Examining the Relationship between Water Quality and Land Use Land Cover in the Huron River Watershed

By**:**

Maya R. Morgan

A thesis submitted
in partial fulfillment of the requirements
for the degree of
Master of Science
(Environment and Sustainability)
in the University of Michigan
May 2024

Thesis Committee:
Professor Andrew Gronewold, Chair
Professor George Kling
Assistant Professor Runzi Wang

Outline

1. Abstract

The nature of the landscape that composes a watershed plays a role in determining the chemistry of runoff that makes its way into a body of water. The goal of this thesis was to assess the impacts of different patterns of land use land cover (LULC) on water quality outcomes in the Huron River Watershed. Focusing on two different subwatersheds, Mill and Allen Creek, I used RStudio to create a series of statistical models for the prediction of five different water quality indicators using percent cover by urban area, percent cover by agriculture, and percent cover by impervious surfaces as explanatory variables. The water quality indicators in question were total dissolved solids, total suspended solids, total phosphorus, nitrite, and nitrate. I also performed additional analyses in the form of two sample t-tests, hierarchical clustering, principal components analysis (PCA), and logistic regression.

I experimented with several different types of models, including generalized additive models, support vector machine regression, random forest regression, neural network regression, and a linear model in OpenBugs. These methods were met with varying degrees of success as indicated by R-squared values and mean squared error (MSE). As the ways in which humans interact with land continue to evolve, it may be useful to devise viable methods of predicting how these changes will affect the quality of the water that we rely on for both drinking and recreation. I concluded that percent cover alone does not provide sufficient information to accurately predict the parameters in question and that it might be helpful to include additional predictors such as temperature, precipitation, and flow in the analysis.

2. **Introduction:**

The composition of the land within a watershed plays an important role in determining the quality and quantity of runoff that makes its way into a body of water (Wilson, 2015). Existing research indicates that there is a relationship between changes in land use and changes in water quality metrics, as well as changes in the amount of runoff (Tong & Chenn, 2002). As water drains from the surface of the land and into a lake or river, it carries residues and debris from the land along with it. Existing research has demonstrated a negative relationship between urbanized land and water quality (Goodspeed et al., 2022; Huang et al., 2016). This is due in part to the impacts of impervious surfaces such as roads, bridges, and sidewalks (Anh et al., 2023). Precipitation runs easily off of these hard surfaces, resulting in large influxes of water that can alter a river's natural flow patterns and deposit excessive debris. Runoff from urbanized areas may contain remnants of harmful materials such as rubber, heavy metals, sodium, and sulfates from roads (Tong and Chenn, 2002). Runoff from agricultural and civil areas alike tends to contain high amounts of nutrients and sediments relative to other categories of land use land cover (LULC) (Huang et al., 2016).

The ultimate goal of this study is to assess the relationship between total dissolved solids, total suspended solids, total phosphorus, nitrite, and nitrate and LULC in Mill and Allen Creek, which are two tributaries of the Huron River. Subsequently, this study will explore different methods of predicting these water quality parameters using LULC variables as predictors. The high levels of imperviousness in Allen Creek may allow for contaminants to be deposited into the river more readily than in Mill Creek. However, it is possible that Mill

Creek may also experience large nutrient and sediment deposits as a result of the amount of agriculture in this area.

My governing hypotheses are that there is a statistically significant difference between the water quality outcomes of Mill and Allen Creek, and that points of measurement that are located in highly developed areas have different effects on the aforementioned water quality parameters than areas that are characterized by agricultural land, and that impervious surfaces are a significant predictor of these outcomes. Using data from a number of stations that are monitored regularly by the Huron River Watershed Council (HRWC) as well as publicly available LULC data, I will examine how the chemistry of the river is related to surrounding land use.

a. Study Area:

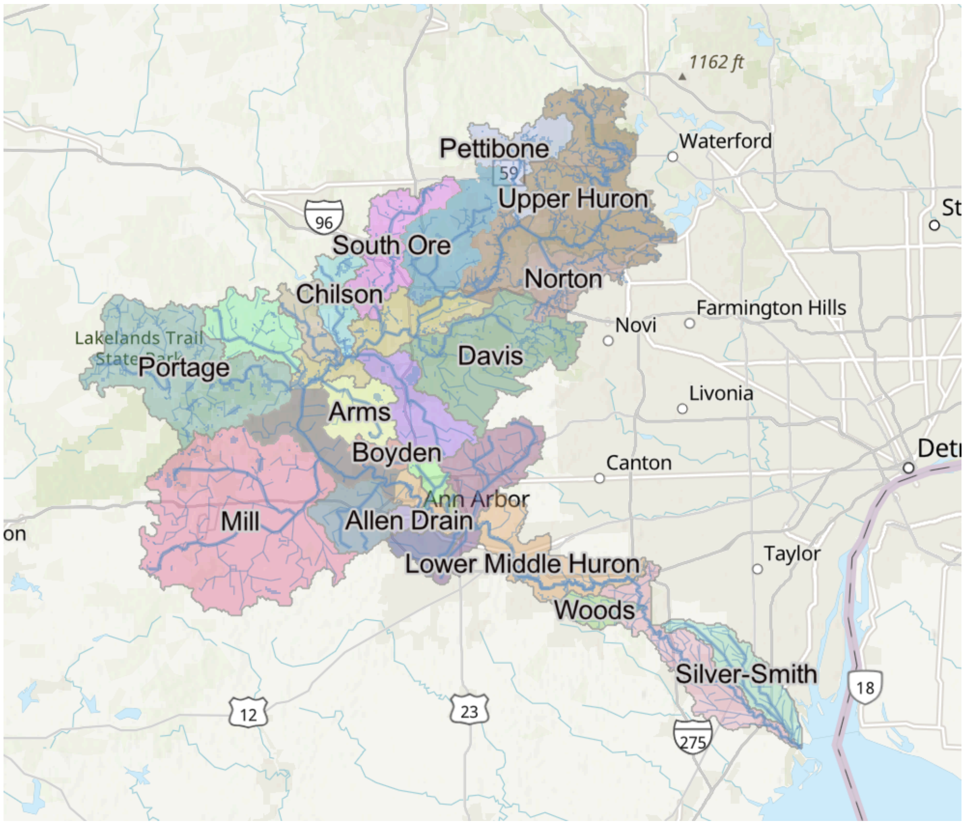


Figure 1: Huron River Watershed (HRWC, n.d.)

The Huron River Watershed is located in southeastern Michigan and spans a distance of roughly 125 miles beginning at Big Lake and ending at Lake Erie (Figure 1). The Huron River watershed has a surface area of approximately 900 square miles, containing hundreds of tributaries that feed into the river's main branch (MDNR, 2002). The land surrounding different portions of the river, and composing these tributaries and their unique sub-watersheds, is characterized by different patterns of LULC. There is also evidence that LULC within the watershed has changed substantially over the course of several years, with many areas becoming increasingly developed (Lei & Zhu, 2017; Figure 2). The landscape of the watershed was once dominated by deciduous forests and prairies but has since transformed. Presently, the primary LULC types within the watershed have shifted towards urban and agriculture (Hay-Chmielewski et al., 1995). These trends were already becoming apparent in the late 1990s, and based on comparisons between land use statistics from 1995 and today it is clear that at least some of them have come to fruition. Mill Creek, for instance, was only 2% urban in 1995 compared to 18% today (Hay-Chmielewski et al., 1995; HRWC, n.d). Some possible reasons for the increase in these two types of land use may include population increases in the area.

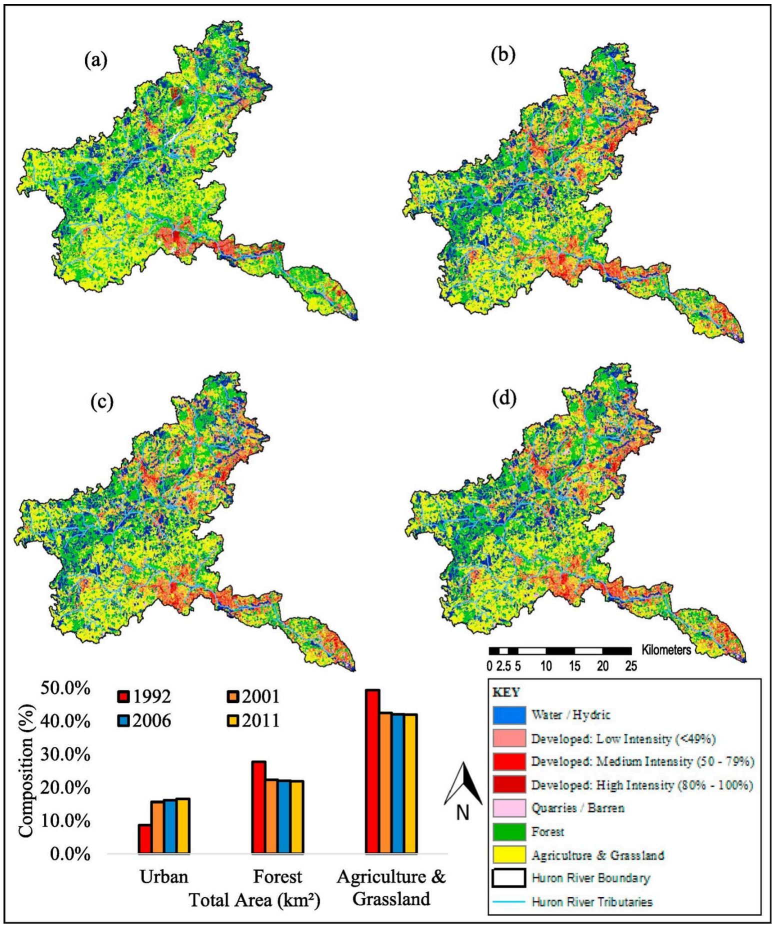


Figure 2: LULC within the Huron River Watershed at different time stamps (Lei & Zhu, 2017)

This thesis will examine how differences in LULC within two different tributaries that feed the main branch of the Huron River affect water quality within these subwatersheds, and attempt to fill the knowledge gap of how changes in LULC might alter the chemistry of the river. This information could help future management decisions that aim to mitigate pollution within the watershed and is particularly relevant to watersheds such as the Huron River which may be prone to future development and changing patterns of LULC. The two creeksheds within the Huron River that will be analyzed are Mill Creek and Allen Creek. Both of these tributaries are located in the middle portion of the river but are very different in LULC composition. The Allen Creek subwatershed is 5 square miles in area, 4.9 of which consist of urban and developed land. Consequently, Allen Creek contains a lot of impervious surfaces. Mill Creek, on the other hand, is 47% agricultural and only 18% urban/residential. It is the largest creekshed within the watershed, covering 143 square miles. It is also the most agricultural creekshed in the Huron system (HRWC, n.d).

The HRWC classifies the ecological health of Allen Creek as “Extremely Disturbed”, giving a score of 7 out of a 100-point scale (Huron River Watershed Council [HRWC], 2014). The HRWC classifies the ecological health of Mill Creek as only “Slightly Impacted”, scoring it a 71 out of 100. Since it is less urban, Mill Creek has much less impervious surface than Allen Creek. Due to these disparities, the expectation is that these two creeksheds will display different water quality outcomes.

b. Water Quality Parameters and LULC

i. TDS:

A study in Raipur, Chhattisgarh, India, attempted to correlate groundwater quantity and quality with LULC (Mondal et al., 2020). The authors of this study found evidence that TDS and some other chemical indicators increased significantly between the years 2000 and 2018, in correlation with rapid urbanization. Agricultural land use can lead to increases in erosion and sedimentation (Huang et al., 2016). This, in turn, could lead to increases in TDS concentrations.

ii. TSS:

Similar to TDS, there is evidence that links changes in TSS to specific changes in land use (Sunardi, et al 2021). For instance, a study of the Cirata reservoir in Indonesia revealed that 92.5% of land-use changes in the form of increasing mixed plantations, cropland, settlements, and grasslands could affect TSS levels in the reservoir. These relationships with these particular types of land use likely exist due to a variety of reasons. The impervious surfaces in settled areas, for example, can lead to increased water discharge and sedimentation of the reservoir. Plantations and cropland, however, can increase soil erosion which results in more pollutants being deposited into the water. On the other hand, the vegetation in forested areas can filter discharges of pollutants into lakes and rivers.

iii. Total Phosphorus

Agricultural and horticulture land practices directly impact phosphorus loading in water bodies (Hassan et al., 2015). This nutrient enrichment can lead to the deterioration of water quality, harmful algal blooms, eutrophication, hypoxia, and fish kills (Morée et al., 2013). Some existing studies have shown that only about 10-15% of agricultural phosphorus is taken up by crops, while 4-5% is carried away to lakes and streams via surface water runoff or infiltration processes (Wu et al., 2008). The remaining phosphorus is fixed within the soil, at which point it becomes inactive. Excessive fertilizer application, and possibly improper nutrient ratios, have contributed to the loss of phosphorus from agricultural fields to aquatic ecosystems. Urban activities can also exert a strong influence on global nitrogen and phosphorus cycles (Morée et al., 2013). Point sources of phosphorus-containing wastewater include human excrement, phosphorus-based detergents, and industry waste products.

iv. Nitrates/nitrites:

A 2020 study of the Yahara River Watershed in Wisconsin revealed statistically significant positive relationships between the percent area of agricultural land in the watershed and nitrate-nitrite concentrations in the river (Li et al., 2021). This suggests that the land cover composition of a watershed may play a role in determining nitrate-nitrite levels in a stream. Additionally, an inverse relationship was observed between nitrate-nitrite concentrations and an area factor, which represented the ratio of the area of woodland, recreational, open, and vacant subdivided land, or wetlands in the riparian zone to agricultural areas in the rest of the watershed. This is an indicator of the effectiveness of the riparian zone as a buffer between pollutants coming off of the catchment and the river.

c. Parameter Concentrations Over Time

The observations included in this study were all collected between 2003 and 2021. As changes in LULC have occurred throughout the watershed during this period, if a relationship does exist between land use and water quality then this should be reflected in the time series analysis of parameter concentrations. Pictured below is a time series plot representing fluctuations in each parameter across the study period. There is quite a bit of fluctuation for each one.

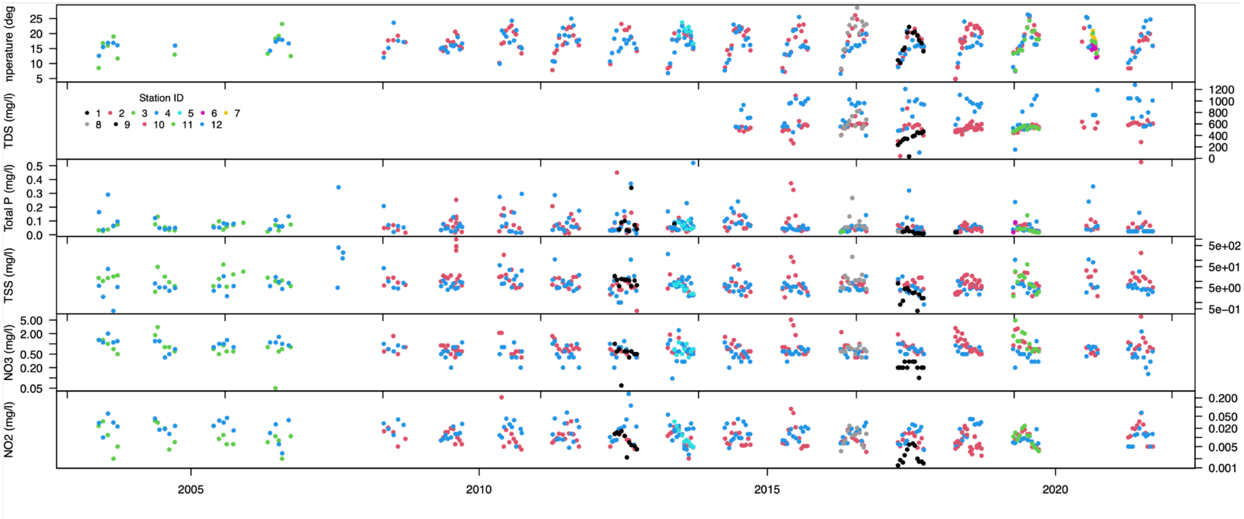


Figure 3: Time series plots showing changes in the concentrations of TDS, TSS, total phosphorus, nitrite, and nitrate between 2003 and 2021. Each monitoring station is represented by a different color.

3. Methods:

I begin this section by describing each of the parameters and how they are detected and measured, I then describe the approach I took to collecting and organizing the data. A subsection is dedicated to each of the exploratory analysis procedures that I performed, namely two-sample t-tests, hierarchical clustering analysis, PCA, and logistic regression. This is followed by a subsection for the Bayesian Inference Models that I created in OpenBugs. Finally, a series of subsections are dedicated to each of the predictive models that I created. These include GAMs, Support Vector Machine Regressions, Random Forest Regressions, and Neural Network Regressions.

a. Variable selection and description

As stated above, this project will focus on TDS, TSS, total phosphorus, nitrite, and nitrate as water quality indicators. TDS can come from a wide variety of sources, and may leach into waterways from sewage, water treatment plants, agricultural runoff, or industrial wastewater (United States Geological Survey [USGS], 2018). It is measured by placing a piece of equipment, known simply as a TDS meter, directly into a sample of water. A TDS rating of less than 300 mg/L is considered excellent, while a rating of over 1200 is considered unacceptable by the World Health Organization (WHO). TSS is a key variable used to describe and control sedimentation dynamics in water bodies (Adjovu et al., 2023). It is the most common pollutant by weight and volume in inland surface waters. There are multiple methods available for measuring TSS, including multiple particle filtration systems and, more recently, remote sensing. TSS is measured in mg/L. Nitrate and nitrite often occur simultaneously because their chemistries are intricately intertwined

(Moorcroft et al., 2001). The fairly inert nitrate ion is reduced to more reactive nitrite. Methods of detecting nitrate and nitrite include spectroscopy, electrochemical detection, chromatography, and capillary electrophoresis. Phosphorus, nitrates, and nitrites are common nutrients found in fertilizers. For this reason, they are often carried into rivers from agricultural runoff (de Oliveira et al., 2016). Phosphorus is a limiting nutrient in many aquatic ecosystems, but excessive levels can be detrimental to the health of aquatic systems by contributing to harmful algal blooms. Each of these nutrients is referred to in mg/L throughout this project.

b. Data organization

As stated above, the data used in this study was collected from the Huron River Watershed Council (HRWC) and contained observations from as far back as May of 2003 to August of 2021. Upon importing the data into RStudio, I noticed a number of issues with the data that initially prevented me from performing any type of statistical analysis. First off, the HRWC monitors a total of 133 stations across the entirety of the watershed. The original dataset that I was given was not organized by station, which made it difficult to isolate observations that were recorded in particular creek sheds. Additionally, the data contained a column that was labeled “Collection.Date”, but it was not sorted in chronological order. Using a GIS map that was created by Ric Lawson at the HRWC, 12 of the 133 chemistry and flow monitoring stations along the river were identified as being located within either Mill or Allen Creek. From there, I used the subset function to create a separate dataset for each station of these stations. Once I had done this, I used the order and as.Date functions to sort the “Collection.Date” column within each individual dataset in ascending order.

c. Bayesian Inference Models (JAGS and OpenBugs)

The first models were created in OpenBugs using RStudio (Appendix A2). They were linear regression models with LogNormal likelihoods. The predictor variables that were included in the model were percent area of developed land, percent covered by agriculture, and percent covered by impervious surfaces. A model was created for each parameter of interest, in which that parameter was the response variable for the model. Since the values of these observations are always positive, a normal likelihood and priors were selected. The process model that was created is as follows, where P represents a given parameter:

P[i] <- **alpha[1]** + **alpha[2]***developed[i] + **alpha[3]***agriculture[i] + **alpha[4]***impervious +**w**[stationNumber[i]]

The priors used for models were:

phi~dunif(0,100)
sig[i]<-1/tau[i]
α1 ~ Normal(0,1000) intercept, total phosphorus independent of land use
α2 ~ Normal(0,1000) slope effect of %developed
α3 ~ Normal(0,1000) slope effect of %agriculture

α4 ~ Normal(0,1000) slope effect of %impervious
tau[i]~dgamma(.01, .01)

The spatial random effect is represented by the variable **w**.

Since the model was designed to compare various water quality parameters between Mill and Allen Creek, all the observations of the parameters of interest from all of the stations within each tributary had to be appended. First, two vectors named “MillCreekTP” and “AllenCreekTP” were created by concatenating the “Total.Phosphorus" columns from each of the datasets that represented stations in Mill and Allen Creek, respectively. I then concatenated all the total phosphorus data into one long vector, titled “TP.all”. This was the vector that was ultimately fed into the model. This procedure was repeated several more times for TDS, TSS, nitrite, and nitrate. Then, the land cover use statistics needed to be converted into a format that could be fed into the model. For each creek, two new objects were created to represent the percent cover by developed and agricultural land in each creek. This was accomplished using the rep function with two arguments, the first of which was the percent cover by each land type, and the second of which was the number of observations for each creek. The latter arguments were equal to the lengths of the parameter vectors.

I created the distance matrix D using the dist function, which calculated the Euclidean distance between (x,y) longitude and latitude coordinates corresponding to each station within the tributaries of interest. These values were taken from a dataframe created from two vectors, x and y, which were composed of the longitude and latitude values respectively. The distance matrix was used to estimate the dependencies between stations based on their relative proximities to each other (Appendix A1). Separate versions of each model were created without these features to assess whether or not they help the models to predict more accurately. I tried for months to make these models work with limited success. The main issue that I kept running into was that most of the models were predicting in the negatives. I tried to remedy this issue by using the na.omit() function to remove NA values from the vectors containing the observations that were fed into the model. I also tried to perform log transformations, both within the models themselves and outside of them. Neither of these methods solved the problem. I was deeply unsatisfied with the performances of these models, which prompted me to abandon them completely and explore different methods of modeling and analyses.

d. Exploratory Analysis

i. Two Sample t-tests

I performed a series of two sample t-tests with a 95% confidence interval in order to compare the average measurements of each parameter between each tributary. I chose to perform a two sample t-test because the independent variable, the tributary in which each measurement was collected, was categorical and the dependent variable, the average concentration of the parameters, was continuous. Another reason that I elected for two sample t-tests was that my objective was to compare the means of two groups, rather than compare the mean of each group to some predetermined standard. In total I performed five Welch's t-tests. The null hypothesis for each test was that there was no statistically significant difference between the mean concentrations of each parameter inMill and Allen Creek. The alternative hypothesis was that there was a statistically significant difference between the mean concentrations of the parameters in the two creeksheds. If the p-value produced by each test was less than the alpha value of .05, then I concluded that I had sufficient evidence to reject the null hypothesis.

A copy of the code that was used to perform these analyses is included at the end of this document (Appendix B3). These two sample t-tests were intended to determine which water quality parameters might be most highly correlated with LULC, based on apparent differences between the two creekshed of very different land use composition. This characteristic would therefore make them better suited for a model that intends to predict water quality based on land use. The original data set that was provided to me by the HRWC contains 27 water quality variables and consists of 5054 rows, though many of them are incomplete. Although the HRWC samples from over 100 stations, only 12 were included in the models. These 12 were pulled from the original dataset.

ii. Hierarchical Clustering Analysis:

The goal of this analysis was to determine similarities between observations of various water quality parameters across various monitoring stations. Since I was only interested in 5 parameters from 12 monitoring stations, I created a new data frame in GoogleSheets for this analysis that only contained observations of those 5 parameters from those 12 stations. I also added three new columns to this new data frame in order to store the LULC statistics. The values in these columns corresponded to the land cover statistics from the tributary that each row of observations were taken from, as indicated by station number. Having the data in this format made it much more workable and allowed me to include only the variables I was interested in without having to filter anything. Hierarchical clustering can be a useful technique for exploring patterns and relationships in water quality and land use data.It allows you to group similar samples or variables based on their similarity in a hierarchical manner.

There aren't observations for each parameter at each station for each time step that is included in the data, which resulted in a number of NA values in the dataset. These NAs interfere with the clustering process, so they were removed using the function na.omit(). I originally tried to standardize the relevant features using the scale() function, but this resulted in negative values for the cluster means so I left the data unscaled. I initially decided to use 3 clusters for my analysis, then applied the clustering algorithm using the hclust() function (Appendix B5). When this produced very asymmetrical clusters, I increased this number to 5. After completing the analysis, the means of the features within each cluster were analyzed.

iii. PCA

After the clustering analysis, I performed a principal components analysis (PCA) with a correlation matrix. PCA is a dimension reduction technique that is intended to decrease the number of variables in a dataset while still explaining a high proportion of the data's variance. It is a multivariate statistical analysis that was created prior to World War II, and popularized during the "Quantitative Revolution" of the 1960s (Mackiewicz & Ratajcak, 1992). I performed this analysis using the princomp() function with the arguments "cor" and "scores" set equal to TRUE in the R statistical software package (R Core Team, 2022). In order to complete the PCA I still had to convert the data into numeric form, as R had categorized it as a list. The results indicated that nine principal components were needed to explain the total variance in the data (Appendix B4).

iv. Logistic regression

The next analysis that I performed was a series of logistic regressions in order to determine how often each water quality parameter is above its approved Environmental Protection Agency (EPA) limit as the percent cover by each LULC category increases. To perform this analysis, it was first necessary to define the EPA limits for each parameter of interest. For total dissolved solids, total suspended solids, total phosphorus, nitrite, and nitrate, these limits are 500 mg/L, 10 mg/L, .1 mg/L, 1 mg/L, and 10 mg/L respectively. The next step was to create binary response variables for each parameter. This was accomplished using the ifelse() function. I began by creating a series of new columns in the dataset titled "ParameterLimit-Exceeded", with "Parameter" corresponding to one of the water quality indicators of interest. Within the ifelse() function, I entered the parameter of interest as the first argument, followed by the > operator and the EPA limit for that parameter, the number 1, and the number 0. I did this for each parameter separately. This tells the function to enter a value of 1 in the "LimitExceeded" column for each parameter if the corresponding observation exceeds the EPA limit, and a 0 if it does not. A copy of this code can be found in Appendices B1.1-B1.5.

After the binary response variable was created for each parameter, a logistic regression model could be created using the glm() function (in the R ‘stats’ package). The models themselves are structurally equivalent to a linear combination, with the binary variable for each parameter as the response variable and the LULC variables as predictors. The main difference between the logistic regression and a generalized linear model is the fact that the “family” argument is set equal to “binomial” rather than “gaussian” (Appendix B1). A series of predicted probabilities plots were created for each parameter and each LULC variable. The code for these plots is located in Appendix B1.6

e. Predictive Models

i. Generalized Additive Models

Generalized additive models (GAMs) follow the general form $n(x) = d + f_1(x_1) + f_2(x_2) + \dots f_p(x_p)$, where the x_i terms are predictors and the f_i terms are functions of the predictors (Hastie & Tibshirani, 1985). These f_i terms may take the form of non-parametric “smooth” terms. In other words, each predictor has a function representing its unique form or shape. This is useful for modeling non-linear predictors. It is possible to test whether or not terms should be linear or not by fitting different versions of the model separately and analyzing the change in deviance relative to the change in degrees of freedom. Alternatively, stepwise model-building algorithms can be used to automatically select terms. Thus, GAMs can be useful because while they may look similar to a traditional linear model, the smooth terms can account for non-linear patterns in the data.

GAMs were used for water quality analysis in a case study of the Chesapeake Bay (Murphy et al.,2019). The study aimed to develop a GAM structure to describe the nonlinear, seasonally varying changes over time while accounting for hydrologic variability via either river flow or salinity. The model in this prior study was developed using the "mgcv" package in R. This package penalizes thin plate regression splines and can fit smooths with more than one variable, and uncertainty estimates using a Bayesian approach. This method showed potential for evaluating both large and small-scale water quality dynamics.

To create the GAMs in R, I began by installing the 'mgcv' package and calling the 'splines' package. I used the same data for these models as I used in the exploratory analyses, as I did for each model described in this section. My predictor variables consist of constant percentages of LULC categories, which caused the model to exclude certain predictors due to singularities. This is something that typically occurs due to collinearity among predictors. Collinearity describes the non-independence of predictor variables, and it can interfere with a model's ability to identify significant predictors (Dormann et al., 2013). To overcome this obstacle, I added jitter to my predictor variables using the runif() function. The models themselves were created using the gam() function with the parameter of interest as the response variable, and the LULC categories as the predictors. Each predictor was fitted with a 'bs' spline and 3 degrees of freedom.

To calculate the Mean Squared Error (MSE) of this model, I first generated a list of predicted values based on each model using the predict() function with the model name as the first argument and newdata set equal to data. The predict() function serves to make predictions from the results of a variety of model fitting functions (Chambers & Hastie, 1992). The newdata argument specifies where to look for explanatory variables to be used to make predictions, and attempts to match the columns in 'newdata' with those in the dataset that is used for fitting. The function checks to see that they are of comparable types and that any factors have the same level set in the same order or can be transformed to be so. The MSE was then calculated using the mean() function to take the average of the squared differences between the actual values of each parameter from the original dataset and the predicted values generated from the predict() function. This process was repeated for each parameter and its corresponding model. The R^2 values are provided automatically in the GAM results, viewed using the summary() function (Appendix A3).

ii. Random Forest Model

Random forest models can be useful tools for prediction. They are quick, accurate, and can handle a large number of input variables without overfitting (Biau, 2012). Classification and regression can be achieved using an aggregation of decision trees (Shaikhina et al., 2019). Inputs are passed through the constituent decision trees and each makes a prediction independently. Final predictions are made based on the aggregate results of the ensemble. There is some evidence that Random Forest Regression (RFR) can be used to predict water quality with greater accuracy than certain traditional linear methods of modeling, based on a prior study on groundwater nitrate concentrations across the African continent (Ouedraogo et al.,2019). One of the primary advantages of RFRs is their non-parametric nature. In other words, RFRs don't need to follow a normal distribution. Random forest models can also predict with exceptional accuracy and can be used to better understand the individual importance and combined effects of explanatory variables. Additionally, RFRs are computationally less demanding than some other methods of modeling such as Multiple Linear Regression(MLR), and parameterization of the models is simple. The authors of this study found that RFR displayed greater predictive power for nitrate concentrations than MLR (R^2 = .92 versus .64). RFR may also be less vulnerable to outliers than other methods because each tree is generated from a different data subset. A potential disadvantage of RFR is its so-called "black-box" nature. This refers to the fact that RFR does not provide direct estimates of the coefficients of explanatory variables. In summation, there is apparent potential for RFR in hydrogeologic studies due to its ability to handle non-linear data and complex relationships.

To create the random forest regression models, I first installed the "randomForest" package. I proceeded to use the randomForest() function to create a model that resembled a linear combination, with the parameter of interest as the response variable and the LULC categories as predictors. Neither MSE or R^2 are included in the results of the model provided by the summary() function. I was, however, able to extract the MSE values produced by each tree of the forest using the $ operator and storing these lists in their own objects. Then, using the mean() function I was able to aggregate all of the MSEs produced by the model. This action serves to provide a better overall look at the performance of the model. To calculate the R^2 for the model, I once again utilized the predict() function, with the model name as the first argument and the newdata argument set equal to 'data'. I then calculated the R^2 for each model using the cor() function, with the parameters as the first argument, followed by the predicted values generated from the predict() function, and subsequently squaring this result. This process was repeated for each model (Appendix A4).

iii. Support Vector Machine

Support Vector Machines (SVMs) are machine learning algorithms that can be used to perform classification or occasionally regression (Pisner & Schnyer). When used for classification, an SVM can separate observations belonging to different classes based on characteristic patterns present in the data. Subsequently, the SVM can be used to determine the most likely categorization for new, incoming data. When used for regression analysis, the goal is to find a “hyperplane” that lies close to as many of the data points as possible in order to minimize the difference between actual and predicted values (Trafalis & Ince). There has been some exploration into the use of Support Vector Regression (SVR) for water quality prediction. A study based on the Gomti River in India attempted to develop an SVR for the prediction of biochemical oxygen demand (BOD) using a set of predictor variables (Singh et al., 2011).

I started building the SVM regression model by first calling the ‘e1071’ package. I also installed the ‘caret’ package. To create the models I used the svm() function with the parameters of interest as the response variable and the LULC categories as the predictors. R^2 and MSE are not included in the model summary for SVM regression, so they once again had to be calculated externally. I did this by once again using the predict() function with the model name as an argument and newdata set equal to “data”. I then used the R2() function from the ‘caret’ package, with the parameter of interest as the first argument and the object storing the predicted values as the second. To calculate the MSE I used the mean() function to take the average of the squared difference between the observed and predicted values for each parameter (Appendix A5).

iv. Neural network

The basic structure of a typical neural network model consists of one layer of input nodes, one layer of “hidden” nodes, and one layer of output nodes (Farragi & Simon,1995). Each node of the input layer is directly connected to all of the nodes in the hidden layer except for one, and each node has an associated weight. The weight represents how strong the influence of the input is on the final result. In the hidden layer, each weight is multiplied by the value of its corresponding input, and then a non-linear transformation is performed on the sum of the weighted inputs. Final predictions are then produced in the output layer. Most neural network models also have a special hidden node (node 0) that is connected to the output nodes but not the input ones, which functions similarly to the constant in a linear regression. Neural networks are feed-forward models, meaning that information only flows in one direction.

Similar to some of the other methods described above, neural networks have been touted for their use in water quality monitoring because they are better equipped to handle non-linear data and the non-stationary nature of water (Chen et al.,2020). Artificial neural networks (ANNs) have displayed competence in handling problems in rivers, lakes, reservoirs, ponds, groundwater, streams, and wastewater treatment plants (WWTPs). ANNs are data-driven models, so obtaining the correct amount of data is essential. For a regression neural network, approximately 100 pieces of data are necessary.

To build the neural network models, I first called the ‘neuralnet’ package. Then, using the neuralnet() function I created the models with the parameter of interest as the response variable and the LULC categories as predictors. To calculate the R^2 values for each model, I first stored the actual values of each parameter in their own unique objects. Then, using the sum() function I took the sum of the squared difference between the observations and the mean of the observed values, storing this sum in a separate object. Using the predict() function, I generated a series of predicted values based on each model, storing each one in its own object. I then calculated the residuals of each model using the sum() function to find the squared difference between the observed and predicted values of each parameter. Finally, I calculated the R^2 value for each model by finding the quotient of the sum of the squared difference between the observations and the mean of the observed values and the residuals. To calculate the MSE for each model, I used the mean() function to find the average of the squared difference between the observed and predicted values (Appendix A6).

4. Results:

In the following sections, I will describe the results of each of the exploratory analyses that I performed prior to attempting to create predictive models. These include two-sample t-tests, a hierarchical clustering analysis, PCA, and a logistic regression for each parameter. I will then proceed to state the results for each predicted model, separated by parameter.

a. OpenBugs Results

i. TDS: Covariance Matrix and Spatial Random Effect

Zero is contained within alphas 1, 3, and 4. These are the intercept of the linear model, the beta coefficient for percent cover by agricultural land, and percent cover by impervious surfaces, respectively. The only alpha that displays statistical significance is alpha 2, which corresponds to the beta coefficient for percent cover by urban/developed land. The convergence of each alpha value can be seen in figures 4 through 7.

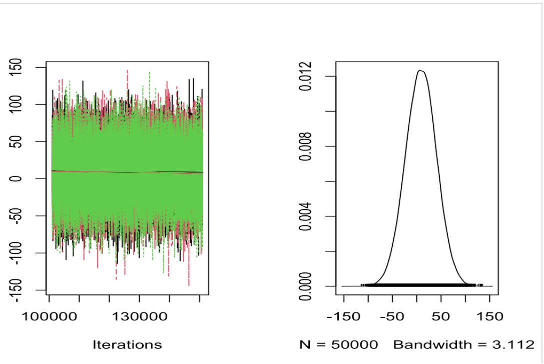


Figure 4: Plot showing the convergence of alpha 1, which is the intercept of the model. In other words, it represents the values of TDS without the influence of other predictor variables. As the plot illustrates, the model converged well but predicted negative values.

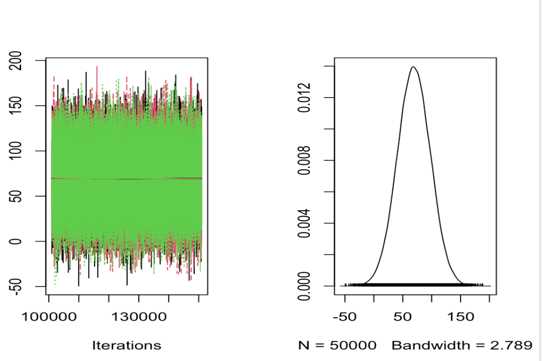


Figure 5: Plot showing the convergence of alpha 2, which is the coefficient for percent cover by urban area. As the plot illustrates, the model converged well but predicted negative values.

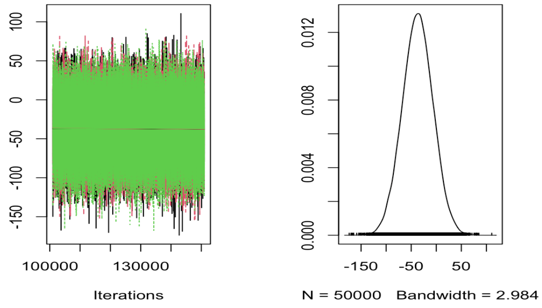


Figure 6: Plot showing the convergence of alpha 3, which is the coefficient for percent cover by agricultural area. As the plot illustrates, the model converged well but predicted negative values.

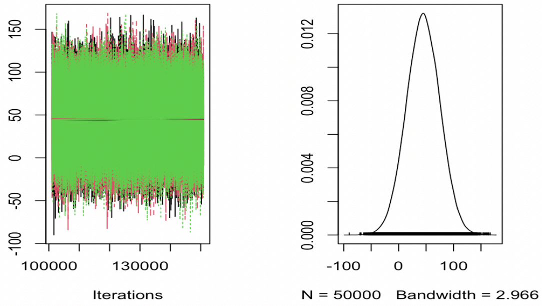


Figure 7: Plot showing the convergence of alpha 4, which is the coefficient for percent cover by impervious surfaces. As the plot illustrates, the model converged well but predicted negative values.

ii. TDS: Without Covariance Matrix and Spatial Random Effect

Zero was not contained within alphas 1, 2, 3, or 4 for this version of the model. However, none of the alphas showed evidence of convergence when plotted. The convergence of each alpha value can be seen in figures 8 through 11.

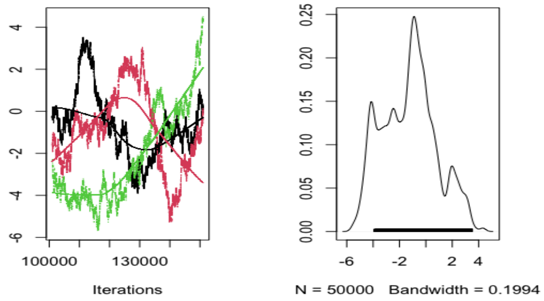


Figure 8: Plot showing the convergence of alpha 1, which is the intercept of the model, without the inclusion of the covariance matrix or spatial random effect. In other words, it represents the values of TDS without the influence of other predictor variables. Clearly this version of the model did not converge at all, indicating that the inclusion of a spatial random effect may help the model to converge if not to predict more accurately.

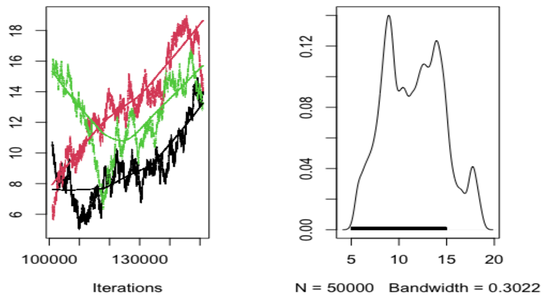


Figure 9: Plot showing the convergence of alpha 2, which is the coefficient for percent cover by urban area, without the inclusion of the covariance matrix or spatial random effect. As the plot illustrates, this version of the model did not converge at all. This suggests that the inclusion of a spatial random effect may help the model to converge if not to predict more accurately.

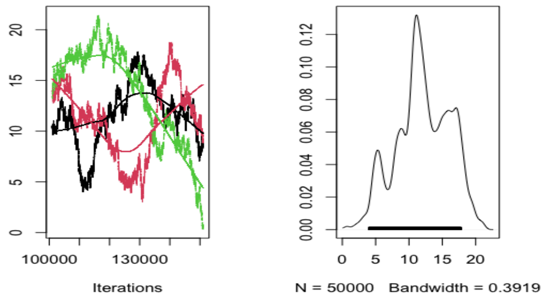


Figure 10: Plot showing the convergence of alpha 3, which is the coefficient for percent cover by agricultural area, without the inclusion of the covariance matrix or spatial random effect. As the plot illustrates, this version of the model also did not converge at all. This suggests that the inclusion of a spatial random effect may help the model to converge if not to predict more accurately.

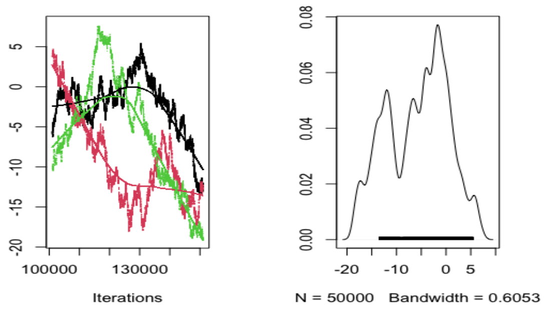


Figure 11: Plot showing the convergence of alpha 4, which is the coefficient for percent cover by impervious surfaces, without the inclusion of the covariance matrix or spatial random effect. As the plot illustrates, this version of the model also did not converge at all. This suggests that the inclusion of a spatial random effect may help the model to converge if not to predict more accurately.

iii. TSS: Covariance Matrix and Spatial Random Effect

Zero is contained within alphas 1, 2, 3, and 4. This indicates that none of the predictors are statistically significant at estimating TSS. The convergence of each alpha value can be seen in figures 12 through 15.

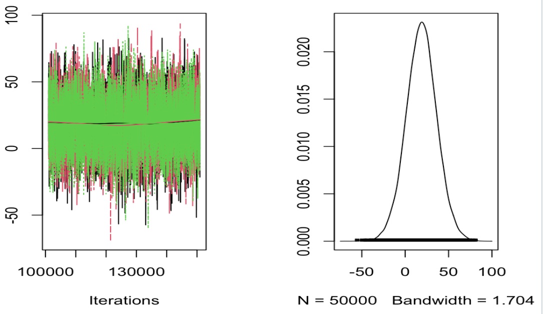


Figure 12: Plot showing the convergence of alpha 1, which is the intercept of the model. In other words, it represents the values of TSS without the influence of other predictor variables. As the plot illustrates, the model converged well but predicted negative values.

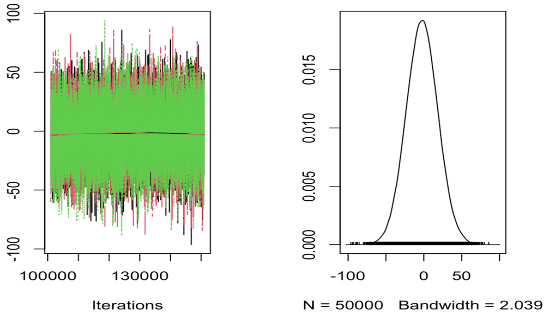


Figure 13: Plot showing the convergence of alpha 2, which is the coefficient for percent cover by urban area. As the plot illustrates, the model converged well but predicted negative values.

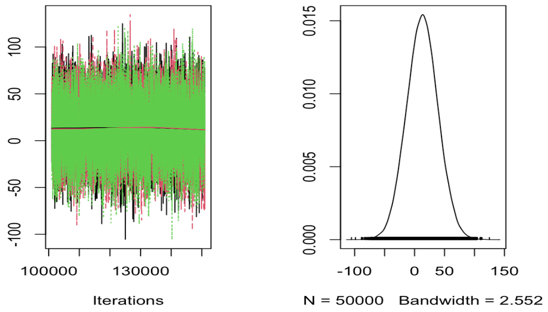


Figure 14: Plot showing the convergence of alpha 3, which is the coefficient for percent cover by agricultural area. As the plot illustrates, the model converged well but predicted negative values.

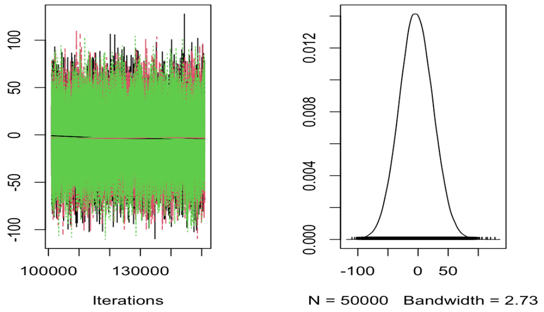


Figure 15: Plot showing the convergence of alpha 4, which is the coefficient for percent cover by impervious surfaces. As the plot illustrates, the model converged well but predicted negative values.

iv. TSS: Without Covariance Matrix and Spatial Random Effect

Zero was not contained within alphas 1, 2, or 3, indicating that these are statistically significant predictors. Zero was contained within alpha 4, suggesting that it is not statistically significant. The convergence of each alpha value can be seen in figures 16 through 19.

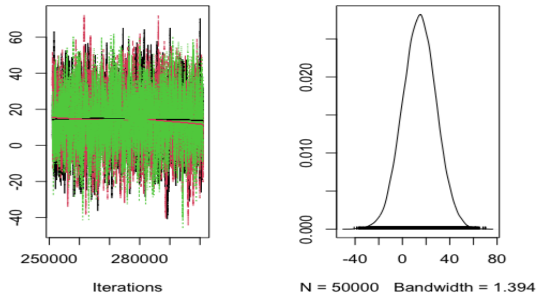


Figure 16: Plot showing the convergence of alpha 1, which is the intercept of the model, without the inclusion of the covariance matrix or spatial random effect. In other words, it represents the values of TSS without the influence of other predictor variables. As the plot illustrates, the model converged well but predicted negative values.

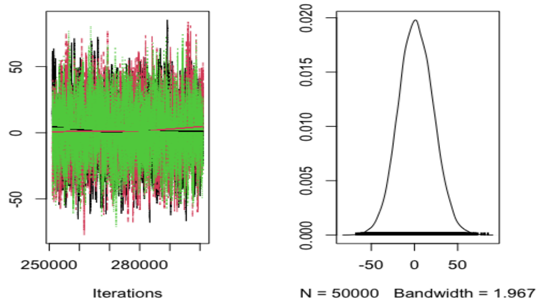


Figure 17: Plot showing the convergence of alpha 2, which is the coefficient for percent cover by urban area, without the inclusion of the covariance matrix or spatial random effect. As the plot illustrates, the model converged well but predicted negative values.

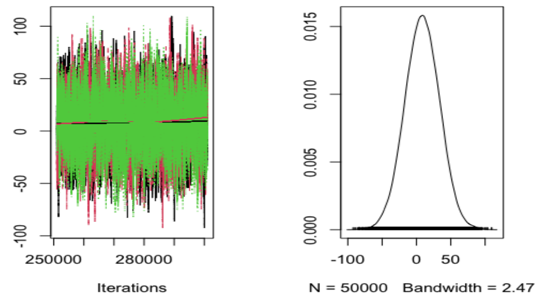


Figure 18: Plot showing the convergence of alpha 3, which is the coefficient for percent cover by agricultural area, without the inclusion of the covariance matrix or spatial random effect. As the plot illustrates, the model converged well but predicted negative values.

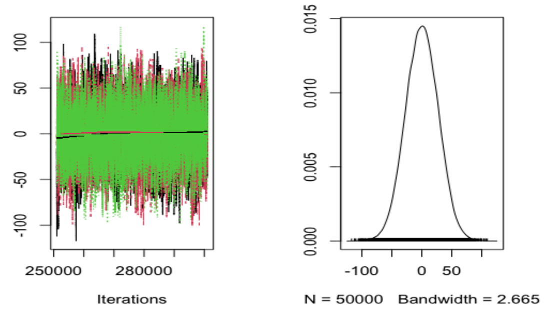


Figure 19: Plot showing the convergence of alpha 4, which is the coefficient for percent cover by impervious surfaces, without the inclusion of the covariance matrix or spatial random effect. As the plot illustrates, the model converged well but predicted negative values.

v. Total Phosphorus: Covariance Matrix and Spatial Random Effect

Zero was contained within each of the alpha values, meaning that they could be equal to zero and are therefore not statistically significant, regardless of the presence of the covariance matrix and spatial random effect. This suggests that the spatial component of the model does help the model predict more effectively and thus does not need to be included. The convergence of each alpha value can be seen in figures 20 through 23.

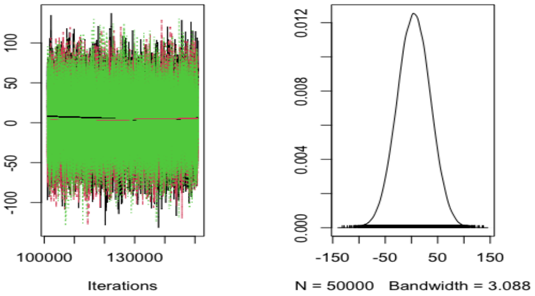


Figure 20: Plot showing the convergence of alpha 1, which is the intercept of the model. In other words, it represents the values of total phosphorus without the influence of other predictor variables. As the plot illustrates, the model converged well but predicted negative values.

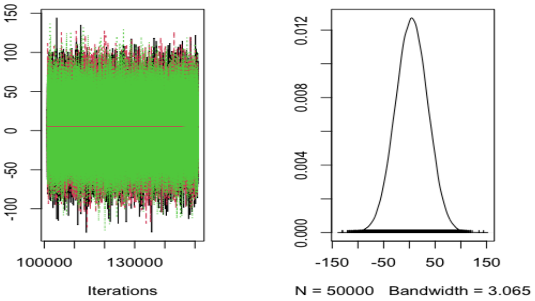


Figure 21: Plot showing the convergence of alpha 2, which is the coefficient for percent cover by urban area. As the plot illustrates, the model converged well but predicted negative values.

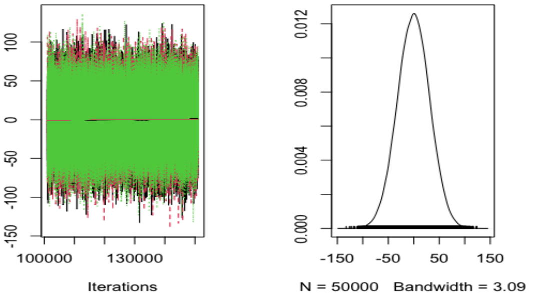


Figure 22: Plot showing the convergence of alpha 3, which is the coefficient for percent cover by agricultural area. As the plot illustrates, the model converged well but predicted negative values.

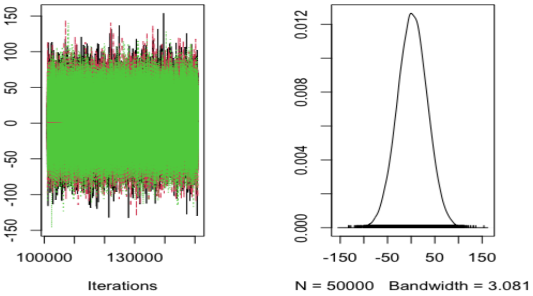


Figure 23: Plot showing the convergence of alpha 4, which is the coefficient for percent cover by impervious surfaces. As the plot illustrates, the model converged well but predicted negative values.

vi. Total Phosphorus: Without Covariance Matrix and Spatial Random Effect

In this version of the model zero was not contained within alphas 1,2, or 3. This suggests that they are statistically significant. Zero was contained within alpha 4, the coefficient for impervious surfaces. This indicates that this predictor is not statistically significant. The convergence of each alpha value can be seen in figures 24 through 27.

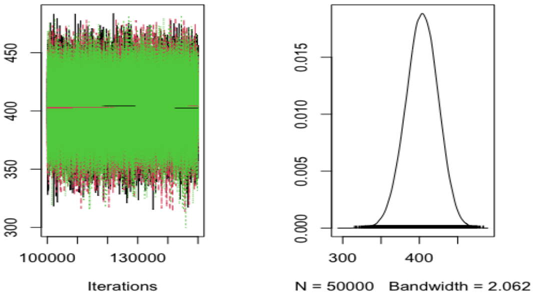


Figure 24: Plot showing the convergence of alpha 1, which is the intercept of the model, without the inclusion of the covariance matrix or spatial random effect. In other words, it represents the values of total phosphorus without the influence of other predictor variables. As the plot illustrates, the model converged well and is not predicting negative values.

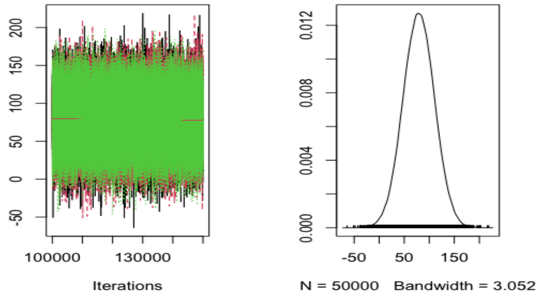


Figure 25: Plot showing the convergence of alpha 2, which is the coefficient for percent cover by urban area, without the inclusion of the covariance matrix or spatial random effect. As the plot illustrates, the model converged well but predicted negative values.

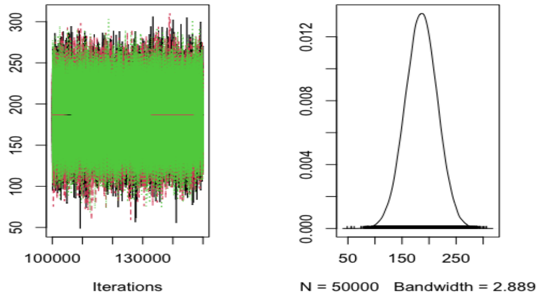


Figure 26: Plot showing the convergence of alpha 3, which is the coefficient for percent cover by agricultural area, without the inclusion of the covariance matrix or spatial random effect. The model converged well and is not predicting negative values.

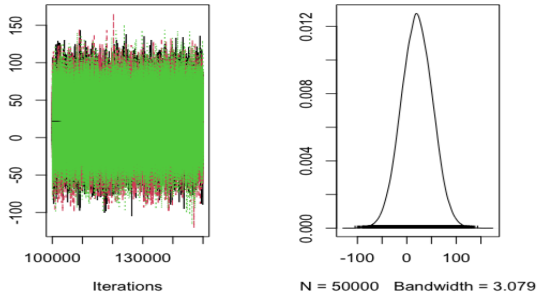


Figure 27: Plot showing the convergence of alpha 4, which is the coefficient for percent cover by impervious surfaces, without the inclusion of the covariance matrix or spatial random effect. As the plot illustrates, the model converged well but predicted negative values.

vii. Nitrate: Covariance Matrix and Spatial Random Effect

Zero is contained within alphas 1,2, and 3, indicating that they are not statistically significant. Zero is not contained within alpha 4, the beta coefficient for percent cover by impervious surfaces. Without the covariance matrix or spatial random effect zero was contained within all four alphas. The convergence of each alpha value can be seen in figures 28 through 31.

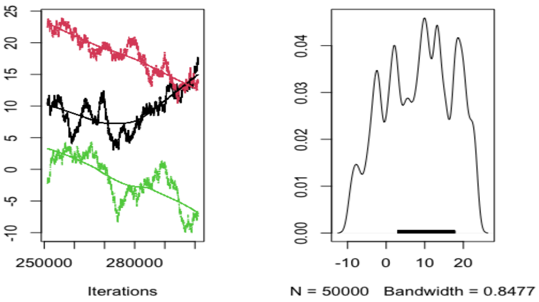


Figure 28: Plot showing the convergence of alpha 1, which is the intercept of the model. In other words, it represents the values of nitrate without the influence of other predictor variables. This model did not converge.

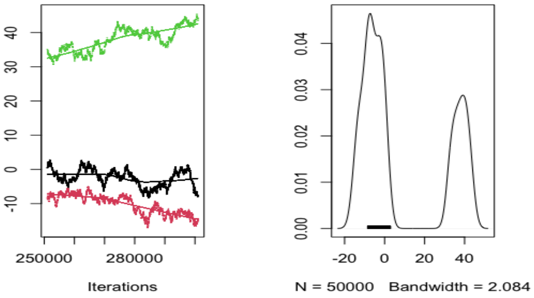


Figure 29: Plot showing the convergence of alpha 2, which is the coefficient for percent cover by urban area. This model did not converge.

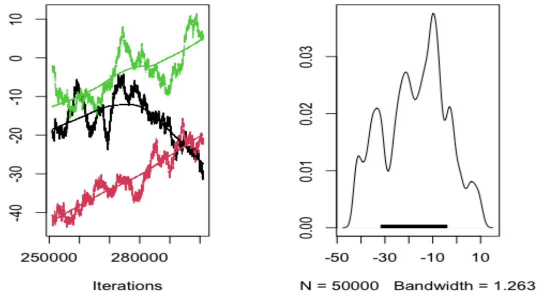


Figure 30: Plot showing the convergence of alpha 3, which is the coefficient for percent cover by agricultural area. This model did not converge.

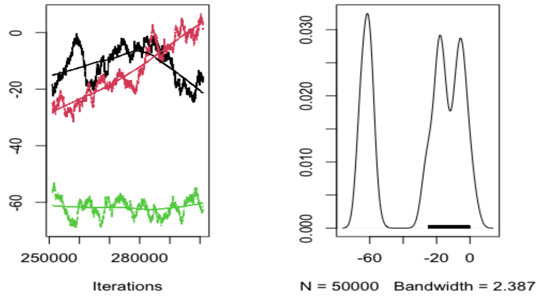


Figure 31: Plot showing the convergence of alpha 4, which is the coefficient for percent cover by impervious surfaces. This model did not converge.

viii. Nitrate: Without Covariance Matrix and Spatial Random Effect

Zero is contained within all four alphas. The convergence of each alpha value can be seen in figures 32 through 35.

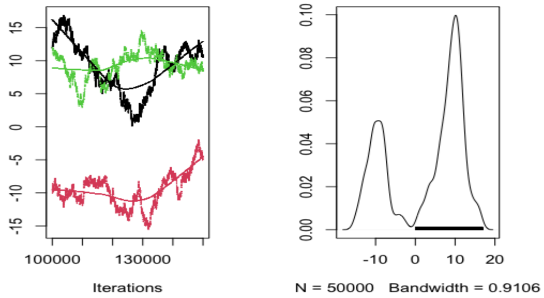


Figure 32: Plot showing the convergence of alpha 1, which is the intercept of the model, without the inclusion of a covariance matrix or spatial random effect. In other words, it represents the values of nitrate without the influence of other predictor variables. This model did not converge.

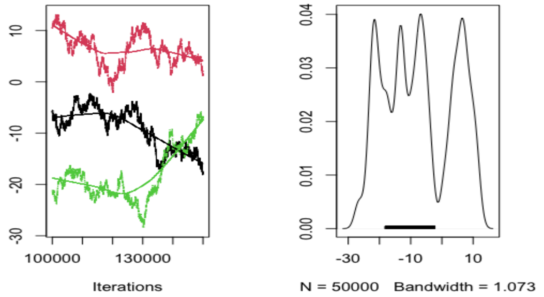


Figure 33: Plot showing the convergence of alpha 2, which is the coefficient for percent cover by urban area, without the inclusion of a covariance matrix or spatial random effect. This model did not converge.

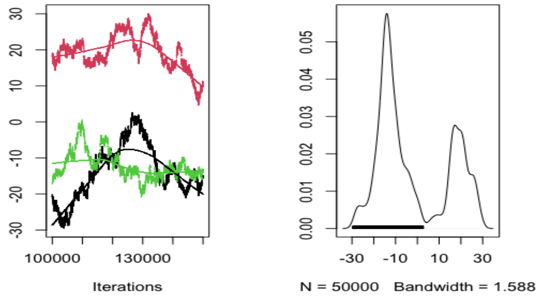


Figure 34: Plot showing the convergence of alpha 3, which is the coefficient for percent cover by agricultural area, without the inclusion of a covariance matrix or spatial random effect. This model did not converge.

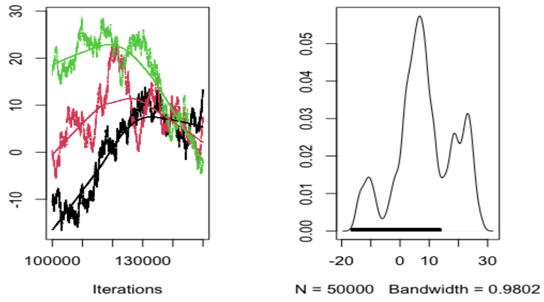


Figure 35: Plot showing the convergence of alpha 4, which is the coefficient for percent cover by impervious surfaces, without the inclusion of a covariance matrix or spatial random effect. This model did not converge.

ix. Nitrite: Covariance Matrix and Spatial Random Effect

Zero is contained within all four alphas with or without covariance matrix and spatial random effect. The convergence of each alpha value can be seen in figures 36 through 39.

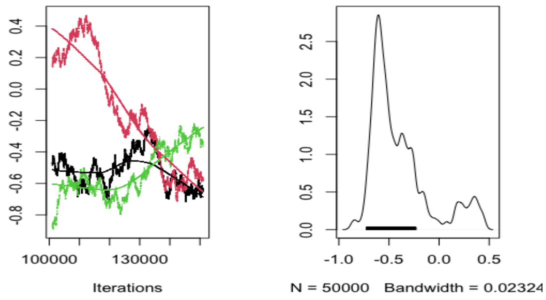


Figure 36: Plot showing the convergence of alpha 1, which is the intercept of the model. In other words, it represents the values of nitrite without the influence of other predictor variables. This model did not converge.

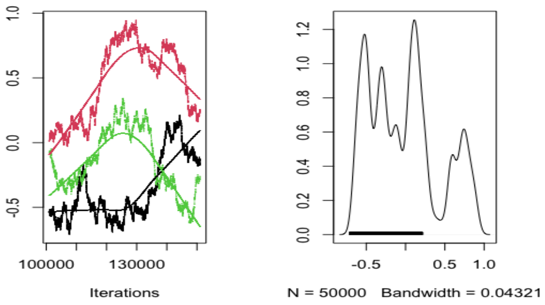


Figure 37: Plot showing the convergence of alpha 2, which is the coefficient for percent cover by urban area. This model did not converge.

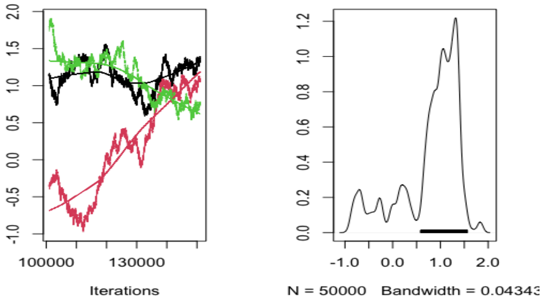


Figure 38: Plot showing the convergence of alpha 3, which is the coefficient for percent cover by agricultural area. This model did not converge.

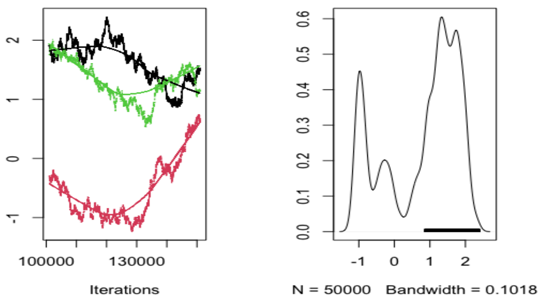


Figure 39: Plot showing the convergence of alpha 4, which is the coefficient for percent cover by impervious surfaces. This model did not converge.

x. Nitrite: Without Covariance Matrix and Spatial Random Effect

As stated above, zero is contained within all 4 alphas. The convergence of each alpha value can be seen in figures 40 through 43.

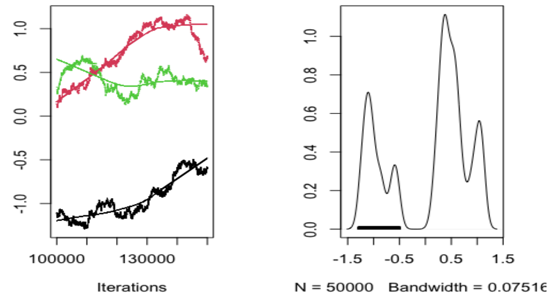


Figure 40: Plot showing the convergence of alpha 1, which is the intercept of the model, without the inclusion of a covariance matrix or spatial random effect. In other words, it represents the values of nitrite without the influence of other predictor variables. This model did not converge.

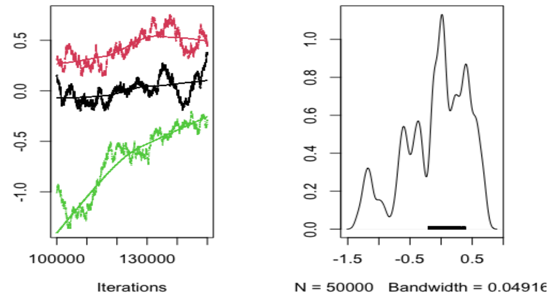


Figure 41: Plot showing the convergence of alpha 2, which is the coefficient for percent cover by urban area, without the inclusion of a covariance matrix or spatial random effect. This model did not converge.

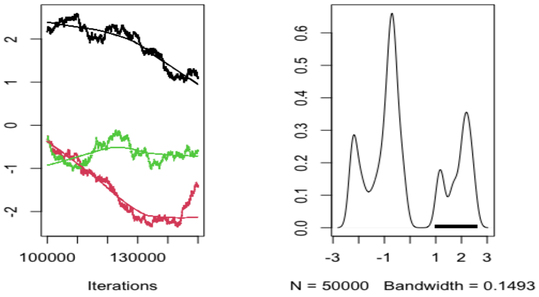


Figure 42: Plot showing the convergence of alpha 3, which is the coefficient for percent cover by agricultural area, without the inclusion of a covariance matrix or spatial random effect. This model did not converge.

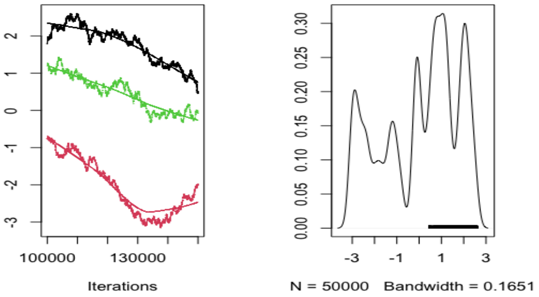


Figure 43: Plot showing the convergence of alpha 4, which is the coefficient for percent cover by impervious surfaces, without the inclusion of a covariance matrix or spatial random effect. This model did not converge.

b. Exploratory Analysis Results

i. Two-Sample t-test Results

A series of two sample t-tests using a 95% confidence interval were performed as part of a separate but related project, comparing the average levels of each parameter between Mill and Allen Creek (Table 1). The t-test for Total Phosphorus returned a p-value of .1366, which was greater than the alpha value of .05, indicating that there was not sufficient evidence to reject the null hypothesis that the difference between the mean concentration of total phosphorus in the two creek sheds was equal to zero. The only other t-test that returned a p-value greater than .05 was the one testing the difference in the average levels of TSS between the creeksheds. This p-value was .2271. The t-tests for TDS, nitrate, and nitrite returned p-values of 2.333e-14, 5.315e-05, and .0003575 respectively. Each of these is less than the alpha value of .05, suggesting that there is sufficient evidence to reject the null hypothesis that the difference between the mean concentration of the parameters in the two creek sheds was equal to zero. As stated above, the initial expectation of this study was that the two creek sheds Mill and Allen Creek would display differences in their water quality outcomes as a result of the fact that they are so drastically different in their land use land cover compositions.

Parameter	P-value (alpha .05)	Ho rejected (yes/no)
Total Dissolved Solids	2.333e-14	yes
Total Suspended Solids	.2271	no
Total Phosphorus	.1366	no
Nitrate	5.315e-05	yes
Nitrite	.0003575	yes

Table 1: Table displaying the results of the two sample t-tests performed for each parameter, and whether or not a statistically significant difference was found between Mill and Allen Creek.

iii. Hierarchical Clustering Analysis Results

For the hierarchical clustering analysis, the initial resulting clusters were quite different in size. Cluster 1 contained 199 observations, while 2 and 3 contained 1 and 5, respectively. The cluster means for TDS were 629, 284, 393, respectively. The cluster means for TP were .050, .53, .27. For TSS they were 8, 222, and 109. For Nitrate they were .76, 6, and 2. Lastly, for Nitrite they are .011, .063, and .036. In an attempt to produce more congruent clusters, the analysis was repeated with five clusters instead. This produced clusters of 116, 39, 20, 4, and 26, respectively (Appendix D). The cluster means for TDS were 571, 417, 1055, 82, 905 respectively. The cluster means for TP were .050, .072, .063, .085, and .060. For TSS they were 10, 19, 6, 29, and 6. For Nitrate they were .73, 1, .72, .53, and .61. Lastly, for Nitrite they are .010, .013, .016, .01 and .02. When looking more closely at the contents of each cluster, I discovered that observations from station MH02B, located in Mill Creek, tended to fall within the same clusters as observations from the Allen Creek station MH04. MH02B is a USGS gage located in Scio Township (HRWC, n.d). It is possible that Scio Township is one of the more developed areas within the Mill Creek subwatershed, which could result in this monitoring station displaying similar water quality outcomes to highly developed Allen Creek.

iv. Principal Analysis Components Results

The PCA resulted in nine principal components. They progressively accounted for less and less of the variance in the data, indicating that some of the later principal components can probably be excluded. 46.6% of the variance was explained by principal component 1, 77.8% by principal components 1 and 2, and 85.7% by 1,2, and 3 (Figure 48). The standard deviations of each of the components also gradually decreased, beginning with 2.05 and ending with 0. The contribution of each variable to principal components 1 and 2 is captured by the biplot below (Figure 49). This plot only includes the first two principal components because most of the variance in the data can be explained without the inclusion of the other seven. The loadings of the PCA are represented by the red arrows, and the points represent the water quality observations.

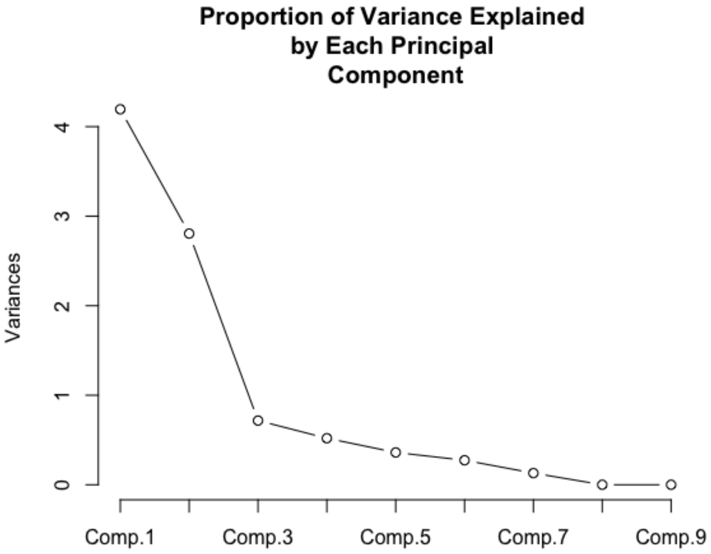


Figure 44: Scree plot showing the proportion of variance explained by each principal component. As one can see, the majority of the variance in the data is explained by the first three principal components. The additional variance that is explained by adding each subsequent component sequentially decreases.

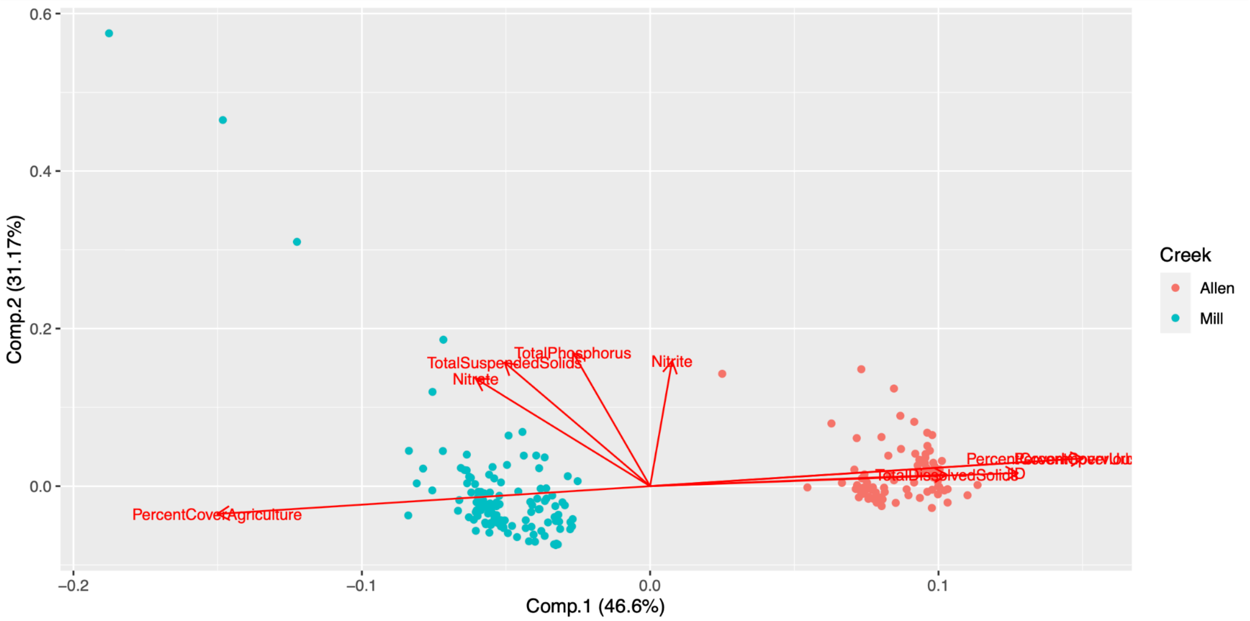


Figure 45: Biplot showing the contribution of each variable to each principal component, as well as the relationships between variables and data points.

v. Logistic Regression Results

1. TDS:

The predicted probabilities plot for TDS and percent cover by urban area, which was created from the results of the logistic regression model for TDS, yielded very low predicted probabilities on the y-axis. According to the results of this analysis, the probability of TDS exceeding its pre-established EPA limit remains at zero until urban cover exceeds 75% (Figure 46). At this point, the plot indicates that the predicted probability of TDS exceeding its limit is $1.5\text{x}10^{-8}$. The fact that this number is so low suggests that the model predicts that the event of TDS exceeding its limit is very unlikely, even when percent coverage by urban area is high. Slightly higher predicted probabilities were evidenced in the plot for TDS and percent cover by agricultural area, but the relationship trended in the opposite direction (Figure 47). Between 0 and approximately 10% agricultural cover, the probability of TDS exceeding its limit dropped from .03 to 0, where it remained up to 100% agricultural cover. The predicted probability plot of the relationship between TDS and percent cover by impervious surfaces showed that predicted probability of TDS exceeding its limit was 1 up to about 10% cover by impervious surfaces (Figure 48). Past this point the predicted probability begins to drop until it reaches 0 at around 20% impervious cover. This is contrary to what I might have expected.

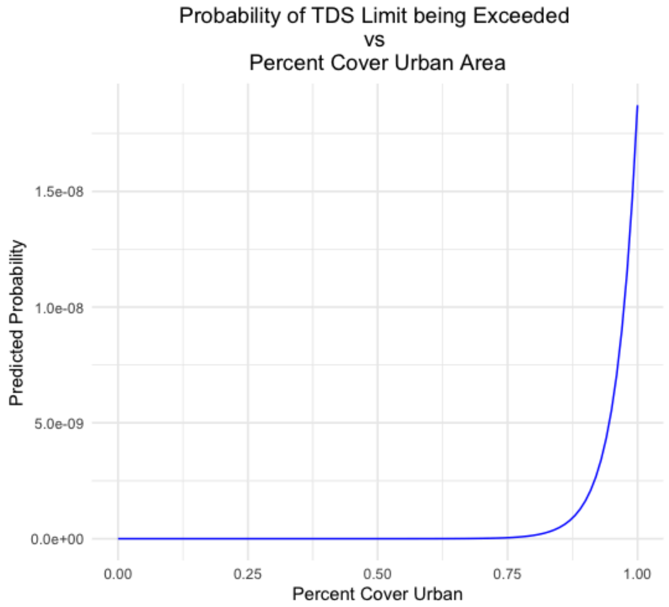


Figure 46: Predicted probabilities plot showing the likelihood of TDS exceeding its EPA limit as percent cover by urban land increases. The probability of exceedance is 0 until 75% urban cover is reached. At this point it increases but only to $1.5\text{x}10^{-8}$.

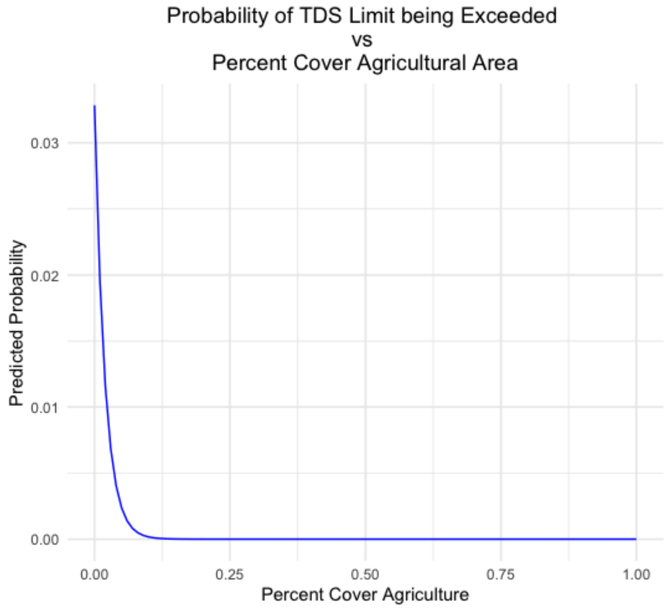


Figure 47: Predicted probabilities plot showing the likelihood of TDS exceeding its EPA limit as percent cover by agricultural land increases. When there is no agricultural cover the probability of the limit being exceeded is .03. This quickly drops to 0 at about 10% agricultural cover.

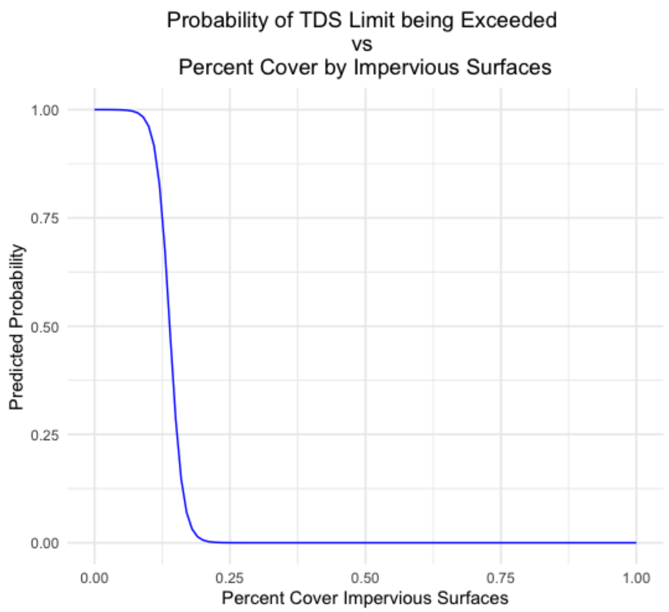


Figure 48: Predicted probabilities plot showing the likelihood of TDS exceeding its EPA limit as percent cover by impervious surfaces increases. The probability of exceedance is 1 when there is no impervious cover, but this begins to drop quickly before reaching 0 at approximately 20% impervious cover.

2. TSS:

The predicted probability of TSS exceeding its established EPA limit with respect to percent cover by urban area follows a logarithmic pattern (Figure 49). The predicted probability of TSS exceeding its limit reaches 1 at about 75% urban cover, at which point the curve flattens. The predicted probability of TSS exceeding its limit with respect to agricultural cover follows an "S" shaped curve (Figure 50). The probability starts to increase at around 10% agricultural cover, then reaches 1 and plateaus at about 25% agricultural cover. The predicted probabilities plot for TSS relative to impervious cover also follows a logarithmic pattern, reaching 1 and plateauing at approximately 25% impervious cover (Figure 51).

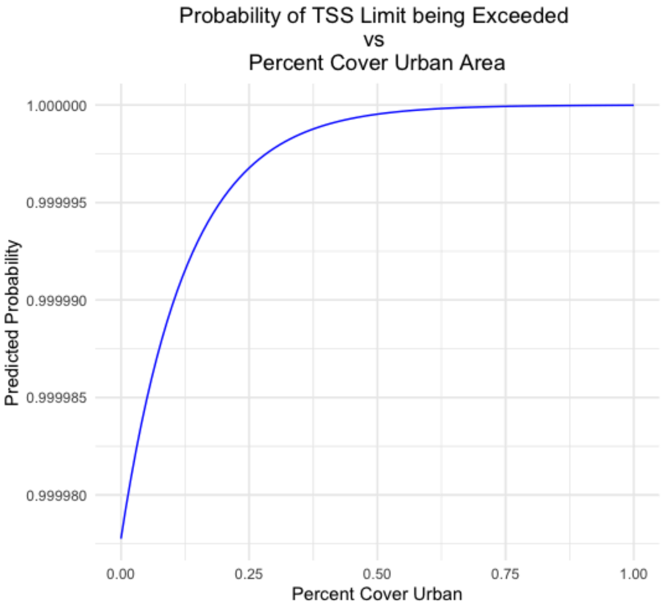


Figure 49: Predicted probabilities plot showing the likelihood of TSS exceeding its EPA limit as percent cover by urban area increases. The probability of exceedance is near 0 when there is no urban cover, but this begins to increase before reaching 1 at approximately 75% urban cover.

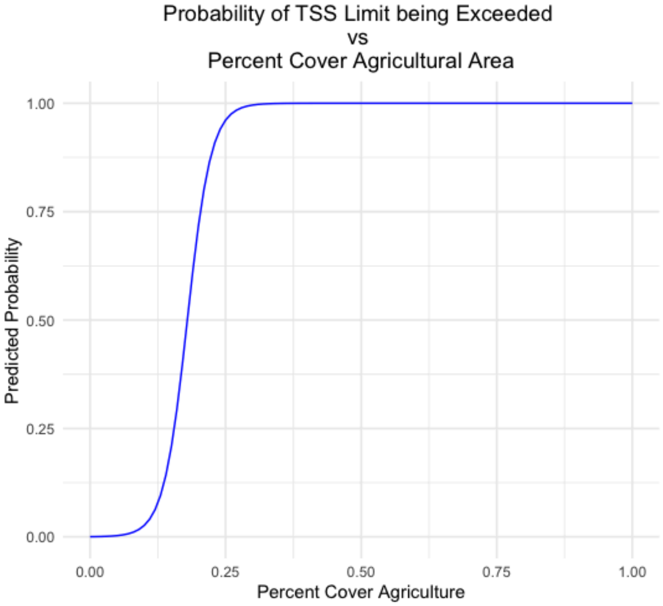


Figure 50: Predicted probabilities plot showing the likelihood of TSS exceeding its EPA limit as percent cover by agricultural land increases. The probability of exceedance is 0 when there is no agricultural cover, begins to increase at roughly 10% cover and reaches 1 at approximately 25%.

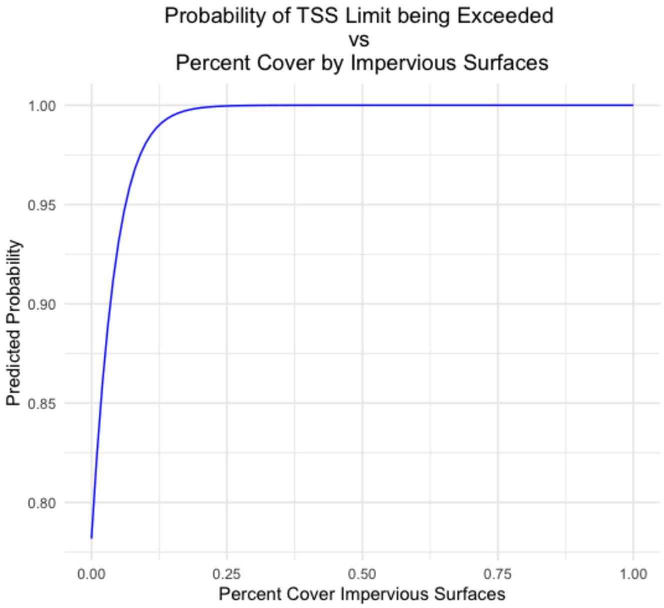


Figure 51: Predicted probabilities plot showing the likelihood of TSS exceeding its EPA limit as percent cover by impervious surfaces increases. The probability of exceedance is near 0 when there is no impervious cover, but this begins to increase quickly before reaching 1 at approximately 20% impervious cover.

3. Total Phosphorus:

The predicted probabilities curve showing the likelihood of total phosphorus exceeding its EPA limit with respect to percent urban cover follows an “S” shape, with an increase in the probability of exceedance beginning at 25% urban cover and culminating at approximately .9 probability of exceedance at 100% urban cover (Figure 52). For percent cover by agricultural land, a similar pattern is observed although it is slightly less pronounced (Figure 53). At 100% urban cover the logistic regression predicts an .85 probability of total phosphorus exceeding its limit. A negative relationship was observed between percent impervious cover and the probability of total phosphorus exceeding its limit (Figure 54). At 100% impervious cover the predicted probability of total phosphorus exceeding its limit is 0.

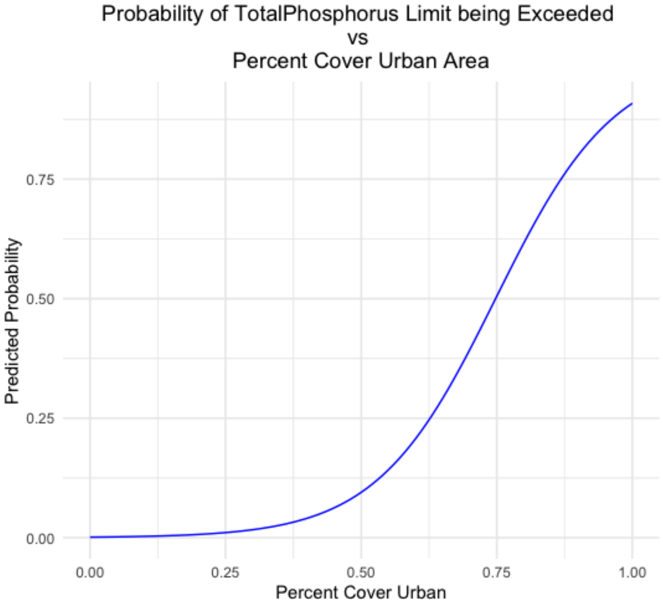


Figure 52: Predicted probabilities plot showing the likelihood of total phosphorus exceeding its EPA limit as percent cover by urban area increases. The probability of exceedance is 0 when there is no urban cover, but this begins to increase and reaches approximately .9 at 100% urban cover.

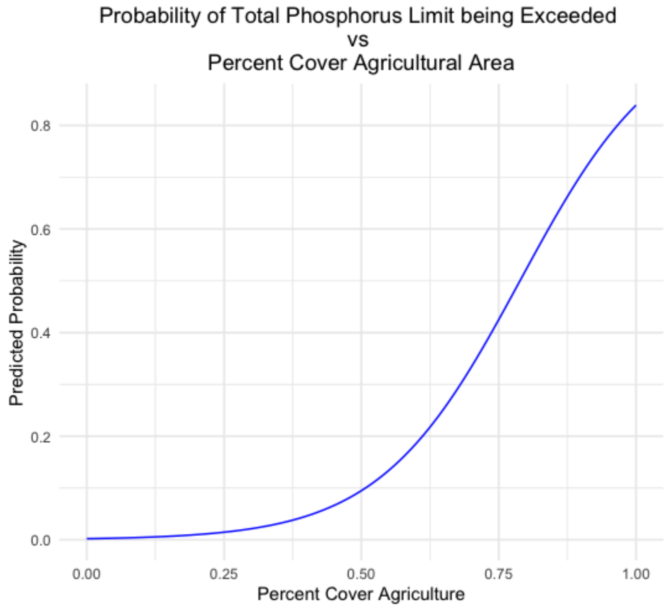


Figure 53: Predicted probabilities plot showing the likelihood of total phosphorus exceeding its EPA limit as percent cover by urban area increases. The probability of exceedance is 0 when there is no agricultural cover, but this begins to increase and reaches approximately .85 at 100% agricultural cover.

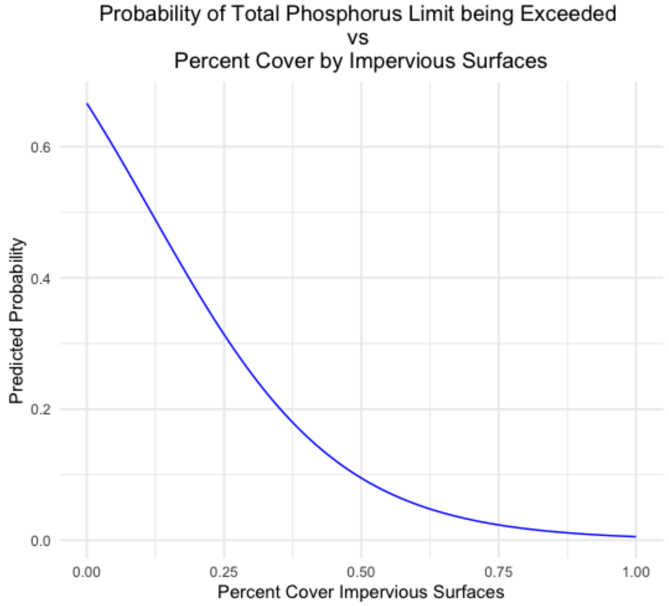


Figure 54: Predicted probabilities plot showing the likelihood of total phosphorus exceeding its EPA limit as percent cover by impervious surfaces increases. The probability of exceedance is roughly when there is no impervious cover, but this begins to decrease and reaches 0 at 100% impervious cover.

4. Nitrite:

The predicted probability of nitrite exceeding its EPA limit stagnates at 0 until roughly 62.5% urban cover is achieved (Figure 55). At this point a steady increase begins until a probability of 1 is reached at about 82.5% urban cover. The predicted probability of nitrite exceeding its limit also remains at 0 until about 62.5% agricultural cover is reached, at which point a steep increase is observed, culminating in a .8 probability of exceedance at 100% agricultural cover (Figure 56). The predicted probability curve for impervious cover is unique in that it indicates that the probability of nitrite exceeding its EPA limit is 1 up until right before 25% impervious cover is reached. Beyond 25% impervious cover the curve drops steeply until it reaches 0 at approximately 40% impervious cover (Figure 57).

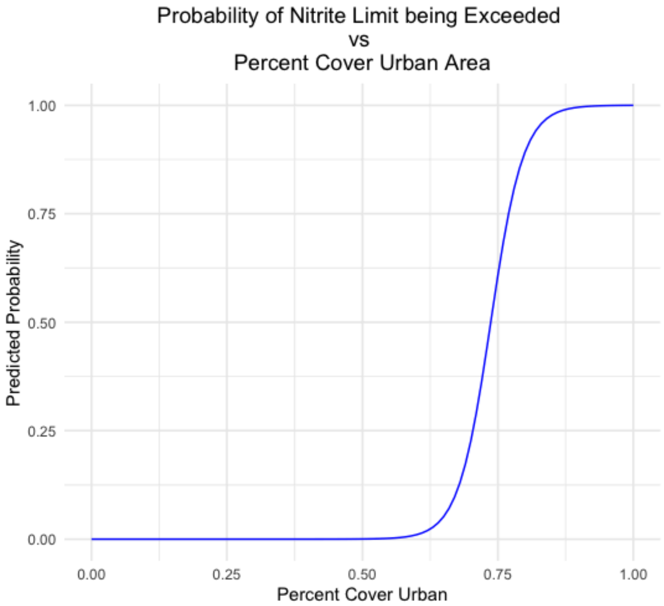


Figure 55: Predicted probabilities plot showing the likelihood of nitrite exceeding its EPA limit as percent cover by urban area increases. The probability of exceedance is 0 until 62.5% cover, where it begins to increase until reaching 1 at 82.5% cover.

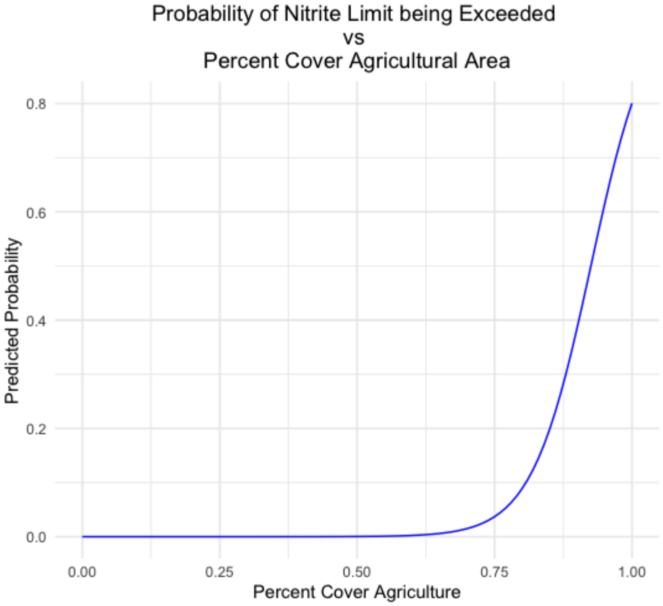


Figure 56: Predicted probabilities plot showing the likelihood of nitrite exceeding its EPA limit as percent cover by agricultural area increases. The probability of exceedance is 0 until 62.5% agricultural cover is reached, at which it increases to .8 at 100% agricultural cover.

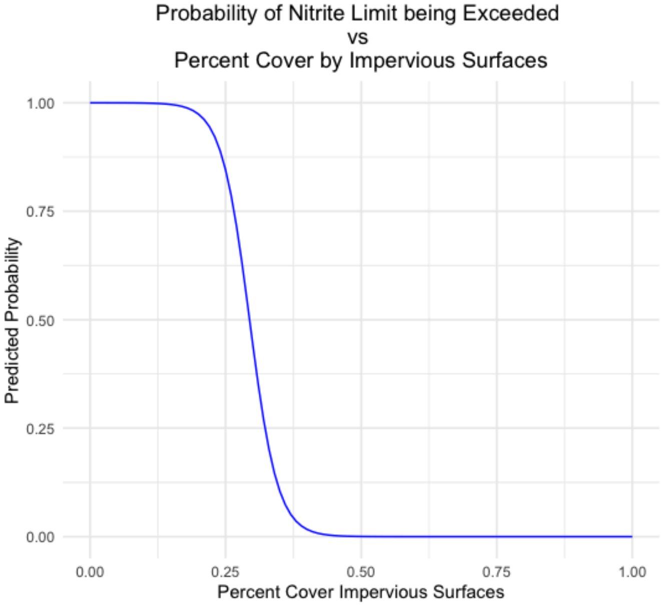


Figure 57: Predicted probabilities plot showing the likelihood of nitrite exceeding its EPA limit as percent cover by agricultural area increases. The probability of exceedance is 1 until just before 25% impervious cover is reached, at which it begins to decrease until reaching 0 at roughly 40% cover.

5. Nitrate:

The predicted probability of nitrate exceeding its EPA limit remains at 2.900701x10^{-12} regardless of percent urban cover. Although the plot resembles a negative linear curve. This is a very small value, suggesting that the logistic regression does not predict the event of nitrate exceeding its limit to be very likely. An identical relationship was observed between the predicted probability of exceedance and percent agricultural cover. Interestingly, the plot for impervious cover reported the same predicted probability but appears as a positive linear curve. These results are not meaningful. This would indicate that a relationship between these land use variables and the likelihood of nitrate exceeding its EPA limit does not exist, but this result would be surprising given the relationships observed among the other parameters. Nitrite, for example, did display a relationship with each of these variables. Nitrate and nitrite are both nitrogen compounds (Moorcroft et al., 2001), but they are produced through different processes. This could result in different relationships with land use. The specific locations from which the nitrate samples were collected, as well as when they were collected, could also be impacting these results. The predicted probabilities plots for these results can be found in Appendix E.

c. Predictive Models

i. Generalized Additive Model Results

The GAM model results yielded fairly low or even negative R^2 values, once again indicating that this may not be an adequate method of modeling this data. The MSEs also spanned a large range again. The highest R^2 value that was obtained was .30 for TDS, corresponding to a high MSE of 34,246. The R^2 for TSS was .003 and the MSE was also quite high at 814. The R^2 for total phosphorus was -.02. The MSE for this model was fairly low at .0036, but given the negative R^2 I would not consider this model a good fit for this data. The R^2 for nitrate was also very low at .013, despite a low MSE of .00015. Lastly, the R^2 for nitrite was .075 and the MSE was .66, once again suggesting that these models do not fit these data well (Table 2).

Parameter	**R^2**	**Mean Squared Error (MSE)**
Total Dissolved Solids	.30	34,246
Total Suspended Solids	.003	814
Total Phosphorus	-.02	.0036
Nitrate	.013	.00015
Nitrite	.075	.66

Table 2: Table summarizing the performances of the generalized additive models for each parameter based on R^2 and Mean Squared Error (MSE).

ii. Support Vector Machine Model Results

The highest R^2 value obtained from the SVM regressions was .32, corresponding to TDS. The other values for R^2 were very low, indicating that SVM regression does not explain very much of the variance in this data. There was also quite a large range in the values for MSE, with the highest value being 32,897 and the lowest being .00013. Given the scale of TDS and TSS, MSEs of 32,897 and 536 are considered quite high. This further indicates that the models may not be predicting accurately, especially coupled with the low R^2 of .022 for TSS. The R^2 values for total phosphorus, nitrate, and nitrite were also very low at .00013, .03, and .06 respectively. The MSE values for these models were .0037, .00013, and .57 respectively.

Parameter	R^2	Mean Squared Error (MSE)
Total Dissolved Solids	.32	32,897
Total Suspended Solids	.022	536
Total Phosphorus	.00013	.0037
Nitrate	.03	.00013
Nitrite	.06	.57

Table 3: Table summarizing the performances of the support vector machine models for each parameter based on R^2 and Mean Squared Error (MSE).

iii. Random Forest Model Results

The random forest regressions returned the highest R^2 of any of the methods presented here, reaching as high as .93 for TSS (Table 4). The MSE for the TSS model was still relatively high, though, at 571. The model for total phosphorus returned an R^2 of .63, indicating that it explains a sufficient amount of the variance in the data. The R^2 values for the other parameters were between .2 and .25, suggesting that RFM regression may not be a good method for modeling these parameters despite the fact that they mostly return very low values for MSE. TDS is the exception to this, having an MSE of 40,563. The TDS model also had a mediocre R^2 at .25. Nitrate and nitrite returned R^2 values of .2 and .21 respectively with MSEs of .00013 each.

Parameter	R^2	Mean Squared Error (MSE)
Total Dissolved Solids	.25	40,563
Total Suspended Solids	.93	571
Total Phosphorus	.63	.0040
Nitrate	.20	.00013
Nitrite	.21	.00013

Table 4: Table summarizing the performances of the random forest models for each parameter based on R^2 and Mean Squared Error (MSE).

iv. Neural Network Models Results

All the R^2 values for the neural network regressions were either negative or just dismally low. The R^2 values for TDS, TSS, total phosphorus, and nitrate were -5.8e-15, -1.8e-07, -.0011, and -.0081 respectively. The MSEs for these models were 47,380, 524, .0035, and .00012. The only model with a positive R^2 of .12 was the one corresponding to nitrite. Although positive, this is still a pretty low R^2 indicating that the model does not explain a significant portion of the data. The MSE for this model is .50.

Parameter	**R^2**	**Mean Squared Error (MSE)**
Total Dissolved Solids	-5.8e-15	47,380
Total Suspended Solids	-1.8e-07	524
Total Phosphorus	-.0011	.0035
Nitrate	-.0081	.00012
Nitrite	.12	.50

Table 5: Table summarizing the performances of the neural network models for each parameter based on R^2 and Mean Squared Error (MSE).

5. Discussion:

The goal of this study was to examine the relationship between land use and water quality in the Huron River Watershed, with a focus on Mill and Allen Creek. Mill and Allen Creek are both tributaries in the Middle Huron, but they are characterized by very different patterns of land use. This paper hypothesized that the LULC composition of a creekshed would impact the quality of runoff in that creekshed, and thus Mill and Allen Creek would experience different water quality outcomes. Specifically, I wanted to investigate how variations in percent cover by urban area, agricultural area, and impervious surfaces would influence Total Phosphorus, Total Dissolved Solids, Total Suspended Solids, Nitrates, and Nitrites. This study attempted to create a model that could predict concentrations of total phosphorus, total dissolved solids, total suspended solids, and nitrates/nitrites in the Huron River Watershed based on land use and land cover. A variety of predictive methods were experimented with, to varying degrees of success.

a. OpenBugs Models Discussion

i. TDS

The inclusion of the covariance matrix and spatial random effect in the model produced different results than the model that did not contain a covariance matrix or spatial random effect. In the model containing D alpha 2 was statistically significant while zero was contained within the other three alphas, indicating that they were not statistically significant. However, in the version of the model that did not include a covariance matrix or spatial random effect, zero was not contained within any of the four alphas, indicating that all of them were statistically significant. This begs the question of why when proximity between stations is not taken into account, percent cover by agriculture and impervious surfaces are significant predictors of TDS in addition to percent cover by urban/developed land. The latter of which was the only significant predictor when D was included. Despite the fact that only alpha 2 showed statistical significance, all the predictors seemed to converge surprisingly well. This means that they settled within a certain margin of error around some final value throughout repeated iterations of the model.

ii. TSS

All the alpha values for these models converged very well and showed a normal distribution. This indicates that the model does a good job at predicting and fits the data well. The presence or absence of the covariance matrix and spatial random effect did not have an impact on which predictors in the TSS were statistically significant, which was not observed for some of the other parameters. In the model without the covariance matrix or spatial random effect, zero is contained within all four alphas. The results for the model with the covariance matrix and spatial random effect were very similar, which suggests that these elements may not help the model predict TSS with greater accuracy. The similarity in the results may be due to the fact that both creeks are located within the Middle Huron and all of the monitoring stations that were included in the model are very close together, as evidenced by the extreme similarities in all of their latitude and longitude values. If this is the case, then the predictive power of the model could perhaps be better assessed by comparing tributaries that are located in different portions of the river. Doing so may provide a better understanding of whether or not the distance between stations plays a role in determining water quality.

iii. Total Phosphorus

As mentioned previously, the effects of the different points of measurement on water quality of the main branch were expected to be strongly spatially dependent. It was initially thought that each point would be affected by any points that are located upstream, but not by any points that are located downstream. Additionally, if a point was located within a tributary, it was assumed that it would not be affected by points upstream even if it is located in the southern portion of the river. For this reason, a distance matrix that described the spatial relationships of all the different points of measurement, as well as a matrix of dependency that described which points had effects on each other and which did not, was initially included in the model. This was used to create a spatial random effect for each station that was included in the model. However, as shown by the results above, the inclusion of the spatial component of the model did not help the model predict total phosphorus concentrations more accurately. None of the predictor variables were statistically significant. Zero was contained within the confidence intervals of all three alpha values, regardless of the spatial random effect. As a result, the distance matrix and spatial random effect were removed in favor of the simpler linear model. The results of this analysis indicate that percent cover of development and percent cover by agricultural land do not significantly influence total phosphorus levels, which contradicts the original hypothesis of this study. None of the predictors show any signs of converging. In other words they are not settling within a certain margin of error around some final value, which indicates that the model is likely not predicting with very much accuracy or that the data does not fit the model well.

To conclude, although phosphorus is an important indicator of river health, given that levels exceeding .1 mg/L are associated with poor water quality outcomes such as harmful algal blooms (Mainstone & Parr, 2002), the results of this study suggest that it may not be the best parameter to use to model any relationship that may exist between LULC and water quality. Based on the results of this analysis, neither the amount of development or agriculture within a watershed appears to be a good predictor of phosphorus within a river, and average levels of phosphorus are not substantially different between watersheds that are predominantly developed or agricultural tributaries.

iv. Nitrates/Nitrites

With the inclusion of the covariance matrix and spatial random effect, zero was contained within alphas 1, 2, and 3. These correspond to the intercept and the coefficients for percent cover by urban area and percent cover by agriculture, respectively. These results suggest that these predictors are not statistically significant. Zero was not contained within the coefficient for percent cover by impervious surfaces, indicating that this might be a statistically significant predictor for nitrate. The presence of the covariance matrix and spatial random effect did to an extent impact which predictors were or were not significant. The results of the model without these features showed zero contained within all four alphas, suggesting that none of them were statistically significant. Zero was contained within all four alphas for the nitrite model with and without the inclusion of the covariance matrix and spatial random effect. This suggests that these features do not assist in the predictive power of the model. This is probably once again due to the fact that the tributaries included in the analysis are so close together. Additionally, it is possible that percent cover variables by themselves are not sufficient predictors for these parameters.

b. Exploratory Analysis Discussion

i. Two Sample t-tests

The results of the two sample t-tests indicate that there are statistically significant differences between the average concentrations of TDS, nitrate, and nitrite between Mill and Allen Creek. There was not sufficient evidence to suggest a difference between the average concentrations of TSS and total phosphorus between the creeksheds. Mill Creek is more than half agricultural, while Allen Creek is almost completely developed. Phosphorus and nitrogen are common nutrients used in agricultural fertilizers, which likely contributes to the total phosphorus and nitrate/nitrite concentrations in Mill Creek. Upon further research, it was discovered that there are also several sources of phosphorus in urban settings that can end up in streams and rivers, including sewage and discharge from wastewater treatment plants (United States Geological Survey [USGS], 2018). Since Allen Creek is so developed, it contains a lot of impervious surfaces, which are conducive to high volumes of runoff that can deposit large amounts of this urban phosphorus into the river. The results of this study imply that agricultural and urban runoff may result in similar amounts of phosphorus deposits. The difference in impervious cover would also account for differences in TDS between the tributaries.

ii. Hierarchical Clustering Analysis

The hierarchical clustering analysis was intended to identify patterns in the data, with the hopes that there would be identifiable similarities and differences between observations from different monitoring stations. The results proposed three very incongruent clusters. I initially considered a number of possible explanations for these imbalances. For instance, I thought that perhaps that the data was naturally skewed or contained outliers, which could affect the clustering process. I also considered that this might just be the natural structure of the clusters in the data, that a large proportion of the observations were simply similar enough to each other that they fall within the same cluster. This explanation would prompt the question of what separates the remaining observations from the large, primary cluster and the additional smaller clusters from each other. Another possibility was that the linkage method I chose was not appropriate. With complete link clustering, which is the method that was employed in this analysis, the similarity of clusters is the similarity of their most dissimilar members (Manning, 2008). This linkage method is particularly sensitive to outliers, so the presence of a dissident value has the potential to dramatically influence the size of the clusters. If this were the case, using a different linkage method might result in more evenly distributed clusters (Manning, 2008).

I decided to take a closer look at which monitoring stations were most prevalent in each cluster in order to identify any discernible patterns in the grouping. What I found was that cluster 1 contained all but a single observation from Allen Creek, and several observations from Mill Creek. Cluster 2 contained a single observation from the Mill Creek station MH02B. Lastly, cluster 3 contained observations from three Mill Creek stations and the final observation from Allen Creek. As far as I can tell, where a sample was taken from does not influence which cluster it gets sorted into. It is possible that since both of these tributaries are located in the Middle Huron and are quite close to each other, their water quality outcomes are similar regardless of the fact that they have very different patterns of LULC.

iii. Principal Components Analysis

The results of the PCA provide valuable insight into the number of components needed to explain most of the variance in the data. Nine principal components were produced, but the first principal component alone accounts for approximately 46% of the variance. The first component also has a standard deviation of about 2.05, which also indicates that it explains a large proportion of the variance. By the sixth component, 98.6% of the variance is explained. However, this does not necessarily mean that it is efficient to retain that many components in the analysis. As we continue along the components, their importance gradually decreases. Component number seven, for instance, only accounts for approximately 1.45% of the variance, eight only accounts for $4.05x10^{-16}$%, and 9 accounts for none at all. For this reason there is really no benefit to including these components in the analysis. The first three components account for roughly 85.7%. Since the goal is to explain as much of the variance as possible with as few components as possible, I would only consider three components in my analysis. Although 77.7% of the data could be explained by only components 1 and 2, I would opt to include the third component in my analysis for good measure. This is a matter of personal preference, despite the fact that it may not strictly be necessary.

The biplot of the PCA demonstrated some interesting relationship between the variables and the principal components, as well as the variables and each other and the data points (Figure 50). The loadings of the PCA, represented by the red arrows, illustrate the strength and relationship between each variable and principal component. The greater the magnitude of the arrow, the greater its contribution to the principal components (Mackiewicz & Ratajcak, 1992). The relationships between the variables themselves are indicated by the angles between the arrows. Smaller angles suggest strong positive relationships between variables.

As the biplot shows, there is a strong relationship between percent urban cover, percent cover by impervious surfaces, and TDS. Station ID was also closely correlated with these variables. The relationship between percent urban cover and percent cover by impervious surfaces is expected due to the fact that urban areas are characterized by high levels of imperviousness. The close association of TDS with these variables suggests that they correlate more strongly with this parameter than the other four. Following TDS, these land use variables display the strongest relationship with nitrite, then total phosphorus, TSS, and nitrate in that order. The arrows representing nitrate, TSS, total phosphorus, and nitrite are much more closely associated with each other than TDS. They follow a clockwise pattern on the plot, separated by acute angles. The arrow representing percent agricultural cover points in the complete opposite direction as the arrows representing the other land use variables, indicating a negative relationship. This land use variable is most closely associated with nitrate, followed by TSS, total phosphorus, nitrite, and TDS.

The points on the biplot are clearly organized into two distinct clusters, each corresponding to one of the two creeksheds. The Mill Creek data points are represented in teal while the Allen Creek data points are represented in orange. The organization of the clusters suggests that observations from within Mill and Allen Creek are overall more similar than observations between the opposing creeksheds. There are two significant outliers in the Mill Creek cluster that I felt warranted a closer look. These two points, located in the top left portion of the graph, correspond to rows 28 and 112 of the data set. Upon further examination, it was discovered that both of these samples were taken during or immediately following precipitation events, which could lead to inflated values of certain parameters.

iv. Logistic Regression Discussion

1. TDS:

The logistic regression models were included in the hopes that it would reveal a relationship between increasing percentage of cover by each land use variable and the likelihood of each water quality parameter exceeding its established EPA limit. These models, however, varied in how informative they were. In the case of TDS and urban cover the predicted probabilities produced by the logistic regression were so low that a relationship between the likelihood of this parameter exceeding its accepted EPA limit and increasing percent urban cover cannot be confidently determined. Predicted probabilities were slightly higher but still fairly low for agricultural cover, also not boding particularly well. The predicted probabilities plot for TDS and impervious cover was more informative than the other two, as it displays a clear separation between when TDS is likely to exceed its limit and when it is not, and this separation corresponds to a specific point along the continuum of percent cover. Given the excess runoff that is produced as a result of impervious surfaces I would have assumed that the predicted probability of TDS exceeding its limit would have increased as impervious cover increased. However, a negative relationship was produced instead.

2. TSS:

The predicted probabilities curves for TSS each showed very clear relationships between the likelihood of TSS exceeding its limit and each of the land use land cover variables. In each case the relationship was positive, and the probability of the EPA limit being exceeded reached 1 in all three cases. This probability is reached much earlier, however, in the cases of percent agricultural cover and impervious cover. In both scenarios the probability of 1 is achieved at or around 25% cover, whereas this does not occur until about 75% urban cover. This suggests that greater increases in TSS may occur at smaller areas of agricultural and impervious cover relative to urban cover. However, given the correlation between urban space and impervious surfaces it is not clear why the predicted probability does not reach 1 until a greater percentage of coverage is reached when land is classified more broadly as just urban.

3. Total Phosphorus:

The relationship between the predicted probability of total phosphorus exceeding its EPA limit and percent urban cover is positive. This is not necessarily surprising given the fact that urban areas are known to increase nutrient loading (Mahmoodi, 2019; Huang et al., 2016). Likewise, there is a positive relationship between the predicted probability of total phosphorus exceeding its limit and percent agricultural cover. This aligns with expectations due to the fact that many of the fertilizers that are used in agriculture contain this nutrient (Anh et al. 2023). However, the relationship between total phosphorus and percent impervious cover was negative, which is surprising since a positive relationship was observed for urban cover. This may indicate that some features of urban areas other than impervious surfaces contribute to phosphorus loading. Such features may include high density residential areas, rapid population growth, and industrial activities that produce waste.

4. Nitrite:

The predicted probabilities plots for nitrite displayed similar trends as the plots for total phosphorus. There was a positive relationship between the predicted probability of nitrite exceeding its limit and both percent urban and agricultural cover. Both of these parameters are nutrients that may originate from common sources, so these results are consistent with expectations. There was also a negative relationship observed between percent impervious cover and the predicted probability of nitrite exceeding its EPA limit. This may once again indicate that some other characteristic or characteristics of urban areas contribute to nutrient loading independent of imperviousness.

5. Nitrate:

The results of the logistic regression analysis for nitrate are confusing. The same, very low, predicted probability is replicated along the y-axis for all three predicted probabilities plots. However, for percent urban and agricultural cover there is an observed negative trend, while impervious cover displays a positive trend. It is unclear why the trends go in opposite directions or why the curves are not flat since the predicted probability of nitrate exceeding its EPA limit appears constant. Regardless, the predicted probability that was produced is very small, suggesting that the likelihood of nitrate exceeding its limit is not very high regardless of percent coverage by any of these land use variables. A clear relationship between these variables and the probability of nitrate exceeding its limit cannot be confidently defined.

c. Predictive Models

i. Generalized Additive Models

The GAMs also returned rather disappointing results. All of the R^2 values that were produced were either low or negative. The highest one once again corresponded to the TDS model, at .30. The MSE of this model was also very high at 34, 246. These numbers suggest that even though this model had a higher R^2 than any of the other ones, it is not particularly well-fitted to the data, and may not be very useful for TDS prediction. The R2 for the TSS model was once again extremely low at .003, and the MSE was quite high at 814. These results cast doubt on the model's efficacy. The R2 for the Total Phosphorus model was even worse at -.02, indicating that this model is not well-fitted to the data at all despite the relatively low MSE of .0036. The R2 values for the Nitrate and Nitrite models were .013 and .075, respectively, and the MSE's were .00015 and .66. Based on these results I would not select this method of modeling these water quality parameters based on these predictors. This is not to say that it is impossible to successfully use this method to model these indicators, but that additional or new predictors might be required to do so.

ii. Support Vector Machine Discussion

The SVM regression models yielded relatively low R^2 values for each parameter. The highest R^2 value produced by any of the models was .32, corresponding to TDS. This indicates that the model explains 32% of the variance in the data. While this is substantially higher than the R^2 for any of the other models, it is still a fairly low number that does not inspire much confidence in the model's predictive power. The MSE of this model was also quite high at 32,897, suggesting a lot of deviance between the predicted and actual values of TDS. The SVM regression for TSS was dismally low at .022, and the MSE fairly high at 536. The SVM regression for Total Phosphorus produced an even lower R^2 of .00013. The MSE for this model was relatively low at .57, but I still would not select this model as a viable choice for phosphorus prediction. The R^2 values for the nitrate and nitrite models were .03 and .06, respectively, and the MSE's were .00013 and .57. None of these numbers are particularly promising or suggest that the models are well-fitted to the data. These results indicate that SVM regression might not be a great choice for modeling these parameters, at least with these predictors.

iii. Random Forest Regression Discussion

Out of all the methods that I tried, Random Forest Regression was perhaps the most promising, at least for certain parameters. The R^2 values for TDS, Nitrate, and Nitrite were still pretty low at .25, .2, and .21 respectively. The corresponding MSEs for these models were 40,563, .00013, and .00013. However, the R^2 values for TSS and Total Phosphorus were .93 and .63, respectively. Despite the very high R^2, the MSE for TSS was still pretty high at 571. However, given that the model is able to explain 93% of the variance in the data, I would still be pretty comfortable choosing this model for TSS prediction. The MSE for Total Phosphorus was fairly low at .004, so I would also feel comfortable selecting this model for phosphorus prediction despite its more modest R^2 value of .63.

iv. Neural Network Regression Discussion

The neural network regression models were perhaps the worst performing models of the group. The R^2 for TDS was -5.8×10^{-15}, the lowest R^2 produced by any of the models included in this paper. This model also produced a higher MSE than any of the others, at 47,380. Overall this model seems like a terrible choice for TDS prediction. The R^2 for TSS was also negative at -1.8x10-07, with an MSE that was more similar to some of the other TSS models but still high at 524. The R^2 for the Total Phosphorus model was -.0011, with an MSE of .0035. The R^2 for the nitrate model was -.0081 and the MSE was .00012. Both of these sets of results are very dismal. The only neural network model to produce a positive R^2, albeit a very low one of .12, was nitrate. The MSE for this model was .5. I would not select this method for modeling any of these parameters. None of these models show much sign of being well-fitted to the data at all.

6. Conclusion:

Throughout this project I experimented with a variety of different methods of modeling water quality parameters using three different land use land cover variables as predictors. These three variables were percent cover by urban area, percent cover by agricultural area, and percent cover by impervious surfaces in each of the subwatersheds that were included in the analysis. The first modeling method that I tried was a linear model in OpenBugs. I used a normal likelihood and included a variance covariance matrix made using the coordinate data of the 12 monitoring stations of interest. This covariance matrix was in turn used to create a spatial random effect to account for spatial dependencies between the stations. For instance, the water chemistry of an upstream station may affect the chemistry of a downstream station. I encountered a number of issues with these models. Specifically, many of the models were predicting negative values, which is clearly not accurate as water quality measurements cannot be negative. I attempted to remedy these effects by taking the logarithm of the predicted values. I experimented with doing so both within and outside of the models, neither of which solved the problem. This prompted me to explore other methods of modeling.

Some of these other methods included Generalized Additive Models, Random Forest Models, Support Vector Machine Regressions, and Neural Network Regressions. I had varying degrees of success with these methods. Most of the models returned relatively low R^2 values and high Mean Squared Errors, although these values varied across parameters and type of model. The only models that I would feel confident moving forward with in my research were the Random Forest Regressions for TSS and Total Phosphorus. The R^2 values for these models were .93 and .63, respectively.

I believe that the poor performances of most of the models had less to do with the methods themselves and more to do with the nature of the predictors that I chose. I feel comfortable concluding that percent cover alone does not provide sufficient information to explain the intricate relationship between land and water. The results of the logistic regressions suggest that a relationship between percent cover and these parameters may exist, but the bigger picture is likely more complicated than percent cover itself can explain. Given the complex nature of the hydrologic cycle, I would argue that the physical characteristics of a particular landscape are more important for predicting water quality than percent cover of any land use category. The physical characteristics to which I am referring may include topography or elevation.

The models may have also failed to account for additional externalities. For instance, many of the streams within Mill Creek have been artificially straightened, which alters the natural flow of the stream and can in turn affect water quality. Additionally, the majority of the wetlands within Mill Creek have been drained for agriculture (Huron River Watershed Council [HRWC], 2014), which could have contributed to poorer water quality outcomes than what might be expected in a creekshed with so little impervious surfaces. Thus, in this case it is difficult to isolate the effects of LULC on water quality because of the effects of another form of human interference. A solution to this problem may be to account for how the natural flow patterns of the creek have been altered by the channelization, which would require an understanding of the relationship between flow and phosphorus and would add complexity to the model. It is likely that this model in its current form is not a particularly effective tool for making predictions about total phosphorus in the Huron River watershed because the relationship between LULC and water quality is more intricate than what was initially expected. Even if the predictor variables had been statistically significant, it would not be possible to say with certainty that phosphorus levels were the result of LULC alone. In order for the model to perform better, additional predictors may need to be included, as well as corrections for externalities.

Additionally, despite the fact that there is a relationship between land use and water quality, there are so many processes involved in determining water quality outcomes that it is difficult to isolate the effects of any one variable. For this reason, I would suggest that anyone attempting to model these indicators using these methods or others include additional predictors. This is something that I would like to do if I were to continue this project. Some variables that I believe would be helpful to include are temperature, flow, and precipitation. Other factors that may be more difficult to model, but can impact water quality, include channelization of the river and recreational water use. Also, I would recommend including more than these two tributaries in the analysis.

Despite the fact that some of my findings were not statistically significant, I still feel like I have accomplished something through this project. This research has allowed me to explore a variety of quantitative methods that I most likely would not have been exposed to otherwise. Before I entered this program, I had little to no experience in R or any other coding languages. While I did learn the basics of R in some of my early coursework at SEAS, I feel like I have been able to develop a much deeper understanding of the language through simply spending independent time in the program and countless hours of trial and error. This project allowed me to expand my skills as a quantitative analyst.

7. Acknowledgements:

I would like to thank my advisor Andrew Gronewold for his continued support throughout this project, the Huron River Watershed Council for providing me with their data, and my friends and loved ones for their continued encouragement. I also want to thank George Kling and Runzi Wang for serving on my committee, and Manish Venumuddula for helping me with my code.

Appendices

APPENDIX A: Model Code

APPENDIX A1: Data collection and organization

```
## (with assistance from Manish Venumuddula and Drew Gronewold)
## Finalized April 2024

setwd("~/Desktop/Thesis")

##Read in the main data set
read.wq0 = read.csv("HRWC_All_ChemFlow_Data.csv")

read.wq<-subset(read.wq0, ID < 13)

print(read.wq)

#write_csv("read.wq")

##Mill Creek Data

Mill01 <- subset(read.wq, Site == "Mill01")
Mill01.ordered = Mill01[order(as.Date(Mill01$Collection.Date,
format = "%m/%d/%Y")),]

MH02B <- subset(read.wq, Site == "MH02B")
MH02B.ordered = MH02B[order(as.Date(MH02B$Collection.Date,
format = "%m/%d/%Y")),]

MH02A <- subset(read.wq, Site == "MH02A")
MH02A.ordered = MH02A[order(as.Date(MH02A$Collection.Date,
format = "%m/%d/%Y")),]

Mill02 <- subset(read.wq, Site == "Mill02")
Mill02.ordered = Mill02[order(as.Date(Mill02$Collection.Date,
format = "%m/%d/%Y")),]

Mill03 <- subset(read.wq, Site == "Mill03")
Mill03.ordered = Mill03[order(as.Date(Mill03$Collection.Date,
format = "%m/%d/%Y")),]

Mill07 <- subset(read.wq, Site == "Mill07")
```

```
Mill07.ordered = Mill07[order(as.Date(Mill07$Collection.Date,
format = "%m/%d/%Y")),]

Mill06 <- subset(read.wq, Site == "Mill06")

Mill06.ordered = Mill06[order(as.Date(Mill06$Collection.Date,
format = "%m/%d/%Y")),]

Mill09 <- subset(read.wq, Site == "Mill09")
Mill09.ordered = Mill09[order(as.Date(Mill09$Collection.Date,
format = "%m/%d/%Y")),]

Mill10 <- subset(read.wq, Site == "Mill10")
Mill10.ordered = Mill10[order(as.Date(Mill10$Collection.Date,
format = "%m/%d/%Y")),]

Mill11 <- subset(read.wq, Site == "Mill11")
Mill11.ordered = Mill11[order(as.Date(Mill11$Collection.Date,
format = "%m/%d/%Y")),]

Mill12 <- subset(read.wq, Site == "Mill12")
Mill12.ordered = Mill12[order(as.Date(Mill12$Collection.Date,
format = "%m/%d/%Y")),]

Mill08 <- subset(read.wq, Site == "Mill08")
Mill08.ordered = Mill08[order(as.Date(Mill08$Collection.Date,
format = "%m/%d/%Y")),]

##Allen Creek Data

MH04 <- subset(read.wq, Site == "MH04")
MH04.ordered <- MH04[order(as.Date(MH04$Collection.Date,
format = "%m/%d/%Y")),]


##Appending Mill Creek stations, removed Mill12
MillCreekTP <- c(Mill01.ordered$Total.Phosphorus,
MH02B.ordered$Total.Phosphorus,
MH02A.ordered$Total.Phosphorus,
  Mill02.ordered$Total.Phosphorus,
Mill03.ordered$Total.Phosphorus,
Mill07.ordered$Total.Phosphorus,
  Mill06.ordered$Total.Phosphorus,
Mill09.ordered$Total.Phosphorus,
Mill10.ordered$Total.Phosphorus,
  Mill11.ordered$Total.Phosphorus,
Mill08.ordered$Total.Phosphorus)

stations.mill <- c(rep(1,
length(Mill01.ordered$Total.Phosphorus)),
              rep(2, length(MH02B.ordered$Total.Phosphorus)),
              rep(3, length(MH02A.ordered$Total.Phosphorus)),

              rep(4, length(Mill02.ordered$Total.Phosphorus)),
              rep(5, length(Mill03.ordered$Total.Phosphorus)),
              rep(6, length(Mill07.ordered$Total.Phosphorus)),
              rep(7, length(Mill06.ordered$Total.Phosphorus)),
              rep(8, length(Mill09.ordered$Total.Phosphorus)),
              rep(9, length(Mill10.ordered$Total.Phosphorus)),
              rep(10,
length(Mill11.ordered$Total.Phosphorus)),
              rep(11,
length(Mill08.ordered$Total.Phosphorus)),
              rep(12, length(MH04.ordered$Total.Phosphorus)))

##Not all stations within Mill Creek have observations for
TDS, many have observations for TSS
MillCreekTDS <- c(MH02B.ordered$Total.Dissolved.Solids,
Mill09.ordered$Total.Dissolved.Solids,
Mill10.ordered$Total.Dissolved.Solids,
                  Mill11.ordered$Total.Dissolved.Solids,
Mill08.ordered$Total.Dissolved.Solids)

MillCreekTemp <- c(MH02B.ordered$Water.Temperature,
MH02A.ordered$Water.Temperature,
Mill02.ordered$Water.Temperature,
Mill03.ordered$Water.Temperature,
Mill07.ordered$Water.Temperature,
Mill06.ordered$Water.Temperature,
Mill09.ordered$Water.Temperature,
Mill10.ordered$Water.Temperature,
Mill11.ordered$Water.Temperature,
Mill08.ordered$Water.Temperature)
length(MillCreekTDS)#211

##TSS for Mill Creek
MillCreekTSS <- c(MH02B.ordered$Total.Suspended.Solids,
Mill09.ordered$Total.Suspended.Solids,
Mill10.ordered$Total.Suspended.Solids,
                  Mill11.ordered$Total.Suspended.Solids,
Mill08.ordered$Total.Suspended.Solids)
length(MillCreekTSS)#211
```

```
##Nitrate for Mill Creek
MillCreekNitrate <- c(Mill01.ordered$Total.Nitrate,
MH02B.ordered$Total.Nitrate, MH02A.ordered$Nitrate,
                      Mill02.ordered$Nitrate,
Mill03.ordered$Nitrate, Mill07.ordered$Nitrate,

                      Mill 06.ordered$Nitrate,
Mill09.ordered$Nitrate, Mill10.ordered$Nitrate,
                      Mill11.ordered$Nitrate,
Mill08.ordered$Nitrate)#103

##Nitrite for Mill Creek
MillCreekNitrite <- c(Mill01.ordered$Total.Nitrite,
MH02B.ordered$Total.Nitrite, MH02A.ordered$Nitrite,
                      Mill02.ordered$Nitrite,
Mill03.ordered$Nitrite, Mill07.ordered$Nitrite,
                      Mill06.ordered$Nitrite,
Mill09.ordered$Nitrite, Mill10.ordered$Nitrite,
                      Mill11.ordered$Nitrite,
Mill08.ordered$Nitrite)#103


##TDS for Allen Creek

AllenCreekTDS <- MH04.ordered$Total.Dissolved.Solids
length(AllenCreekTDS)#183

##TSS for Allen Creek

AllenCreekTSS <- MH04.ordered$Total.Suspended.Solids
length(AllenCreekTSS)#183

##Nitrate for Allen Creek

AllenCreekNitrate <- MH04.ordered$Nitrate #183

##Temperature for Allen Creek

##Nitrite for Allen Creek

AllenCreekNitrite <- MH04.ordered$Nitrite #183

AllenCreekTemp <- MH04.ordered$Water.Temperature


##Concatenating TP from Mill and Allen Creek
```

```
TP.all <- c(MillCreekTP, MH04.ordered$Total.Phosphorus)

##Concatenating TDS from Mill and Allen Creek
TDS.all <- c(MillCreekTDS, AllenCreekTDS)
length(TDS.all)#394

##Concatenating TSS from Mill and Allen Creek

TSS.all <- c(MillCreekTSS, AllenCreekTSS)

##Concatenating Nitrate data from Mill and Allen Creek
Nitrate.all <- c(MillCreekNitrate, AllenCreekNitrate)

##Concatenating Nitrite data from Mill and Allen Creek
Nitrite.all <- c(MillCreekNitrite, AllenCreekNitrite)

##Concatenating temperature observations from Mill and Allen
Creek
Temp.all <- c(MillCreekTemp, AllenCreekTemp)

mean(Temp.all, na.rm = TRUE)

##Percent cover by Urban/Residential, Imperviousness, Ag, and
Forest in Mill Creek
length(MillCreekTP) #283

MillCreek.Developed <- rep(.18, 283)

MillCreek.Forest <- rep(.1, 283)

MillCreek.Impervious <- rep(.04, 283)

MillCreek.Ag <- rep(.47, 283)

##May need to use different LULC objects for different
parameters because the number of observations are different
##LULC for TDS
MillCreekTDS.Developed <- rep(.18, 211)

MillCreekTDS.Forest <- rep(.1, 211)

MillCreekTDS.Impervious <- rep(.04, 211)

MillCreekTDS.Ag <- rep(.47, 211)

##LULC objects for Nitrate
MillCreekNitrate.Developed <- rep(.18, 103)
MillCreekNitrate.Forest <- rep(.1, 103)

MillCreekNitrate.Impervious <- rep(.04, 103)

MillCreekNitrate.Ag <- rep(.47, 103)

#Don't need to repeat for nitrite, number of observations is
the same

##Percent cover by Urban/Residential, Imperviousness,  and
Forest in Allen Creek

AllenCreek.Developed <- rep(.96, 183)

AllenCreek.Forest <- rep(.01, 183)

AllenCreek.Impervious <- rep(.55, 183)

AllenCreek.Ag <- rep(0, 183)


##Predictors

Developed.all <- c(MillCreek.Developed, AllenCreek.Developed)
length(Developed.all)

Forest.all <- c(MillCreek.Forest, AllenCreek.Forest)
length(Forest.all)

Impervious.all <- c(MillCreek.Impervious,
AllenCreek.Impervious)

Ag.all <- c(MillCreek.Ag, AllenCreek.Ag)

##Predictors for TDS model
TDSDeveloped.all <- c(MillCreekTDS.Developed,
AllenCreek.Developed)
length(Developed.all)

TDSForest.all <- c(MillCreekTDS.Forest, AllenCreek.Forest)
length(Forest.all)

TDSImpervious.all <- c(MillCreekTDS.Impervious,
AllenCreek.Impervious)

TDSAg.all <- c(MillCreekTDS.Ag, AllenCreek.Ag)
```

```
##Predictors for Nitrate and Nitrite models
NDeveloped.all <- c(MillCreekNitrate.Developed,
AllenCreek.Developed)

NForest.all <- c(MillCreekNitrate.Forest, AllenCreek.Forest)


NImpervious.all <- c(MillCreekNitrate.Impervious,
AllenCreek.Impervious)

NAg.all <- c(MillCreekNitrate.Ag, AllenCreek.Ag)
```

APPENDIX A2: OpenBugs Model code

```
##Creating the model

#install.packages("rjags")
#install.packages("coda")
#install.packages("runjags")
#install.packages("MASS")
#install.packages("MCMCpack")


library(rjags)
library(coda)
library(runjags)
library(MASS)
library(MCMCpack)


##Calculating Euclidean distance between station coordinates
#Stations <- read.csv("SiteMetadata.csv")

#create a matrix with all distances

stations <- c("Mill01","MH02B","MH02A","Mill02","Mill03","Mill07",
"Mill06","Mill09","Mill10","Mill11","Mill08", "MH04")

#install.packages("tidyverse")
library(tidyverse)
row.col <- data.frame( x =
c(42.339353,42.29989,42.290259,42.281577,42.287667,42.29,42.295,
42.32223,42.32949,42.27219,42.2665194, 42.28994),
                       y =
c(-83.89056,-83.89849,-83.908518,-83.922922,-83.937157,-83.94,
-83.95,-83.97946,-84.04447,-83.93649,-83.9555855,-83.74597))
```

```
#Stations = c("Mill01","MH02B","MH02A","Mill02","Mill03","Mill07",
"Mill06","Mill09","Mill10","Mill11","Mill08", "MH04"),
#row.col <- data.frame(Stations$Latitude, Stations$Longitude)
#coordinates for Mill06 and Mill07 made up, data not available
D <- dist(row.col, method = "euclidean", diag = TRUE,
 upper = TRUE)

D <- as.matrix(D)

NS<-12

ID <- read.wq$ID

TDS.all <- na.omit(TDS.all)
```

APPENDIX A2.1.1: OpenBugs TDS with Covariance Matrix and Spatial Random Effect

```
##TDS model with covariance matrix and spatial random effect
d <- read.jagsdata("TDSModelData.R")

##Initial values, 3 chains

inits1<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits2<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits3<-list(tau = c(1,1,1), alpha = c(1,1,1))

#file.exists("TDSModel_1Oct2023.bug")

Model <- jags.model("TDSModel_2Jan2024.bug", d, n.chains = 3)

##Updating the model

update(Model, 100000) ##burn in
parameters <- coda.samples(Model, c("alpha", "tau", "sig"),
n.iter = 50000, thin = 1)
summary(parameters)

plot(parameters[,'tau[1]'])
plot(parameters[,'tau[2]'])
plot(parameters[,'tau[3]'])
plot(parameters[,'alpha[1]'])
plot(parameters[,'alpha[2]'])
plot(parameters[,'alpha[3]'])
plot(parameters[,'alpha[4]'])

dev.off()
```

```
y<- coda.samples( Model, c("y.pred"), n.iter = 50000, thin =
1)
```

```
c1 <- rgb(173,216,230,max = 255, alpha = 80, names =
"lt.blue")
c2 <- rgb(255,192,203,max = 255, alpha = 80, names =
"lt.pink")
c3 <- rgb(144,238,144,max = 255, alpha = 80, names =
"lt.green")
c4 <- rgb(160,32,240,max = 255, alpha = 80, names = "purple")
```

APPENDIX A2.1.2: OpenBugs TDS without Covariance Matrix and Spatial Random Effect

```
##TDS model without covariance matrix or spatial random effect

d1.1 <- read.jagsdata("TDSModelData_nocm.R")

inits1<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits2<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits3<-list(tau = c(1,1,1), alpha = c(1,1,1))

Model1.1 <- jags.model("TDSModel_2Jan2024_nocm.bug", d1.1,
 n.chains = 3)

update(Model1.1, 100000) ##burn in
parameters1.1 <- coda.samples(Model1.1, c("alpha", "tau"),
 n.iter = 50000, thin = 1)
summary(parameters1.1)

y1.1 <- coda.samples( Model1.1, c("y.pred"), n.iter = 50000,
 thin = 1)
```

APPENDIX A2.2.1: OpenBugs TSS with Covariance Matrix and Spatial Random Effect

```
TSS.all <- na.omit(TSS.all)
TSS.all <- TSS.all[TSS.all != 980]
##TSS model with covariance matrix and spatial random effect
d2 <- read.jagsdata("TSSModelData.R")


inits1<-list(tau = c(1,1,1), alpha = c(1,1,1), sig = c(1,1))
```

```
inits2<-list(tau = c(1,1,1), alpha = c(1,1,1), sig = c(1,1))
inits3<-list(tau = c(1,1,1), alpha = c(1,1,1), sig = c(1,1))

Model2 <- jags.model("TSSModel_2Jan2024.bug", d2, n.chains = 3)

update(Model2, 100000) ##burn in
parameters2 <- coda.samples(Model2, c("alpha", "tau", "sig"),

 n.iter = 50000, thin = 1)
summary(parameters2)


plot(parameters2[,'tau[1]'])
plot(parameters2[,'tau[2]'])
plot(parameters2[,'tau[3]'])
plot(parameters2[,'alpha[1]'])
plot(parameters2[,'alpha[2]'])
plot(parameters2[,'alpha[3]'])
plot(parameters2[,'alpha[4]'])

y2 <- coda.samples( Model2, c("y.pred"), n.iter = 50000,
thin = 1)

hist(TSS.all, main = "All TSS Observations \nin Mill and
 Allen Creek", xlab = "TSS", col = c1, freq = FALSE, xlim =
 c(-200,1000))
lines(density(as.numeric(y2[[1]])))
```

APPENDIX A2.2.2: OpenBugs TSS without Covariance Matrix and Spatial Random Effect

```
##TSS model without covariance matrix or spatial random effect
d2.1 <- read.jagsdata("TSSModelData_nocm.R")


inits1<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits2<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits3<-list(tau = c(1,1,1), alpha = c(1,1,1))

Model2.1 <- jags.model("TSSModel_3Jan2024_nocm.bug", d2.1,
 n.chains = 3)

update(Model2.1, 100000) ##burn in
parameters2.1 <- coda.samples(Model2.1, c("alpha", "tau", "sig"),
 n.iter = 50000, thin = 1)
summary(parameters2.1)

y2.1 <- coda.samples( Model2.1, c("y.pred"), n.iter = 50000,
 thin = 1)
```

APPENDIX A2.3.1: OpenBugs Nitrate with Covariance Matrix and Spatial Random Effect

```
##Nitrate model with covariance matrix and spatial random effect
d3 <- read.jagsdata("NitrateModelData.R")


inits1<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits2<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits3<-list(tau = c(1,1,1), alpha = c(1,1,1))

Model3 <- jags.model("NitrateModel_3Jan2024.bug", d3,
 n.chains = 3)

update(Model3, 100000) ##burn in
parameters3 <- coda.samples(Model3, c("alpha", "tau", "sig")
, n.iter = 50000, thin = 1)
summary(parameters3)

y3 <- coda.samples( Model3, c("y.pred"), n.iter = 50000,
 thin = 1)

hist(Nitrate.all, main = "All Nitrate Observations \nin
 Mill and Allen Creek", xlab = "Nitrate", col = c1,
 freq = FALSE, xlim = c(-2,5))
lines(density(as.numeric(y3[[1]])))
```

APPENDIX A2.3.2: OpenBugs Nitrate without Covariance Matrix and Spatial Random Effect

```
##Nitrate model without covariance matrix or spatial random
 effect
d3.1 <- read.jagsdata("NitrateModelData_nocm.R")

inits1<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits2<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits3<-list(tau = c(1,1,1), alpha = c(1,1,1))

Model3.1 <- jags.model("NitrateModel_1Oct2023_nocm.bug", d3,
n.chains = 3)

update(Model3.1, 100000) ##burn in
parameters3.1 <-
  coda.samples(Model3.1,
               c("alpha", "tau", "sig"),
               n.iter = 50000,
               thin = 1)
summary(parameters3.1)

y3.1 <- coda.samples( Model3.1, c("y.pred"), n.iter = 50000,
thin = 1)


hist(Nitrate.all, main = "All Nitrate Observations \nin Mill
and Allen Creek", xlab = "Nitrate", col = c1, freq = FALSE,
xlim = c(-2,5))
lines(density(as.numeric(y3.1[[1]])))
```

APPENDIX A2.4.1: OpenBugs Nitrite with Covariance Matrix and Spatial Random Effect

```
##Nitrite model with covariance matrix and spatial random
effect
d4 <- read.jagsdata("NitriteModelData.R")

inits1<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits2<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits3<-list(tau = c(1,1,1), alpha = c(1,1,1))

Model4 <- jags.model("NitriteModel_1Oct2023.bug", d4, n.chains
= 3)

update(Model4, 100000) ##burn in
parameters4 <- coda.samples(Model4, c("alpha", "tau", "sig"),
n.iter = 50000, thin = 1)
summary(parameters4)

y4 <- coda.samples( Model4, c("y.pred"), n.iter = 50000, thin
= 1)

hist(Nitrite.all, main = "All Nitrite Observations \nin Mill
and Allen Creek", xlab = "Nitrite", col = c1, freq = FALSE,
xlim = c(-.25,.5))
lines(density(as.numeric(y4[[1]])))
```

APPENDIX A2.4.2: OpenBugs Nitrite without Covariance Matrix and Spatial Random Effect

```
##Nitrite model without covariance matrix or spatial random
effect
d4.1 <- read.jagsdata("NitriteModelData_nocm.R")
```

```
inits1<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits2<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits3<-list(tau = c(1,1,1), alpha = c(1,1,1))

Model4.1 <- jags.model("NitriteModel_1Oct2023_nocm.bug", d4.1,
n.chains = 3)


update(Model4.1, 100000) ##burn in
parameters4.1 <- coda.samples(Model4.1, c("alpha", "tau",
"sig"), n.iter = 50000, thin = 1)
summary(parameters4.1)

y4.1 <- coda.samples( Model4.1, c("y.pred"), n.iter = 50000,
thin = 1)

hist(Nitrite.all, main = "All Nitrite Observations \nin Mill
and Allen Creek", xlab = "Nitrite", col = c1, freq = FALSE,
xlim = c(-.25,.5))
lines(density(as.numeric(y4.1[[1]])))

##Trying to make multi-panel plots

par(mfrow = c(2,3))
plot(parameters[,'alpha[1]'])
plot(parameters[,'alpha[2]'])
plot(parameters[,'alpha[3]'])
plot(parameters2[,'alpha[1]'])
plot(parameters2[,'alpha[2]'])
plot(parameters2[,'alpha[3]'])



#dev.off()
```

APPENDIX A2.5.1: OpenBugs Phosphorus with Covariance Matrix and Spatial Random Effect

```
##TP with covariance matrix
d5 <- read.jagsdata("TPModelData.R")

inits1<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits2<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits3<-list(tau = c(1,1,1), alpha = c(1,1,1))

Model5 <- jags.model("TPModel_24March2024.bug", d, n.chains =
3)
```

```
##Updating the model

update(Model5, 100000) ##burn in
parameters5 <- coda.samples(Model5, c("alpha", "tau", "sig"),
n.iter = 50000, thin = 1)

summary(parameters5)

plot(parameters[,'tau[1]'])
plot(parameters[,'tau[2]'])
plot(parameters[,'tau[3]'])
plot(parameters[,'alpha[1]'])
plot(parameters[,'alpha[2]'])
plot(parameters[,'alpha[3]'])
plot(parameters[,'alpha[4]'])
```

APPENDIX A2.5.2: OpenBugs Phosphorus without Covariance Matrix and Spatial Random Effect

```
##TP without covariance matrix
d5.1 <- read.jagsdata("TPModelData_nocm.R")

inits1<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits2<-list(tau = c(1,1,1), alpha = c(1,1,1))
inits3<-list(tau = c(1,1,1), alpha = c(1,1,1))


Model5.1 <- jags.model("TPModel_24March2024_nocm.bug", d,
n.chains = 3)

##Updating the model

update(Model5.1, 100000) ##burn in
parameters5.1 <- coda.samples(Model5.1, c("alpha", "tau",
"sig"), n.iter = 50000, thin = 1)
summary(parameters)


plot(parameters[,'tau[1]'])
plot(parameters[,'tau[2]'])
plot(parameters[,'tau[3]'])
plot(parameters[,'alpha[1]'])
plot(parameters[,'alpha[2]'])
plot(parameters[,'alpha[3]'])
plot(parameters[,'alpha[4]'])
```

```
##Log transforming each chain of the predicted values, did not
work
#epsilon <- 1e-10

#y[[1]]

# Check if any values are non-positive before log
transformation
#if (any(y[[1]] <= 0)) {
  #warning("Some values in 'y[[1]]' are non-positive.
Excluding them before log transformation.")

  # Exclude non-positive values and then apply log
transformation
 # log.y1 <- log(y[[1]][y[[1]] > 0] + epsilon)
#} else {
  #log.y1 <- log(y[[1]] + epsilon)
#}
##y[[2]]
#if (any(y[[2]] <= 0)) {
 # warning("Some values in 'y[[2]]' are non-positive.
Excluding them before log transformation.")

  # Exclude non-positive values and then apply log
transformation
 # log.y2 <- log(y[[2]][y[[2]] > 0] + epsilon)
#} else {
#  log.y2 <- log(y[[2]] + epsilon)
#}

##Exporting Predicted vs actual histograms to a single pdf
pdf(file = "AllFigures")

##TDS with Covariance Matrix
par(mfrow=c(2,3), oma = c(0, 2, 0, 0))
hist(TDS.all, main = "Histogram of TDS Observed vs. Predicted
Values\n With Covariance Matrix", xlab = "TDS", col = c1, freq
= FALSE, ylim = c(0,.006), cex.main = .8)
lines(density(as.numeric(y[[1]])))

hist(TDS.all, main = "Histogram of TDS Observed vs. Predicted
Values\n With Covariance Matrix", xlab = "TDS", col = c1, freq
= FALSE, ylim = c(0,.006), cex.main = .8)
lines(density(as.numeric(y[[2]])), col = "blue")
```

```
hist(TDS.all, main = "Histogram of TDS Observed vs. Predicted
Values\n With Covariance Matrix", xlab = "TDS", col = c1, freq
= FALSE, ylim = c(0,.006), cex.main = .8)
lines(density(as.numeric(y[[3]])), col = "red")
legend(x = "topright",
       col = c("black", "blue", "red"), lty = 1, lwd = 1,
       legend = c('Chain 1', 'Chain 2', 'Chain 3'), cex = .5)



##TDS without Covariance Matrix
dev.off()
par(mfrow=c(2,3), oma = c(0, 2, 0, 0))

hist(TDS.all, main = "Histogram of TDS\n Observed vs.
Predicted Values\n Without Covariance Matrix", xlab = "TDS",
col = c1, freq = FALSE, cex.main = .8)
lines(density(as.numeric(y1.1[[1]])))

hist(TDS.all, main = "Histogram of TDS\n Observed vs.
Predicted Values\n Without Covariance Matrix", xlab = "TDS",
col = c1, freq = FALSE, cex.main = .8)
lines(density(as.numeric(y1.1[[2]])), col = "blue")

hist(TDS.all, main = "Histogram of TDS\n Observed vs.
Predicted Values\n Without Covariance Matrix", xlab = "TDS",
col = c1, freq = FALSE, cex.main = .8)
lines(density(as.numeric(y1.1[[3]])), col = "red")
legend(x = "topright",
       col = c("black", "blue", "red"), lty = 1, lwd = 1,
       legend = c('Chain 1', 'Chain 2', 'Chain 3'), cex = .5)

##TSS with CM having problems right now

##TSS without CM
dev.off()
par(mfrow=c(2,3), oma = c(0, 2, 0, 0))

hist(TSS.all, main = "Histogram of TSS Observed vs. Predicted
Values\n With Covariance Matrix", xlab = "TSS", col = c1, freq
= FALSE, xlim = c(-200,1000), cex.main = .8)
lines(density(as.numeric(logy2.1.1)))

hist(TSS.all, main = "Histogram of TSS Observed vs. Predicted
Values\n With Covariance Matrix", xlab = "TSS", col = c1, freq
= FALSE, xlim = c(-200,1000), cex.main = .8)
lines(density(as.numeric(y2.1[[2]])))
```

```
hist(TSS.all, main = "Histogram of TSS Observed vs. Predicted
Values\n With Covariance Matrix", xlab = "TSS", col = c1, freq
= FALSE, xlim = c(-200,1000), cex.main = .8)
lines(density(as.numeric(y2.1[[3]])))
legend(x = "topright",
       col = c("black", "blue", "red"), lty = 1, lwd = 1,
       legend = c('Chain 1', 'Chain 2', 'Chain 3'), cex = .5)
```

APPENDIX A3: Generalized Additive Model code

```
##New methods
data <- read.csv("newdata.csv")
head(data)
data <- na.omit(data)
cor(data[c("Percent.Cover.Urban", "Percent.Cover.Agriculture",
"Percent.Cover.Impervious.Surfaces")])

model <- lm(Total.Dissolved.Solids ~ Percent.Cover.Urban +
Percent.Cover.Agriculture + Percent.Cover.Impervious.Surfaces,
 data = data)

# Print the summary of the model
summary(model)
##Singularities produced due to multicollinearity in
predictors

##GAM
# Install and load the mgcv package
install.packages("mgcv")
library(mgcv)
library(splines)
# Fit a GAM model

# Added jitter to the constant predictors
data$Percent.Cover.Urban <- data$Percent.Cover.Urban +
 runif(nrow(data), min = 0, max = 0.01)
data$Percent.Cover.Agriculture <- data$Percent.Cover.Agriculture
 + runif(nrow(data), min = 0, max = 0.01)
data$Percent.Cover.Impervious.Surfaces <-
data$Percent.Cover.Impervious.Surfaces + runif(nrow(data),
 min = 0, max = 0.01)
```

APPENDIX A3.1: GAM TDS

```
##TDS
gam_model1 <- gam(Total.Dissolved.Solids ~
                    bs(Percent.Cover.Urban, degree = 3) +
                    bs(Percent.Cover.Agriculture, degree = 3) +
                    bs(Percent.Cover.Impervious.Surfaces,
 degree = 3),
                  data = data)
# Summary of the model
summary(gam_model1)

print(results1)

##MSE
predictions.gam <- predict(gam_model1, newdata = data)
mse.gam <- mean((data$Total.Dissolved.Solids -
 predictions.gam)^2)
print(mse.gam)
```

APPENDIX A3.2: GAM TSS

```
##TSS
gam_model2 <- gam(Total.Suspended.Solids ~
                    bs(Percent.Cover.Urban, degree = 3) +
                    bs(Percent.Cover.Agriculture, degree = 3) +
                    bs(Percent.Cover.Impervious.Surfaces,
 degree = 3),
                  data = data)
summary(gam_model2)

##MSE
predictions.gam2 <- predict(gam_model2, newdata = data)
mse.gam2 <- mean((data$Total.Suspended.Solids -
 predictions.gam2)^2)
print(mse.gam2)
```

APPENDIX A3.3: GAM Nitrate

```
##Nitrate
gam_model3 <- gam(Nitrate ~ bs(Percent.Cover.Urban, degree = 3) +
                    bs(Percent.Cover.Agriculture, degree = 3) +
                    bs(Percent.Cover.Impervious.Surfaces,
 degree = 3), data = data)
summary(gam_model3)

##MSE
predictions.gam3 <- predict(gam_model3, newdata = data)
mse.gam3 <- mean((data$Nitrate - predictions.gam3)^2)
```

```
print(mse.gam3)
```

APPENDIX A3.4: GAM Nitrite

```
##Nitrite
gam_model4 <- gam(Nitrite ~ bs(Percent.Cover.Urban, degree = 3) +
                    bs(Percent.Cover.Agriculture, degree = 3) +
                    bs(Percent.Cover.Impervious.Surfaces,
 degree = 3), data = data)

summary(gam_model4)

##MSE
predictions.gam4 <- predict(gam_model4, newdata = data)
mse.gam4 <- mean((data$Nitrite - predictions.gam4)^2)
print(mse.gam4)
```

APPENDIX A3.5: GAM Total Phosphorus

```
##Total Phosphorus
gam_model5 <- gam(Total.Phosphorus ~ bs(Percent.Cover.Urban,
 degree = 3) +
                    bs(Percent.Cover.Agriculture, degree = 3) +
                    bs(Percent.Cover.Impervious.Surfaces,
 degree = 3), data = data)
results5 <- summary(gam_model5)
print(results5)

##MSE
predictions.gam5 <- predict(gam_model5, newdata = data)
mse.gam5 <- mean((data$Total.Phosphorus - predictions.gam5)^2)
print(mse.gam5)
```

APPENDIX A4: Random Forest Model code

```
##Random forest model
install.packages("randomForest")
library(randomForest)
```

APPENDIX A4.1: Random Forest Regression TDS

```
##TDS
rf_model1 <- randomForest(Total.Dissolved.Solids ~
 Percent.Cover.Urban + Percent.Cover.Agriculture +
Percent.Cover.Impervious.Surfaces, data = data)
summary(rf_model1)
```

```
mse1 <- rf_model1$mse
rsq1 <- rf_model1$rsq
print(mse1)
mean(mse1)
print(rsq1)

# Make predictions
predicted_valuesrf <- predict(rf_model1, newdata = data)


# Calculate R-squared
r_squaredrf <- cor(data$Total.Dissolved.Solids,
 predicted_valuesrf)^2

# Print the R-squared value
print(r_squaredrf)
```

APPENDIX A4.2: Random Forest Regression TSS

```
##TSS
rf_model2 <- randomForest(Total.Suspended.Solids ~
Percent.Cover.Urban + Percent.Cover.Agriculture +
Percent.Cover.Impervious.Surfaces, data = data)
summary(rf_model2)

# Make predictions
predicted_valuesrf2 <- predict(rf_model2, newdata = data)

# Calculate R-squared
r_squaredrf2 <- cor(data$Total.Suspended.Solids,
predicted_valuesrf2)^2

# Print the R-squared value
print(r_squaredrf2)

##MSE
mse2 <- rf_model2$mse
mean(mse2)
```

APPENDIX A4.3: Random Forest Regression Total Phosphorus

```
##Total Phosphorus
rf_model3 <- randomForest(Total.Phosphorus ~
Percent.Cover.Urban + Percent.Cover.Agriculture +
Percent.Cover.Impervious.Surfaces, data = data)
summary(rf_model3)
```

```
# Make predictions
predicted_valuesrf3 <- predict(rf_model3, newdata = data)

# Calculate R-squared
r_squaredrf3 <- cor(data$Total.Suspended.Solids,
predicted_valuesrf3)^2

# Print the R-squared value
print(r_squaredrf3)


##MSE
mse3 <- rf_model3$mse
mean(mse3)
```

APPENDIX A4.4: Random Forest Regression Nitrate

```
##Nitrate
rf_model4 <- randomForest(Nitrate ~ Percent.Cover.Urban +
Percent.Cover.Agriculture +
Percent.Cover.Impervious.Surfaces, data = data)
summary(rf_model4)

# Make predictions
predicted_valuesrf4 <- predict(rf_model4, newdata = data)

# Calculate R-squared
r_squaredrf4 <- cor(data$Total.Suspended.Solids,
predicted_valuesrf4)^2

# Print the R-squared value
print(r_squaredrf4)

##MSE
mse4 <- rf_model4$mse
mean(mse4)
```

APPENDIX A4.5: Random Forest Regression Nitrite

```
##Nitrite
rf_model5 <- randomForest(Nitrate ~ Percent.Cover.Urban +
Percent.Cover.Agriculture + Percent.Cover.Impervious.Surfaces,
 data = data)
summary(rf_model5)

# Make predictions
predicted_valuesrf5 <- predict(rf_model5, newdata = data)
```

```
# Calculate R-squared
r_squaredrf5 <- cor(data$Total.Suspended.Solids,
predicted_valuesrf5)^2

# Print the R-squared value
print(r_squaredrf5)

##MSE

mse5 <- rf_model5$mse
mean(mse5)
```

APPENDIX A5: Support Vector Machine Model code

```
##Support Vector Machine
library(e1071)

##TDS
install.packages("caret")
library(caret)
```

APPENDIX A5.1: Support Vector Machine Regression TDS

```
#SVM model
svm_model <- svm(Total.Dissolved.Solids ~ Percent.Cover.Urban +
Percent.Cover.Agriculture + Percent.Cover.Impervious.Surfaces,
 data = data)

# Make predictions
predicted_values <- predict(svm_model, newdata = data)

# Calculate R-squared using the 'caret' package
r_squared1 <- R2(data$Total.Dissolved.Solids, predicted_values)

# Print the R-squared value
print(r_squared1)

# Calculate Mean Squared Error (MSE)
mse.svm <- mean((data$Total.Dissolved.Solids -
predicted_values)^2)
print(mse.svm)


par(mfrow=c(1,2))
plot(data$Total.Dissolved.Solids, predicted_values,
     xlab = "Actual Values", ylab = "Predicted Values",
main = "Actual vs. Predicted")
abline(0, 1, col = "red")

residuals1 <- data$Total.Dissolved.Solids - predicted_values
plot(predicted_values, residuals,
     xlab = "Predicted Values", ylab = "Residuals", main =
"Residuals vs. Predicted")
abline(h = 0, col = "red", lty = 2)  # Adds a horizontal

line at y = 0
```

APPENDIX A5.2: Support Vector Machine Regression TSS

```
##TSS
svm_model2 <- svm(Total.Suspended.Solids ~ Percent.Cover.Urban
 + Percent.Cover.Agriculture +
Percent.Cover.Impervious.Surfaces, data = data)
summary(svm_model2)

# Make predictions
predicted_values2 <- predict(svm_model2, newdata = data)

# Calculate R-squared using the 'caret' package
r_squared2 <- R2(data$Total.Suspended.Solids, predicted_values2)

# Print the R-squared value
print(r_squared2)

##MSE
mse.svm2 <- mean((data$Total.Suspended.Solids -
predicted_values2)^2)
print(mse.svm2)

par(mfrow=c(1,2))
plot(data$Total.Suspended.Solids, predicted_values2,
     xlab = "Actual Values", ylab = "Predicted Values",
 main = "Actual vs. Predicted")
abline(0, 1, col = "red")

residuals1 <- data$Total.Dissolved.Solids - predicted_values
plot(predicted_values, residuals,
     xlab = "Predicted Values", ylab = "Residuals", main =
"Residuals vs. Predicted")
abline(h = 0, col = "red", lty = 2)  # Adds a horizontal line
 at y = 0
```

APPENDIX A5.3: Support Vector Machine Regression Total Phosphorus

```
##Total Phosphorus
svm_model3 <- svm(Total.Phosphorus ~ Percent.Cover.Urban +
Percent.Cover.Agriculture + Percent.Cover.Impervious.Surfaces,
 data = data)
summary(svm_model3)

# Make predictions

predicted_values3 <- predict(svm_model3, newdata = data)

# Calculate R-squared using the 'caret' package
r_squared3 <- R2(data$Total.Phosphorus, predicted_values3)

# Print the R-squared value
print(r_squared3)

##MSE
mse.svm3 <- mean((data$Total.Phosphorus - predicted_values3)^2)
print(mse.svm3)
```

APPENDIX A5.4: Support Vector Machine Regression Nitrate

```
##Nitrates
svm_model4 <- svm(Nitrate ~ Percent.Cover.Urban +
 Percent.Cover.Agriculture +
Percent.Cover.Impervious.Surfaces, data = data)
summary(svm_model4)

# Make predictions
predicted_values4 <- predict(svm_model4, newdata = data)

# Calculate R-squared using the 'caret' package
r_squared4 <- R2(data$Nitrate, predicted_values4)

# Print the R-squared value
print(r_squared4)

##MSE
mse.svm4 <- mean((data$Nitrate - predicted_values4)^2)
print(mse.svm4)
```

APPENDIX A5.5: Support Vector Machine Regression Nitrite

```
##Nitrites
svm_model5 <- svm(Nitrite ~ Percent.Cover.Urban +
 Percent.Cover.Agriculture +
 Percent.Cover.Impervious.Surfaces, data = data)
summary(svm_model5)

# Make predictions
predicted_values5 <- predict(svm_model5, newdata = data)

# Calculate R-squared using the 'caret' package
r_squared5 <- R2(data$Nitrite, predicted_values5)


# Print the R-squared value
print(r_squared5)

##MSE
mse.svm5 <- mean((data$Nitrite - predicted_values5)^2)
print(mse.svm5)
```

APPENDIX A6: Neural Network model code

```
##Neural Networks
library(neuralnet)
```

APPENDIX A6.1: Neural Network Regression TDS

```
##TDS
nn_model <- neuralnet(Total.Dissolved.Solids ~ Percent.Cover.
Urban + Percent.Cover.Agriculture +
Percent.Cover.Impervious.Surfaces, data = data)
predicted_valuesnn <- predict(nn_model, data)

# Extract the observed values
observed_values1 <- data$Total.Dissolved.Solids

# Calculate R-squared
ss_total1 <- sum((observed_values1 - mean(observed_values1))^2)
ss_residual1 <- sum((observed_values1 - predicted_valuesnn)^2)

r_squarednn <- 1 - (ss_residual1 / ss_total1)
print(r_squarednn)
#negative r-squared

##MSE
mse_nn <- mean((observed_values1 - predicted_valuesnn)^2)
print(mse_nn)
```

APPENDIX A6.2: Neural Network Regression TSS

```
##TSS
nn_model2 <- neuralnet(Total.Suspended.Solids ~
Percent.Cover.Urban + Percent.Cover.Agriculture +
Percent.Cover.Impervious.Surfaces, data = data)
predicted_valuesnn2 <- predict(nn_model2, data)
# Extract the observed values

observed_values2 <- data$Total.Suspended.Solids

# Calculate R-squared
ss_total2 <- sum((observed_values2 - mean(observed_values2))^2)
ss_residual2 <- sum((observed_values2 - predicted_valuesnn2)^2)

r_squarednn2 <- 1 - (ss_residual2 / ss_total2)
print(r_squarednn2)

##MSE
mse_nn2 <- mean((observed_values2 - predicted_valuesnn2)^2)
print(mse_nn2)
```

APPENDIX A6.3: Neural Network Regression Total Phosphorus

```
##Total Phosphorus
nn_model3 <- neuralnet(Total.Phosphorus ~ Percent.Cover.Urban +
Percent.Cover.Agriculture + Percent.Cover.Impervious.Surfaces,
 data = data)
predicted_valuesnn3 <- predict(nn_model3, data)
# Extract the observed values
observed_values3 <- data$Total.Phosphorus

# Calculate R-squared
ss_total3 <- sum((observed_values3 - mean(observed_values3))^2)
ss_residual3 <- sum((observed_values3 - predicted_valuesnn3)^2)

r_squarednn3 <- 1 - (ss_residual3 / ss_total3)
print(r_squarednn3)

##MSE
mse_nn3 <- mean((observed_values3 - predicted_valuesnn3)^2)
print(mse_nn3)
```

APPENDIX A6.4: Neural Network Regression Nitrate

```
##Nitrate
nn_model4 <- neuralnet(Nitrate ~ Percent.Cover.Urban +
Percent.Cover.Agriculture + Percent.Cover.Impervious.Surfaces
, data = data)
predicted_valuesnn4 <- predict(nn_model4, data)
# Extract the observed values
observed_values4 <- data$Nitrate

# Calculate R-squared

ss_total4 <- sum((observed_values4 - mean(observed_values4))^2)
ss_residual4 <- sum((observed_values4 - predicted_valuesnn4)^2)

r_squarednn4 <- 1 - (ss_residual4 / ss_total4)
print(r_squarednn4)

##MSE
mse_nn4 <- mean((observed_values4 - predicted_valuesnn4)^2)
print(mse_nn4)
```

APPENDIX A6.5: Neural Network Regression Nitrite

```
##Nitrite
nn_model5 <- neuralnet(Nitrite ~ Percent.Cover.Urban +
Percent.Cover.Agriculture + Percent.Cover.Impervious.Surfaces,
 data = data)
predicted_valuesnn5 <- predict(nn_model5, data)
# Extract the observed values
observed_values5 <- data$Nitrite

# Calculate R-squared
ss_total5 <- sum((observed_values5 -
mean(observed_values5))^2)
ss_residual5 <- sum((observed_values5 -
predicted_valuesnn5)^2)

r_squarednn5 <- 1 - (ss_residual5 / ss_total5)
print(r_squarednn5)

##MSE
mse_nn5 <- mean((observed_values5 - predicted_valuesnn5)^2)
print(mse_nn5)
```

APPENDIX B: Exploratory data analysis

APPENDIX B1: Logistic Regression and Predicted Probabilities Plots

```
##Recreating logistic regression
```

APPENDIX B1.1: Logistic Regression TDS

```
#TDS
# Create a binary variable indicating whether TDS exceeds the
EPA
 limit (500)
data$TDSLimitExceeded <- ifelse(data$Total.Dissolved.Solids >
500,
 1, 0)

# Create a logistic regression model
logistic_model1 <- glm(TDSLimitExceeded ~ Percent.Cover.Urban
+ Percent.Cover.Agriculture +
Percent.Cover.Impervious.Surfaces,
                      data = data,
                      family = binomial)
summary(logistic_model1)
```

APPENDIX B1.2: Logistic Regression TSS

```
#TSS
data$TSSLimitExceeded <- ifelse(data$Total.Suspended.Solids >
10,
1, 0)

# Create a logistic regression model
logistic_model2 <- glm(TSSLimitExceeded ~ Percent.Cover.Urban
+ Percent.Cover.Agriculture +
Percent.Cover.Impervious.Surfaces,
                       data = data,
                       family = binomial)
summary(logistic_model2)
```

APPENDIX B1.3: Logistic Regression Total Phosphorus

```
#Total Phosphorus
data$TPLimitExceeded <- ifelse(data$Total.Phosphorus > .1, 1,
0)

# Create a logistic regression model
```

```
logistic_model3 <- glm(TPLimitExceeded ~ Percent.Cover.Urban +
Percent.Cover.Agriculture + Percent.Cover.Impervious.Surfaces,
                       data = data,
                       family = binomial)
summary(logistic_model3)
```

APPENDIX B1.4: Logistic Regression Nitrite

```
#Nitrite
data$NitriteLimitExceeded <- ifelse(data$Nitrite > 1, 1, 0)

# Create a logistic regression model
logistic_model4 <- glm(NitriteLimitExceeded ~
Percent.Cover.Urban
+ Percent.Cover.Agriculture +
Percent.Cover.Impervious.Surfaces,
                       data = data,
                       family = binomial)
summary(logistic_model4)
```

APPENDIX B1.5: Logistic Regression Nitrate

```
#Nitrate
data$NitrateLimitExceeded <- ifelse(data$Nitrate > 10, 1, 0)

# Create a logistic regression model
logistic_model5 <- glm(NitrateLimitExceeded ~
Percent.Cover.Urban
+ Percent.Cover.Agriculture +
Percent.Cover.Impervious.Surfaces,
                       data = data,
                       family = binomial)
summary(logistic_model5)
```

APPENDIX B1.6: Predicted Probabilities Plots

```
table.TDS <- table(data$TDSLimitExceeded)
print(table.TDS)

table.TSS <- table(data$TSSLimitExceeded)
print(table.TSS)

table.TP <- table(data$TPLimitExceeded)
print(table.TP)
```

```
table.Nitrate <- table(data$NitrateLimitExceeded)

print(table.Nitrate)

table.Nitrite <- table(data$NitriteLimitExceeded)
print(table.Nitrite)

##Creating logistic regression curves for each parameter
# Generate data for prediction
new_data <- data.frame(
  Percent.Cover.Urban = seq(0, 1, by = 0.01),  # Adjust the range
 as needed
  Percent.Cover.Agriculture = 0.5,              # Use a constant
 value or adjust as needed
  Percent.Cover.Impervious.Surfaces = 0.5      # Use a constant
value or adjust as needed
)

new_data2 <- data.frame(
  Percent.Cover.Agriculture = seq(0, 1, by = 0.01),  # Adjust
the range as needed
  Percent.Cover.Urban = 0.5,              # Use a constant value
 or adjust as needed
  Percent.Cover.Impervious.Surfaces = 0.5      # Use a constant
value or adjust as needed
)

new_data3 <- data.frame(
  Percent.Cover.Impervious.Surfaces = seq(0, 1, by = 0.01),
# Adjust the range as needed
  Percent.Cover.Agriculture = 0.5,              # Use a constant
value or adjust as needed
  Percent.Cover.Urban = 0.5      # Use a constant value or adjust
as needed
)
```

APPENDIX B1.6.1: Predicted Probabilities TDS

```
##Predicted probabilities plot for TDS based on logistic
regression
pdf(file = "PredictedProbabilitiesPlots")
new_data$Predicted_ProbTDS <- predict(logistic_model1, newdata
 = new_data, type = "response")
new_data2$Predicted_ProbTDS <- predict(logistic_model1, newdata
 = new_data2, type = "response")
```

```
new_data3$Predicted_ProbTDS <- predict(logistic_model1, newdata
 = new_data3, type = "response")


library(ggplot2)

#Percent Cover Urban
ggplot(new_data, aes(x = Percent.Cover.Urban,
y = Predicted_ProbTDS)) +
  geom_line(color = "blue") +
  labs(title = "Probability of TDS Limit being Exceeded \nvs
\nPercent Cover Urban Area",
       x = "Percent Cover Urban",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))

#Percent Cover Agriculture
ggplot(new_data2, aes(x = Percent.Cover.Agriculture, y =
Predicted_ProbTDS)) +
  geom_line(color = "blue") +
  labs(title = "Probability of TDS Limit being Exceeded \nvs
\nPercent Cover Agricultural Area",
       x = "Percent Cover Agriculture",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))

#Percent Cover Impervious Surfaces
ggplot(new_data3, aes(x = Percent.Cover.Impervious.Surfaces, y =
Predicted_ProbTDS)) +
  geom_line(color = "blue") +
  labs(title = "Probability of TDS Limit being Exceeded \nvs
 \nPercent Cover by Impervious Surfaces",
       x = "Percent Cover Impervious Surfaces",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))
```

APPENDIX B1.6.2: Predicted Probabilities TSS

```
##Predicted probabilities plot for TSS based on logistic
regression
new_data$Predicted_ProbTSS <- predict(logistic_model2, newdata
 = new_data, type = "response")
new_data2$Predicted_ProbTSS <- predict(logistic_model2, newdata
 = new_data2, type = "response")
```

```
new_data3$Predicted_ProbTSS <- predict(logistic_model2, newdata
 = new_data3, type = "response")


#Percent Cover Urban
ggplot(new_data, aes(x = Percent.Cover.Urban,
y = Predicted_ProbTSS)) +
  geom_line(color = "blue") +
  labs(title = "Probability of TSS Limit being Exceeded \nvs
 \nPercent Cover Urban Area",
       x = "Percent Cover Urban",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))

#Percent Cover Agriculture
ggplot(new_data2, aes(x = Percent.Cover.Agriculture, y =
Predicted_ProbTSS)) +
  geom_line(color = "blue") +
  labs(title = "Probability of TSS Limit being Exceeded \nvs
 \nPercent Cover Agricultural Area",
       x = "Percent Cover Agriculture",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))

#Percent Cover Impervious Surfaces
ggplot(new_data3, aes(x = Percent.Cover.Impervious.Surfaces,
y = Predicted_ProbTSS)) +
  geom_line(color = "blue") +
  labs(title = "Probability of TSS Limit being Exceeded \nvs
\nPercent Cover by Impervious Surfaces",
       x = "Percent Cover Impervious Surfaces",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))
```

APPENDIX B1.6.3: Predicted Probabilities Total Phosphorus

```
##Predicted probabilities plot for Total Phosphorus based on
 logistic regression
new_data$Predicted_ProbTP <- predict(logistic_model3, newdata
 = new_data, type = "response")
new_data2$Predicted_ProbTP <- predict(logistic_model3, newdata
 = new_data2, type = "response")
new_data3$Predicted_ProbTP <- predict(logistic_model3, newdata
 = new_data3, type = "response")
```

```
#Percent Cover Urban

ggplot(new_data, aes(x = Percent.Cover.Urban,
y = Predicted_ProbTP)) +
  geom_line(color = "blue") +
  labs(title = "Probability of TotalPhosphorus Limit being
Exceeded \nvs \nPercent Cover Urban Area",
       x = "Percent Cover Urban",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))

#Percent Cover Agriculture
ggplot(new_data2, aes(x = Percent.Cover.Agriculture, y =
Predicted_ProbTP)) +
  geom_line(color = "blue") +
  labs(title = "Probability of Total Phosphorus Limit being
 Exceeded \nvs \nPercent Cover Agricultural Area",
       x = "Percent Cover Agriculture",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))

#Percent Cover Impervious Surfaces
ggplot(new_data3, aes(x = Percent.Cover.Impervious.Surfaces,
 y = Predicted_ProbTP)) +
  geom_line(color = "blue") +
  labs(title = "Probability of Total Phosphorus Limit being
 Exceeded \nvs \nPercent Cover by Impervious Surfaces",
       x = "Percent Cover Impervious Surfaces",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))
```

APPENDIX B1.6.4: Predicted Probabilities Nitrite

```
##Predicted probabilities plot for Nitrite based on logistic
 regression
new_data$Predicted_ProbNitrite <- predict(logistic_model4,
newdata = new_data, type = "response")
new_data2$Predicted_ProbNitrite <- predict(logistic_model4,
newdata = new_data2, type = "response")
new_data3$Predicted_ProbNitrite <- predict(logistic_model4,
newdata = new_data3, type = "response")

#Percent Cover Urban
```

```
ggplot(new_data, aes(x = Percent.Cover.Urban,
y = Predicted_ProbNitrite)) +

  geom_line(color = "blue") +
  labs(title = "Probability of Nitrite Limit being Exceeded
 \nvs \nPercent Cover Urban Area",
       x = "Percent Cover Urban",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))

#Percent Cover Agriculture
ggplot(new_data2, aes(x = Percent.Cover.Agriculture, y =
Predicted_ProbNitrite)) +
  geom_line(color = "blue") +
  labs(title = "Probability of Nitrite Limit being Exceeded
 \nvs \nPercent Cover Agricultural Area",
       x = "Percent Cover Agriculture",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))

#Percent Cover Impervious Surfaces
ggplot(new_data3, aes(x = Percent.Cover.Impervious.Surfaces,
 y = Predicted_ProbNitrite)) +
  geom_line(color = "blue") +
  labs(title = "Probability of Nitrite Limit being Exceeded
\nvs \nPercent Cover by Impervious Surfaces",
       x = "Percent Cover Impervious Surfaces",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))
```

APPENDIX B1.6.5: Predicted Probabilities Nitrate

```
##Predicted probabilities plot for Nitrate based on logistic
 regression
new_data$Predicted_ProbNitrate <- predict(logistic_model5,
 newdata = new_data, type = "response")
new_data2$Predicted_ProbNitrate <- predict(logistic_model5,
 newdata = new_data2, type = "response")
new_data3$Predicted_ProbNitrate <- predict(logistic_model5,
 newdata = new_data3, type = "response")

#Percent Cover Urban
ggplot(new_data, aes(x = Percent.Cover.Urban,
y = Predicted_ProbNitrate)) +
  geom_line(color = "blue") +
  labs(title = "Probability of Nitrate Limit being Exceeded

 \nvs \nPercent Cover Urban Area",
       x = "Percent Cover Urban",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))

#Percent Cover Agriculture
ggplot(new_data2, aes(x = Percent.Cover.Agriculture, y =
Predicted_ProbNitrate)) +
  geom_line(color = "blue") +
  labs(title = "Probability of Nitrate Limit being Exceeded
 \nvs \nPercent Cover Agricultural Area",
       x = "Percent Cover Agriculture",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))

#Percent Cover Impervious Surfaces
ggplot(new_data3, aes(x = Percent.Cover.Impervious.Surfaces,
 y = Predicted_ProbNitrate)) +
  geom_line(color = "blue") +
  labs(title = "Probability of Nitrate Limit being Exceeded
-oi \nvs \nPercent Cover by Impervious Surfaces",
       x = "Percent Cover Impervious Surfaces",
       y = "Predicted Probability") +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))
```

APPENDIX B2: Time Series Plot

```
##Creating time series plots for all parameters on the same
scale

##Removing an outlier from Total Phosphorus
# Value to replace with NA
value_to_replace <- c(18,15)

# Replace the specified value with NA in the column
read.wq$Total.Phosphorus <- ifelse(read.wq$Total.Phosphorus ==
value_to_replace, NA, read.wq$Total.Phosphorus)


sum(is.na(read.wq$Collection.Date))
str(read.wq$Collection.Date)
```

```
read.wq$Collection.Date <- as.Date(read.wq$Collection.Date,
format = "%m/%d/%Y")
read.wq <- read.wq[order(read.wq$Collection.Date), ]

##Time Series plot

pdf(file = "time_series_21april2024.pdf", width = 12, height =
5, paper = "special")
  par(mfrow = c(6,1))
  par(mar = c(0,4,0,4))
  par(oma = c(4,0,.2,0))
  plot(     x = as.Date(read.wq[,1], format = "%m/%d/%Y"),
            y = read.wq$temp, ylab = "Temperature (deg C)",
            pch = 20, axes = F, las = 1,
            col = read.wq$ID); box()
       axis(2, las = 1); axis(1, labels = FALSE)
  # plot(   x = as.Date(read.wq[,1], format = "%m/%d/%Y"),
            # y = read.wq$DO, ylab = "DO (mg/l)",
            # pch = 20, axes = F,
            # col = read.wq$ID); box()
       # axis(4); axis(1, labels = FALSE)
  plot(     x = as.Date(read.wq[,1], format = "%m/%d/%Y"),
            y = read.wq$tds, ylab = "TDS (mg/l)",
            pch = 20, axes = F, las = 1,
            col = read.wq$ID); box()
       axis(4, las = 1); axis(1, labels = FALSE)
  legend(   x = as.Date(read.wq[,1], format =
"%m/%d/%Y")[1]-1200,
            y = 1200,
            title = "Station ID",
            xjust = 0, pch = 20,
            horiz = T, cex = 0.85,
            legend    = sort(unique(read.wq$ID))[1:7],
            col  = sort(unique(read.wq$ID))[1:7],
            bty = "n")
  legend(   x = as.Date(read.wq[,1], format =
"%m/%d/%Y")[1]-1200,
            y = 800,
            # title = "Station ID",
            pch = 20,
            horiz = T, cex = 0.85,
            legend    = sort(unique(read.wq$ID))[8:12],
            col  = sort(unique(read.wq$ID))[8:12],
            bty = "n")
  plot(     x = as.Date(read.wq[-c(362,363),1], format =
"%m/%d/%Y"),
            y = read.wq$tp[-c(362,363)], ylab = "Total P
(mg/l)",
            pch = 20, axes = F, las = 1,
            col = read.wq$ID); box()
       axis(2, las = 1); axis(1, labels = FALSE)
  plot(     x = as.Date(read.wq[,1], format = "%m/%d/%Y"),
            y = read.wq$tss, ylab = "TSS (mg/l)",log = "y",
            pch = 20, axes = F, las = 1,
            col = read.wq$ID); box()
       axis(4, las = 1); axis(1, labels = FALSE)
  plot(     x = as.Date(read.wq[,1], format = "%m/%d/%Y"),
            y = read.wq$nitrate, ylab = "NO3 (mg/l)",log =
"y",
            pch = 20, axes = F, las = 1,
            col = read.wq$ID); box()
       axis(2, las = 1); axis(1, labels = FALSE)
  plot(     x = as.Date(read.wq[,1], format = "%m/%d/%Y"),
            y = read.wq$nitrite, ylab = "NO2 (mg/l)", log =
"y",
            pch = 20, yaxt = "n", las = 1,
            col = read.wq$ID); box()
       axis(4, las = 1)
dev.off()
```

APPENDIX B3: Two Sample t-test

```
##t tests

#Ho: There is no statistically significant difference between
the means of the two samples
#Ha: There is a statistically significant difference between
the means of the two samples
```

APPENDIX B 3.1: Total Phosphorus

```
#TP
t.test(MillCreekTP, AllenCreekTP)


Welch Two Sample t-test

data:  MillCreekTP and AllenCreekTP
t = -1.4953, df = 174.37, p-value = 0.1366
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  -0.4616047  0.0636543
sample estimates:
  mean of x  mean of y
0.06611585 0.26509105
```

APPENDIX B3.2: Nitrate

```
#Nitrate
t.test(MillCreekNitrate, AllenCreekNitrate)

Welch Two Sample t-test

data:  MillCreekNitrate and AllenCreekNitrate
t = 4.0891, df = 369.94, p-value = 5.315e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  0.1104605 0.3151092
sample estimates:
  mean of x mean of y
0.8738795 0.6610947
```

APPENDIX B3.3: Nitrite

```
#Nitrite
t.test(MillCreekNitrite, AllenCreekNitrite)

Welch Two Sample t-test

data:  MillCreekNitrite and AllenCreekNitrite
t = -3.6166, df = 264.36, p-value = 0.0003575
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  -0.012029200 -0.003548311
sample estimates:
  mean of x  mean of y
0.01202774 0.01981650
```

APPENDIX B3.4: TDS

```
#TDS

t.test(MillCreekTDS, AllenCreekTDS)

Welch Two Sample t-test

data:  MillCreekTDS and AllenCreekTDS
t = -8.8143, df = 107.65, p-value = 2.333e-14
```

```
alternative hypothesis: true difference in means is not equal
to 0
95 percent confidence interval:
  -314.2595 -198.8630
sample estimates:
  mean of x mean of y
526.8122  783.3735
```

APPENDIX B3.5: TSS

```
#TSS
t.test(MillCreekTSS, AllenCreekTSS)

Welch Two Sample t-test

data:  MillCreekTSS and AllenCreekTSS
t = 1.2104, df = 299.58, p-value = 0.2271
alternative hypothesis: true difference in means is not equal
to 0
95 percent confidence interval:
  -5.166538 21.675602
sample estimates:
  mean of x mean of y
24.07340  15.81887
```

APPENDIX B4: Principal Component Analysis

```
##PCA
data <- data[, sapply(data, is.numeric)]
pca <- princomp(data, cor = T, scores = T)
summary(pca)

plot(pca, type = "l", main = "Cumulative Proportion of
Variance Explained \nby Each Principal \nComponent")
```

APPENDIX B5: Hierarchical Clustering

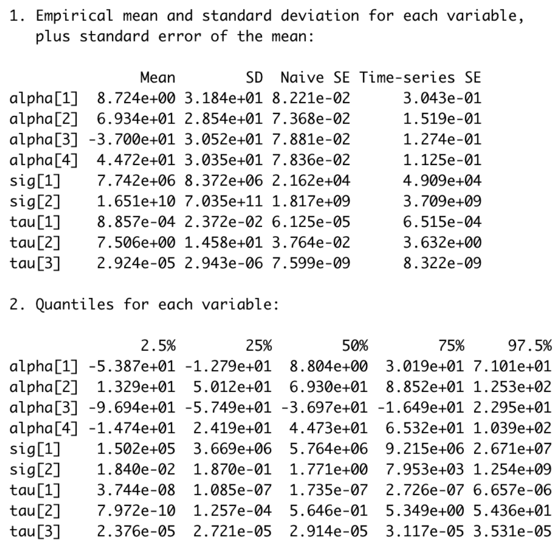
```
##Hierarchical cluster

# Select numeric columns for clustering
data <- na.omit(data)

numeric_data <- data[, sapply(data, is.numeric)]
complete_cases <- complete.cases(numeric_data)
numeric_data <- numeric_data[complete_cases, ]
```

```
# Perform hierarchical clustering using Euclidean distance and
complete linkage

hierarchical_result <- hclust(dist(data[1:6]), method =
"complete")

num_clusters <- 5  # Replace with your desired number of
clusters

# Cut the dendrogram into clusters
clusters <- cutree(hierarchical_result, k = num_clusters)

# Colors for each cluster
cluster_colors <- c("red", "blue", "green")  # Adjust the
colors as needed

hierarchical_result$labels  =
data.cluster[hierarchical_result$order,1]
dend <- as.dendrogram(hierarchical_result)

pdf("dendrogram_20april2024.pdf", paper="special", width=8,
height=5.5)
par(cex = 0.25)
plot(dend, cex = 0.2)                #, horiz = TRUE)
cluster_colors    = c("red", "blue", "green", "orange",
"purple")                   # Colors for each cluster; adjust
as needed
rect.hclust( hierarchical_result,
             k = num_clusters,
             border = cluster_colors)
dev.off()
```

APPENDIX C: R Output - OpenBugs Models

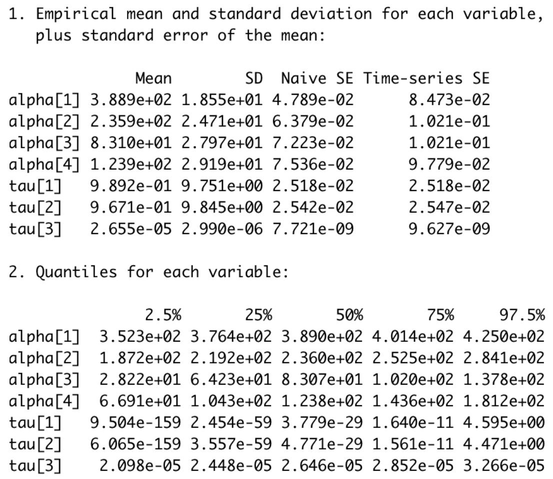
TDS:

Model Output with Covariance Matrix and Spatial Random Effect



```
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

              Mean        SD  Naive SE Time-series SE
alpha[1]  8.724e+00 3.184e+01 8.221e-02      3.043e-01
alpha[2]  6.934e+01 2.854e+01 7.368e-02      1.519e-01
alpha[3] -3.700e+01 3.052e+01 7.881e-02      1.274e-01
alpha[4]  4.472e+01 3.035e+01 7.836e-02      1.125e-01
sig[1]    7.742e+06 8.372e+06 2.162e+04      4.909e+04
sig[2]    1.651e+10 7.035e+11 1.817e+09      3.709e+09
tau[1]    8.857e-04 2.372e-02 6.125e-05      6.515e-04
tau[2]    7.506e+00 1.458e+01 3.764e-02      3.632e+00
tau[3]    2.924e-05 2.943e-06 7.599e-09      8.322e-09

2. Quantiles for each variable:

              2.5%        25%        50%        75%     97.5%
alpha[1] -5.387e+01 -1.279e+01  8.804e+00  3.019e+01 7.101e+01
alpha[2]  1.329e+01  5.012e+01  6.930e+01  8.852e+01 1.253e+02
alpha[3] -9.694e+01 -5.749e+01 -3.697e+01 -1.649e+01 2.295e+01
alpha[4] -1.474e+01  2.419e+01  4.473e+01  6.532e+01 1.039e+02
sig[1]    1.502e+05  3.669e+06  5.764e+06  9.215e+06 2.671e+07
sig[2]    1.840e-02  1.870e-01  1.771e+00  7.953e+03 1.254e+09
tau[1]    3.744e-08  1.085e-07  1.735e-07  2.726e-07 6.657e-06
tau[2]    7.972e-10  1.257e-04  5.646e-01  5.349e+00 5.436e+01
tau[3]    2.376e-05  2.721e-05  2.914e-05  3.117e-05 3.531e-05
```

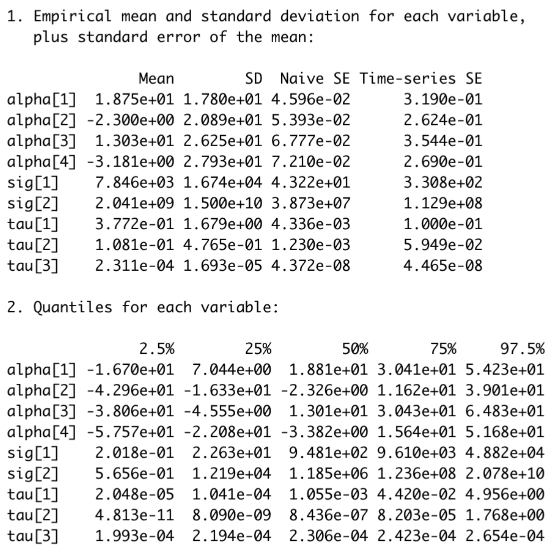
Model Output without Covariance Matrix and Spatial Random Effect



```
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

             Mean        SD  Naive SE Time-series SE
alpha[1] 3.889e+02 1.855e+01 4.789e-02      8.473e-02
alpha[2] 2.359e+02 2.471e+01 6.379e-02      1.021e-01
alpha[3] 8.310e+01 2.797e+01 7.223e-02      1.021e-01
alpha[4] 1.239e+02 2.919e+01 7.536e-02      9.779e-02
tau[1]   9.892e-01 9.751e+00 2.518e-02      2.518e-02
tau[2]   9.671e-01 9.845e+00 2.542e-02      2.547e-02
tau[3]   2.655e-05 2.990e-06 7.721e-09      9.627e-09

2. Quantiles for each variable:

              2.5%       25%       50%       75%     97.5%
alpha[1]  3.523e+02 3.764e+02 3.890e+02 4.014e+02 4.250e+02
alpha[2]  1.872e+02 2.192e+02 2.360e+02 2.525e+02 2.841e+02
alpha[3]  2.822e+01 6.423e+01 8.307e+01 1.020e+02 1.378e+02
alpha[4]  6.691e+01 1.043e+02 1.238e+02 1.436e+02 1.812e+02
tau[1]   9.504e-159 2.454e-59 3.779e-29 1.640e-11 4.595e+00
tau[2]   6.065e-159 3.557e-59 4.771e-29 1.561e-11 4.471e+00
tau[3]    2.098e-05 2.448e-05 2.646e-05 2.852e-05 3.266e-05
```

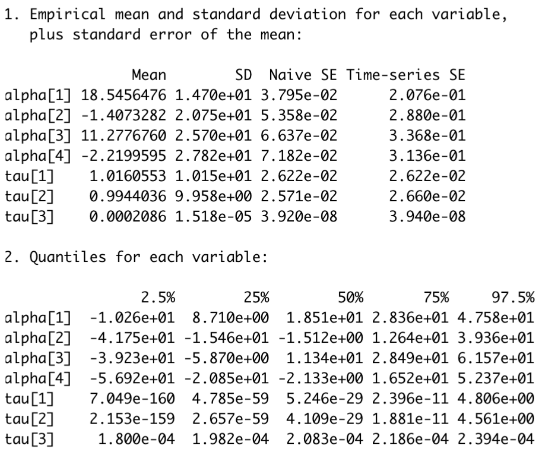
TSS:

Model Output with Covariance Matrix and Spatial Random Effect



```
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

              Mean        SD  Naive SE Time-series SE
alpha[1]  1.875e+01 1.780e+01 4.596e-02      3.190e-01
alpha[2] -2.300e+00 2.089e+01 5.393e-02      2.624e-01
alpha[3]  1.303e+01 2.625e+01 6.777e-02      3.544e-01
alpha[4] -3.181e+00 2.793e+01 7.210e-02      2.690e-01
sig[1]    7.846e+03 1.674e+04 4.322e+01      3.308e+02
sig[2]    2.041e+09 1.500e+10 3.873e+07      1.129e+08
tau[1]    3.772e-01 1.679e+00 4.336e-03      1.000e-01
tau[2]    1.081e-01 4.765e-01 1.230e-03      5.949e-02
tau[3]    2.311e-04 1.693e-05 4.372e-08      4.465e-08

2. Quantiles for each variable:

              2.5%        25%        50%       75%     97.5%
alpha[1] -1.670e+01  7.044e+00  1.881e+01 3.041e+01 5.423e+01
alpha[2] -4.296e+01 -1.633e+01 -2.326e+00 1.162e+01 3.901e+01
alpha[3] -3.806e+01 -4.555e+00  1.301e+01 3.043e+01 6.483e+01
alpha[4] -5.757e+01 -2.208e+01 -3.382e+00 1.564e+01 5.168e+01
sig[1]    2.018e-01  2.263e+01  9.481e+02 9.610e+03 4.882e+04
sig[2]    5.656e-01  1.219e+04  1.185e+06 1.236e+08 2.078e+10
tau[1]    2.048e-05  1.041e-04  1.055e-03 4.420e-02 4.956e+00
tau[2]    4.813e-11  8.090e-09  8.436e-07 8.203e-05 1.768e+00
tau[3]    1.993e-04  2.194e-04  2.306e-04 2.423e-04 2.654e-04
```

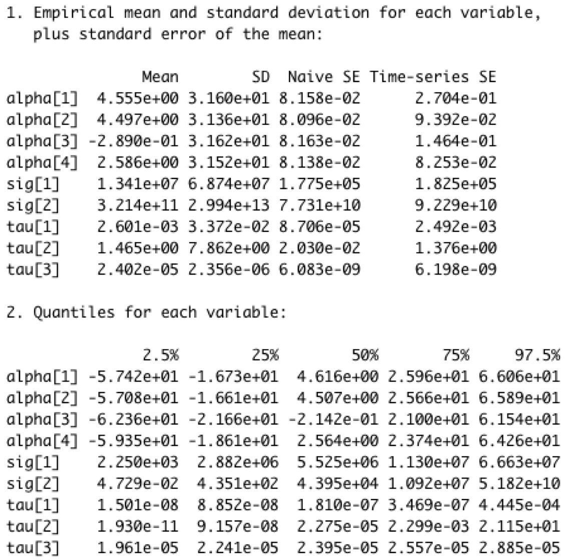
Model Output without Covariance Matrix and Spatial Random Effect



```
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

              Mean        SD  Naive SE Time-series SE
alpha[1] 18.5456476 1.470e+01 3.795e-02      2.076e-01
alpha[2] -1.4073282 2.075e+01 5.358e-02      2.880e-01
alpha[3] 11.2776760 2.570e+01 6.637e-02      3.368e-01
alpha[4] -2.2199595 2.782e+01 7.182e-02      3.136e-01
tau[1]    1.0160553 1.015e+01 2.622e-02      2.622e-02
tau[2]    0.9944036 9.958e+00 2.571e-02      2.660e-02
tau[3]    0.0002086 1.518e-05 3.920e-08      3.940e-08

2. Quantiles for each variable:

              2.5%        25%        50%       75%     97.5%
alpha[1]  -1.026e+01  8.710e+00  1.851e+01 2.836e+01 4.758e+01
alpha[2]  -4.175e+01 -1.546e+01 -1.512e+00 1.264e+01 3.936e+01
alpha[3]  -3.923e+01 -5.870e+00  1.134e+01 2.849e+01 6.157e+01
alpha[4]  -5.692e+01 -2.085e+01 -2.133e+00 1.652e+01 5.237e+01
tau[1]    7.049e-160  4.785e-59  5.246e-29 2.396e-11 4.806e+00
tau[2]    2.153e-159  2.657e-59  4.109e-29 1.881e-11 4.561e+00
tau[3]     1.800e-04  1.982e-04  2.083e-04 2.186e-04 2.394e-04
```

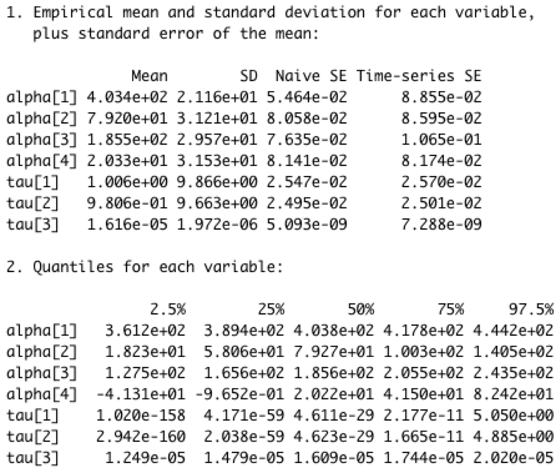
Total Phosphorus:

Model Output with Covariance Matrix and Spatial Random Effect



```
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

              Mean        SD  Naive SE Time-series SE
alpha[1]  4.555e+00 3.160e+01 8.158e-02      2.704e-01
alpha[2]  4.497e+00 3.136e+01 8.096e-02      9.392e-02
alpha[3] -2.890e-01 3.162e+01 8.163e-02      1.464e-01
alpha[4]  2.586e+00 3.152e+01 8.138e-02      8.253e-02
sig[1]    1.341e+07 6.874e+07 1.775e+05      1.825e+05
sig[2]    3.214e+11 2.994e+13 7.731e+10      9.229e+10
tau[1]    2.601e-03 3.372e-02 8.706e-05      2.492e-03
tau[2]    1.465e+00 7.862e+00 2.030e-02      1.376e+00
tau[3]    2.402e-05 2.356e-06 6.083e-09      6.198e-09

2. Quantiles for each variable:

              2.5%        25%        50%       75%     97.5%
alpha[1] -5.742e+01 -1.673e+01  4.616e+00 2.596e+01 6.606e+01
alpha[2] -5.708e+01 -1.661e+01  4.507e+00 2.566e+01 6.589e+01
alpha[3] -6.236e+01 -2.166e+01 -2.142e-01 2.100e+01 6.154e+01
alpha[4] -5.935e+01 -1.861e+01  2.564e+00 2.374e+01 6.426e+01
sig[1]    2.250e+03  2.882e+06  5.525e+06 1.130e+07 6.663e+07
sig[2]    4.729e-02  4.351e+02  4.395e+04 1.092e+07 5.182e+10
tau[1]    1.501e-08  8.852e-08  1.810e-07 3.469e-07 4.445e-04
tau[2]    1.930e-11  9.157e-08  2.275e-05 2.299e-03 2.115e+01
tau[3]    1.961e-05  2.241e-05  2.395e-05 2.557e-05 2.885e-05
```

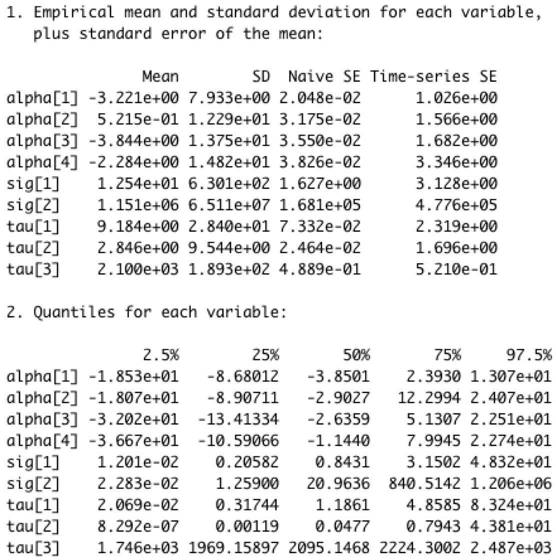
Model Output without Covariance Matrix and Spatial Random Effect



```
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

             Mean        SD  Naive SE Time-series SE
alpha[1] 4.034e+02 2.116e+01 5.464e-02      8.855e-02
alpha[2] 7.920e+01 3.121e+01 8.058e-02      8.595e-02
alpha[3] 1.855e+02 2.957e+01 7.635e-02      1.065e-01
alpha[4] 2.033e+01 3.153e+01 8.141e-02      8.174e-02
tau[1]   1.006e+00 9.866e+00 2.547e-02      2.570e-02
tau[2]   9.806e-01 9.663e+00 2.495e-02      2.501e-02
tau[3]   1.616e-05 1.972e-06 5.093e-09      7.288e-09

2. Quantiles for each variable:

               2.5%        25%       50%       75%     97.5%
alpha[1]   3.612e+02  3.894e+02 4.038e+02 4.178e+02 4.442e+02
alpha[2]   1.823e+01  5.806e+01 7.927e+01 1.003e+02 1.405e+02
alpha[3]   1.275e+02  1.656e+02 1.856e+02 2.055e+02 2.435e+02
alpha[4]  -4.131e+01 -9.652e-01 2.022e+01 4.150e+01 8.242e+01
tau[1]    1.020e-158  4.171e-59 4.611e-29 2.177e-11 5.050e+00
tau[2]    2.942e-160  2.038e-59 4.623e-29 1.665e-11 4.885e+00
tau[3]     1.249e-05  1.479e-05 1.609e-05 1.744e-05 2.020e-05
```

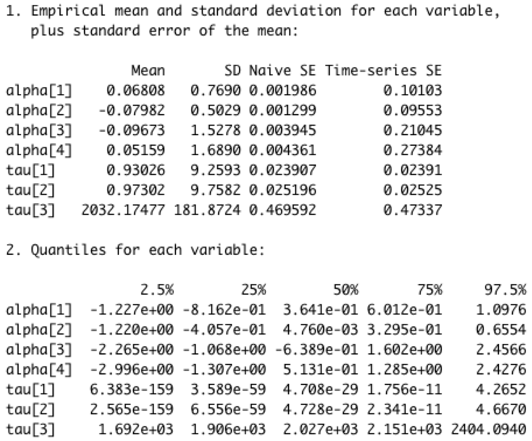
Nitrite:

Model Output with Covariance Matrix and Spatial Random Effect



```
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

              Mean        SD  Naive SE Time-series SE
alpha[1] -3.221e+00 7.933e+00 2.048e-02      1.026e+00
alpha[2]  5.215e-01 1.229e+01 3.175e-02      1.566e+00
alpha[3] -3.844e+00 1.375e+01 3.550e-02      1.682e+00
alpha[4] -2.284e+00 1.482e+01 3.826e-02      3.346e+00
sig[1]    1.254e+01 6.301e+02 1.627e+00      3.128e+00
sig[2]    1.151e+06 6.511e+07 1.681e+05      4.776e+05
tau[1]    9.184e+00 2.840e+01 7.332e-02      2.319e+00
tau[2]    2.846e+00 9.544e+00 2.464e-02      1.696e+00
tau[3]    2.100e+03 1.893e+02 4.889e-01      5.210e-01

2. Quantiles for each variable:

              2.5%        25%       50%       75%     97.5%
alpha[1] -1.853e+01   -8.68012   -3.8501    2.3930 1.307e+01
alpha[2] -1.807e+01   -8.90711   -2.9027   12.2994 2.407e+01
alpha[3] -3.202e+01  -13.41334   -2.6359    5.1307 2.251e+01
alpha[4] -3.667e+01  -10.59066   -1.1440    7.9945 2.274e+01
sig[1]    1.201e-02    0.20582    0.8431    3.1502 4.832e+01
sig[2]    2.283e-02    1.25900   20.9636  840.5142 1.206e+06
tau[1]    2.069e-02    0.31744    1.1861    4.8585 8.324e+01
tau[2]    8.292e-07    0.00119    0.0477    0.7943 4.381e+01
tau[3]    1.746e+03 1969.15897 2095.1468 2224.3002 2.487e+03
```

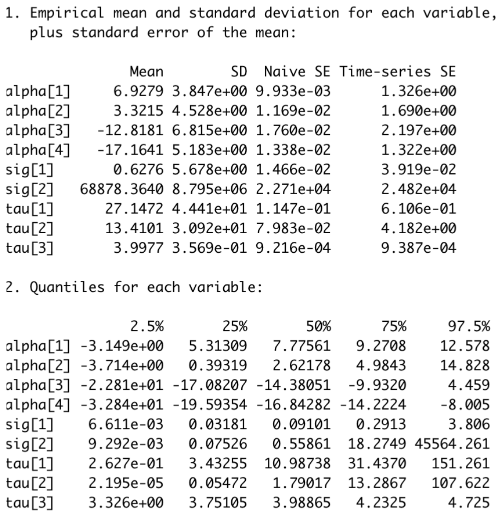
Model Output without Covariance Matrix and Spatial Random Effect



```
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

              Mean       SD Naive SE Time-series SE
alpha[1]    0.06808   0.7690 0.001986        0.10103
alpha[2]   -0.07982   0.5029 0.001299        0.09553
alpha[3]   -0.09673   1.5278 0.003945        0.21045
alpha[4]    0.05159   1.6890 0.004361        0.27384
tau[1]      0.93026   9.2593 0.023907        0.02391
tau[2]      0.97302   9.7582 0.025196        0.02525
tau[3]   2032.17477 181.8724 0.469592        0.47337

2. Quantiles for each variable:

               2.5%        25%        50%       75%     97.5%
alpha[1]  -1.227e+00 -8.162e-01  3.641e-01 6.012e-01    1.0976
alpha[2]  -1.220e+00 -4.057e-01  4.760e-03 3.295e-01    0.6554
alpha[3]  -2.265e+00 -1.068e+00 -6.389e-01 1.602e+00    2.4566
alpha[4]  -2.996e+00 -1.307e+00  5.131e-01 1.285e+00    2.4276
tau[1]    6.383e-159  3.589e-59  4.708e-29 1.756e-11    4.2652
tau[2]    2.565e-159  6.556e-59  4.728e-29 2.341e-11    4.6670
tau[3]     1.692e+03  1.906e+03  2.027e+03 2.151e+03 2404.0940
```

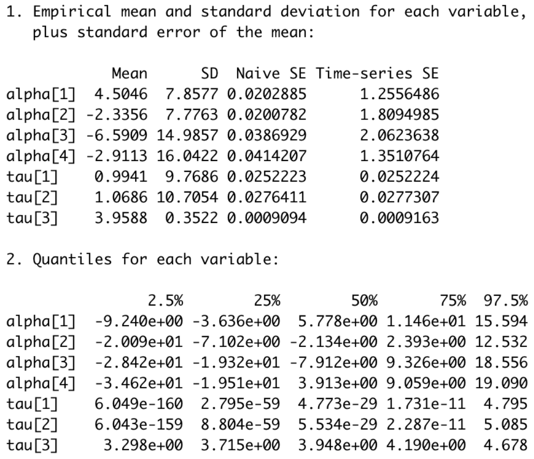
Nitrate:

Model Output with Covariance Matrix and Spatial Random Effect



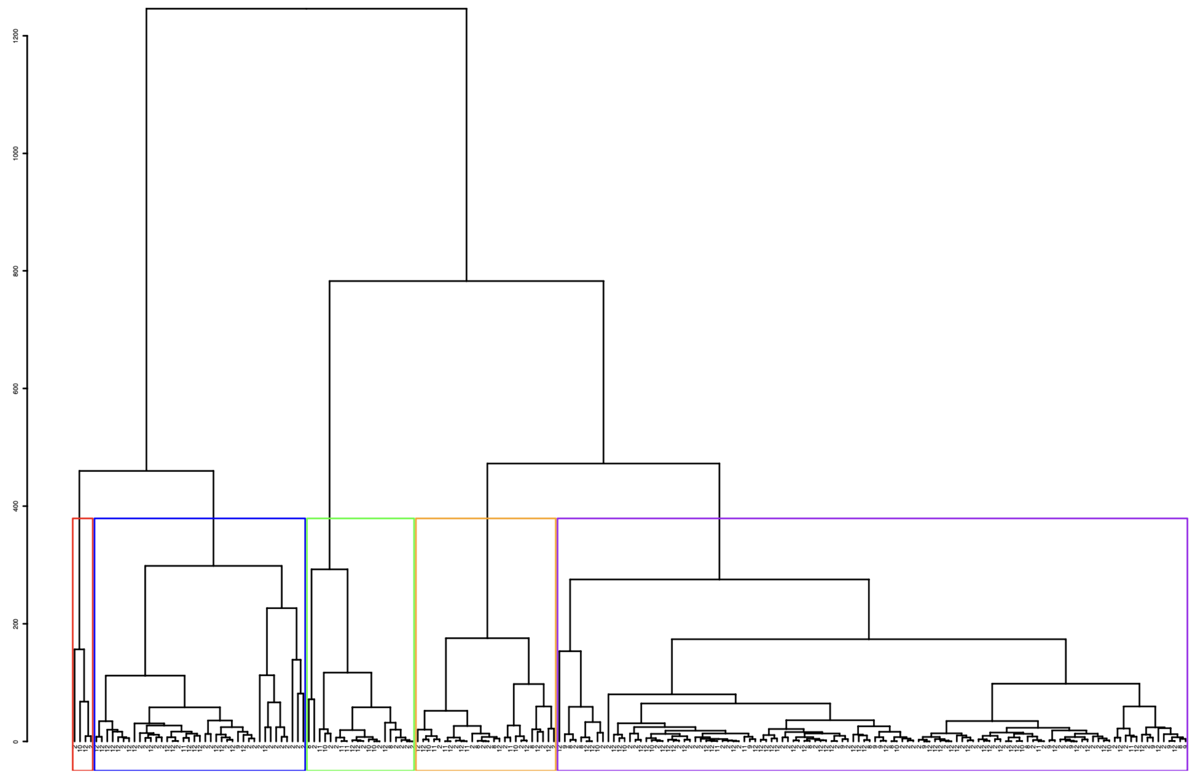
```
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

             Mean        SD  Naive SE Time-series SE
alpha[1]     6.9279 3.847e+00 9.933e-03      1.326e+00
alpha[2]     3.3215 4.528e+00 1.169e-02      1.690e+00
alpha[3]   -12.8181 6.815e+00 1.760e-02      2.197e+00
alpha[4]   -17.1641 5.183e+00 1.338e-02      1.322e+00
sig[1]       0.6276 5.678e+00 1.466e-02      3.919e-02
sig[2]   68878.3640 8.795e+06 2.271e+04      2.482e+04
tau[1]      27.1472 4.441e+01 1.147e-01      6.106e-01
tau[2]      13.4101 3.092e+01 7.983e-02      4.182e+00
tau[3]       3.9977 3.569e-01 9.216e-04      9.387e-04

2. Quantiles for each variable:

              2.5%       25%       50%      75%     97.5%
alpha[1] -3.149e+00   5.31309   7.77561   9.2708    12.578
alpha[2] -3.714e+00   0.39319   2.62178   4.9843    14.828
alpha[3] -2.281e+01 -17.08207 -14.38051  -9.9320     4.459
alpha[4] -3.284e+01 -19.59354 -16.84282 -14.2224    -8.005
sig[1]    6.611e-03   0.03181   0.09101   0.2913     3.806
sig[2]    9.292e-03   0.07526   0.55861  18.2749 45564.261
tau[1]    2.627e-01   3.43255  10.98738  31.4370   151.261
tau[2]    2.195e-05   0.05472   1.79017  13.2867   107.622
tau[3]    3.326e+00   3.75105   3.98865   4.2325     4.725
```

Model Output without Covariance Matrix and Spatial Random Effect



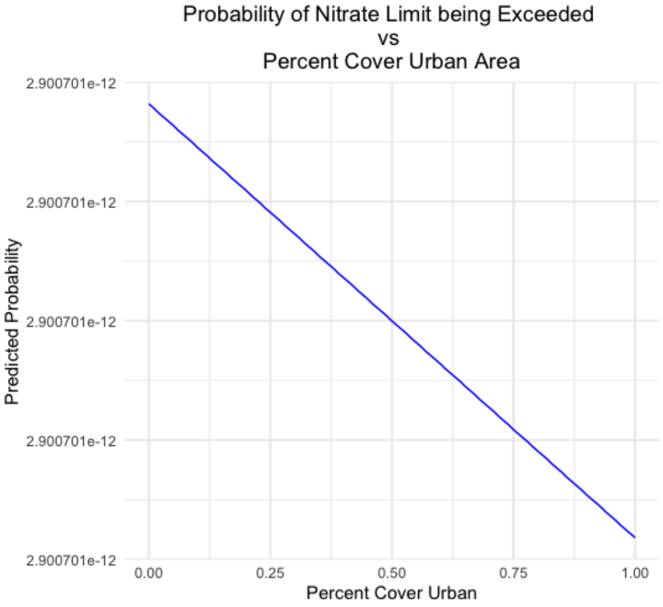
```
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

           Mean      SD  Naive SE Time-series SE
alpha[1]  4.5046  7.8577 0.0202885      1.2556486
alpha[2] -2.3356  7.7763 0.0200782      1.8094985
alpha[3] -6.5909 14.9857 0.0386929      2.0623638
alpha[4] -2.9113 16.0422 0.0414207      1.3510764
tau[1]    0.9941  9.7686 0.0252223      0.0252224
tau[2]    1.0686 10.7054 0.0276411      0.0277307
tau[3]    3.9588  0.3522 0.0009094      0.0009163

2. Quantiles for each variable:

               2.5%        25%        50%       75%  97.5%
alpha[1]  -9.240e+00 -3.636e+00  5.778e+00 1.146e+01 15.594
alpha[2]  -2.009e+01 -7.102e+00 -2.134e+00 2.393e+00 12.532
alpha[3]  -2.842e+01 -1.932e+01 -7.912e+00 9.326e+00 18.556
alpha[4]  -3.462e+01 -1.951e+01  3.913e+00 9.059e+00 19.090
tau[1]    6.049e-160  2.795e-59  4.773e-29 1.731e-11  4.795
tau[2]    6.043e-159  8.804e-59  5.534e-29 2.287e-11  5.085
tau[3]     3.298e+00  3.715e+00  3.948e+00 4.190e+00  4.678
```

APPENDIX D: Hierarchical Clustering Dendrogram

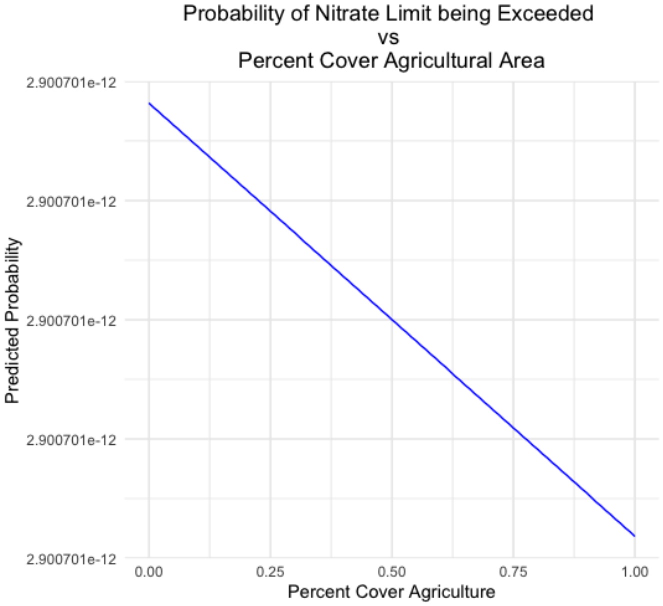


Dendrogram displaying the relative sizes of each cluster in the analysis. Cluster 1 is shown in purple, 2 in blue, 3 in green, 4 in red, and 5 in orange.

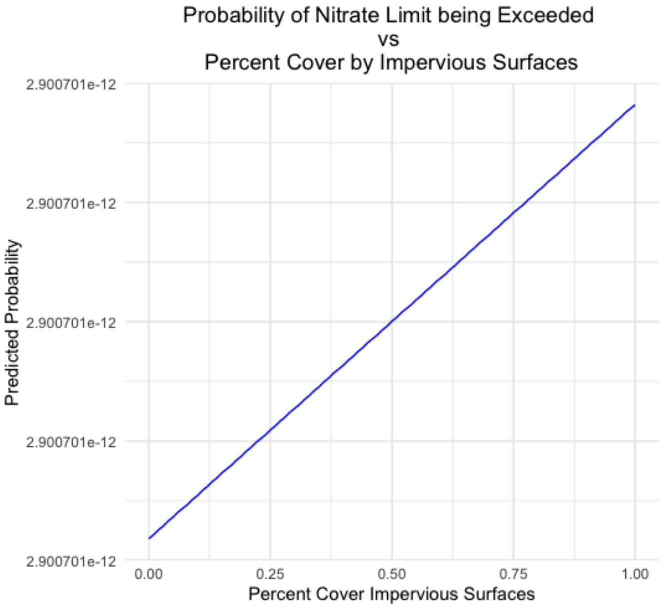
APPENDIX E: Nitrate Logistic Regression Results



Predicted probabilities plot showing the likelihood of nitrate exceeding its EPA limit as percent cover by urban area increases. The probability of exceedance displays a negative relationship but remains at 2.900701x10^{-12} throughout.



Predicted probabilities plot showing the likelihood of nitrate exceeding its EPA limit as percent cover by agricultural area increases. The probability of exceedance displays a negative relationship but remains at 2.900701x10^{-12} throughout.



Predicted probabilities plot showing the likelihood of nitrate exceeding its EPA limit as percent cover by impervious surfaces increases. The probability of exceedance displays a positive relationship but remains at 2.900701x10^{-12} throughout.

Bibliography

Adjovu, G. E., Stephen, H., James, D., & Ahmad, S. (2023). Measurement of Total Dissolved Solids and Total Suspended Solids in Water Systems: A Review of the Issues, Conventional, and Remote Sensing Techniques. *Remote Sensing*, *15*(14), 3534. https://doi.org/10.3390/rs15143534

Anh, N. T., Can, L. D., Nhan, N. T., Schmalz, B., & Luu, T. L. (2023). Influences of key factors on river water quality in urban and rural areas: A review. *Case Studies in Chemical and Environmental Engineering*, *8*, 100424. https://doi.org/10.1016/j.cscee.2023.100424

Biau, G.´rard. (2012). Analysis of a Random Forests Model. *Journal of Machine Learning Research*, *13*, 1063–1095.

Chambers, J. M. and Hastie, T. J. (1992) *Statistical Models in S*. Wadsworth & Brooks/Cole

Chen, Y., Song, L., Liu, Y., Yang, L., & Li, D. (2020). A Review of the Artificial Neural Network Models for Water Quality Prediction. *Applied Sciences*, *10*(17), 5776. https://doi.org/10.3390/app10175776

De Oliveira, L. M., Maillard, P., & De Andrade Pinto, É. J. (2016). Modeling the effect of land use/land cover on nitrogen, phosphorous and dissolved oxygen loads in the Velhas River using the concept of exclusive contribution area. *Environmental Monitoring and Assessment*, *188*(6), 333. https://doi.org/10.1007/s10661-016-5323-2

Dormann, C. F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., Marquéz, J. R. G., Gruber, B., Lafourcade, B., Leitão, P. J., Münkemüller, T., McClean, C., Osborne, P. E., Reineking, B., Schröder, B., Skidmore, A. K., Zurell, D., & Lautenbach, S. (2013). Collinearity: A review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, *36*(1), 27–46. https://doi.org/10.1111/j.1600-0587.2012.07348.x

Estimated Nitrate Concentrations in Groundwater Used for Drinking. (n.d.). Environmental Protection Agency.

Faraggi, D., & Simon, R. (1995). A neural network model for survival data. *Statistics in Medicine*, *14*(1), 73–82. https://doi.org/10.1002/sim.4780140108

Goodspeed, R., Wang, R., Lizundia, C., Du, L., & Jaipuria, S. (2023). Incorporating water quality into land use scenario analysis with random forest models. *Environment and Planning B: Urban Analytics and City Science*, *50*(6), 1518–1533. https://doi.org/10.1177/23998083221138842

Hassan, Z. U., Shah, J. A., Kanth, T. A., & Pandit, A. K. (2015). Influence of land use/land cover on the water chemistry of Wular Lake in Kashmir Himalaya (India). *Ecological Processes*, *4*(1), 9. https://doi.org/10.1186/s13717-015-0035-z

Hastie, T., & Tibshirani, R. (1987). Generalized Additive Models: Some Applications. *Journal of the American Statistical Association*, *82*(398), 371–386. https://doi.org/10.1080/01621459.1987.10478440

Hay-Chmielewski, E. M., Seelbach, P. W., Whelan, G. E., & Jester Jr., D. B. (1995). *Huron River Assessment* (Fisheries Special Report 16). Michigan State Department of Natural Resources

Huang, J., Zhan, J., Yan, H., Wu, F., & Deng, X. (2013). Evaluation of the Impacts of Land Use on Water Quality: A Case Study in The Chaohu Lake Basin. *The Scientific World Journal*, *2013*, 1–7. https://doi.org/10.1155/2013/329187

Lei, C., & Zhu, L. (2018). Spatio-temporal variability of land use/land cover change (LULCC) within the Huron River: Effects on stream flows. *Climate Risk Management*, *19*, 35–47. https://doi.org/10.1016/j.crm.2017.09.002

Li, Y., Boswell, E., & Thompson, A. (2021). Correlations between land use and stream nitrate-nitrite concentrations in the Yahara River Watershed in south-central Wisconsin. *Journal of Environmental Management*, *278*, 111535. https://doi.org/10.1016/j.jenvman.2020.111535

Litke, D. W. (1999). *Review of Phosphorus Control Measures in the United States and Their Effects on Water Quality*. U.S. GEOLOGICAL SURVEY. https://pubs.usgs.gov/wri/wri994007/pdf/wri99-4007.pdf

Maćkiewicz, A., & Ratajczak, W. (1993). Principal components analysis (PCA). *Computers & Geosciences*, *19*(3), 303–342. https://doi.org/10.1016/0098-3004(93)90090-R

Mahmoodi, M. (2019). LINKING LAND USE CHANGES TO VARIATION IN SURFACE WATER QUALITY: EVIDENCE FROM 36 CATCHMENTS IN IRAN. *Applied Ecology and Environmental Research*, *17*(4). https://doi.org/10.15666/aeer/1704_81518169

Mainstone, C. P., & Parr, W. (2002). Phosphorus in rivers—Ecology and management. *Science of The Total Environment*, *282–283*, 25–47. https://doi.org/10.1016/S0048-9697(01)00937-8

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge university press.

MDNR. 2002. Huron River Plan.in M. D. o. N. Resources, editor.

Mondal, K. C., Rathod, K. G., Joshi, H. M., Mandal, H. S., Khan, R., Rajendra, K., Mawale, Y. K., Priya, K., & Jhariya, D. C. (2020). Impact of Land-use and Land-cover Change on Groundwater Quality and Quantity in the Raipur, Chhattisgarh, India: A Remote Sensing and GIS approach. *IOP Conference Series: Earth and Environmental Science*, *597*(1), 012011. https://doi.org/10.1088/1755-1315/597/1/012011

Moorcroft, M. (2001). Detection and determination of nitrate and nitrite: A review. *Talanta*, *54*(5), 785–803. https://doi.org/10.1016/S0039-9140(01)00323-X

Morée, A. L., Beusen, A. H. W., Bouwman, A. F., & Willems, W. J. (2013). Exploring global nitrogen and phosphorus flows in urban wastes during the twentieth century. *Global Biogeochemical Cycles*, *27*(3), 836–846. https://doi.org/10.1002/gbc.20072

Murphy, R. R., Perry, E., Harcum, J., & Keisman, J. (2019). A Generalized Additive Model approach to evaluating water quality: Chesapeake Bay case study. *Environmental Modelling & Software*, *118*, 1–13. https://doi.org/10.1016/j.envsoft.2019.03.027

Ouedraogo, I., Defourny, P., & Vanclooster, M. (2019). Application of random forest regression and comparison of its performance to multiple linear regression in modeling groundwater nitrate concentration at the African continent scale. *Hydrogeology Journal*, *27*(3), 1081–1098. https://doi.org/10.1007/s10040-018-1900-5

"Our Watershed." Huron River Watershed Council.https://www.hrwc.org/our-watershed/

Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine. In *Machine Learning* (pp. 101–121). Elsevier. https://doi.org/10.1016/B978-0-12-815739-8.00006-7

R Core Team (2022). R: A language and environment for statistical computing. R Foundation for

Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Secondary drinking water standards: Guidance for nuisance ... - US EPA. (n.d.-c).

https://www.epa.gov/sdwa/secondary-drinking-water-standards-guidance-nuisance-chemicals

Shaikhina, T., Lowe, D., Daga, S., Briggs, D., Higgins, R., & Khovanova, N. (2019). Decision tree and random forest models for outcome prediction in antibody incompatible kidney transplantation. *Biomedical Signal Processing and Control*, *52*, 456–462. https://doi.org/10.1016/j.bspc.2017.01.012

Singh, K. P., Basant, N., & Gupta, S. (2011). Support vector machines in water quality management. *Analytica Chimica Acta*, *703*(2), 152–162. https://doi.org/10.1016/j.aca.2011.07.027

Sunardi, S., Nursamsi, I., Dede, M., Paramitha, A., Arief, M. C. Wi., Ariyani, M., & Santoso, P. (2022). Assessing the Influence of Land-Use Changes on Water Quality Using Remote

Sensing and GIS: A Study in Cirata Reservoir, Indonesia. *Science and Technology Indonesia*, *7*(1), 106–114. https://doi.org/10.26554/sti.2022.7.1.106-114

Table 1. summary of contract required detection limits ... - US EPA. (n.d.-b).

https://19january2017snapshot.epa.gov/sites/production/files/2015-06/documents/160_2.pdf

Tong, S. T. Y., & Chen, W. (2002). Modeling the relationship between land use and surface water quality. *Journal of Environmental Management*, *66*(4), 377–393. https://doi.org/10.1006/jema.2002.0593

Trafalis, T. B., & Ince, H. (2000). Support vector machine for regression and applications to financial forecasting. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, 348–353 vol.6. https://doi.org/10.1109/IJCNN.2000.859420

Umwali, E. D., Kurban, A., Isabwe, A., Mind'je, R., Azadi, H., Guo, Z., Udahogora, M., Nyirarwasa, A., Umuhoza, J., Nzabarinda, V., Gasirabo, A., & Sabirhazi, G. (2021). Spatio-seasonal variation of water quality influenced by land use and land cover in Lake Muhazi. *Scientific Reports*, *11*(1), 17376. https://doi.org/10.1038/s41598-021-96633-9

Wilson, C. O. (2015). Land use/land cover water quality nexus: Quantifying anthropogenic influences on surface water quality. *Environmental Monitoring and Assessment*, *187*(7), 424. https://doi.org/10.1007/s10661-015-4666-4

Wu, C., Wu, J., Qi, J., Zhang, L., Huang, H., Lou, L., & Chen, Y. (2010). Empirical estimation of total phosphorus concentration in the mainstream of the Qiantang River in

China using Landsat TM data. *International Journal of Remote Sensing*, *31*(9), 2309–2324. https://doi.org/10.1080/01431160902973873