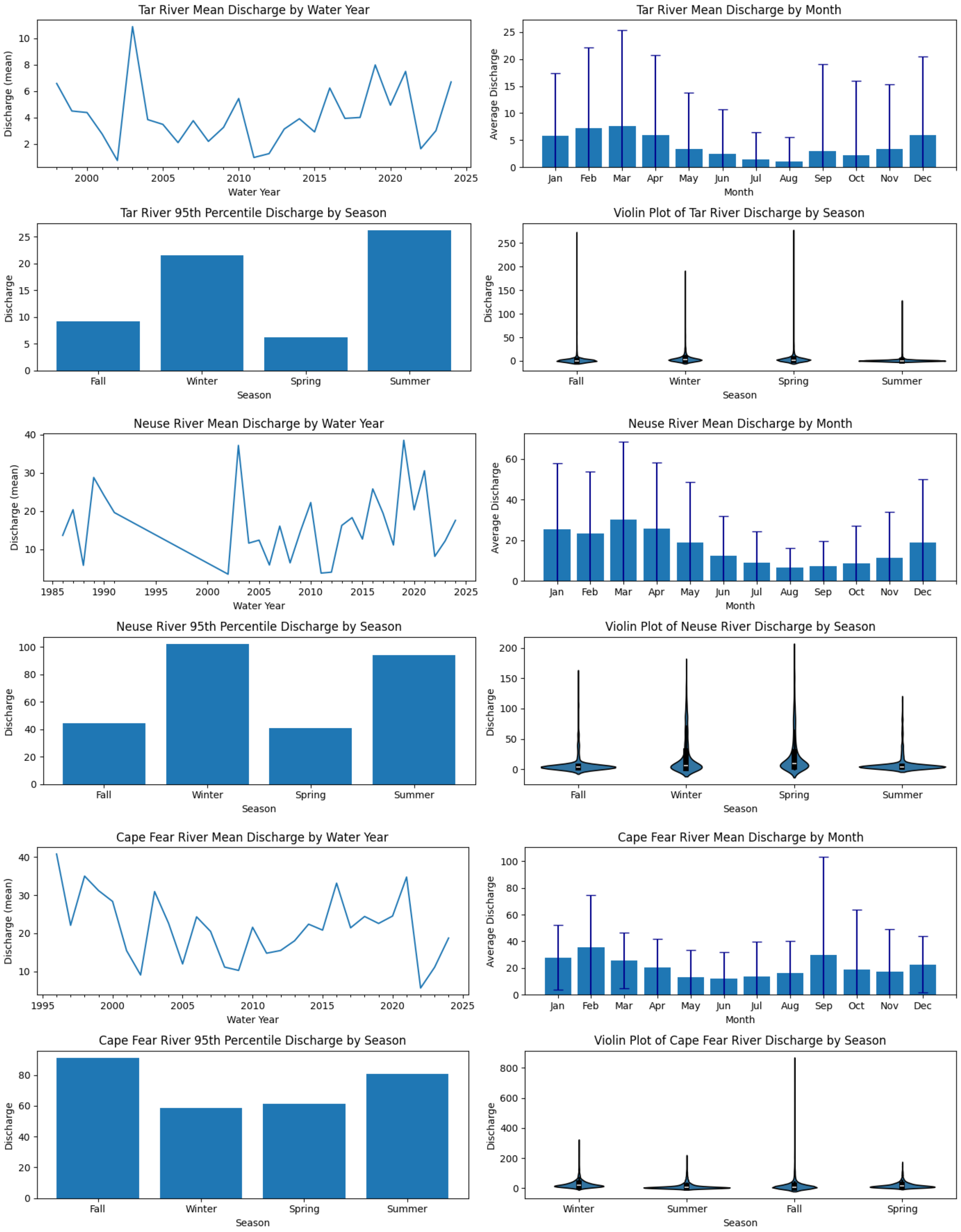
For data collection, I used R scripts and the dataRetrieval package from USGS to pull the specific gages I wanted. From here, I was able to export the data as csv files and load them into a Python environment. Each gage had two files: one for discharge, and one for gage height.

In Python, I created functions that would load in, clean, and combine the files for each gage. From the discharge file, I kept the following variables: ['Date','Month','Day','Q','waterYear','Q7','Q30']. I renamed "Day" and "Q" to be more descriptive; "Day" counts the day of the year, and "Q" is the gage discharge. From the gage height file, I kept "Date" and "X_00065_00003". I renamed "X_00065_00003" to "gageHeight" to be more descriptive. I then merged the two dataframes on "Date" to create a combined dataframe with all the data I wanted to keep. Since I put this code in a function, I could easily apply it to all three gages. As to not repeat myself, for each process described further, unless otherwise stated, I created a function to easily repeat the task through all three gages.

After cleaning, I wanted to add a "Season" variable to see if later, when creating a linear model, "Season" would be a significant predictor of gage discharge. For this, I used the following method to assign the season:

If the month number is...	The season is...
12, 1, 2	Winter
3, 4, 5	Spring
6, 7, 8	Summer
9, 10, 11	Fall

For data visualization, I created four plots per gage: mean discharge by water year, mean discharge by month, 95th percentile discharge by season, and a discharge by season violin plot. It was at this stage where I could see that there was a significant gap in data for the Neuse gage.



Tar River Mean Discharge by Water Year

Tar River Mean Discharge by Month

Tar River 95th Percentile Discharge by Season

Violin Plot of Tar River Discharge by Season

Neuse River Mean Discharge by Water Year

Neuse River Mean Discharge by Month

Neuse River 95th Percentile Discharge by Season

Violin Plot of Neuse River Discharge by Season

Cape Fear River Mean Discharge by Water Year

Cape Fear River Mean Discharge by Month

Cape Fear River 95th Percentile Discharge by Season

Violin Plot of Cape Fear River Discharge by Season

From here, I began the process of filling in missing data. Since the Tar and Cape Fear gages had relatively small amounts of missing data, I was able to use pd.date_range() and the interpolate method to fill in the gaps. Because of the magnitude of missing data in the Neuse gage data, I opted to use data from the gages closest to the Neuse gage (Crabtree Creek & Marsh Creek). While these gages were also missing data, they were able to fill in about half of the missing data points for the Neuse gage.

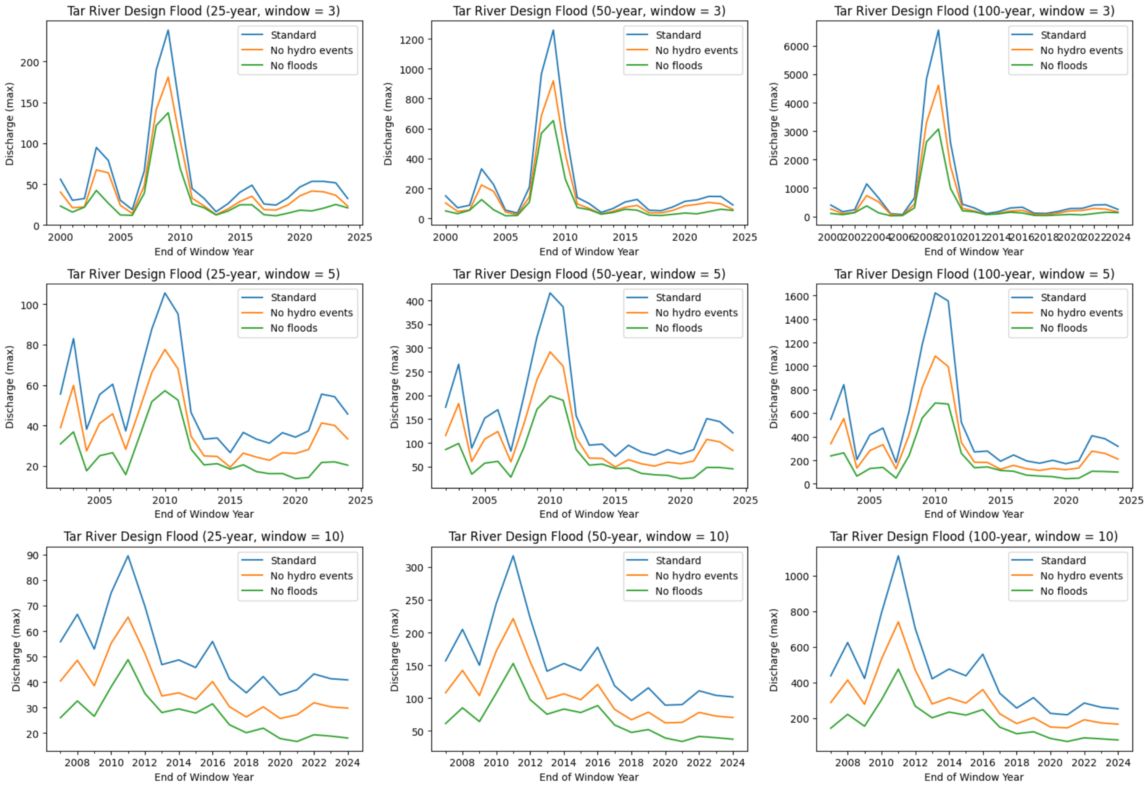
The last data engineering step I took before analysis was introducing two new variables: “Flood” and “HydroEvent”. I wanted a way to define in the data when a flood was occurring, and when another significant hydrologic event was occurring. I defined a flood as any point where the “gageHeight” variable was greater than or equal to the 95th percentile of “gageHeight” for that gage. Hydrologic events were any point where the discharge was greater than or equal to the 90th percentile of discharge for that gage, and “gageHeight” being *less than* or equal to the 95th percentile of “gageHeight” for that gage.

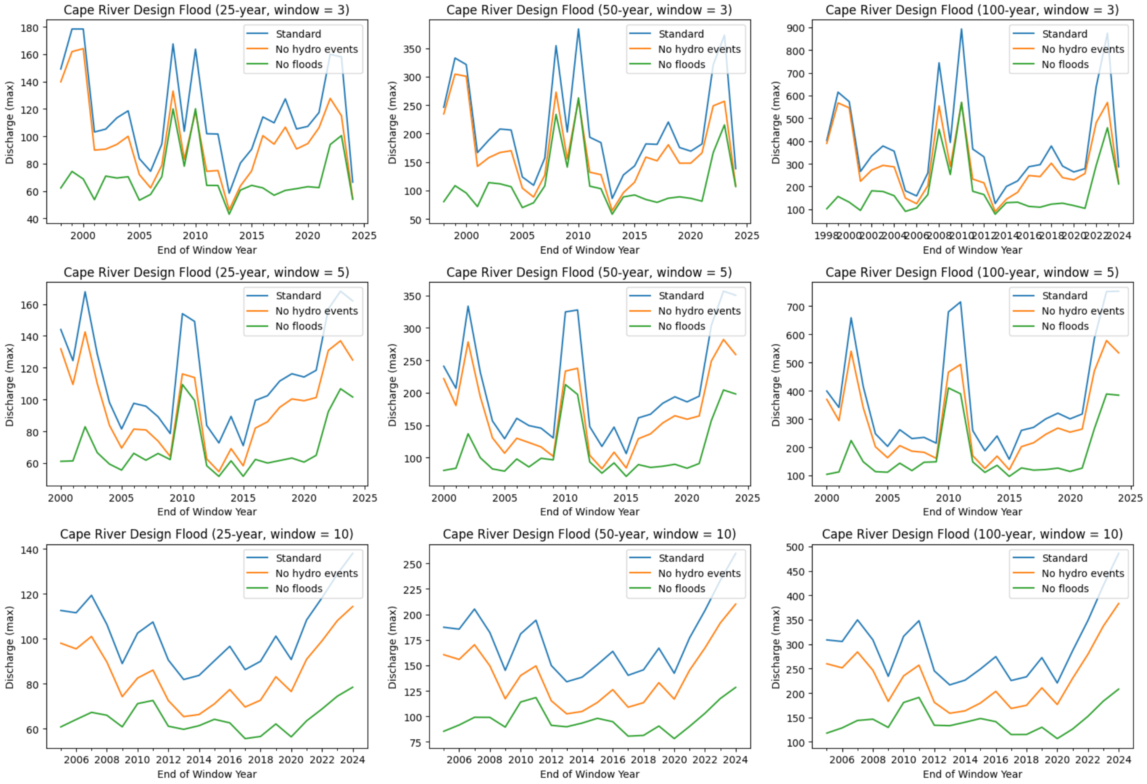
To look at correlations between predictors for discharge, I used statsmodels’ Ordinary Least Squares linear model. For each gage, I looked at the following variables: ['waterYear', 'Q7', 'Q30', 'gageHeight', 'Season_Spring', 'Season_Summer','Season_Winter']. The “Season” variable had to be one-hot encoded, as OLS doesn’t work with non-numeric data. One season was dropped to prevent multicollinearity in the model. For these models, I used the imputed/filled data sets. The VIF for all variables was less than 4 in all datasets, so I did not remove any variables.

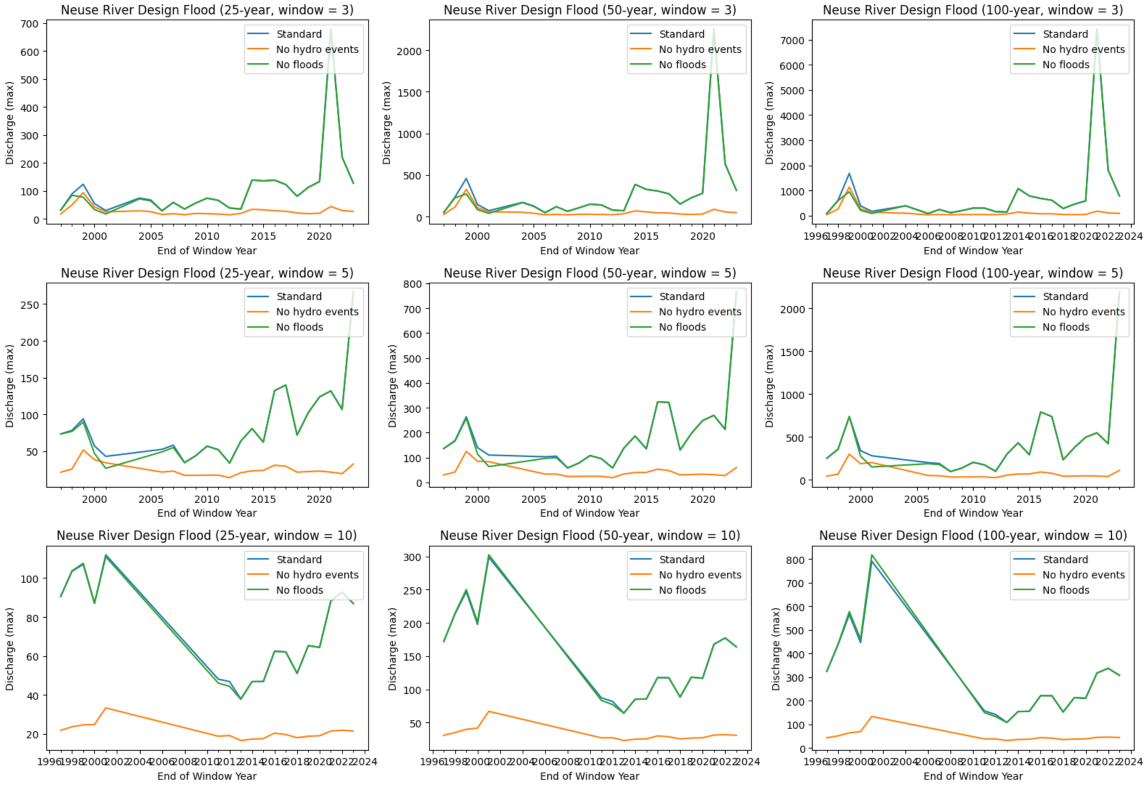
Gage	R-squared	Significant Predictors (p<0.05)	
Tar River	0.841	Flood HydroEvent Q30 gageHeight	Season_Spring Season_Summer Season_Winter
Neuse River	0.813	waterYear Flood HydroEvent Q7 Q30 gageHeight	Season_Spring
Cape Fear River	0.806	waterYear Flood HydroEvent Q7 Q30 gageHeight	Season_Spring Season_Summer Season_Winter

When implementing design flood, I noticed that the data I pulled from USGS seemed to be missing a lot of the historical (pre-1990s) data, which caused the range to be too small to run a window of 30 years, so I got to explore how smaller rolling windows affected the design flood outcomes. I wanted to experiment with rolling windows of 3, 5, and 10 years. I kept the return periods (25-/50-/100-year) the same. I referenced the Natural Resources Conservation Service's National Engineering Handbook (Part 654) a *lot* in this step. I opted to use a generalized extreme value distribution, as I was struggling to get the return period to work using scipy's gumbel_r() function.

The design flood generally matched Sarah's findings in regards to return period: increasing the return period has the same trends with a higher magnitude. Comparing window sizes, smaller windows seem to have a larger effect on the design flood, and larger window sizes tend to have more similar trends between data sets (data vs removed hydrologic events vs floods).







Future Work

Trying to fill the years-long gap in Neuse discharge proved difficult, as the gaps weren’t small and spread throughout; I couldn’t use linear interpolation like I had in the other gages. With the limited data

I had, it was difficult to come up with an interpolation/ML-based solution. Additionally, when running the imputed data in the design flood code, there were some issues with a gap that couldn't be filled in the Neuse data, and I couldn't figure out why.

I plan to expand on this project in the following year (2024-2025) in an honors thesis. I will be working with Dr. Sebastian to build a model that can help predict floods in data-poor regions using a variety of datasets, including the USGS dataset that I worked with in this project.

Self-Reflection

I found this project very enlightening. I learned a lot about myself and my planning habits, particularly how to work around them to stay on track and balancing this project with my other courses. I also really enjoyed getting to clean and engineer data sets instead of being handed a pre-processed set.

I believe the most insightful part of this project was seeing how my goals shifted over time as I re-evaluated my goals throughout the semester. I started off with a very large, lofty goal, but I had to adjust to meet the time constraints and availability of data. I also realized that I don't understand machine learning as well as I thought; I went into this project assuming I could do something cool with ML, but soon came to realize that with the given time frame and data, it would require a lot more investigation and learning to really grasp how to fit (or force) ML into my project.