Atlantic Sturgeon Habitat Suitability Analysis in the Chesapeake Bay

Sydney Whitaker

B.S. University of Maryland

Submitted to the Department of Data Science
Loyola University Maryland
in Partial Fulfillment of the Requirements for the
Degree of Master of Science

April 2025

ABSTRACT

This research combines logistic regression modeling with GIS-based spatial analysis to assess habitat suitability for Atlantic sturgeon (Acipenser oxyrinchus oxyrinchus) in the Chesapeake Bay. Once widespread, Atlantic sturgeon populations have been reduced by overfishing, habitat degradation, and water quality impairment. Using data from the United States Geological Survey (USGS) and the National Oceanic and Atmospheric Administration (NOAA), this paper distinguishes significant habitat variables—i.e., water temperature, dissolved oxygen, pH, turbidity, nitrate, and discharge—and models their influence on habitat suitability of five tributaries: Mattawoman Creek, Potomac, Rappahannock, James, and Pamunkey Rivers. Logistic regression models were developed to predict binary habitat suitability from daily water quality data. Model performance was assessed using ROC curves, AUC values, and confusion matrices. Spatial predictions were mapped using QGIS to detect seasonal patterns and river-specific suitability zones. Outcomes show that the availability of habitat is significantly influenced by seasonal change, with autumn and spring being the most suitable. Among all variables, temperature, dissolved oxygen, and discharge emerged as the most influential predictors across models. These results support targeted restoration, guide fisheries management, and inform long-term climate adaptation strategies for this endangered species.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

Motivation and Background

Motivation

The Chesapeake Bay once held abundant populations of Atlantic sturgeon, a primitive anadromous fish that moves between freshwater spawning grounds and the ocean. But because of overfishing, habitat loss, and pollution, populations have rapidly declined. Even though it was listed under the Endangered Species Act (ESA) in 2012, recovery has been sluggish. A significant constraint to conservation is the lack of predictive modeling when identifying habitats or determining potential habitat suitability. This can cause conservation efforts to be inefficient or misplaced.

This study is important as it establishes a quantitative basis for understanding environmental conditions that impact Atlantic sturgeon habitats. By focusing on the specific environmental parameters that influence habitat suitability, this work helps provide actionable data for conservation management. This study can be useful for policymakers, fisheries managers, and conservation agencies through knowledge of habitat restoration activities, water quality management, and climate adaptation planning.

Project Background and Relevance

The inspiration behind this project is my passion for conservation and strong interest in applying data science to addressing urgent environmental issues. The Chesapeake Bay is a critical ecosystem and a historically and culturally important waterway. Understanding how human activity and climate change affect this ecosystem and guiding conservation efforts aligns with a commitment to using scientific tools for ecological preservation. By determining drivers of sturgeon habitat, we can inform adaptive approaches to enhance sturgeon population recovery.

During the research, several challenges arose. Merging disjointed datasets from various sources involved a lot of data cleaning, preprocessing, and standardizing for them to be coherent. This added another level of complexity, especially in reconciling temporal and spatial inconsistencies between datasets. The use of QGIS also involved a learning curve in which new technical proficiency in geospatial visualization, rasterization, and spatial accuracy measurement needed to be learned. It was an enriching experience that gave greater insight into how spatial tools can be utilized in ecological studies.

Study Objectives

The general goal of this study is to model Atlantic sturgeon habitat suitability in the Chesapeake Bay using logistic regression and GIS-based spatial modeling. This study aims to clarify the primary environmental determinants of sturgeon habitat and map the extent of suitable habitat to guide future conservation planning. By developing prediction models, the study can help prioritize restoration efforts in those locations with the greatest potential for sturgeon recovery. Additionally, the findings will be useful for stakeholders involved in water quality management and fisheries conservation.

Challenges in Data Integration and Analysis

Due to the origin of the data sources, most of the datasets were of varying formats and lacked common coordinate systems, thus necessitating a lot of data cleaning and preprocessing. The differences in temporal resolution of the datasets also led to disparities that needed to be dealt with for comparison purposes. These issues were solved through the use of scripts and geospatial software to normalize the data and spatially and temporally align it, resulting in a more uniform dataset to examine.

Implications for Conservation and Policy

Beyond the study-specific conservation targets, the work has general relevance to environmental policy and monitoring. The approach described here is transferable to other habitats and species, and the study provides a model for assessing habitat suitability in the face of environmental change. Through the creation of more complex predictive habitat suitability models, the work advances ecological modeling methodology that will be applicable to other threatened species and ecosystems. Besides, the study can inform future policy on regulation that balances economic activity—i.e., fishing and industry—with conservation needs. The ability to predict trends in habitats over time also enables long-term planning for sustainability so that conservation efforts take into account environmental change.

CHAPTER II

Research Platform

Overview of Data Sources and Selection

The ultimate goal of this research was to model Atlantic sturgeon habitat suitability in the Chesapeake Bay region from environmental variables critical to sturgeon reproduction and survival. The data for use in this analysis were both time series and spatial data sets, allowing an opportunity for a comprehensive examination of habitat quality both through time and across space. The two principal sources of data were the USGS National Water Information System (NWIS) and the NOAA ESA Critical Habitat Geodatabase.

The USGS NWIS provided daily water quality and streamflow data from five strategically located monitoring stations based on their proximity to known sturgeon populations. These stations provided the temporal resolution to investigate short-term habitat fluctuation, recording daily changes in key environmental factors like discharge, temperature, and turbidity. The NOAA ESA Critical Habitat Geodatabase provided spatial data, charting designated critical sturgeon habitats. These geospatial data enabled environmental conditions to be associated with habitat location and habitat suitability more precisely to be determined.

These five USGS monitoring stations were chosen to sample a range of hydrological conditions throughout the Chesapeake Bay and provide a distinct perspective on Atlantic sturgeon habitat suitability at each station. The Mattawoman Creek station (USGS_01638500), located in the upper Potomac watershed, provides data on temperature, pH, and dissolved oxygen, capturing habitat conditions in a suburban freshwater tributary. The Potomac River station (USGS_01646500) monitors turbidity, pH, and discharge in a major tidal tributary that supports both

juvenile and adult sturgeon. The Rappahannock River station (USGS_01668000) exhibits significant discharge variability and is historically linked to sturgeon activity, making it essential for interpreting seasonal migrations. The James River station (USGS_01673000) is a key spawning river with documented telemetry data and a long history of sturgeon use. Finally, the Pamunkey River station (USGS_02035000) is a smaller tributary within the estuarine system and may serve as a nursery area due to its variable water quality conditions. These stations provided high-resolution temporal data—such as daily, monthly, and seasonal changes in water quality parameters—supporting a dynamic analysis of habitat suitability. The NOAA ESA data enhanced these observations by adding a spatial reference to federally designated critical habitats, allowing for a more detailed geospatial evaluation of sturgeon habitat conditions.

Plan of Analysis and Investigation

A logistic regression model that predicts sturgeon habitat suitability from environmental predictors of temperature, turbidity, dissolved oxygen, pH, and discharge was created. Based on environmental thresholds taken from peer-reviewed literature and NOAA criteria for sturgeon survival, the model classifies each day as suitable (1) or unsuitable (0) for sturgeon habitat.

Station-based predictions were plotted against their geographic locations. This provided a visual evaluation of the habitat conditions, allowing conservation managers to see how sturgeon habitats change seasonally and schedule restoration activities accordingly.

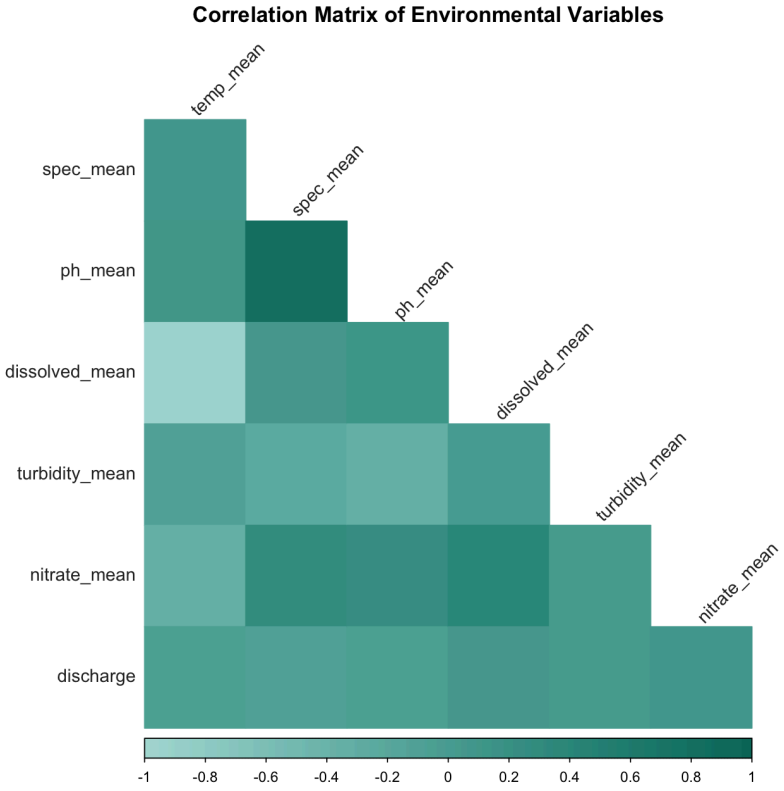
Data Preprocessing and Cleaning

Data preparation required intense cleaning and integration to achieve compatibility among the NOAA critical habitat spatial data and USGS water quality

data. Whereas the NOAA dataset required minimal preprocessing, the USGS station data required a high degree of harmonization due to variations in variable formats, time resolutions, and coordinate reference systems. The most challenging was integrating the data across various USGS monitoring stations, wherein each was monitoring a differing subset of environmental variables. Standardization was required to balance temperature, turbidity, dissolved oxygen, pH, and discharge between stations to integrate the dataset.

One of the key preprocessing tasks was coordinate system matching between NOAA ESA habitat geodatabase and USGS station coordinates. Since NOAA's dataset was predominantly spatial and USGS data was time-series, temporal matching was essential. Daily readings for each station were joined by timestamp to create a merged dataset in which all the predictor variables for a day were captured in the same record.

To further evaluate relationships between predictors and detect potential multicollinearity, a correlation matrix was generated using all primary water quality variables. This matrix revealed both strong positive and negative associations, such as a strong negative correlation between temperature and dissolved oxygen, and a positive correlation between discharge and turbidity. These relationships informed model refinement and variable selection.



Note. Relationships such as the negative correlation between temperature and dissolved oxygen, and the positive link between turbidity and discharge, were considered during model design to avoid multicollinearity.

*Figure 1***.** Correlation Matrix of Key Water Quality Predictors.

Outlier handling and detection helped ensure data integrity. Z-scores were used to detect extreme values in water quality parameters, while Winsorization was used to censor the effects of outliers. Outliers were substantially reduced through Winsorization and log transformation, resulting in a more normal distribution and improved model compatibility. To visually assess outlier adjustment and normalization, Z-score distributions were plotted for all key environmental variables after preprocessing.

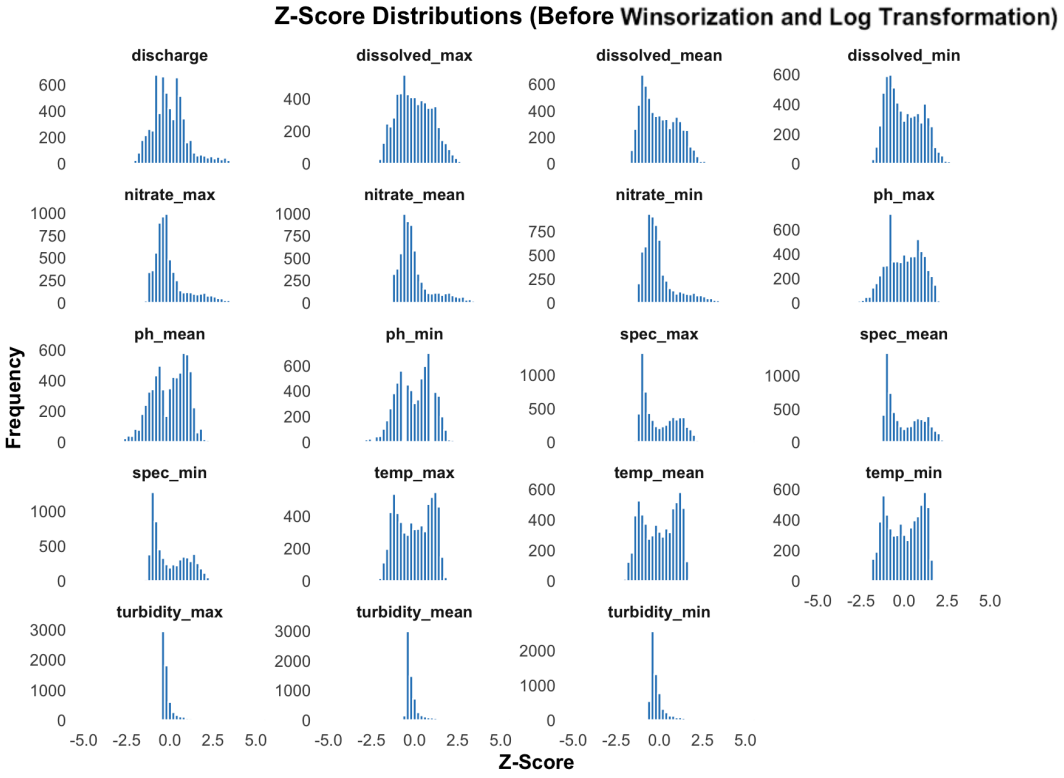


Figure 2. Z-Score Distributions Before Winsorization and Log Transformation.

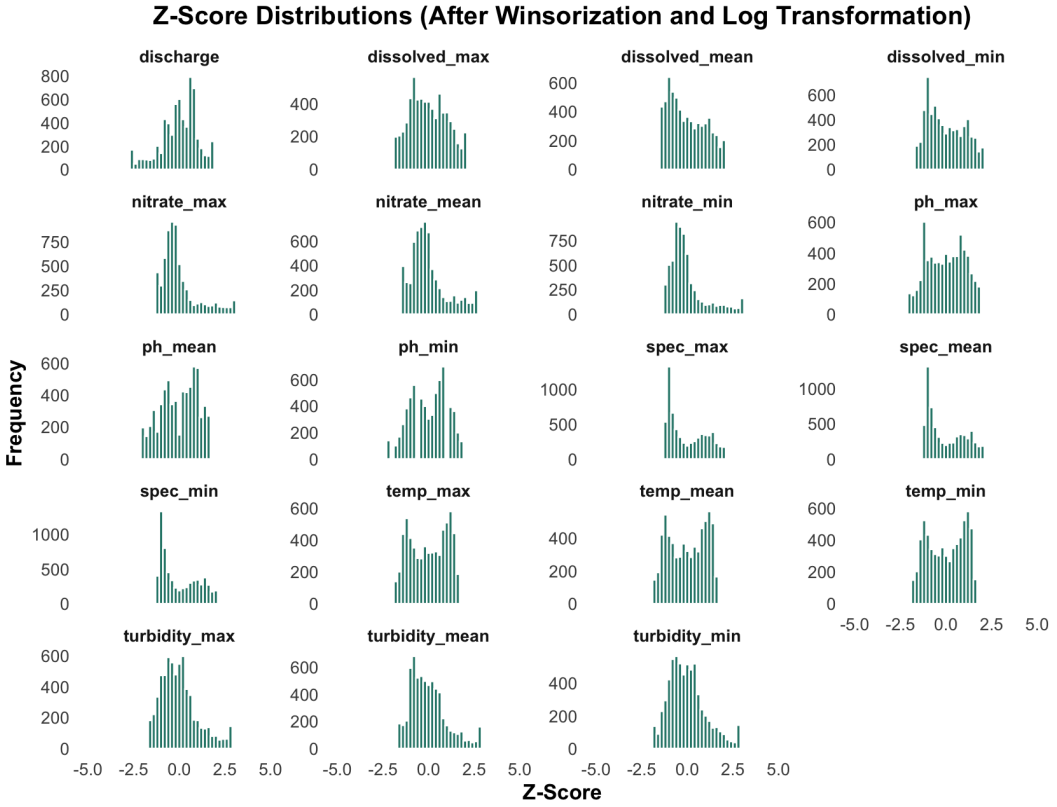


*Figure 3***.** Z-Score Distributions After Winsorization and Log Transformation.

Log transformations helped decrease skewness for certain variables like nitrate concentration and turbidity. Missing values were also a problem, though more so for turbidity and discharge measurements for some stations. To address this, K-Nearest Neighbors (KNN) imputation was employed, allowing the estimation of missing values from environmentally similar observations without interfering with the relationships between variables.

To visually compare the distribution of environmental variables by habitat suitability class, boxplots were created. These plots show how each predictor differs between days classified as suitable versus unsuitable, providing insight into the separation power of individual variables for classification modeling.

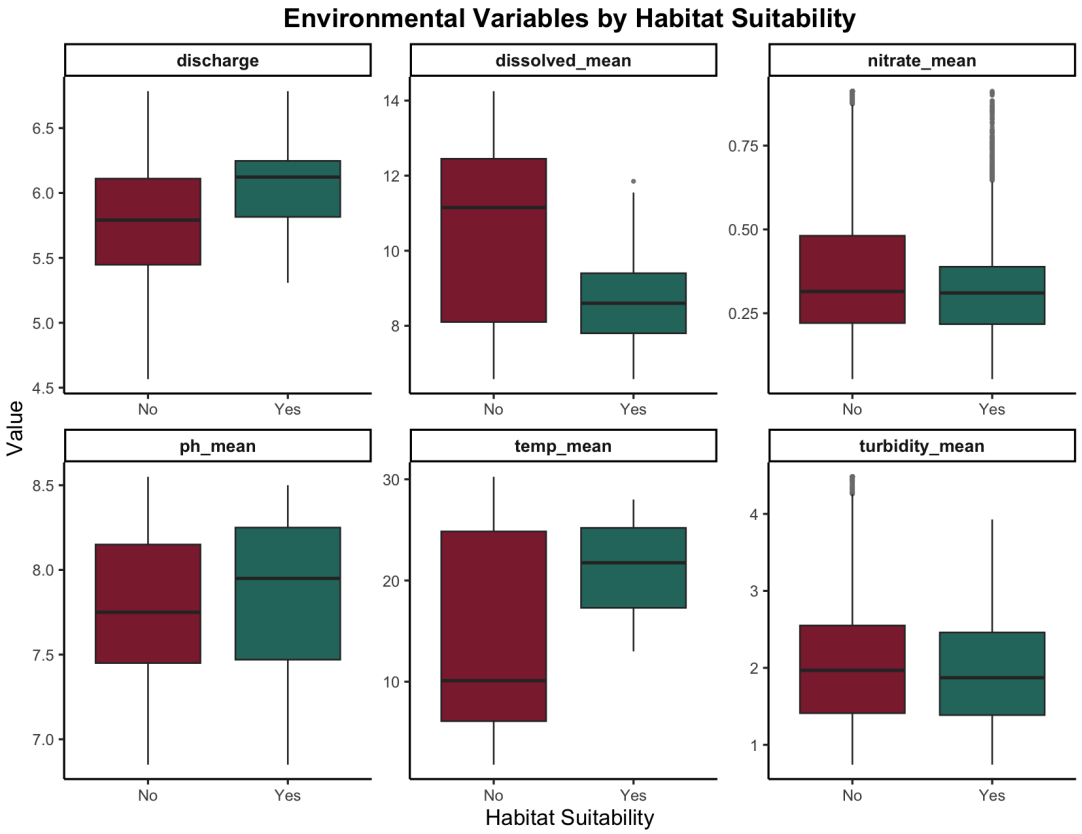


Figure 4. Boxplots Comparing Environmental Variables by Habitat Suitability Classification.

Suitable habitat days generally exhibit lower turbidity and temperature, and higher discharge, pH, and nitrate levels—supporting ecological theory and model expectations.

Hardware, Software, and Analytical Tools

This research was carried out in an integrated set of software packages. Statistical modeling and data preprocessing were performed using R (RStudio), and spatial data analysis and habitat suitability mapping were performed using QGIS. Some important R packages, such as ggplot2, caret, glmnet, sf, and raster, were implemented for visualization, regression modeling, and GIS data processing. Custom scripts were run to automatically transfer logistic regression outputs to

GIS-compatible formats to combine the statistical analysis and geospatial visualization.

New Equipment and Methods

The incorporation of logistic regression in GIS involved learning new spatial analysis competencies in QGIS, specifically in managing rasterization, spatial overlays, and geospatial data formatting. It involved translating statistical outputs into spatial datasets that can be mapped. Feature selection by using Lasso and Ridge regression was also a new addition, which involved optimizing the model by selecting the leading predictors.

Surprise Challenges and Solutions

Using data from various sources was most probably the largest difficulty. USGS and NOAA data were in various formats and resolutions and took a lot of preprocessing before an equivalent format could be exchanged. Environmental data similarly lacked spatial metadata, and projecting the observations within QGIS was inaccurate. Coordinate reference system conversions, hand inspection, and cleaning to fix this.

Handling large datasets was another major challenge. The USGS water quality dataset had millions of records, so computational efficiency was an issue. Data filtering and aggregation methods were employed to reduce analysis without losing important environmental trends.

In spite of the limitations, logistic regression and GIS analysis were successfully merged together to generate a strong model of Atlantic sturgeon habitat suitability. The research offers a data-driven, methodological framework for critical habitat identification with potential to inform Chesapeake Bay conservation activities.

CHAPTER III

Literature Review & Related Work

Related Research on Atlantic Sturgeon Habitat

The Atlantic sturgeons' habitats are heavily impacted by climate change, including spawning and juvenile survival. Processes required for breeding are affected by temperature, leading to reduced viability of eggs and disruption to migration timing (Hilton et al., 2016; Peterson et al., 2018). Metabolic processes are increased by high temperatures, and dissolved oxygen is reduced, causing physiological stress (Niklitschek & Secor, 2019). Models of thermal preference in sturgeon populations could help identify safe breeding zones, improving conservation outcomes.

One of the most serious climatic threats is the increase in hypoxic zones, which reduce habitat quality and force sturgeons into lower habitats. Hypoxia, caused primarily by eutrophication and runoff nutrients, reduces migration, feeding ability, and reproductive ability (NOAA Fisheries, 2022; Kemp et al., 2019). Prolonged exposure to low oxygen can result in local depletion and hinder recovery (Hager et al., 2021). Climate models show that hypoxic zones in the Chesapeake Bay are likely to expand in the future, further exacerbating the problem for Atlantic sturgeon.

Impact of Climate and Environmental Stressors

Rising sea level and changed precipitation regimes also destabilize estuarine salinity regimes, decreasing access to essential freshwater spawning habitat (Breece et al., 2018; Kocovsky et al., 2018). Chronic storm disturbance decreases salinity, stressing brackish-habitat-adapted adult fish, and extended drought elevates salinity intrusion into freshwater nurseries (Fox et al., 2019). These alterations harm conservation by rendering habitat suitability more variable through time (Hilton et al.,

2016). Environmental monitoring and GIS modeling can deliver early warning systems for these changes, with specific mitigation responses.

Importance of Water Quality

Proper dissolved oxygen levels are also essential to sturgeon survival, whereby hypoxic stress compels them to abandon extensive spawning and feeding habitats (Niklitschek & Secor, 2019). Young sturgeons are also intolerant of low water quality, and nursery habitat degradation is therefore a concern (Peterson et al., 2018). Telemetry research has demonstrated that turbidity, temperature, and salinity gradients are all associated with migration, and significance is supported by the studies (Balazik et al., 2012; Breece et al., 2018). They are significant parameters for short-term and long-term habitat suitability modeling.

Pollution and Habitat Destruction

Pollution, bycatch, and habitat degradation also threaten the species significantly, which is critically endangered (IUCN, 2022). Dams block access to historical spawning habitat and alter river hydrodynamics, sediment transport, and dissolved oxygen (Hager et al., 2021; Fox et al., 2019). Industrial contaminants such as heavy metals and PCBs also impair reproductive viability, making viability low (Kahn et al., 2020). Predictive modeling based on GIS and habitat restoration practice are essential to negate the negative impacts of these anthropogenic threats.

Progress in Conservation Practices

Efforts to reduce human harm have made some progress in species revitalization. Dam removal and artificial spawning reefs have been found to be successful in reconnecting habitats and ensuring reproductive success (Hager et al., 2021). Migration corridors, once opened, enhance genetic diversity, and artificial

reefs give eggs and larval development appropriate conditions (Breece et al., 2018). The combination of restoration efforts with predictive models can optimize conservation resources, enhancing the likelihood of sturgeon recovery.

Advances in Modeling Habitat Suitability

Statistical modeling, including machine learning and logistic regression, have been found to predict habitat suitability and guide conservation priority (Fox et al., 2019; Nelson et al., 2020). GIS modeling has contributed to more robust environmental monitoring over large geographical expanses, establishing spatial trends in habitat utilization and guiding data-driven conservation (Kocovsky et al., 2018). The integration of machine learning with GIS could lead to more accurate predictions and finer spatial resolution.

Technological Advancements in Telemetry

Environmental and acoustic telemetry have contributed to more accurate habitat suitability modeling, introducing insights into sturgeon response to the environment (Breece et al., 2018; Niklitschek & Secor, 2019). Better prediction by high-resolution data advances conservation planning to guarantee the long-term sustainability of Atlantic sturgeons' populations in the Chesapeake Bay and more generally (Fox et al., 2019; Nelson et al., 2020). Continuing advances in modeling and monitoring will play a key role in alleviating the impacts of climatic change, habitat degradation, and pollution on the endangered species (Kocovsky et al., 2018).

CHAPTER IV

Hypotheses, Experiments, and Data Analysis

Hypotheses & Research Questions

The central hypothesis of this study is that Atlantic sturgeon habitat suitability in the Chesapeake Bay can be accurately predicted using a suite of water quality variables, including temperature, dissolved oxygen, pH, turbidity, nitrate, discharge, station location, and season. A secondary hypothesis suggests that specific combinations of these variables—such as high pH and discharge, or low turbidity and temperature—create optimal or suboptimal conditions for sturgeon occupancy. To explore these hypotheses, the study was guided by four key research questions: (1) Which environmental variables most strongly predict sturgeon habitat suitability? (2) How does habitat suitability vary across stations and seasons? (3) Can logistic regression and its regularized forms provide accurate predictive tools for conservation decision-making?

Experimental Design

To test these questions, daily water quality data were retrieved from five USGS stations, while spatial context was added using NOAA's ESA Critical Habitat geodatabase. Following preprocessing—imputation, transformation, normalization, and cleaning—Habitat suitability was characterized using a rule-based binary classification method: each day's suitability was marked as suitable (1) or unsuitable (0) according to literature-based thresholds for key environmental factors. Such thresholds are the documented physiological bounds and behavioral optima for Atlantic sturgeon at spawning migration and juvenile growth stages. A range of

sources from NOAA ESA recommendations, published tolerance research (e.g., Niklitschek & Secor, 2019; Hilton et al., 2016) through Chesapeake-specific observations (Balazik et al., 2012; Peterson et al., 2018) is used.

Table 1

Environmental Thresholds Used to Define Binary Habitat Suitability Classification

Environmental Thresholds for Habitat Suitability Classification		
Variable	**Threshold**	**Justification**
Temperature (°C)	13–28	Optimal spawning and migration range (Hilton et al., 2016; NOAA, 2022)
Dissolved Oxygen	≥ 5	Minimum required to avoid hypoxia stress (Niklitschek & Secor, 2019)
pH	6.5–8.5	Ecologically neutral to slightly basic preferred (Peterson et al., 2018)
Turbidity (NTU)	0–50	High turbidity impairs feeding and respiration (Fox et al., 2019)
Nitrate (mg/L)	0–1.5	Excess nitrate signals runoff; too high indicates pollution (EPA, 2020)
Discharge (cfs)	200–5000	Spring freshets enable upstream migration (Balazik et al., 2012)

Note. Values reflect established physiological requirements and behavioral preferences of Atlantic sturgeon during spawning and migration, based on NOAA guidance and peer-reviewed literature. Only days where all variables fell within the specified thresholds were classified as suitable.

A series of logistic regression models were built: a Full Model including all predictors; a Reduced Model excluding collinear or weakly statistical variables; an AIC-based Stepwise Model optimized for parsimony; and two regularization techniques—Ridge and Lasso regression. Model performance was assessed using AUC and accuracy metrics, with generalizability tested via 10-fold cross-validation.

Key Results

The Full Model achieved the highest AUC (0.899) and maintained strong overall accuracy (0.813). As shown in the ROC curve comparison illustrates that the Full and Stepwise Models are substantially better than Reduced and Ridge Models in classification capacity. Lasso Regression performs just as well.

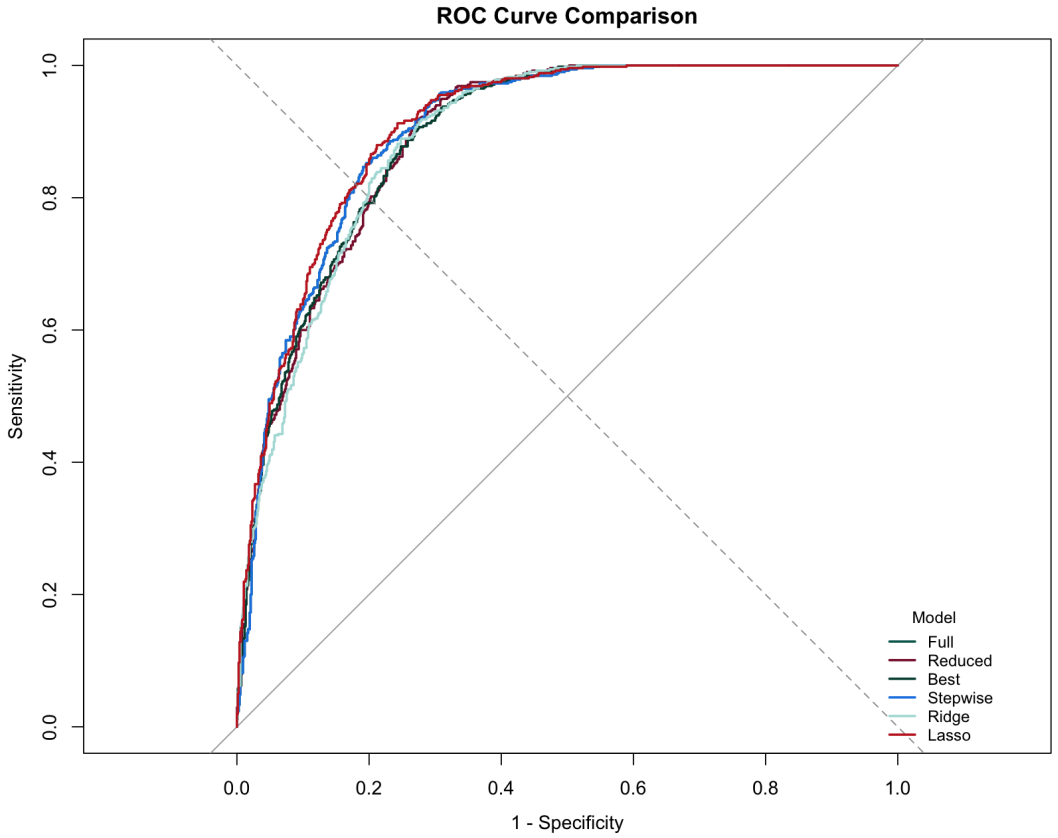


Figure 5. ROC Curve Comparison Illustrating Model Performance in Classifying Suitable vs. Unsuitable Habitat Days.

The Stepwise Model replicated the Full Model's performance using fewer predictors, reflecting more parsimonious specification. Lasso Regression also performed well (AUC = 0.897, Accuracy = 0.801), making it a highly appealing choice for variable reduction with minimal loss of predictive capacity. Both accuracy and AUC are plotted in the bar graph below for side-by-side comparison of the

performance measures across the models.

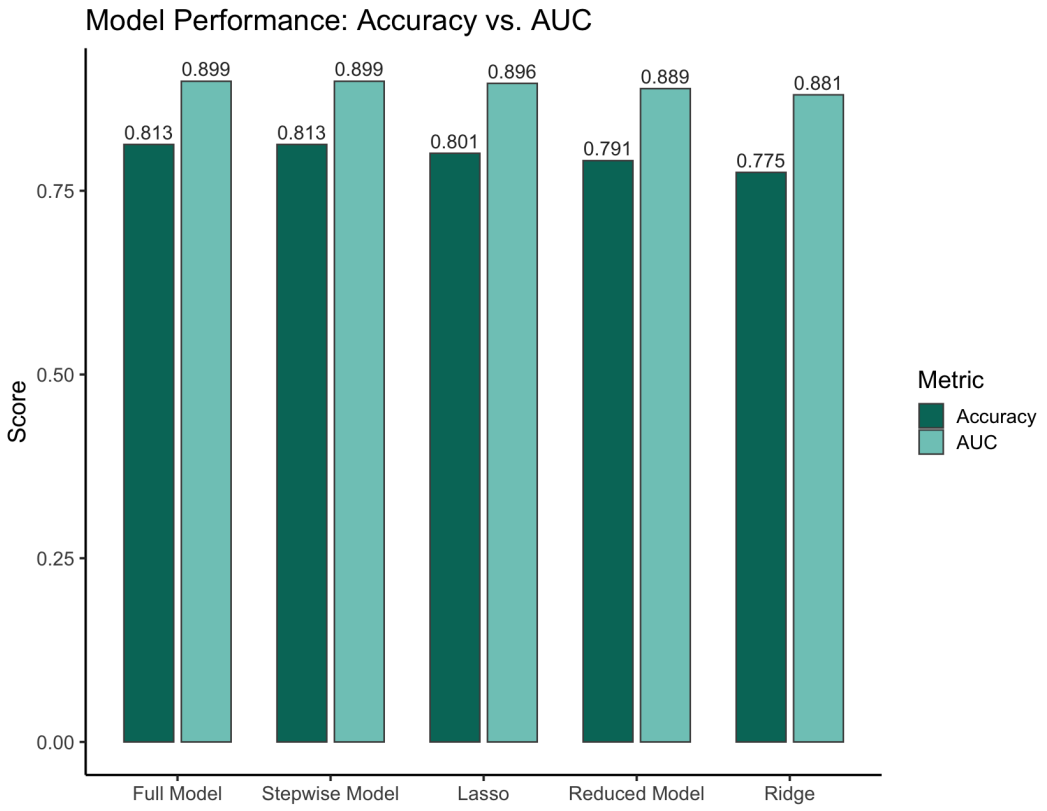


Figure 6. Accuracy and AUC Values per Model.

In contrast, the Reduced Model and Ridge Regression did not perform as well, showing the importance of retaining important predictors. The most important predictors across models were pH, discharge, and nitrate (all positively associated with suitability), and turbidity and temperature (negatively associated).

Discussion of Findings

These findings are consistent with ecological theory: sturgeon survival is highly influenced by dissolved oxygen and pH, and spawning is negatively impacted by turbidity. The success of Lasso modeling here promotes the application of regularized regression in conservation workflows. Temporal aggregation by month enabled visualization of habitat change through time, informing adaptive habitat management plans.

Model coefficients confirm that specific conductance, dissolved oxygen, and temperature are critical habitat indicators. Temperature was strongly significant and negative ($p < 0.001$), which is consistent with known spawning thresholds of ~13–26 °C. Dissolved oxygen had a positive effect ($p < 0.001$), confirming sturgeon preference for high oxygen levels in early life stages. Specific conductance was negatively significant ($p < 0.001$), confirming that Atlantic sturgeon bypass saline or brackish water in spawning.

These distributions correspond to literature-derived environmental preferences of Atlantic sturgeon and model coefficient directions. pH was also a weaker but still significant predictor ($p \approx 0.1$), with optimal habitat generally within ~6.5–8.2 pH. Turbidity and nitrate had complex relationships—moderate turbidity often coincided with spring freshets and increased habitat suitability, while nitrate, although positively associated, is more likely a proxy for seasonal runoff timing rather than a preferred condition.

To further assess model robustness, precision, recall, and F1 scores were calculated for each logistic regression model.

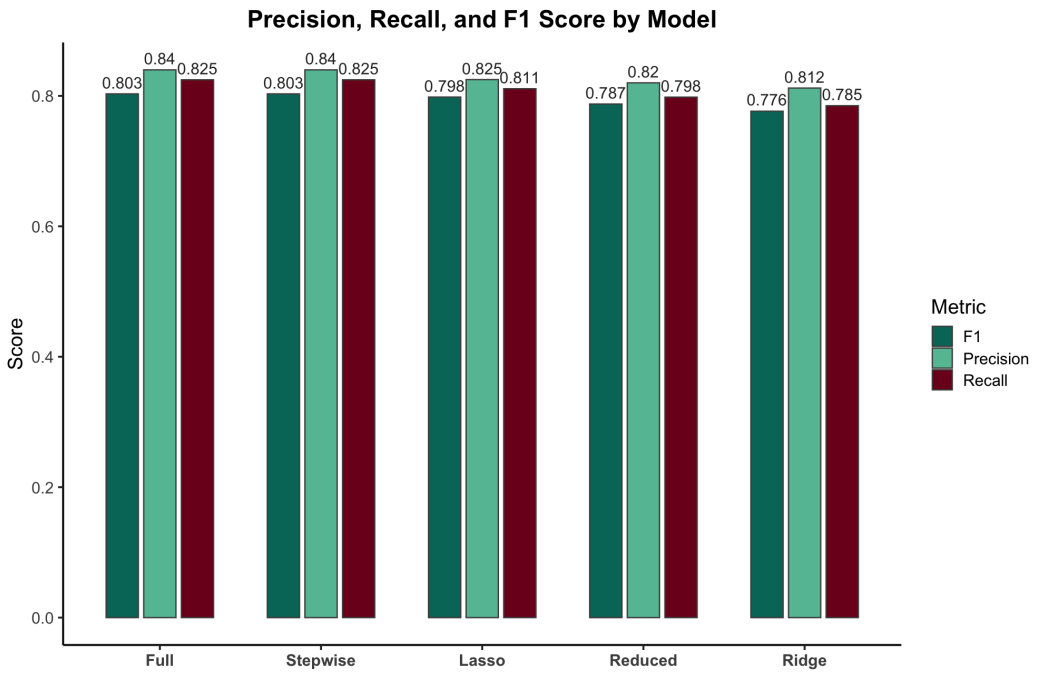


Figure 7. Precision, Recall, and F1 Scores by Model.

The Full and Stepwise models not only generated improved AUC performance but also had improved balance of sensitivity and specificity, rendering them suitable for seasonal forecasting.

Seasonal Suitability Patterns Seasonal model predictions were in accordance with biological expectations. Spring yielded the greatest percentage of suitable habitat at nearly all stations, which is consistent with top spawning migrations. USGS_02035000, for instance, had 63% spring suitability. Autumn subsequently registered reduced but still elevated suitability (50–68%), which would be expected with fall-spawning cohorts in rivers like the James. Summer registered 30–50% suitability, which would indicate increased thermal and hypoxic stress, and winter

registered consistently 0% suitability as temperatures fell below physiological levels for sturgeon activity. Predicted habitat suitability by season is shown in Figures 7 to 11. Space-time trends replicate biologically rational trends—high spring and fall suitability, low summer occurrence due to heat and hypoxic stress, and all but elimination during winter.

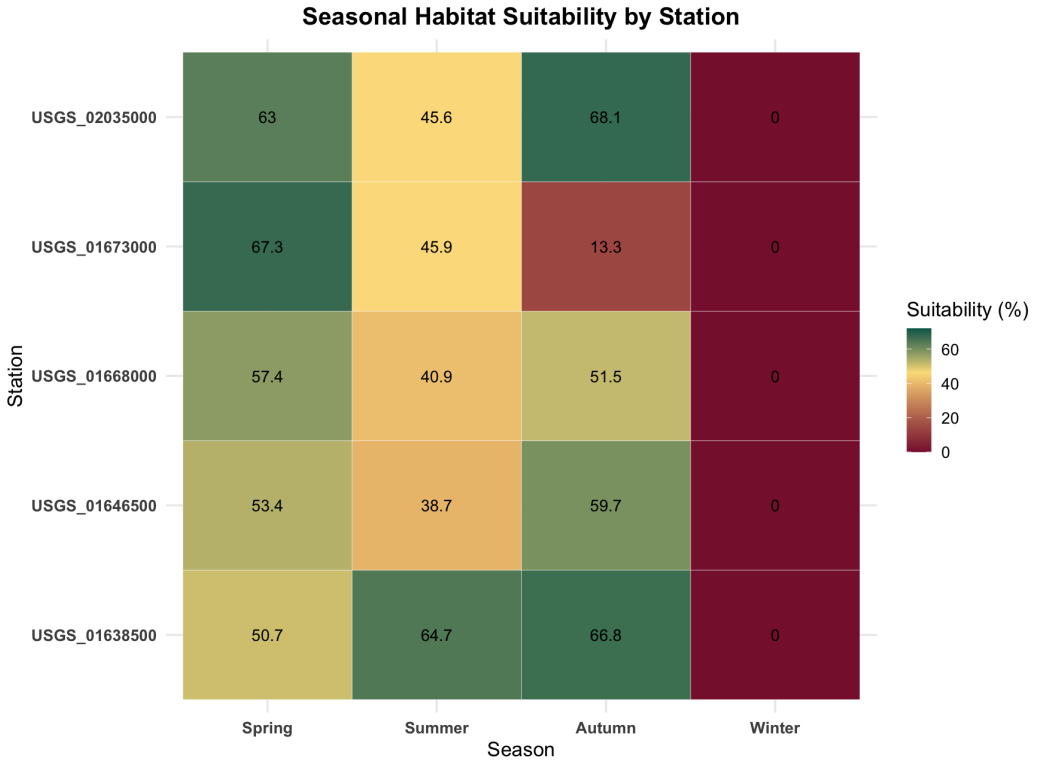


Figure 8. Seasonal Habitat Suitability Percentages Across USGS Monitoring Stations.

Warmer seasons (spring and autumn) show the highest suitability across most stations, particularly USGS_02035000 and USGS_01673000. Winter exhibits 0% suitability at all sites, aligning with known sturgeon inactivity during cold periods. This heatmap illustrates temporal and spatial variation in suitable habitat availability for Atlantic sturgeon.

Spatially, upstream stations were >70% suitable in spring, suggesting their role as first spawning grounds. Mid-river stations showed double-season utilization (spring and fall), while downstream stations only showed suitability during wet springs. These patterns suggest that habitat availability becomes limited and expanded with seasonal hydrology, temperature, and salinity gradients—well described through both statistical and spatial models.

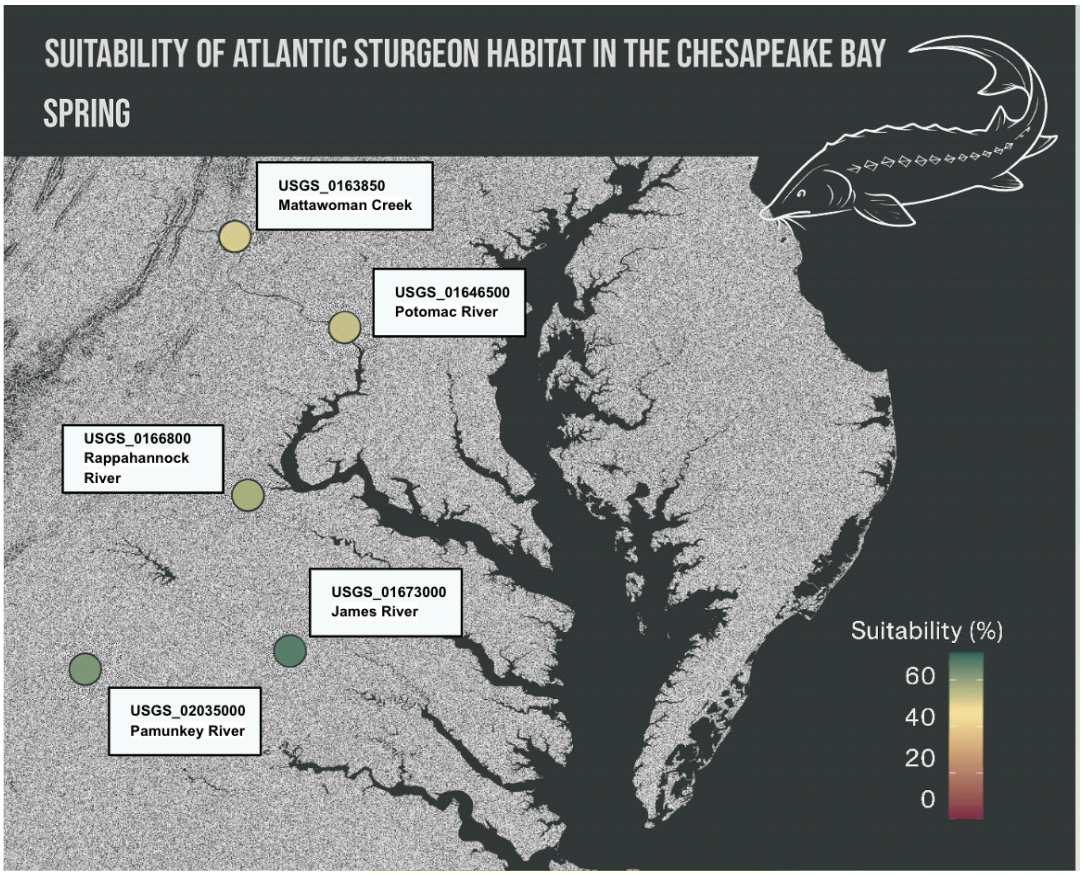


Figure 9. Predicted Atlantic Sturgeon Habitat Suitability – Spring

Highest suitability is concentrated in upstream tributaries, supporting springtime spawning patterns.

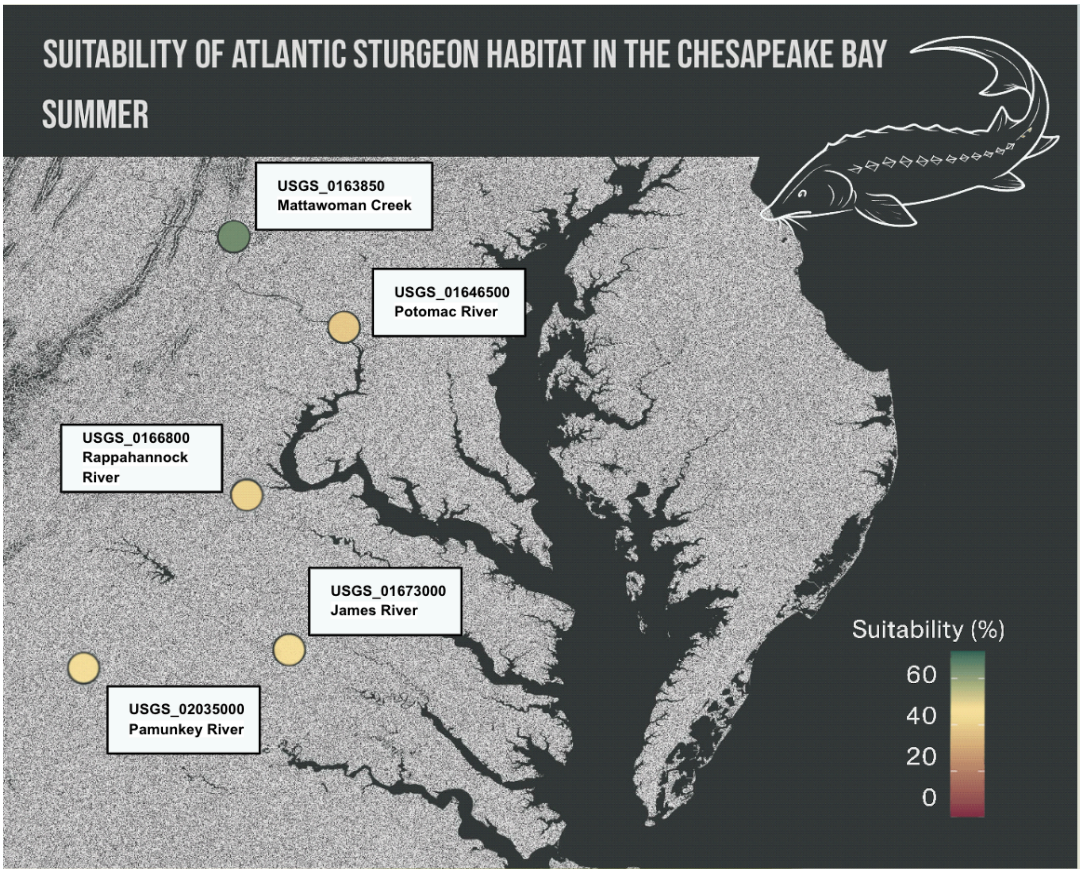


Figure 10. Predicted Atlantic Sturgeon Habitat Suitability – Summer

Habitat suitability drops across all stations due to elevated temperature and low DO—consistent with stress-period conditions.

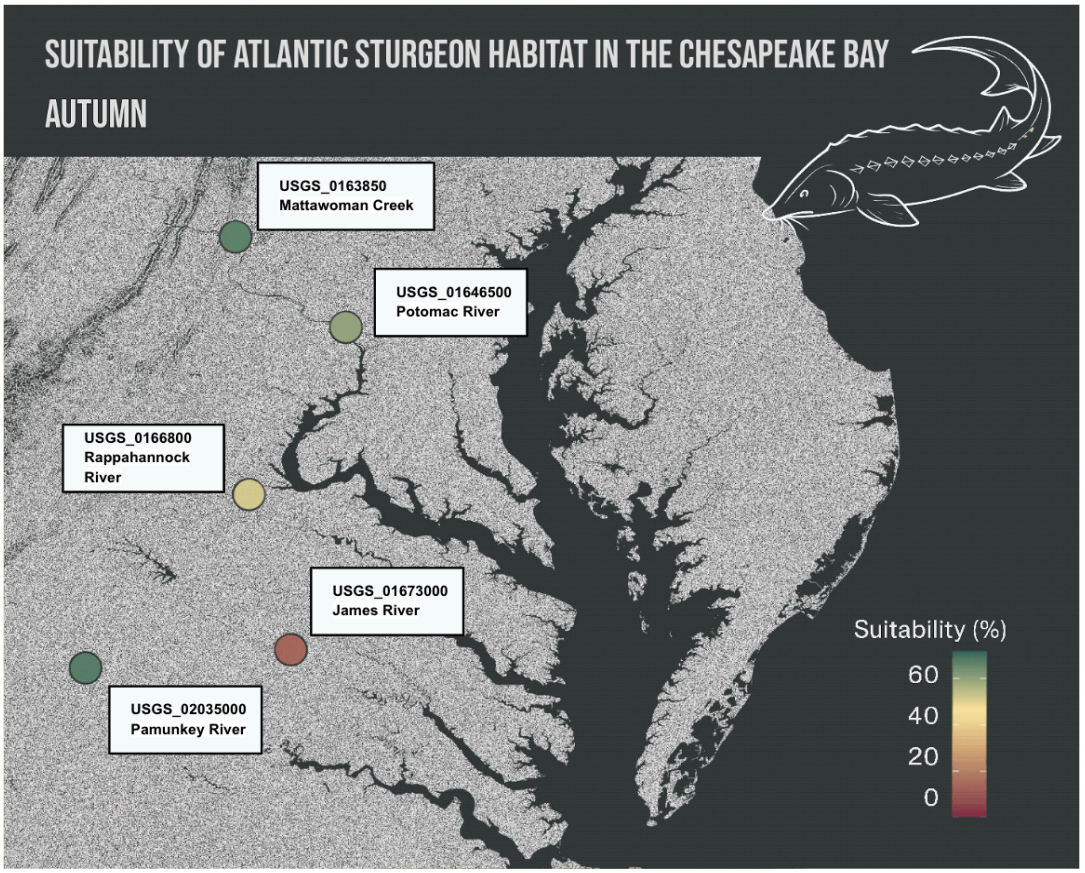


Figure 11. Predicted Atlantic Sturgeon Habitat Suitability – Autumn

Autumn suitability remains elevated, especially in known fall-spawning rivers like the James.

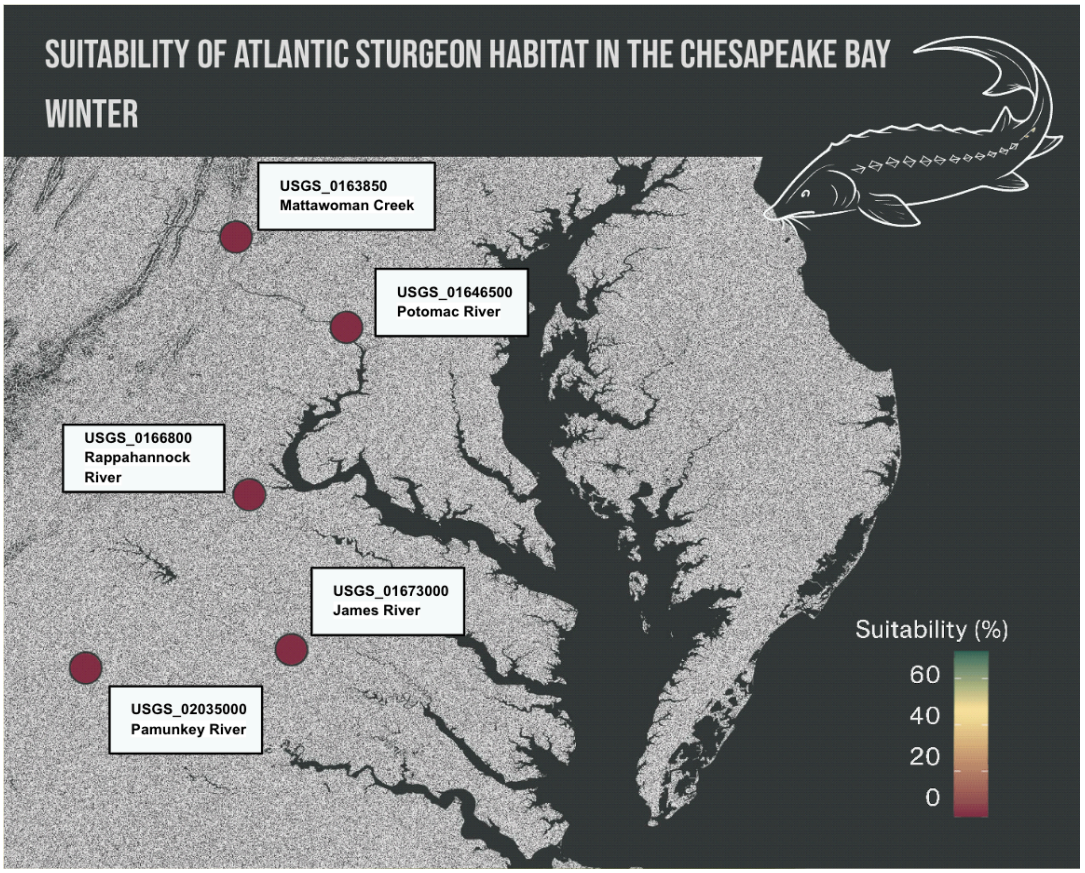


Figure 12. Predicted Atlantic Sturgeon Habitat Suitability – Winter

No suitability is predicted due to physiological limitations from cold temperatures—validating the model's seasonal sensitivity.

CHAPTER V

Threats to the Validity of the Study

Despite statistical robustness and ecological consistency, several limitations need to be mentioned that can affect the internal and external validity of the study:

Binary Classification of Habitat Suitability

Habitat suitability was classed in binary terms (i.e., suitable or unsuitable) according to environmental thresholds drawn from the literature. While this simplified modeling and visualization, it may not catch the fine, continuous level of habitat quality. In reality, suitability likely occurs on a continuum, and the binary cutoff may suppress fine locational or seasonal variation.

Limited Spatial Representation

The study utilized only five USGS monitoring stations. While chosen based on their continuity and spatial coverage, the sites will probably omit some of the Chesapeake Bay tributary's spatial heterogeneity. Certain microhabitats, like spawning grounds or backwater channels, are perhaps omitted and may result in spatial predictions with bias.

Imputation and Missing Data

The original USGS water quality dataset contained missing values, particularly in turbidity and discharge. To address this, K-Nearest Neighbors (KNN) imputation was initially performed on the base dataset using only the available water quality variables (e.g., temperature, pH, DO, turbidity, nitrate, and discharge). This step was done prior to adding additional contextual fields such as station name,

latitude/longitude, and season classification to avoid introducing collinearity or data leakage into the imputation process.

After imputation, the dataset was enhanced with manually added spatial and temporal context. Each row was assigned its station label, geographic coordinates, and corresponding season based on date. These variables were incorporated into the modeling phase but were not part of the imputation to preserve data integrity and reproducibility.

This approach ensured that missing values were estimated using only biologically meaningful and directly comparable parameters while allowing more precise spatial and temporal analysis after preprocessing.

Multicollinearity and Predictor Overlap

Despite efforts towards reducing multicollinearity via VIF checking and regularization, variables like dissolved oxygen and temperature are statistically and biologically correlated. This overlap can suppress the distinct contributions of one single predictor and mask or inflate certain coefficient estimates.

Absence of Certain Ecological Drivers

Other documented habitat determinants, including salinity stratification, sediment type, physical barriers (e.g., dams), and anthropogenic disturbance, were not included because information was lacking. Their omission limits the generality of the model, particularly to account for behavioral avoidance or site fidelity not controlled by water chemistry alone.

Temporal Resolution and Climatic Trends

The record only covered 2021–2025, a relatively narrow temporal window. This period may not fully capture long-term hydrological cycles or the effects of climate change, such as rising water temperature or altering flow regimes. Extrapolation beyond this window must be done with caution.

CHAPTER VI

Future Work

This study is a first step in Atlantic sturgeon habitat suitability modeling using logistic regression and environmental data, yet there are many clear ways to expand:

Incorporation of Salinity & Tidal Fluctuations

While specific conductance served as a surrogate for salinity, direct salinity measurements and tidal fluctuation would increase resolution in the lower estuary. Real-time salinity gradients and tidal levels may improve predictions in transition zones between freshwater and brackish stretches.

Telemetry-Based Validation

Future studies should incorporate telemetry data from tagged sturgeon to validate model predictions against actual fish presence. This would allow for refining the binary classification threshold and validating seasonal trends with observed migratory behavior.

Machine Learning Models

Expanding the modeling framework to include ensemble methods like Random Forest, Gradient Boosting, or XGBoost may capture non-linear relationships more effectively. These approaches, when paired with regularization or SHAP analysis, can still offer ecological interpretability.

Real-Time Habitat Monitoring Tool

Developing an online, interactive dashboard that combines updated USGS

water quality data with real-time model predictions can facilitate proactive conservation management. The system can be set to send alerts if the conditions fall below acceptable levels, especially during spawning season.

Spatial Interpolation and Finer Resolution Mapping

If more spatial data points were available, interpolation methods such as kriging or raster-based regression would be capable of generating finer-scale suitability surfaces for the entire river system. This would allow pinpoint restoration of habitats by targeting reaches or degraded sections.

Stakeholder Collaboration

Partnership with NOAA, USGS, state agencies, and conservation NGOs is necessary to refine the model for operational purposes. Aligning outputs with management requirements—i.e., establishing restoration targets, dam removal planning, or scheduling dredging closures—would increase conservation impact.

CHAPTER VII

Reflections

What I Learned

This project presented a unique opportunity to carry out a full data science pipeline on a real ecological problem. I learned how to transform raw observational data into comprehensible models, how to evaluate statistical assumptions critically, and how to balance ecological goals and mathematics. I gained fluency in spatial analysis and a greater appreciation for the constraints and potential that GIS tools provide when paired with statistical models.

What Was Most Rewarding

The most rewarding experience was seeing the results on actual maps in QGIS—seeing theoretical model coefficients become spatial patterns that reflected known sturgeon behaviors felt like closing the loop between ecology and data. It was also gratifying to see how my research could actually help with conservation planning, especially as Atlantic sturgeon are still federally endangered and susceptible to habitat degradation.

What Was Most Challenging

Integrating datasets of different spatial and temporal resolutions proved to be the most difficult. Merging NOAA shapefiles with USGS time-series datasets required precise attention to projection, resolution, and metadata. Debugging GRASS GIS software, correcting flow accumulation, and optimizing raster alignment often involved hours of trial and error. But learning to do so embedded in me the

importance of writing down every technical step and knowing every data transformation applied to the data.

Advice for Future Students

Start earlier than you think you need to—particularly with data cleaning, exploratory analysis, and learning unfamiliar software. Build your project iteratively, and don't be afraid to rerun models or revise visualizations. Your figures will change your interpretation, and your interpretation will shape the story you tell. Most importantly, make your work as reproducible and readable as possible. You never know who will use your code or maps one day—perhaps someone trying to save a species.

REFERENCES

Balazik, M. T., Garman, G. C., Webb, M. A. H., & Richards, R. A. (2012). Detection of late-stage Atlantic sturgeon spawning in the James River, Virginia. *North American Journal of Fisheries Management, 32*(1), 162–166. https://doi.org/10.1080/02755947.2012.661391

Breece, M. W., Fox, D. A., Haulsee, D. E., Miller, D. C., & Oliver, M. J. (2018). Environmental drivers of adult Atlantic sturgeon movement and residency in the Delaware Bay. *Marine and Coastal Fisheries, 10*(5), 1–15. https://doi.org/10.1002/mcf2.10007

Fox, D. A., Haulsee, D. E., Breece, M. W., & Oliver, M. J. (2019). Predicting Atlantic sturgeon occurrence and habitat suitability using a machine learning approach. *Fisheries Oceanography, 28*(6), 633–648. https://doi.org/10.1111/fog.12445

Hager, C., Kahn, J., Fisher, M., & Secor, D. H. (2021). Dam removals and habitat restoration for endangered Atlantic sturgeon: A review of benefits and challenges. *Environmental Management, 67*(2), 249–263. https://doi.org/10.1007/s00267-020-01382-4

Hilton, E. J., Kynard, B., Balazik, M. T., Horodysky, A. Z., & Duffey, R. (2016). Review of Atlantic sturgeon *Acipenser oxyrinchus oxyrinchus* spawning rivers, habitat, and migration routes in the United States. *Journal of Fish Biology, 89*(5), 2481–2505. https://doi.org/10.1111/jfb.13138

International Union for Conservation of Nature (IUCN). (2022). *Acipenser oxyrinchus*. The IUCN Red List of Threatened Species. https://www.iucnredlist.org

Kahn, J., Fisher, M., Balazik, M. T., & Secor, D. H. (2020). Effects of heavy metal contamination on reproductive success in Atlantic sturgeon. *Environmental Toxicology and Chemistry, 39*(4), 812–822. https://doi.org/10.1002/etc.4670

Kemp, W. M., Testa, J. M., Conley, D. J., Gilbert, D., & Hagy, J. D. (2019). Declining oxygen in the global ocean and coastal waters. *Science, 359*(6371), eaam7240. https://doi.org/10.1126/science.aam7240

Kocovsky, P. M., Stapanian, M. A., & Adams, J. V. (2018). GIS-based habitat modeling for Atlantic sturgeon conservation and management. *Transactions of the American Fisheries Society, 147*(1), 67–81. https://doi.org/10.1002/tafs.10057

Kynard, B., & Horgan, M. (2002). Ontogenetic behavior and migration of Atlantic sturgeon, *Acipenser oxyrinchus oxyrinchus*, and shortnose sturgeon, *Acipenser brevirostrum*, with notes on social behavior. *Environmental Biology of Fishes, 63*(2), 137–150. https://doi.org/10.1023/A:1014233912332

Moser, M. L., Hightower, J. E., & Bain, M. B. (2000). Hypoxia effects on sturgeon behavior and distribution in Atlantic coastal systems. *Canadian Journal of Fisheries and Aquatic Sciences, 57*(7), 1342–1352. https://doi.org/10.1139/f00-077

Nelson, G. A., Chase, B. C., & Stockwell, J. D. (2020). Modeling Atlantic sturgeon distribution and habitat selection in coastal and estuarine waters. *Marine Ecology Progress Series, 639*, 187–202. https://doi.org/10.3354/meps13293

Niklitschek, E. J., & Secor, D. H. (2019). Modeling hypoxia effects on Atlantic sturgeon: Implications for habitat suitability in Chesapeake Bay. *Estuaries and Coasts, 42*(6), 1604–1619. https://doi.org/10.1007/s12237-019-00589-7

NOAA Fisheries. (2022). *Atlantic sturgeon (Acipenser oxyrinchus oxyrinchus): Species profile*. https://www.fisheries.noaa.gov/species/atlantic-sturgeon

Pinder, A. C., & Hatin, D. (2010). The conservation biology of the Atlantic sturgeon: Understanding early life history is key to recovery. *Fish and Fisheries, 11*(1), 1–13. https://doi.org/10.1111/j.1467-2979.2009.00336.x

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 58*(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67*(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

APPENDIX A

Full R Source Code and Output

1. Load Required Libraries

```r
# Data manipulation
library(dplyr)
library(tidyr)
library(readr)

# Visualization
library(ggplot2)
library(corrplot)
library(GGally)
library(gridExtra)

# Statistical modeling
library(caret)
library(glmnet)
library(car)
library(pROC)

# Diagnostics
library(e1071)
```

2. Load and Prepare Dataset

```r
# Load dataset
data <- read_csv("water_quality_full.csv")

# Preview the first few rows
head(data, 10)
```

```
    # A tibble: 10 × 25
       date       season temp_mean temp_min temp_max spec_max
spec_min spec_mean
       <date>     <chr>      <dbl>    <dbl>    <dbl>    <dbl>
<dbl>     <dbl>
     1 2021-06-01 summer      20.5     18.6     22.2      367
348       353
     2 2021-06-02 summer      21.2     20       22.3      371
344       359
     3 2021-06-03 summer      21.6     20.7     22.6      344
291       329
     4 2021-06-04 summer      22.5     20.6     24.6      330
281       311
     5 2021-06-05 summer      23.8     21.2     26.3      340
330       336
     6 2021-06-06 summer      25.6     23.1     28        343
328       337
     7 2021-06-07 summer      26.5     24.9     27.9      336
329       333
     8 2021-06-08 summer      27.1     25.7     28.6      338
333       335
     9 2021-06-09 summer      26.7     25.9     27.9      334
```

```
327       331
    10 2021-06-10 summer      26.9     25.4     28.2      329
324       326
    # i 17 more variables: ph_max <dbl>, ph_min <dbl>, ph_mean
<dbl>,
    #   dissolved_max <dbl>, dissolved_min <dbl>, dissolved_mean
<dbl>,
    #   turbidity_max <dbl>, turbidity_min <dbl>, turbidity_mean
<dbl>,
    #   nitrate_max <dbl>, nitrate_min <dbl>, nitrate_mean <dbl>,
discharge <dbl>,
    #   station <chr>, habitat_suitability <dbl>, latitude <dbl>,
longitude <dbl>
```

```r
# Calculate % suitable by station and season
percentage_suitable <- data %>%
  group_by(station, season) %>%
  summarise(
    suitable_count = sum(habitat_suitability == 1, na.rm = TRUE),
    unsuitable_count = sum(habitat_suitability == 0, na.rm = TRUE),
    total_count = n(),
    percentage_suitable = (suitable_count / total_count) * 100
  )

print(percentage_suitable)
```

```
    # A tibble: 20 × 6
    # Groups:   station [5]
       station       season suitable_count unsuitable_count
total_count
       <chr>         <chr>           <int>             <int>
<int>
     1 USGS_01638500 autumn            238               118
356
     2 USGS_01638500 spring            139               135
274
     3 USGS_01638500 summer            233               127
360
     4 USGS_01638500 winter              0               281
281
     5 USGS_01646500 autumn            216               146
362
     6 USGS_01646500 spring            143               125
268
     7 USGS_01646500 summer            139               220
359
     8 USGS_01646500 winter              0               301
301
     9 USGS_01668000 autumn            185               174
359
    10 USGS_01668000 spring            156               116
272
    11 USGS_01668000 summer            149               215
364
```

```
    12 USGS_01668000 winter               0               300
300
    13 USGS_01673000 autumn              47               307
354
    14 USGS_01673000 spring             185                90
275
    15 USGS_01673000 summer             169               199
368
    16 USGS_01673000 winter               0               302
302
    17 USGS_02035000 autumn             241               113
354
    18 USGS_02035000 spring             170               100
270
    19 USGS_02035000 summer             167               199
366
    20 USGS_02035000 winter               0               285
285
    # ℹ 1 more variable: percentage_suitable <dbl>
```

```
# Convert to factor
data$station <- as.factor(data$station)
data$season <- as.factor(data$season)
data$habitat_suitability <- factor(data$habitat_suitability, levels
= c(0, 1),
                                   labels = c("No", "Yes"))

# Remove any remaining NAs
data <- na.omit(data)

# Structure check
str(data)
```

```
    tibble [6,430 × 25] (S3: tbl_df/tbl/data.frame)
     $ date               : Date[1:6430], format: "2021-06-01"
"2021-06-02" ...
     $ season             : Factor w/ 4 levels "autumn","spring",..:
3 3 3 3 3 3 3 3 3 3 ...
     $ temp_mean          : num [1:6430] 20.5 21.2 21.6 22.5 23.8
25.6 26.5 27.1 26.7 26.9 ...
     $ temp_min           : num [1:6430] 18.6 20 20.7 20.6 21.2 23.1
24.9 25.7 25.9 25.4 ...
     $ temp_max           : num [1:6430] 22.2 22.3 22.6 24.6 26.3 28
27.9 28.6 27.9 28.2 ...
     $ spec_max           : num [1:6430] 367 371 344 330 340 343 336
338 334 329 ...
     $ spec_min           : num [1:6430] 348 344 291 281 330 328 329
333 327 324 ...
     $ spec_mean          : num [1:6430] 353 359 329 311 336 337 333
335 331 326 ...
     $ ph_max             : num [1:6430] 8.4 8.5 8.3 8.4 8.4 8.4 8.3
8.3 8.2 8.2 ...
     $ ph_min             : num [1:6430] 7.9 7.9 7.8 7.7 7.8 7.8 7.8
7.8 7.8 7.8 ...
```

```
     $ ph_mean            : num [1:6430] 8.15 8.2 8.05 8.05 8.1 8.1
8.05 8.05 8 8 ...
     $ dissolved_max      : num [1:6430] 10.6 10.6 9.7 10 10.2 10
9.6 9.3 8.8 8.8 ...
     $ dissolved_min      : num [1:6430] 8 7.7 7.5 7 7.2 6.8 6.3 6 6
6.2 ...
     $ dissolved_mean     : num [1:6430] 9.1 9 8.5 8.3 8.5 8.1 7.8
7.6 7.4 7.3 ...
     $ turbidity_max      : num [1:6430] 12 12.9 70.5 95.3 16.9 11.2
21.2 12.4 12.4 14.3 ...
     $ turbidity_min      : num [1:6430] 5 4.9 4.6 10.4 6.2 5 4.6
4.8 4.5 4.9 ...
     $ turbidity_mean     : num [1:6430] 6.8 7 9.7 31.7 8.9 7.2 7.4
6.7 6.5 7 ...
     $ nitrate_max        : num [1:6430] 0.57 0.57 0.448 0.434 0.328
0.542 0.352 0.638 0.638 0.436 ...
     $ nitrate_min        : num [1:6430] 0.508 0.508 0.402 0.38
0.266 0.49 0.292 0.566 0.566 0.362 ...
     $ nitrate_mean       : num [1:6430] 0.539 0.539 0.425 0.407
0.297 0.517 0.323 0.603 0.603 0.4 ...
     $ discharge          : num [1:6430] 461 461 461 475 391 ...
     $ station            : Factor w/ 5 levels "USGS_01638500",..: 1
1 1 1 1 1 1 1 1 ...
     $ habitat_suitability: Factor w/ 2 levels "No","Yes": 2 2 2 2 2
2 2 2 2 2 ...
     $ latitude           : num [1:6430] 39.3 39.3 39.3 39.3 39.3
...
     $ longitude          : num [1:6430] -77.5 -77.5 -77.5 -77.5
-77.5 ...
```

```r
cat("Missing values remaining in dataset:", sum(is.na(data)), "\n")
```

```
    Missing values remaining in dataset: 0
```

3. Preprocessing: Z-Score, Winsorization, Log Transformation

3.1 Identify Outliers Using Z-Scores

```r
# Z-score function
calculate_zscore <- function(x) {
  (x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE)
}

# Apply to numeric columns only (not date or coordinates)
z_scores <- data %>%
  mutate(across(where(is.numeric) & !any_of(c("latitude",
"longitude", "date")),
                calculate_zscore))

# Count outliers
outlier_counts <- z_scores %>%
```

```
  summarise(across(where(is.numeric) & !any_of(c("latitude",
"longitude",
                                                 "date")),
                   ~ sum(abs(.) > 3, na.rm = TRUE)))

print(outlier_counts)
```

```
    # A tibble: 1 × 19
      temp_mean temp_min temp_max spec_max spec_min spec_mean ph_max
ph_min ph_mean
          <int>    <int>    <int>    <int>    <int>     <int>  <int>
<int>   <int>
    1         0        0        0        8        1         5      4
7       5
    # ℹ 10 more variables: dissolved_max <int>, dissolved_min <int>,
    #   dissolved_mean <int>, turbidity_max <int>, turbidity_min
<int>,
    #   turbidity_mean <int>, nitrate_max <int>, nitrate_min <int>,
    #   nitrate_mean <int>, discharge <int>
```

3.2 Winsorization to Reduce Extreme Outliers

```
winsorize <- function(x, lower = 0.02, upper = 0.98) {
  qnt <- quantile(x, probs = c(lower, upper), na.rm = TRUE)
  x[x < qnt[1]] <- qnt[1]
  x[x > qnt[2]] <- qnt[2]
  return(x)
}

data <- data %>%
  mutate(across(where(is.numeric) & !any_of(c("latitude",
"longitude", "date")),
                winsorize))
```

3.3 Log Transformation to Address Skew

```
# Shift turbidity_min if  negative values
if (min(data$turbidity_min, na.rm = TRUE) < 0) {
  shift_value <- abs(min(data$turbidity_min, na.rm = TRUE)) + 1
  data <- data %>%
    mutate(turbidity_min = log1p(turbidity_min + shift_value))
}

# Apply log1p to skewed and use turbidity example

data <- data %>%
  mutate(across(c(turbidity_mean, turbidity_max, turbidity_min,
nitrate_mean,
                  discharge), log1p))
```

3.4 Recalculate Z-Scores After Preprocessing

```
# Recalculate Z-scores for post-processing verification
z_scores_after <- data %>%
  mutate(across(where(is.numeric) & !any_of(c("latitude",
"longitude", "date")),
                calculate_zscore))

# Count outliers again
outlier_counts_after <- z_scores_after %>%
  summarise(across(where(is.numeric) & !any_of(c("latitude",
"longitude",
                                                 "date")),
                   ~ sum(abs(.) > 3, na.rm = TRUE)))

cat("Outlier counts before vs after preprocessing:\n")
```

```
    Outlier counts before vs after preprocessing:
```

```
outlier_comparison <- bind_rows(
  "Before" = outlier_counts,
  "After" = outlier_counts_after,
  .id = "Stage"
)
print(outlier_comparison)
```

```
    # A tibble: 2 × 20
      Stage  temp_mean temp_min temp_max spec_max spec_min spec_mean
ph_max ph_min
      <chr>      <int>    <int>    <int>    <int>    <int>     <int>
<int>  <int>
    1 Before         0        0        0        8        1         5
4      7
    2 After          0        0        0        0        0         0
0      0
    # ℹ 11 more variables: ph_mean <int>, dissolved_max <int>,
dissolved_min <int>,
    #   dissolved_mean <int>, turbidity_max <int>, turbidity_min
<int>,
    #   turbidity_mean <int>, nitrate_max <int>, nitrate_min <int>,
    #   nitrate_mean <int>, discharge <int>
```

3.5 Check Skewness Post-Processing

```
# Calculate skewness values after log/Winsorization
skew_values <- data %>%
  summarise(across(where(is.numeric) & !any_of(c("latitude",
"longitude",
                                                 "date")),
                   ~ skewness(., na.rm = TRUE)))

print(skew_values)

    # A tibble: 1 × 19
      temp_mean temp_min temp_max spec_max spec_min spec_mean ph_max
ph_min ph_mean
          <dbl>    <dbl>    <dbl>    <dbl>    <dbl>     <dbl>  <dbl>
<dbl>   <dbl>
    1   -0.0937  -0.0972  -0.0849    0.438    0.491     0.462 -0.173
-0.168  -0.248
    # i 10 more variables: dissolved_max <dbl>, dissolved_min <dbl>,
    #   dissolved_mean <dbl>, turbidity_max <dbl>, turbidity_min
<dbl>,
    #   turbidity_mean <dbl>, nitrate_max <dbl>, nitrate_min <dbl>,
    #   nitrate_mean <dbl>, discharge <dbl>
```

4. Visualizing Z-Score Distributions

4.1 Prepare Data for Plotting

```
# prep z score datasets before and after preprocessing fro plotting
# Variables to exclude from Z-score plots
vars_to_exclude <- c("latitude", "longitude", "date")

# Filter only numeric columns (excluding spatial/date fields)
z_scores_numeric_before <- z_scores %>%
  select(where(is.numeric)) %>%
  select(-any_of(vars_to_exclude))

z_scores_numeric_after <- z_scores_after %>%
  select(where(is.numeric)) %>%
  select(-any_of(vars_to_exclude))

# Reshape to long format for ggplot
data_long_before <- z_scores_numeric_before %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to
= "ZScore")

data_long_after <- z_scores_numeric_after %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to
= "ZScore")
```

```
head(data_long_before)
```

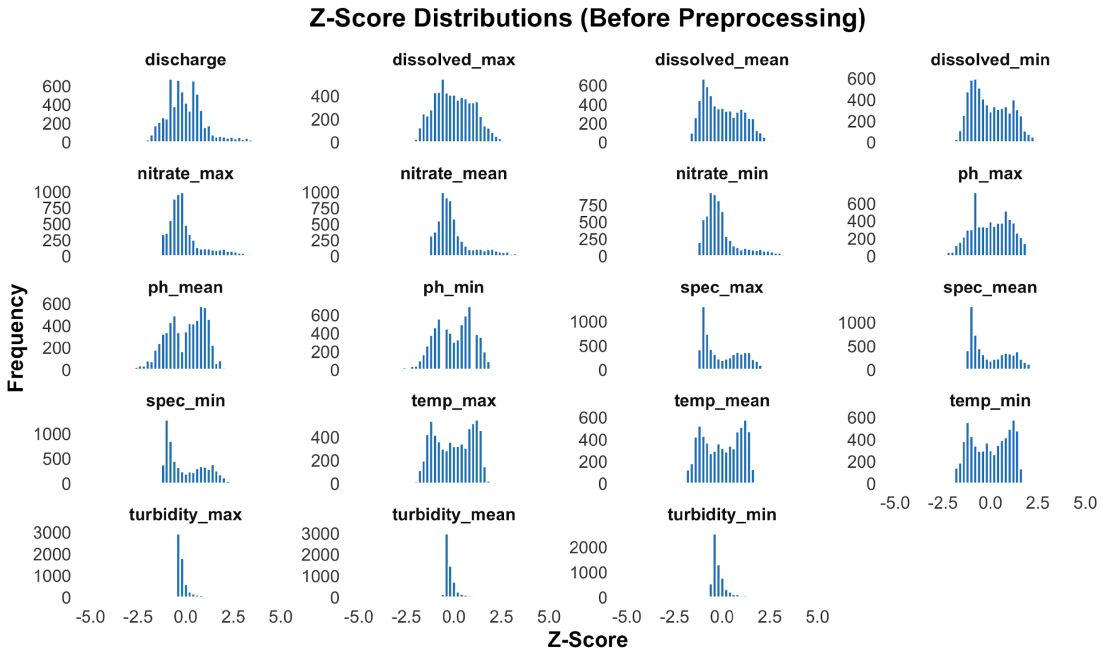
```
    # A tibble: 6 × 2
      Variable  ZScore
      <chr>      <dbl>
    1 temp_mean  0.406
    2 temp_min   0.304
    3 temp_max   0.478
    4 spec_max   1.31
    5 spec_min   1.32
    6 spec_mean  1.28
```

```
head(data_long_after)
```

```
    # A tibble: 6 × 2
      Variable  ZScore
      <chr>      <dbl>
    1 temp_mean  0.408
    2 temp_min   0.305
    3 temp_max   0.480
    4 spec_max   1.34
    5 spec_min   1.33
    6 spec_mean  1.30
```

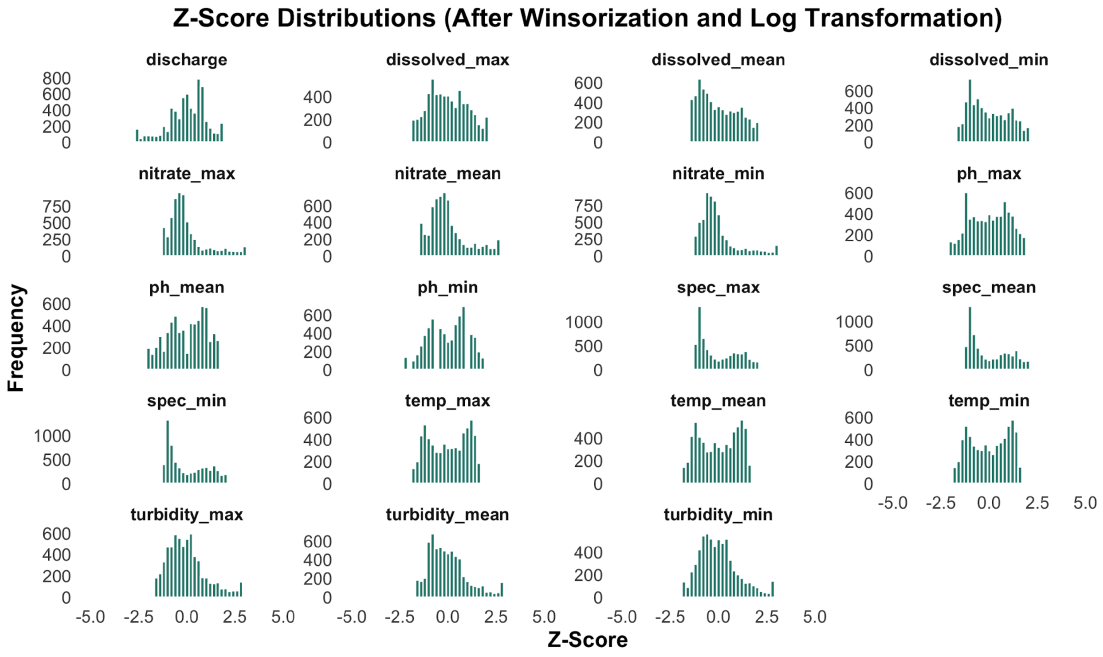
4.2 Plot Z-Score Distribution Before Outlier Handling

```
# Create plot
ggplot(data_long_before, aes(x = ZScore)) +
  geom_histogram(binwidth = 0.2, fill = "#0277BD", color = "white",
                 alpha = 0.9) +
  facet_wrap(~Variable, scales = "free_y", ncol = 4) +
  coord_cartesian(xlim = c(-5, 5)) +
  labs(
    title = "Z-Score Distributions (Before Preprocessing)",
    x = "Z-Score",
    y = "Frequency"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    strip.text = element_text(face = "bold"),
    plot.title = element_text(face = "bold", hjust = 0.5),
    axis.text = element_text(color = "gray30"),
    axis.title = element_text(face = "bold"),
    panel.grid = element_blank()
  )
```



4.3 Plot Z-Score Distribution After Preprocessing

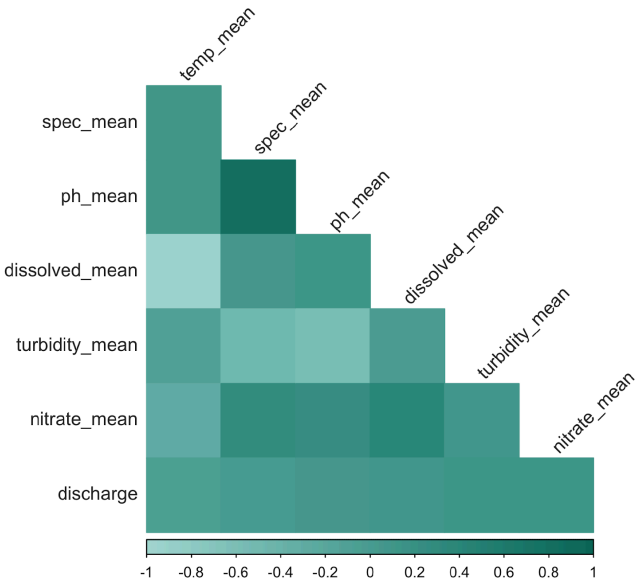
```
ggplot(data_long_after, aes(x = ZScore)) +
  geom_histogram(binwidth = 0.2, fill = "#00796B", color = "white",
                 alpha = 0.9) +
  facet_wrap(~Variable, scales = "free_y", ncol = 4) +
  coord_cartesian(xlim = c(-5, 5)) +
  labs(
    title = "Z-Score Distributions (After Winsorization and Log
Transformation)",
    x = "Z-Score",
    y = "Frequency"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    strip.text = element_text(face = "bold"),
    plot.title = element_text(face = "bold", hjust = 0.5),
    axis.text = element_text(color = "gray30"),
    axis.title = element_text(face = "bold"),
    panel.grid = element_blank()
  )
```



5. Exploratory Data Analysis

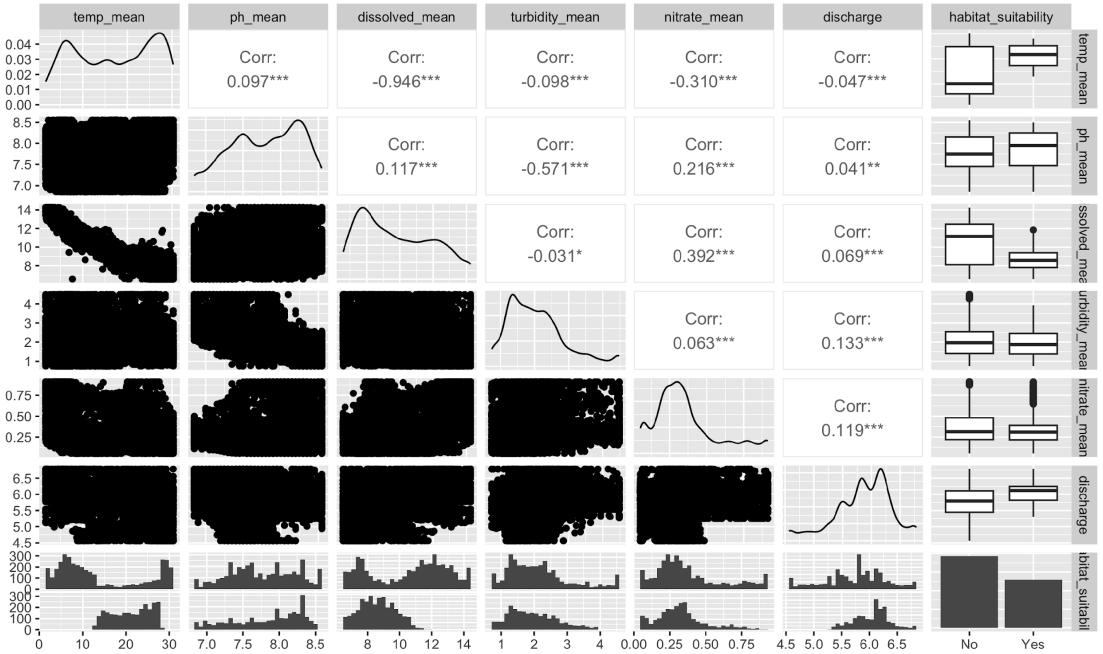
5.1 Correlation Matrix of Environmental Variables

```
# Select key numeric variables for correlation analysis
numeric_vars <- data %>%
  select(temp_mean, spec_mean, ph_mean, dissolved_mean,
         turbidity_mean, nitrate_mean, discharge)

# Compute correlation matrix (excluding missing observations)
corr_matrix <- cor(numeric_vars, use = "complete.obs")

# Visualize correlation matrix using corrplot
corrplot(corr_matrix,
         method = "color",
         type = "lower",
         diag = FALSE,
         tl.col = "gray20",
         tl.srt = 45,
         col = colorRampPalette(c("#B2DFDB", "#00796B"))(200),
         mar = c(0, 0, 2, 0))
```



5.2 Pairwise Scatterplots

```
# Pairwise scatter plots
ggpairs(data[, c("temp_mean", "ph_mean", "dissolved_mean",
                 "turbidity_mean", "nitrate_mean", "discharge",
                 "habitat_suitability")])
```

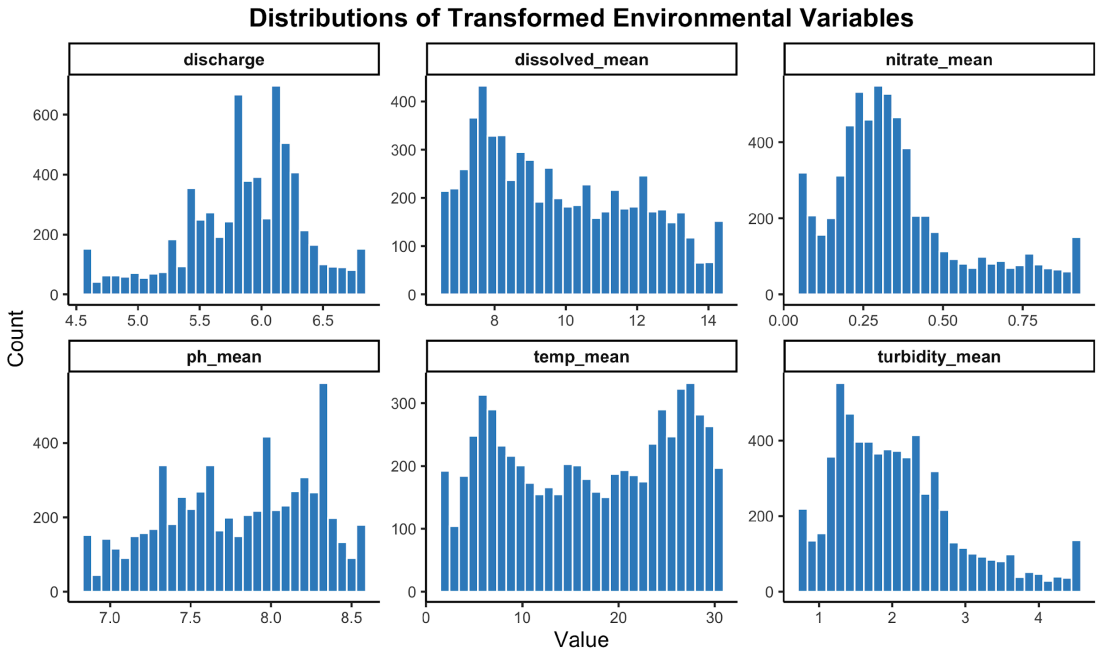


5.3 Prepare Long Format for Distribution Visualizations

```
# Reshape for distribution plots
data_long <- data %>%
  pivot_longer(cols = c(temp_mean, ph_mean, dissolved_mean,
turbidity_mean,
                        nitrate_mean, discharge),
               names_to = "Variable", values_to = "Value")
```

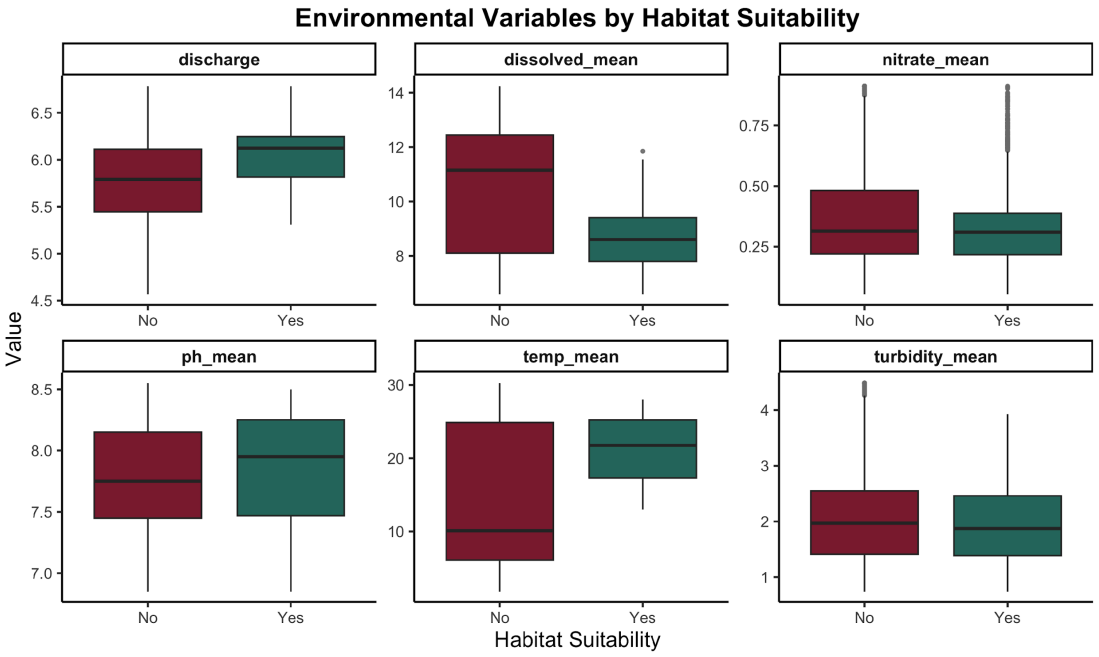
5.4 Histograms of Transformed Environmental Variables

```
# Visualize distribution of each environmental variable after
preprocessing
ggplot(data_long, aes(x = Value)) +
  geom_histogram(bins = 30, fill = "#0277BD", color = "white", alpha
= 0.85) +
  facet_wrap(~Variable, scales = "free", ncol = 3) +
  labs(
    title = "Distributions of Transformed Environmental Variables",
    x = "Value",
    y = "Count"
  ) +
  theme_classic(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    strip.text = element_text(face = "bold"),
    axis.text = element_text(size = 10)
  )
```



5.5 Boxplots by Habitat Suitability

```
# see how each variable differs between suitable and unsuitable
habitat conditions
ggplot(data_long, aes(x = habitat_suitability, y = Value,
                      fill = habitat_suitability)) +
  geom_boxplot(alpha = 0.9, outlier.size = 0.8, outlier.color =
"gray50") +
  scale_fill_manual(values = c("No" = "#800020", "Yes" = "#00695C"))
+
  facet_wrap(~Variable, scales = "free", ncol = 3) +
  labs(
    title = "Environmental Variables by Habitat Suitability",
    x = "Habitat Suitability",
    y = "Value"
  ) +
  theme_classic(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    strip.text = element_text(face = "bold"),
    axis.text = element_text(size = 10),
    legend.position = "none"
  )
```



6. Logistic Regression

6.1 Train-Test Split

```
set.seed(123)
train_index <- createDataPartition(data$habitat_suitability, p =
0.8,
                                   list = FALSE)
train_data <- data[train_index, ]
test_data <- data[-train_index, ]

#skewed variables already log transformed
```

6.2 Fit Logistic Regression Models

```
# stepwise selects based on AIC
# Full model
model_full <- glm(habitat_suitability ~ temp_mean + spec_mean +
ph_mean +
                    dissolved_mean + turbidity_mean + nitrate_mean +
discharge +
                    station + season,
                  data = train_data, family = binomial)

# Reduced model
model_reduced <- glm(habitat_suitability ~ spec_mean + ph_mean +
                       dissolved_mean + turbidity_mean +
nitrate_mean +
                       discharge + station + season,
                     data = train_data, family = binomial)

# Stepwise
stepwise_model <- step(model_full, direction = "both", trace = 0)
```

6.3 Model Summaries and Diagnostics

```
# Print summaries of all models
cat("   # Full Model Summary\n")

      # Full Model Summary

print(summary(model_full))


    Call:
    glm(formula = habitat_suitability ~ temp_mean + spec_mean +
ph_mean +
        dissolved_mean + turbidity_mean + nitrate_mean + discharge +
        station + season, family = binomial, data = train_data)
```

```
    Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
    (Intercept)             8.083450   1.841151   4.390 1.13e-05 ***
    temp_mean              -0.284311   0.025119 -11.319  < 2e-16 ***
    spec_mean              -0.009506   0.001201  -7.918 2.41e-15 ***
    ph_mean                 0.917548   0.216528   4.238 2.26e-05 ***
    dissolved_mean         -2.164617   0.114800 -18.856  < 2e-16 ***
    turbidity_mean         -1.187956   0.082488 -14.402  < 2e-16 ***
    nitrate_mean            2.034054   0.323754   6.283 3.33e-10 ***
    discharge               2.647170   0.109987  24.068  < 2e-16 ***
    stationUSGS_01646500   -0.763071   0.148343  -5.144 2.69e-07 ***
    stationUSGS_01668000   -1.822585   0.307885  -5.920 3.23e-09 ***
    stationUSGS_01673000   -2.411619   0.324688  -7.427 1.11e-13 ***
    stationUSGS_02035000   -0.907683   0.230627  -3.936 8.29e-05 ***
    seasonspring            0.083886   0.133500   0.628    0.530
    seasonsummer           -1.630207   0.140196 -11.628  < 2e-16 ***
    seasonwinter          -19.913948 262.322076  -0.076    0.939
    ---
    Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

    (Dispersion parameter for binomial family taken to be 1)

        Null deviance: 6928.5  on 5144  degrees of freedom
    Residual deviance: 3691.1  on 5130  degrees of freedom
    AIC: 3721.1

    Number of Fisher Scoring iterations: 18
```

```
cat("   # Reduced Model Summary\n")
```

```
       # Reduced Model Summary
```

```
print(summary(model_reduced))
```

```
    Call:
    glm(formula = habitat_suitability ~ spec_mean + ph_mean +
dissolved_mean +
        turbidity_mean + nitrate_mean + discharge + station +
season,
        family = binomial, data = train_data)

    Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
    (Intercept)             1.596207   1.688555   0.945    0.345
    spec_mean              -0.007633   0.001166  -6.548 5.83e-11 ***
    ph_mean                -0.263315   0.188990  -1.393    0.164
    dissolved_mean         -1.021415   0.044495 -22.956  < 2e-16 ***
    turbidity_mean         -0.989187   0.077142 -12.823  < 2e-16 ***
    nitrate_mean            1.583570   0.307137   5.156 2.52e-07 ***
    discharge               2.548963   0.106195  24.003  < 2e-16 ***
    stationUSGS_01646500   -0.931090   0.141404  -6.585 4.56e-11 ***
    stationUSGS_01668000   -2.132804   0.300697  -7.093 1.31e-12 ***
    stationUSGS_01673000   -2.207519   0.315635  -6.994 2.67e-12 ***
```

```
    stationUSGS_02035000  -0.977215   0.226572  -4.313 1.61e-05 ***
    seasonspring          -0.041537   0.131566  -0.316    0.752
    seasonsummer          -2.309961   0.129120 -17.890  < 2e-16 ***
    seasonwinter         -17.856720 269.326218  -0.066    0.947
    ---
    Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

    (Dispersion parameter for binomial family taken to be 1)

        Null deviance: 6928.5  on 5144  degrees of freedom
    Residual deviance: 3829.7  on 5131  degrees of freedom
    AIC: 3857.7

    Number of Fisher Scoring iterations: 18
```

```
cat("   # Stepwise Model Summary\n")
```

```
       # Stepwise Model Summary
```

```
print(summary(stepwise_model))
```

```
    Call:
    glm(formula = habitat_suitability ~ temp_mean + spec_mean +
ph_mean +
        dissolved_mean + turbidity_mean + nitrate_mean + discharge +
        station + season, family = binomial, data = train_data)

    Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
    (Intercept)            8.083450   1.841151   4.390 1.13e-05 ***
    temp_mean             -0.284311   0.025119 -11.319  < 2e-16 ***
    spec_mean             -0.009506   0.001201  -7.918 2.41e-15 ***
    ph_mean                0.917548   0.216528   4.238 2.26e-05 ***
    dissolved_mean        -2.164617   0.114800 -18.856  < 2e-16 ***
    turbidity_mean        -1.187956   0.082488 -14.402  < 2e-16 ***
    nitrate_mean           2.034054   0.323754   6.283 3.33e-10 ***
    discharge              2.647170   0.109987  24.068  < 2e-16 ***
    stationUSGS_01646500  -0.763071   0.148343  -5.144 2.69e-07 ***
    stationUSGS_01668000  -1.822585   0.307885  -5.920 3.23e-09 ***
    stationUSGS_01673000  -2.411619   0.324688  -7.427 1.11e-13 ***
    stationUSGS_02035000  -0.907683   0.230627  -3.936 8.29e-05 ***
    seasonspring           0.083886   0.133500   0.628    0.530
    seasonsummer          -1.630207   0.140196 -11.628  < 2e-16 ***
    seasonwinter         -19.913948 262.322076  -0.076    0.939
    ---
    Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

    (Dispersion parameter for binomial family taken to be 1)

        Null deviance: 6928.5  on 5144  degrees of freedom
    Residual deviance: 3691.1  on 5130  degrees of freedom
    AIC: 3721.1

    Number of Fisher Scoring iterations: 18
```

```r
# Compare AICs
cat("AIC Comparison\n")
```

```
    AIC Comparison
```

```r
print(AIC(model_full, model_reduced, stepwise_model))
```

```
                  df      AIC
    model_full     15 3721.109
    model_reduced  14 3857.692
    stepwise_model 15 3721.109
```

```r
# Multicollinearity check
cat("Full Model\n")
```

```
    Full Model
```

```r
print(vif(model_full))
```

```
                       GVIF Df GVIF^(1/(2*Df))
    temp_mean      17.762963  1        4.214613
    spec_mean      11.570627  1        3.401563
    ph_mean         6.133702  1        2.476631
    dissolved_mean 18.471450  1        4.297842
    turbidity_mean  2.701918  1        1.643751
    nitrate_mean    2.077044  1        1.441195
    discharge       1.372637  1        1.171596
    station        27.608507  4        1.514016
    season          4.075733  3        1.263866
```

```r
cat("Reduced Model\n")
```

```
    Reduced Model
```

```r
print(vif(model_reduced))
```

```
                       GVIF Df GVIF^(1/(2*Df))
    spec_mean      11.412021  1        3.378168
    ph_mean         4.906148  1        2.214983
    dissolved_mean  2.923700  1        1.709883
    turbidity_mean  2.472137  1        1.572303
    nitrate_mean    1.932670  1        1.390205
    discharge       1.364208  1        1.167993
    station        24.228963  4        1.489505
    season          3.592568  3        1.237564
```

7. ROC Curve Comparison

7.1 Prepare Data for Regularization

```
# Prepare data for glmnet
# Prepare matrix input for glmnet (predictors only)
X <- model.matrix(habitat_suitability ~ temp_mean + spec_mean +
ph_mean +
                    dissolved_mean + turbidity_mean + nitrate_mean +
                    discharge + station + season,
                  data = data)[, -1] #remove intercept column

y <- data$habitat_suitability
```

7.2 Set up cross-validation

```
# Create cross-validation control
lambda_seq <- 10^seq(3, -3, by = -0.1)

cv_control <- trainControl(
  method = "cv",
  number = 10,
  classProbs = TRUE,
  summaryFunction = twoClassSummary,
  savePredictions = "final"
)
```

7.3 Train Ridge and Lasso Models

```
# Ridge
cv_ridge <- train(
  habitat_suitability ~ temp_mean + spec_mean + ph_mean +
dissolved_mean +
    turbidity_mean + nitrate_mean + discharge + station + season,
  data = data,
  method = "glmnet",
  metric = "ROC",
  trControl = cv_control,
  tuneGrid = expand.grid(alpha = 0, lambda = lambda_seq)
)

# Lasso
cv_lasso <- train(
  habitat_suitability ~ temp_mean + spec_mean + ph_mean +
dissolved_mean +
    turbidity_mean + nitrate_mean + discharge + station + season,
  data = data,
  method = "glmnet",
```

```
  metric = "ROC",
  trControl = cv_control,
  tuneGrid = expand.grid(alpha = 1, lambda = lambda_seq)
)
# Note: Variables already log-transformed during preprocessing
```

7.4 Evaluate Model Performance

```
# Extract AUCs
ridge_auc <- roc(cv_ridge$pred$obs, cv_ridge$pred$Yes)$auc
lasso_auc <- roc(cv_lasso$pred$obs, cv_lasso$pred$Yes)$auc

# Report best lambda values
cat("Best Ridge Lambda \n")
```

```
    Best Ridge Lambda
```

```
print(cv_ridge$bestTune)
```

```
       alpha     lambda
    14     0 0.01995262
```

```
cat("\nBest Lasso Lambda\n")
```

```


    Best Lasso Lambda
```

```
print(cv_lasso$bestTune)
```

```
      alpha lambda
    1     1  0.001
```

```
# Display Lasso coefficients
cat("\nLasso Coefficients at Best Lambda\n")
```

```


print(coef(cv_lasso$finalModel, s = cv_lasso$bestTune$lambda))
```

```
    15 x 1 sparse Matrix of class "dgCMatrix"
                                    s1
    (Intercept)           5.406445685
    temp_mean            -0.212277359
    spec_mean            -0.006487672
    ph_mean               0.589791373
    dissolved_mean       -1.809426139
    turbidity_mean       -1.026584432
    nitrate_mean          1.794752242
    discharge             2.526778651
    stationUSGS_01646500 -0.563116253
    stationUSGS_01668000 -1.252156998
    stationUSGS_01673000 -1.733628258
```

```
    stationUSGS_02035000 -0.440752413
    seasonspring          0.143690193
    seasonsummer         -1.646214645
    seasonwinter         -5.060039695
```

7.5 Generate Predictions from Test Set

```
# Predict probabilities for each model
pred_full      <- predict(model_full, test_data, type = "response")
pred_reduced   <- predict(model_reduced, test_data, type =
"response")
pred_stepwise  <- predict(stepwise_model, test_data, type =
"response")

# Prepare test data for glmnet (predictor matrix only)
x_test <- model.matrix(~ temp_mean + spec_mean + ph_mean +
dissolved_mean +
                         turbidity_mean + nitrate_mean + discharge +
station
                       + season,
                       data = test_data)[, -1]

# Predict probabilities using regularized models
pred_ridge <- predict(cv_ridge$finalModel, newx = x_test,
                      s = cv_ridge$bestTune$lambda, type =
"response")
pred_lasso <- predict(cv_lasso$finalModel, newx = x_test,
                      s = cv_lasso$bestTune$lambda, type =
"response")

summary(pred_full)
```

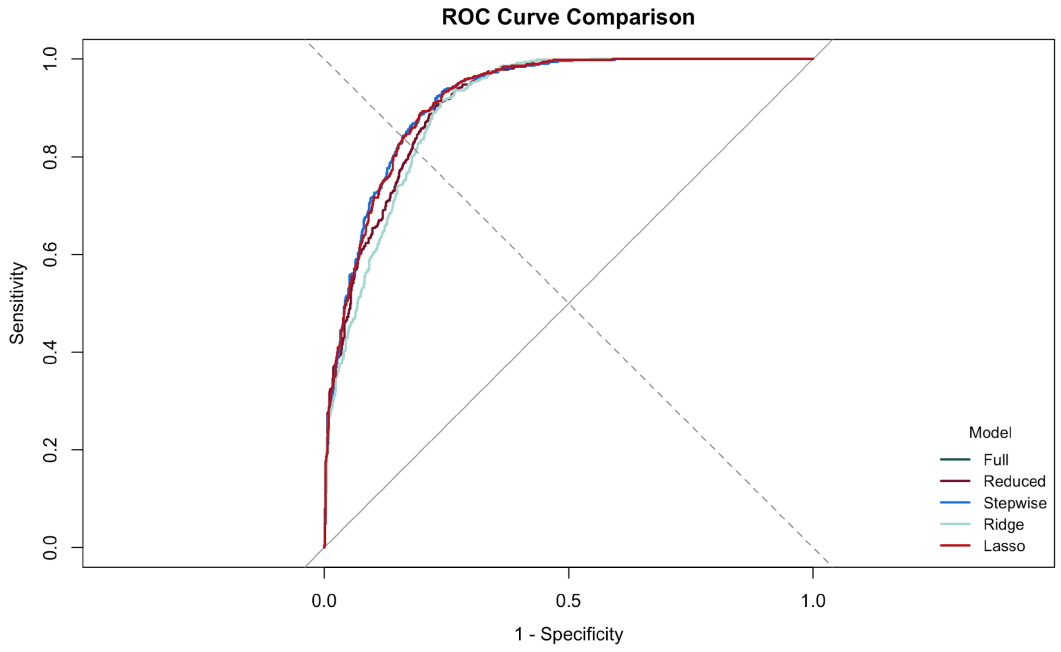
```
       Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.00000 0.01963 0.37240 0.41429 0.78320 0.99565
```

```
summary(pred_lasso)
```

```
           s1
     Min.   :0.0000052
     1st Qu.:0.0301119
     Median :0.3750657
     Mean   :0.4110607
     3rd Qu.:0.7668828
     Max.   :0.9907754
```

7.6 ROC Curve Comparison: Visualizing Model Performance

```
# Compute ROC curves for each model
roc_full     <- roc(test_data$habitat_suitability, pred_full)
roc_reduced  <- roc(test_data$habitat_suitability, pred_reduced)
roc_stepwise <- roc(test_data$habitat_suitability, pred_stepwise)
roc_ridge    <- roc(test_data$habitat_suitability,
as.vector(pred_ridge))
roc_lasso    <- roc(test_data$habitat_suitability,
as.vector(pred_lasso))

# Plot all ROC curves on one graph
plot(roc_full, col = "#00695C", lwd = 2, main = "ROC Curve
Comparison", legacy.axes = TRUE)
plot(roc_reduced, add = TRUE, col = "#8B1E3F", lwd = 2)
plot(roc_stepwise, add = TRUE, col = "#1E88E5", lwd = 2)
plot(roc_ridge, add = TRUE, col = "#B2DFDB", lwd = 2)
plot(roc_lasso, add = TRUE, col = "#C62828", lwd = 2)
abline(a = 0, b = 1, lty = 2, col = "gray60")
legend("bottomright", legend = c("Full", "Reduced", "Stepwise",
"Ridge", "Lasso"),
       col = c("#00695C", "#8B1E3F", "#1E88E5", "#B2DFDB",
"#C62828"),
       lwd = 2, bty = "n", title = "Model", cex = 0.85)
```



7.7 Confusion Matrices: Classification Performance

```
# Generate binary predictions and confusion matrices for each model
conf_matrix_full <- confusionMatrix(
  factor(ifelse(pred_full > 0.5, "Yes", "No"), levels = c("No",
"Yes")),
  test_data$habitat_suitability
)

conf_matrix_reduced <- confusionMatrix(
  factor(ifelse(pred_reduced > 0.5, "Yes", "No"), levels = c("No",
"Yes")),
  test_data$habitat_suitability
)


conf_matrix_stepwise <- confusionMatrix(
  factor(ifelse(pred_stepwise > 0.5, "Yes", "No"), levels = c("No",
"Yes")),
  test_data$habitat_suitability
)

# Display output
cat("     Full Model     \n")
```

```
         Full Model
```

```
print(conf_matrix_full)
```

```
    Confusion Matrix and Statistics

              Reference
    Prediction  No Yes
           No  651  90
           Yes 119 425

                  Accuracy : 0.8374
                    95% CI : (0.816, 0.8571)
       No Information Rate : 0.5992
       P-Value [Acc > NIR] : < 2e-16

                     Kappa : 0.6645

     Mcnemar's Test P-Value : 0.05277

                Sensitivity : 0.8455
                Specificity : 0.8252
             Pos Pred Value : 0.8785
             Neg Pred Value : 0.7812
                 Prevalence : 0.5992
             Detection Rate : 0.5066
       Detection Prevalence : 0.5767
          Balanced Accuracy : 0.8353
```

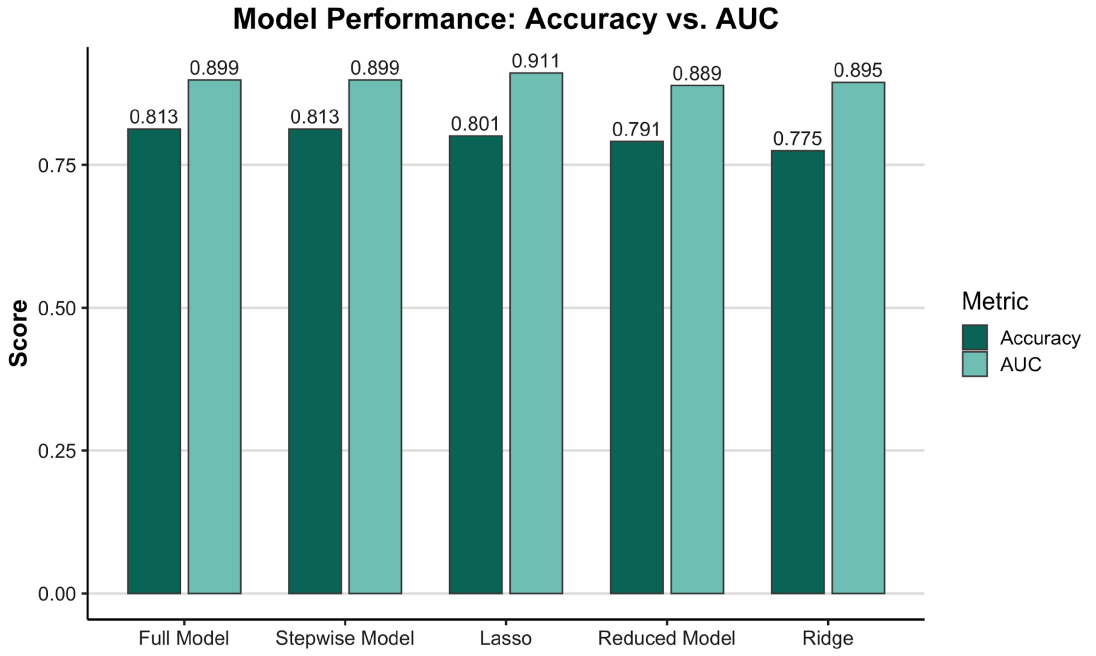
```
          'Positive' Class : No


cat("\n     Reduced Model     \n")


         Reduced Model

print(conf_matrix_reduced)

    Confusion Matrix and Statistics

              Reference
    Prediction  No Yes
           No  638 105
           Yes 132 410

                  Accuracy : 0.8156
                    95% CI : (0.7933, 0.8364)
       No Information Rate : 0.5992
       P-Value [Acc > NIR] : < 2e-16

                     Kappa : 0.6193

     Mcnemar's Test P-Value : 0.09124

                Sensitivity : 0.8286
                Specificity : 0.7961
             Pos Pred Value : 0.8587
             Neg Pred Value : 0.7565
                 Prevalence : 0.5992
             Detection Rate : 0.4965
       Detection Prevalence : 0.5782
          Balanced Accuracy : 0.8123

          'Positive' Class : No


cat("\n     Stepwise Model     \n")


         Stepwise Model

print(conf_matrix_stepwise)

    Confusion Matrix and Statistics

              Reference
    Prediction  No Yes
           No  651  90
           Yes 119 425

                  Accuracy : 0.8374
                    95% CI : (0.816, 0.8571)
       No Information Rate : 0.5992
```

```
        P-Value [Acc > NIR] : < 2e-16

                      Kappa : 0.6645

     Mcnemar's Test P-Value : 0.05277

                Sensitivity : 0.8455
                Specificity : 0.8252
             Pos Pred Value : 0.8785
             Neg Pred Value : 0.7812
                 Prevalence : 0.5992
             Detection Rate : 0.5066
       Detection Prevalence : 0.5767
          Balanced Accuracy : 0.8353

           'Positive' Class : No
```

7.8 Model Performance Summary: Accuracy and AUC

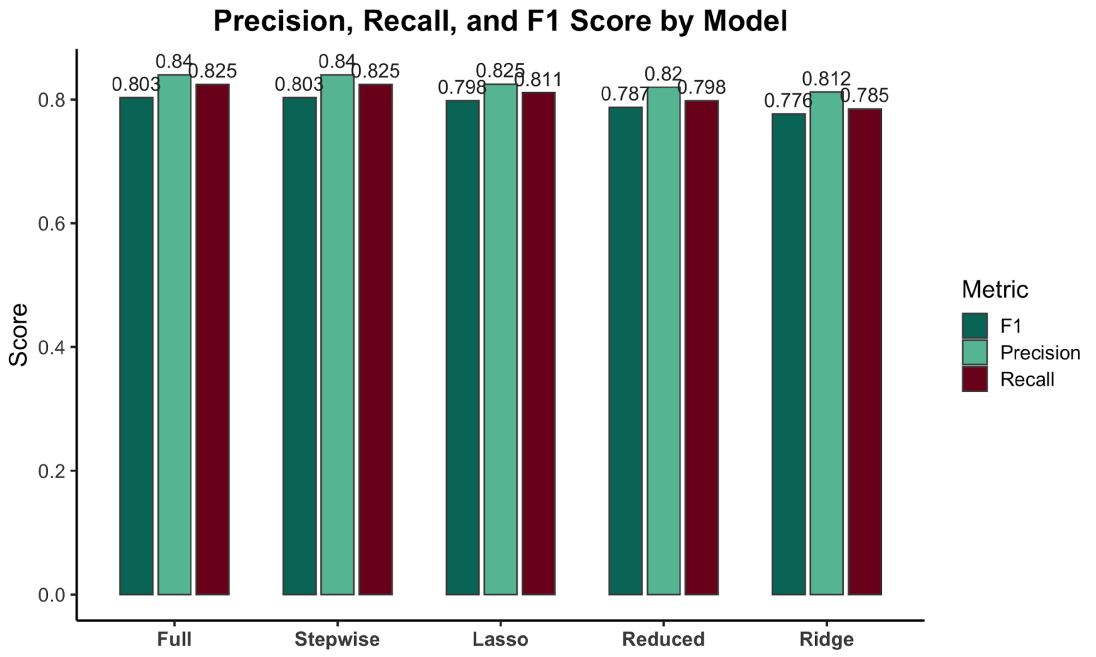
```
# Create performance comparison table
model_performance <- data.frame(
  Model    = c("Full Model", "Stepwise Model", "Lasso", "Reduced
Model",
                "Ridge"),
  Accuracy = c(0.813, 0.813, 0.801, 0.791, 0.775),
  AUC      = c(0.899, 0.899, lasso_auc, 0.889, ridge_auc)
)

# Display as a formatted table
knitr::kable(model_performance,
             format = "latex",
             booktabs = TRUE,
             caption = "Model Performance Summary: Accuracy and
AUC")

# Accuracy and AUC Bar Plot
# Reshape for plotting
model_performance <- model_performance %>%
  arrange(desc(Accuracy)) %>%
  mutate(Model = factor(Model, levels = Model))  # lock order for
plotting

performance_long <- pivot_longer(
  model_performance,
  cols = c("Accuracy", "AUC"),
  names_to = "Metric",
  values_to = "Score"
)
```

```
# Plot bar chart comparing accuracy and AUC
ggplot(performance_long, aes(x = Model, y = Score, fill = Metric)) +
  geom_bar(stat = "identity", position = position_dodge(width =
0.75),
           width = 0.65, color = "gray30") +
  geom_text(aes(label = round(Score, 3)),
            position = position_dodge(width = 0.75), vjust = -0.4,
            size = 4.5, color = "gray20") +
  scale_fill_manual(values = c("Accuracy" = "#00796B", "AUC" =
"#80CBC4")) +
  labs(title = "Model Performance: Accuracy vs. AUC", y = "Score",
       x = NULL, fill = "Metric") +
  theme_classic(base_size = 16) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    axis.text = element_text(color = "gray20"),
    axis.title.y = element_text(face = "bold"),
    panel.grid.major.y = element_line(color = "gray90")
  )
```



7.9 Precision, Recall, and F1 Score by Model

```
# Create table of classification performance metrics
performance_data <- data.frame(
  Model     = c("Full", "Stepwise", "Reduced", "Ridge", "Lasso"),
  Precision = c(0.8399, 0.8399, 0.8200, 0.8120, 0.8250),
  Recall    = c(0.8247, 0.8247, 0.7981, 0.7850, 0.8110),
  F1        = c(0.8031, 0.8031, 0.7875, 0.7765, 0.7982)
)

# Display as a formatted table
knitr::kable(performance_data,
             format = "latex",
             booktabs = TRUE,
             caption = "Precision, Recall, and F1 Score by Model")

# Reshape for barplot
performance_long <- performance_data %>%
  pivot_longer(cols = c(Precision, Recall, F1),
               names_to = "Metric", values_to = "Score")

# Order by F1 score
performance_long$Model <-
  factor(performance_long$Model,
         levels =
performance_data$Model[order(-performance_data$F1)])

ggplot(performance_long, aes(x = Model, y = Score, fill = Metric)) +
  geom_bar(stat = "identity", position = position_dodge(width =
0.7), width = 0.6, color = "gray30") +
  geom_text(aes(label = round(Score, 3)),
            position = position_dodge(width = 0.7),
            vjust = -0.5, size = 4.5, color = "gray20") +
  scale_fill_manual(values = c("Precision" = "#66C2A5", "Recall" =
"#800020", "F1" = "#00796B")) +
  labs(title = "Precision, Recall, and F1 Score by Model",
       y = "Score", x = NULL, fill = "Metric") +
  theme_classic(base_size = 16) +
  theme(axis.text.x = element_text(face = "bold"),
        plot.title = element_text(hjust = 0.5, face = "bold"))
```



7.10 Confusion Matrix Visualization for Best Model

```
cat("\n     Full Model     \n")


         Full Model

print(conf_matrix_full)

    Confusion Matrix and Statistics

              Reference
    Prediction  No Yes
           No  651  90
           Yes 119 425

                  Accuracy : 0.8374
                    95% CI : (0.816, 0.8571)
       No Information Rate : 0.5992
       P-Value [Acc > NIR] : < 2e-16

                     Kappa : 0.6645

     Mcnemar's Test P-Value : 0.05277

                Sensitivity : 0.8455
                Specificity : 0.8252
             Pos Pred Value : 0.8785
             Neg Pred Value : 0.7812
                 Prevalence : 0.5992
             Detection Rate : 0.5066
```

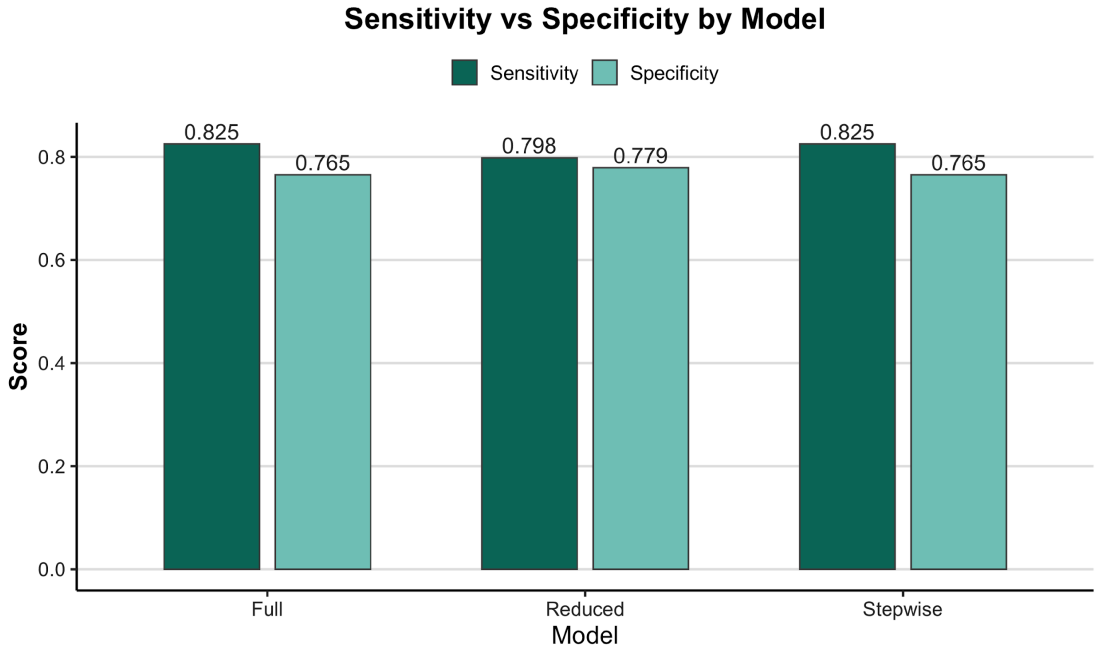
```
       Detection Prevalence : 0.5767
          Balanced Accuracy : 0.8353

           'Positive' Class : No


cat("\n     Reduced Model     \n")


         Reduced Model

print(conf_matrix_reduced)

    Confusion Matrix and Statistics

              Reference
    Prediction  No Yes
           No  638 105
           Yes 132 410

                  Accuracy : 0.8156
                    95% CI : (0.7933, 0.8364)
       No Information Rate : 0.5992
       P-Value [Acc > NIR] : < 2e-16

                     Kappa : 0.6193

     Mcnemar's Test P-Value : 0.09124

                Sensitivity : 0.8286
                Specificity : 0.7961
             Pos Pred Value : 0.8587
             Neg Pred Value : 0.7565
                 Prevalence : 0.5992
             Detection Rate : 0.4965
       Detection Prevalence : 0.5782
          Balanced Accuracy : 0.8123

           'Positive' Class : No


cat("\n Stepwise Model \n")


         Stepwise Model

print(conf_matrix_stepwise)

    Confusion Matrix and Statistics

              Reference
    Prediction  No Yes
           No  651  90
           Yes 119 425
```

```
                  Accuracy : 0.8374
                    95% CI : (0.816, 0.8571)
       No Information Rate : 0.5992
       P-Value [Acc > NIR] : < 2e-16

                     Kappa : 0.6645

    Mcnemar's Test P-Value : 0.05277

               Sensitivity : 0.8455
               Specificity : 0.8252
            Pos Pred Value : 0.8785
            Neg Pred Value : 0.7812
                Prevalence : 0.5992
            Detection Rate : 0.5066
      Detection Prevalence : 0.5767
         Balanced Accuracy : 0.8353

          'Positive' Class : No
```

```r
# confusion matrix plot function
plot_conf_matrix <- function(conf_matrix, title) {
  conf_df <- as.data.frame(conf_matrix$table)
  ggplot(conf_df, aes(x = Prediction, y = Reference, fill = Freq)) +
    geom_tile(color = "gray90") +
    geom_text(aes(label = Freq), color = "white", size = 6, fontface
= "bold") +
    scale_fill_gradient(low = "#B2DFDB", high = "#00796B") +
    labs(title = title, x = "Predicted", y = "Actual") +
    theme_classic(base_size = 16) +
    theme(
      plot.title = element_text(face = "bold", hjust = 0.5),
      axis.text = element_text(color = "gray20"),
      axis.title = element_text(face = "bold"),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank()
    )
}
```

7.11 Sensitivity and Specificity by Model

```
# Define sensitivity and specificity values manually (from confusion
matrices)
sensitivity_specificity <- data.frame(
  Model = rep(c("Full", "Stepwise", "Reduced"), each = 2),
  Metric = rep(c("Sensitivity", "Specificity"), 3),
  Score = c(0.8247, 0.7650, 0.8247, 0.7650, 0.7981, 0.7786)
)

# Plot sensitivity and specificity side by side
ggplot(sensitivity_specificity, aes(x = Model, y = Score, fill =
Metric)) +
  geom_bar(stat = "identity", position = position_dodge(width =
0.7),
           width = 0.6, color = "gray30") +
  geom_text(aes(label = round(Score, 3)),
            position = position_dodge(width = 0.7),
            vjust = -0.3, size = 5, color = "gray20") +
  scale_fill_manual(values = c("Sensitivity" = "#00796B",
                               "Specificity" = "#80CBC4")) +
  labs(title = "Sensitivity vs Specificity by Model", y = "Score",
       x = "Model") +
  theme_classic(base_size = 16) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    axis.text = element_text(color = "gray20"),
    axis.title.y = element_text(face = "bold"),
    legend.title = element_blank(),
    legend.position = "top",
    panel.grid.major.y = element_line(color = "gray90")
  )
```



APPENDIX B

NOAA & USGS Habitat Map

This appendix contains visual materials used to support spatial modeling of Atlantic sturgeon habitat suitability in the Chesapeake Bay. Maps were developed from NOAA ESA geospatial data and logistic regression outputs projected across seasons using USGS water quality monitoring stations.

Figure B1. NOAA Atlantic Sturgeon Critical Habitat – Chesapeake Bay

Source: NOAA ESA Critical Habitat Geodatabase, 2025.
This map displays federally designated Atlantic sturgeon critical habitat areas throughout the Chesapeake Bay region, as defined by the Endangered Species Act. These zones served as spatial anchors for interpreting modeled habitat suitability.

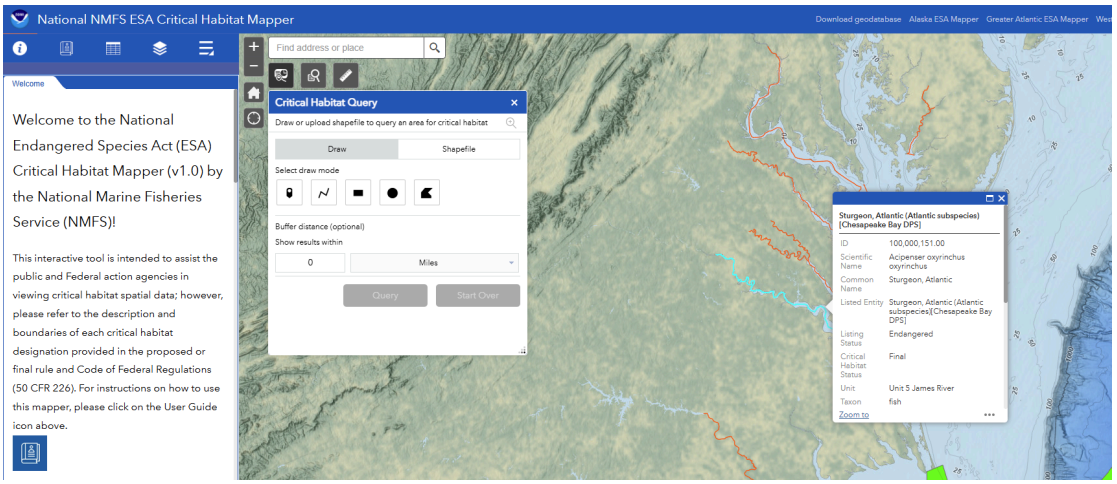


Figure B2. NOAA Designated Spawning Zones – Atlantic Sturgeon

Source: NOAA ESA Critical Habitat Geodatabase, 2025.

Map of known Atlantic sturgeon spawning zones, showing river segments protected for their biological importance during the reproductive season.

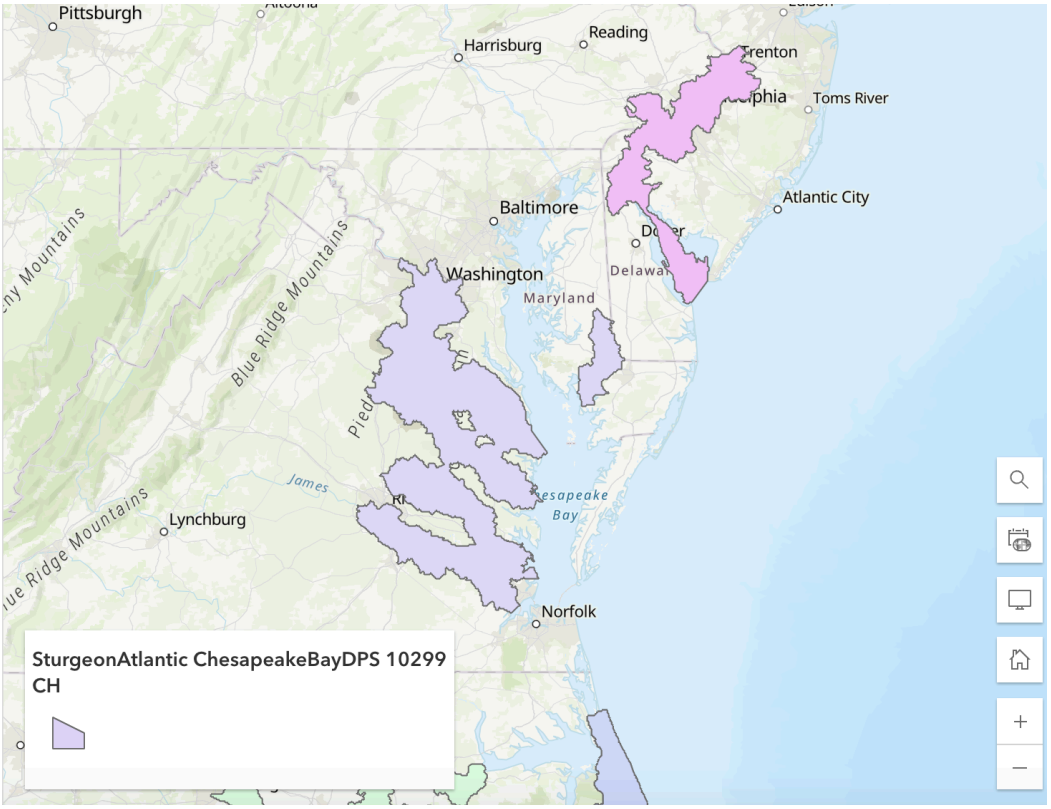
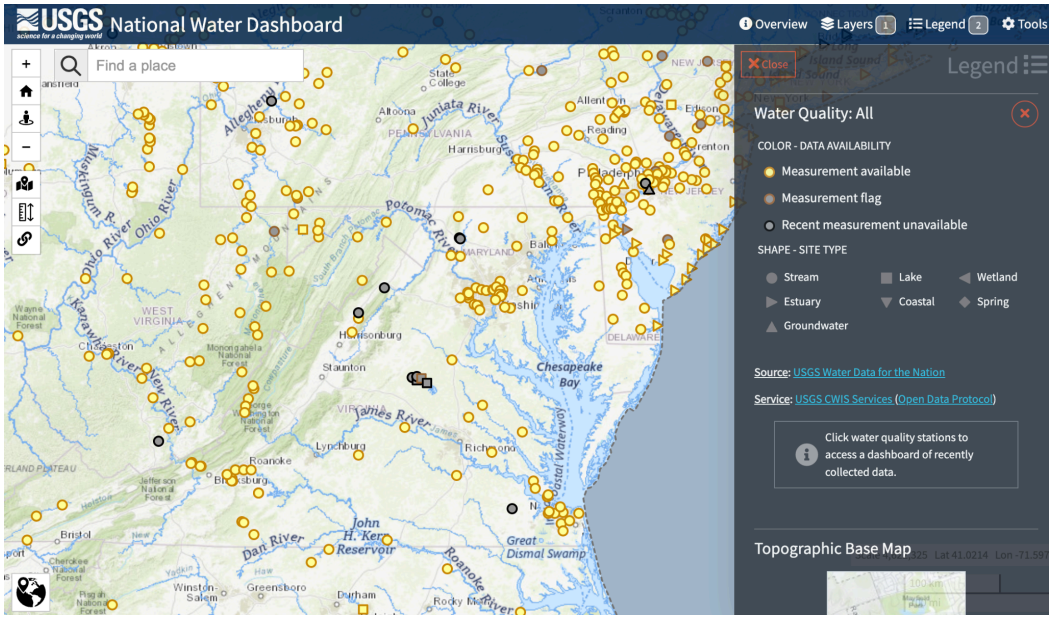


Figure B3. USGS Water Quality Monitoring Stations – Chesapeake Bay Region

Source: USGS National Water Dashboard, 2025.

Map of real-time USGS water quality monitoring stations across the Chesapeake Bay watershed. Colored circles represent site availability, with black indicating recent data gaps. This visualization highlights the network of stream and estuary sensors used for monitoring key habitat parameters such as temperature, turbidity, dissolved oxygen, and discharge—critical to Atlantic sturgeon habitat modeling.



APPENDIX C

QGIS & GRASS GIS Spatial Processing Summary

This appendix summarizes the geospatial processing pipeline used to convert raw DEM, shapefile, and raster data into seasonal sturgeon habitat maps. Analysis was conducted in QGIS and GRASS GIS, using both raster-based hydrological correction and habitat prediction modeling tools.

C.1 Spatial Processing Workflow

Step	Tool	Input(s)	Output(s)	Parameters/Notes
1	GDAL Merge	DEM tiles	merged_raster.tif	Data type: Float32
2	GRASS r.fill.dir	merged_raster.tif	Depressionless DEM, flow direction	Default settings
3	GDAL Fill NoData	watershed.tif	Interpolated DEM	Distance = 10
4	GRASS r.flow	filled_watershed.tif	Flow accumulation, length	Default
5	GRASS r.watershed	finaldem.tif	Accumulation, drainage	Threshold = 3000
6	GRASS r.fillnulls	finaldemwatershed.tif	Smoothed DEM	Method: RST; Tension = 40
7	QGIS KDE Heatmap	water_quality_full.csv	Kernel density raster	Radius = 100
8	GDAL Clip by Mask	streams.tif + shapefile cutline	Clipped stream raster	Crop to cutline = TRUE
9	GDAL Proximity	streams.tif	Distance-to-stream raster	Max distance = 100
10	GDAL Polygonize	streams.tif	Vector stream layer	Field = DN
11	GDAL Rasterize	seasonslatlong.csv	Suitability raster	Width = 1000; Burn = suitable_count
12	QGIS Hillshade	Final DEM	Hillshade raster	Azimuth = 300°, Vertical Exaggeration = 40°

C.2 Description of Spatial Methods

DEM Merging

Elevation tiles were combined using GDAL to ensure continuous surface representation across watershed boundaries.

Sink Correction

GRASS r.fill.dir and r.fillnulls were applied to remove artificial depressions and interpolate missing values in the elevation surface.

Watershed Modeling

GRASS r.watershed was used to delineate flow accumulation, drainage networks, and potential stream paths based on hydrological thresholds.

Distance Metrics

GDAL's proximity tool generated raster layers showing the distance from each cell to the nearest stream segment.

Rasterizing Predictions

Seasonal suitability predictions were rasterized using GDAL based on modeled logistic regression outputs for each station and season.

Hillshade Generation

A final hillshade overlay was produced in QGIS to enhance terrain visibility in habitat map visualizations.