Patch Reservoir Data Mining Project

By Jefferson Bourguignon Coutinho Project Advisor DR. Elena Braynova

### Abstract:

This project was aimed at analyzing the data collected by Worcester State from Patch Reservoir. The idea for choosing this data set comes from the direct geographical and social association of this data set and all my peers, as well as the availability of experts in the field directly connected with the data collection. My goal was if possible, make a positive increment to the body of data by showing where the data is useful in yielding insights, and where the data collection or structure could be changed to yield a better data set more friendly to processing. My question was broad and general because my idea was to extract meaning from a field, I’m not familiar. How are my target variables changing over time and how do they associate with each other if any association can be inferred at all? My findings were inconclusive, pending further analysis. The data showed weak associations of important variables such as water temperature, dissolved oxygen, and PH. The timeline shows interesting results of a downward trend in all the fore-mentioned variables. I decided to focus on temperature and found that bias was introduced in 2021 by adding data from Cook Pond where the temperature data is considerably lower than Patch Reservoir. This does not invalidate the insight but raises more questions and introduce the chance to evaluate and improve the collection process.

### Introduction

The data for this project is closely related to the climate and the environmental changes in our immediate surrounding. The choice of this dataset was made purposely for this reason. The Patch Reservoir water body is located very close to the campus, and it is a good sample of the hydrographical health of our immediate surroundings. Weather patterns are changing rapidly so finding out how these changes affect us locally can help us plan for a most likely uncertain future. My goal in this project is to analyze the most important values in the system, Temperature, Dissolved Oxygen and PH. To show a timeline of how these features change over time and why they relate to each if any relation exists at all.

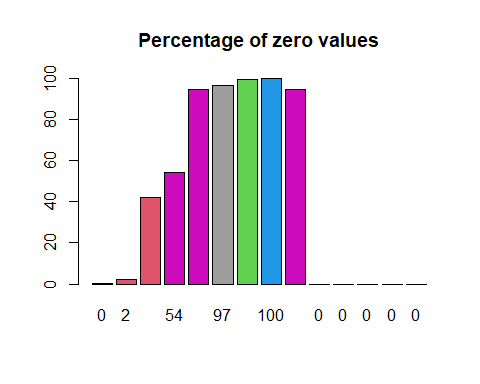
### Dataset

df <- read.csv('fullData.csv')  
glimpse(df)

## Rows: 582  
## Columns: 14  
## $ index <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16~  
## $ Temperature\_C <dbl> 24.0, 25.1, 24.0, 24.1, 24.3, 23.3, 25.1, 24.2, 22.9, 23~  
## $ pH <dbl> 0.00, 7.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.~  
## $ DO <dbl> 9.28, 10.00, 9.35, 9.74, 9.60, 9.06, 9.80, 8.80, 8.74, 9~  
## $ Depth\_Ft <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ Secchi\_Depth <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ Phosphorus <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ Nitrates <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ Conductivi <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ GlobalID <chr> "{229D5C15-B68E-45F2-945F-BF9216EEF7C1}", "{58DD7E0E-E77~  
## $ CreationDa <chr> "9/13/2017", "9/13/2017", "9/13/2017", "9/13/2017", "9/1~  
## $ EditDate <chr> "9/13/2017", "9/13/2017", "9/13/2017", "9/13/2017", "9/1~  
## $ Latitude <dbl> 42.26913, 42.27107, 42.26744, 42.26887, 42.26870, 42.266~  
## $ Longitude <dbl> -71.85201, -71.84932, -71.85319, -71.85174, -71.85155, -~

My data set consisted of 582 instances out of 20 attributes of which 11 are numeric and 9 nominals. This version of the data is the final version in which I engineered some of the datum to generate new attributes suitable for working with the models that I wanted to use.

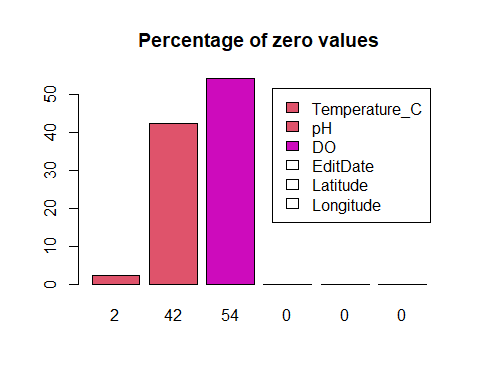
#png(file="percentageofzeroesTest.png")  
zerosPercentage <- colSums(df==0)/nrow(df)\*100 #getting the percentage of zeroes  
  
barplot(zerosPercentage, main = "Percentage of zero values", col = zerosPercentage, names.arg = round(zerosPercentage, 0))



#dev.off()#saving png file

One of the problems with my original dataset was the number of zeroes and the fact that these zeroes were most likely missing data. In fact my project advisor DR. Braynova and field expert DR. Hansen have advised me that these datum were missing from the data and do not represent a value, as well as a few other values well over their upper boundaries. After creating a bar graph to visualize the percentage of missing data I decided to remove these columns.

#removing unwanted columns  
removeUnwantedColumns <- c('index', 'Phosphorus', 'Nitrates', 'Depth\_Ft', 'Secchi\_Depth', 'Conductivi', 'CreationDa', 'GlobalID')  
cleanedDf = df[!(names(df)) %in% removeUnwantedColumns]  
  
#png(file="cleanedpercentageofzeroes.png")  
cleanedzerosPercentage <- colSums(cleanedDf==0)/nrow(cleanedDf)\*100 #getting the percentage of zeroes  
  
barplot(cleanedzerosPercentage, main = "Percentage of zero values", col = cleanedzerosPercentage, legend = TRUE, names.arg = round(cleanedzerosPercentage, 0) )



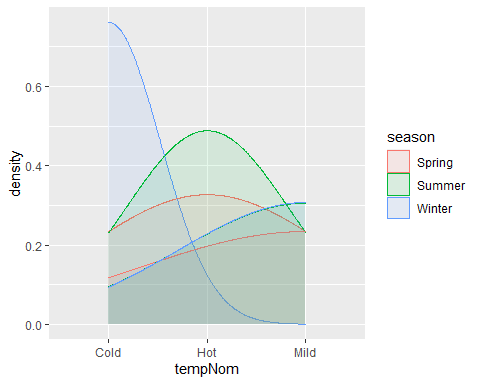
#dev.off()#saving png file

After generating a barplot of the new data set while removing all of the data that had more than 70 percent of missing data, I was left with two very important attributes, PH and dissolved Oxygen. They shared each about half of its values as missing data. These attributes would have to be dealt with separately to maintain the integrity of the data and to extract as much meaning out of it as possible.

df <- read.csv("fullData.csv")  
  
df$date <- as.Date(df$CreationDa, "%m/%d/%Y")#remember uppercase Y  
monthNom <- format(df$date, "%b")  
df$monthNom <- monthNom  
month <- format(df$date, "%m")  
df$month <- month  
year <- format(df$date, "%y")  
df$year <- year  
days <- format(df$date, "%d")  
df$days <- days  
monthYear <- format(df$date, "%b-%y")  
df$monthYear <- monthYear  
  
#write.csv(df, file = "machineLearning.csv")

One of the things I had to work with was duplicate date with misleading names; date of Creation, and Date of update, so I removed one of them. Next, I realize that the data was missing categorical values which are very important to run certain models. The date presented a very good solution to this problem because it could be grouped and divided into discrete ranges of values. I decided to extract as much as I could out of date; month as a nominal; year as a number; days as a number; month and year as a nominal.

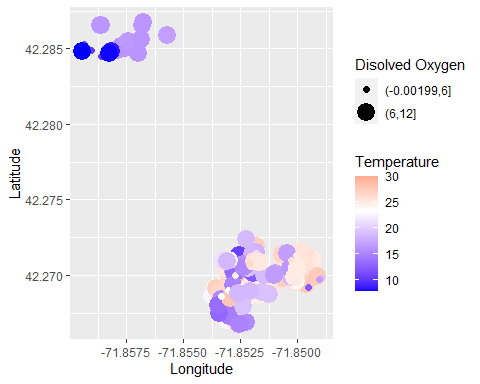
df$tempNom <- as.factor(ifelse(df$Temperature\_C < 15, "Cold",  
 ifelse(df$Temperature\_C <= 25, "Mild",  
 ifelse(df$Temperature\_C < 38, "Hot", "Too hot"))))  
  
df$phNom <- as.factor(ifelse(df$pH < 4, "Really bad, the fish will die",  
 ifelse(df$pH < 5, "Bad; frog eggs, tadpoles, and mayflies die",  
 ifelse(df$pH < 6.0, "Rainbow trout begin to die",  
 ifelse(df$pH < 8, "healthy water",  
 ifelse(df$pH < 9, "Sea Water", "Toxic"))))))  
df$doNom <- as.factor(ifelse(df$DO < 4, "No fish can live",  
 ifelse(df$DO < 6.5, "Few fish can live",  
 ifelse(df$DO < 9.5, "Most fish can live", "All fish can live"))))  
  
df$num <- as.numeric(df$month)  
df$season <- as.factor(ifelse(df$num < 5, "Winter",  
 ifelse(df$num < 7, "Spring",  
 ifelse(df$num < 10, "Summer", "Winter"))))  
  
drop <- c("num")  
df = df[!(names(df) %in% drop)]  
  
#data engineering creating nominal  
write.csv(df, file = "machineLearningcompleteFinal.csv")  
  
  
df %>%  
 filter(DO < 12) %>%  
 filter(DO > 0) %>%  
 ggplot(aes(x = tempNom, fill = season, color = season)) +   
 geom\_density(alpha = 0.1)



I realize that many other values could be discretized, and this turned into a great wealth of material to be ran and analyzed by the models. The string data used in the classification of the ranges were taken from the USGS and other altorities. The date was again used to generate seasons. I chose the months that described them better, not the official beginning and end of season. Please refer to the code to define seasons in df$seasons. The above plot shows the density of the discrete temperature set as cold, hot, and mild by season. Please refer to the code for definition of discrete temperature.

#png("latLongTempDo.png", height=3000, width=3000)  
DOnoZeoes <- read.csv("dissolvedOxygenOnlyGoodValues.csv")  
mid = median(DOnoZeoes$Temperature\_C)  
  
sp <- ggplot(DOnoZeoes, aes(x = Longitude, y = Latitude, color = Temperature\_C)) +  
 geom\_point(aes(size = cut(DO, breaks = 2)))   
sp + scale\_color\_gradient2(midpoint = mid, low="blue", mid = "white", high="red") +   
 labs(color = "Temperature", size = "Disolved Oxygen")

## Warning: Using size for a discrete variable is not advised.



#dev.off()

This data has been corrected for extraneous values in PH, it preserves only values greater than 0 and less than or equal to 14; the temperature preserves only values greater than 0 or equal to or less than 40 degrees Celsius.

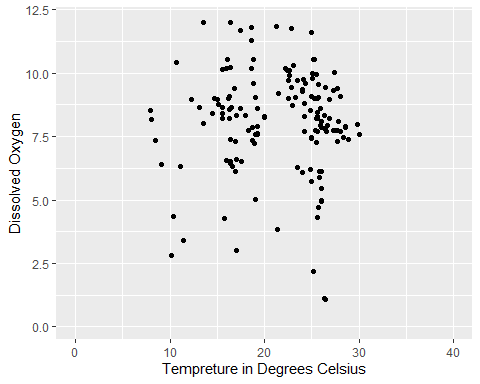
Map

Description automatically generated

The distribution shows a discrepancy that is explained by the image above. Some of the data was collected from Cook Pond. This data was also taken at a particular year which leads me to believe that because the water temperature in Cook Pond is colder it introduced bias to my model.

#png("TempDo.png", height=600, width=700)  
  
df %>%   
 filter(Temperature\_C > 0 | Temperature\_C < 40) %>%  
 ggplot(aes(x = Temperature\_C, y = DO)) +  
 geom\_point() +  
 xlim(c(0, 40)) +  
 ylim(c(0.1, 12)) +   
 ylab("Dissolved Oxygen") +  
 xlab("Tempreture in Degrees Celsius")

## Warning: Removed 428 rows containing missing values (geom\_point).

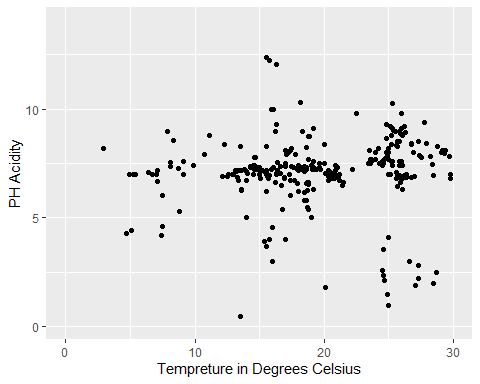


#dev.off()

Temperature as a function of dissolved oxygen. There is no clear relation besides the fact that they appear to loosely cluster at some point.

#png("TempPh.png", height=600, width=700)  
  
df %>%   
 filter(Temperature\_C > 0 | Temperature\_C < 40) %>%  
 ggplot(aes(x = Temperature\_C, y = pH)) +  
 geom\_point() +  
 xlim(c(0, 30)) +  
 ylim(c(0.1, 14)) +   
 ylab("PH Acidity") +  
 xlab("Tempreture in Degrees Celsius")

## Warning: Removed 284 rows containing missing values (geom\_point).

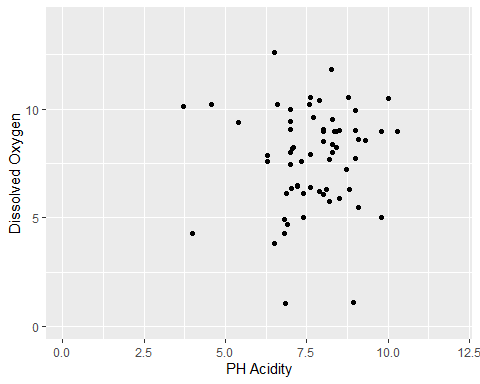


#dev.off()

Temperature as a function of PH. There seems to be a pattern that shows that PH doesn’t change much over the temperature span. This is good, PH should be regular, too much or too little is unhealthy. Please refer to the code on page 5 for details on healthy PH.

#png("phDo.png", height=600, width=700)  
  
df %>%   
 filter(DO > 0 | DO < 12) %>%  
 ggplot(aes(x = pH, y = DO)) +  
 geom\_point() +  
 xlim(c(0.1, 12)) +  
 ylim(c(0.1, 14)) +   
 ylab("Dissolved Oxygen") +  
 xlab("PH Acidity")

## Warning: Removed 515 rows containing missing values (geom\_point).



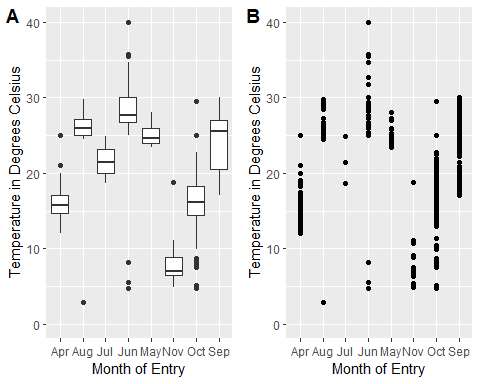
#dev.off()

PH as a function of dissolved Oxygen. They appear to have a centroid, which indicates regularity. It also shows that dissolved oxygen has a greater range relative to PH.

#png("monthTemp.png", height=600, width=700)  
  
plt1 <- df %>%   
 ggplot(aes(x = monthNom, y = Temperature\_C)) +  
 geom\_boxplot() +  
 ylim(c(0.1, 40)) +   
 ylab("Temperature in Degrees Celsius") +  
 xlab("Month of Entry")  
  
plt2 <- df %>%   
 ggplot(aes(x = monthNom, y = Temperature\_C)) +  
 geom\_point() +  
 ylim(c(0.1, 40)) +   
 ylab("Temperature in Degrees Celsius") +  
 xlab("Month of Entry")  
  
plot\_grid(plt1, plt2, labels = "AUTO")

## Warning: Removed 28 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 28 rows containing missing values (geom\_point).



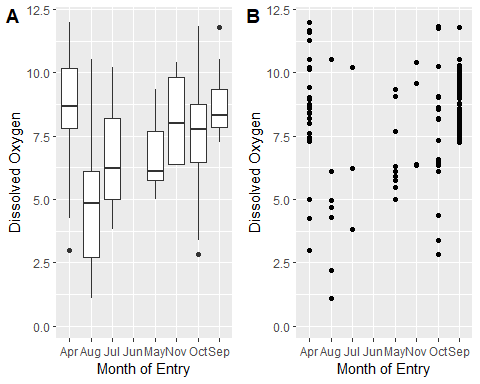
#dev.off()

Here are temperature boxplots and scatter plot for density. The data has been grouped by months, but the months contain every year. Not good for a timeline view.

#png("monthDO.png", height=600, width=700)  
plt1 <- df %>%   
 ggplot(aes(x = monthNom, y = DO)) +  
 geom\_boxplot() +  
 ylim(c(0.1, 12)) +   
 ylab("Dissolved Oxygen") +  
 xlab("Month of Entry")  
  
plt2 <- df %>%   
 ggplot(aes(x = monthNom, y = DO)) +  
 geom\_point() +  
 ylim(c(0.1, 12)) +   
 ylab("Dissolved Oxygen") +  
 xlab("Month of Entry")  
  
plot\_grid(plt1, plt2, labels = "AUTO")

## Warning: Removed 428 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 428 rows containing missing values (geom\_point).



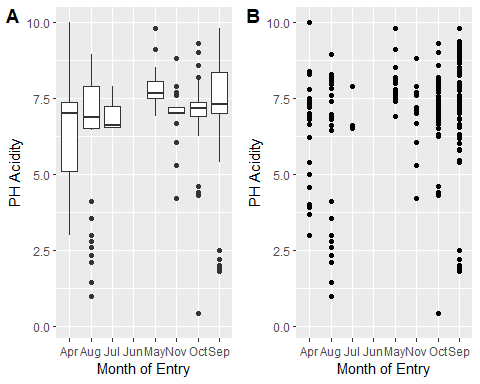
#dev.off()

We also have dissolved Oxygen boxplots and scatter plot for density. The data has been grouped by months, but the months contain every year. Not good for a timeline view.

#png("monthPh.png", height=600, width=700)  
plt1 <- df %>%   
 ggplot(aes(x = monthNom, y = pH)) +  
 geom\_boxplot() +  
 ylim(c(0.1, 10)) +   
 ylab("PH Acidity") +  
 xlab("Month of Entry")   
  
plt2 <- df %>%   
 ggplot(aes(x = monthNom, y = pH)) +  
 geom\_point() +  
 ylim(c(0.1, 10)) +   
 ylab("PH Acidity") +  
 xlab("Month of Entry")   
  
plot\_grid(plt1, plt2, labels = "AUTO")

## Warning: Removed 289 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 289 rows containing missing values (geom\_point).



#dev.off()

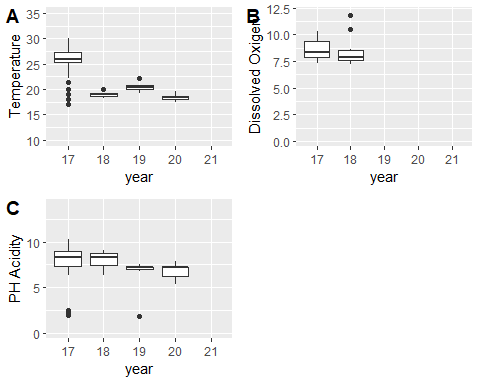
Next we have PH boxplots and scatter plot for density. The data has been grouped by months, but the months contain every year. Not good for a timeline view.

#png("allYear.png", height=600, width=700)  
  
plt1 <- df %>%  
 filter(monthNom == "Sep") %>%  
 ggplot(aes(x = year, y = Temperature\_C)) +  
 geom\_boxplot() +  
 ylim(c(10, 35)) +  
 ylab("Temperature")  
  
plt2 <- df %>%  
 filter(monthNom == "Sep") %>%  
 ggplot(aes(x = year, y = DO)) +  
 geom\_boxplot() +  
 ylim(c(0.1, 12)) +   
 ylab("Dissolved Oxigen")  
  
plt3 <- df %>%  
 filter(monthNom == "Sep") %>%  
 ggplot(aes(x = year, y = pH)) +  
 geom\_boxplot() +  
 ylim(c(0.1, 14)) +   
 ylab("PH Acidity")  
  
plot\_grid(plt1, plt2, plt3, labels = "AUTO")

## Warning: Removed 2 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 120 rows containing non-finite values (stat\_boxplot).

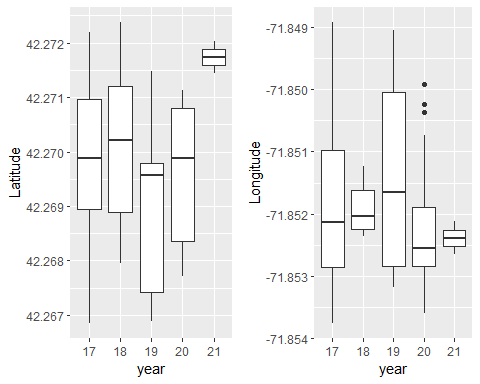
## Warning: Removed 107 rows containing non-finite values (stat\_boxplot).



#dev.off()

This is the target data boxplot over the years. This is appropriate as a timeline but is missing too much data. The temperature, dissolved oxygen and PH seemed to have a downward trend over the years in September but there is not enough data to tell that for sure.

#png("latLongYear.png", height=600, width=700)  
plt1 <- df %>%  
 filter(monthNom == "Sep") %>%  
 ggplot(aes(x = year, y = Latitude)) +  
 geom\_boxplot()   
  
plt2 <- df %>%  
 filter(monthNom == "Sep") %>%  
 ggplot(aes(x = year, y = Longitude)) +  
 geom\_boxplot()   
  
plot\_grid(plt1, plt2)

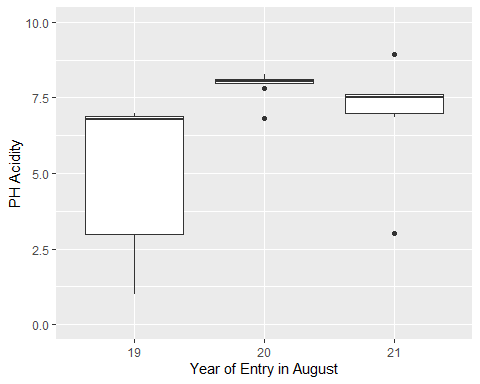


#dev.off()

For fun I just wanted to see what the spatial distribution of their sampling over the years was. It looks as if they moved northwards in 2021; and in 2020 they went west as indicated by the outliers. this could be a good tool to improve on their data collection system over the course of their research. this data may not be useful to us but may help generating more useful data as collection goes on. later this same data helped make sense of a trend in water temperature. By showing the exact year their collection changed.

#png("phAug.png", height=600, width=700)  
df %>%  
 filter(monthNom == "Aug") %>%  
 ggplot(aes(x = year, y = pH)) +  
 geom\_boxplot() +  
 ylim(c(0.01, 10))+  
 ylab("PH Acidity") +  
 xlab("Year of Entry in August")

## Warning: Removed 13 rows containing non-finite values (stat\_boxplot).



#dev.off

Looking back at all the years, august had an abnormal number of outliers for lower PH values. A closer look show that in 2019 the interquartile values were all below 7.5; the full interquartile range of 2020 was above both 2019 and 2021; 2021 showed a less diverging range. Nevertheless, when the zeroes are not filtered 2021 shows an alarming lower range but those values were most certainly null at zero.

## Timeline Description automatically generated with medium confidence

2019 shows PH at a dangerously low level but 2020 and 2021 had a more regular PH which indicates a positive change to the environment. It would be interesting to investigate further the reason why 2019 had such bad interquartile range even though the median is a good level.

## Data Mining

### Cluster

Using simple K-means with 66 % split yielded an interesting result, it appears to show that there is a difference over the years because they were clustered as such. This was a good result, but I decided to use mainly classifiers for models.

Table, calendar

Description automatically generated

### Classifier

Finding the best parameters for SMOreg

10-fold cross-validation and 66% split yield a similar error, so I decided to use the ignore class unknow instances with a greater value because I think it would give a cleaner result.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Using 10-cross fold SMOreg I generated the models for prediction of Temperature, PH, Dissolved Oxygen and Year.

These models show well a pattern when visualized, which we’ll see next.

Table

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedTable

Description automatically generated

A picture containing text, screenshot, indoor

Description automatically generated

The error in the model is high so any pattern perceived by their prediction needs to be taken into account separately and investigated using different tools to validate any inferences.

A picture containing text, screenshot, indoor, computer

Description automatically generated

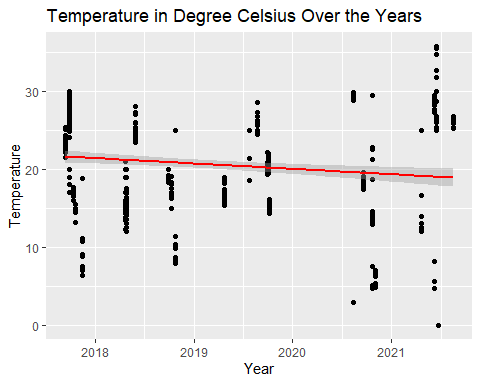
Graphical user interface, chart, scatter chart

Description automatically generated

What stood out in SMOreg was the year predicted by temperature which clearly shows that the water temperature is lowering over the years. Because the error was so high I used an R regression in order to validate this assumption.

#png("tempoveryears.png", height=600, width=700)  
graph <- df %>%   
 filter(Temperature\_C < 40) %>%  
 filter(Temperature\_C > 0) %>%  
 ggplot(aes(x = date, y = Temperature\_C)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", col = "red") +  
 xlab("Year") +   
 ylab("Temperature") +  
 ggtitle("Temperature in Degree Celsius Over the Years")  
   
  
graph

## `geom\_smooth()` using formula 'y ~ x'



This plot generated in R validates the Weka models it shows a downward trend in water temperature. But we must remember the introduction of data from Cook Pond and its difference in water temperature in 2021.

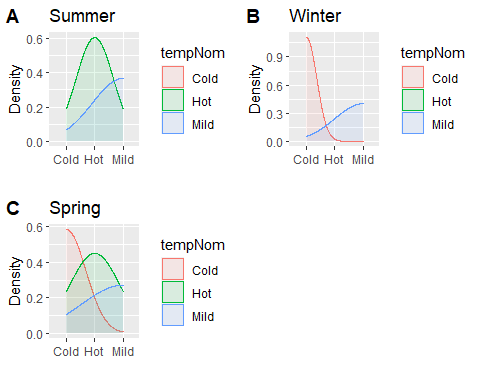
#dev.off  
#png("densitytempoverseasons.png", height=600, width=700)  
plt1 <- df %>%   
 filter(Temperature\_C < 40) %>%  
 filter(Temperature\_C > 0) %>%  
 filter(season == "Summer") %>%  
 ggplot(aes(x = tempNom, fill = tempNom, color = tempNom)) +  
 geom\_density(alpha = 0.1) +  
 ylab("Density") +  
 xlab("") +  
 ggtitle("Summer")  
  
  
plt2 <- df %>%   
 filter(Temperature\_C < 40) %>%  
 filter(Temperature\_C > 0) %>%  
 filter(season == "Winter") %>%  
 ggplot(aes(x = tempNom, fill = tempNom, color = tempNom)) +  
 geom\_density(alpha = 0.1) +  
 ylab("Density") +  
 xlab("") +  
 ggtitle("Winter")  
  
plt3 <- df %>%   
 filter(Temperature\_C < 40) %>%  
 filter(Temperature\_C > 0) %>%  
 filter(season == "Spring") %>%  
 ggplot(aes(x = tempNom, fill = tempNom, color = tempNom)) +  
 geom\_density(alpha = 0.1) +  
 ylab("Density") +  
 xlab("") +  
 ggtitle("Spring")  
  
plot\_grid(plt1, plt2, plt3, labels = "AUTO")

## Warning: Groups with fewer than two data points have been dropped.

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning  
## -Inf

## Warning: Groups with fewer than two data points have been dropped.

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning  
## -Inf



#dev.off  
#graph + stat\_regline\_equation(label.x = 2000, label.y = 40)

The above graph shows discrete water temperature over the seasons. This distribution makes logical sense but holds no important value.

### Rules

Using part for nominal dissolved oxygen I found that using cross validation 20 gave me the best correctly classified instances 73.8% and lowest root relative square error 78% which is quite high and unreliable.Table

Description automatically generated with medium confidence

But the detail class accuracy showed that the class “No fish can live” had 84% accuracy and the confusion matrix confirmed that indeed many values in that class were classified correctly. These are some of the rules it generated for this class

–-PART decision list

------------------

monthYear = Jun-21: No fish can live (43.0)

phNom = healthy water AND

monthYear = Oct-19: No fish can live (22.0/1.0)

phNom = healthy water AND

monthYear = Sep-20: No fish can live (21.0)

phNom = Toxic: No fish can live (64.0/15.0)

phNom = healthy water AND

monthYear = May-18: No fish can live (19.0)

phNom = healthy water AND

monthYear = Oct-20: No fish can live (15.0)

monthYear = Aug-19: No fish can live (27.0/4.0)

monthYear = Oct-19: All fish can live (17.0/5.0)

monthYear = Apr-21 AND

tempNom = Cold: No fish can live (17.0/2.0)

phNom = healthy water AND

monthYear = Sep-19: No fish can live (14.0/1.0)

phNom = Really bad, the fish will die AND

monthYear = Sep-17 AND

tempNom = Hot: All fish can live (66.0/30.0)

monthYear = Aug-20: No fish can live (13.0)

phNom = healthy water AND

monthYear = Oct-17: No fish can live (11.0)

phNom = Really bad, the fish will die AND

monthYear = Sep-17: Most fish can live (22.0/10.0)

monthYear = Sep-17: No fish can live (26.0/12.0)

monthYear = Nov-17: No fish can live (12.0/4.0)

monthYear = Apr-19 AND

phNom = Really bad, the fish will die: All fish can live (12.0/1.0)

monthYear = Aug-21: All fish can live (12.0/5.0)

monthYear = Apr-21: All fish can live (10.0/3.0)

monthYear = Oct-17 AND

phNom = Really bad, the fish will die: All fish can live (9.0/3.0)

monthYear = Apr-18 AND

phNom = Really bad, the fish will die: Most fish can live (20.0/9.0)

monthYear = Oct-18 AND

phNom = healthy water: No fish can live (22.0/8.0)

monthYear = Apr-19: No fish can live (10.0/4.0)

monthYear = Oct-20: All fish can live (9.0/1.0)

monthYear = Nov-20: No fish can live (9.0)

monthYear = Apr-18 AND

phNom = healthy water AND

tempNom = Cold: No fish can live (8.0/2.0)

monthYear = Sep-18: Most fish can live (8.0/2.0)

monthYear = Apr-18 AND

phNom = healthy water: All fish can live (5.0/2.0)

monthYear = Oct-18: Most fish can live (9.0/5.0)

monthNom = Sep: No fish can live (8.0)

monthYear = May-18 AND

phNom = Sea Water: Few fish can live (6.0/2.0)

monthYear = Oct-17: No fish can live (4.0)

monthNom = Jul: No fish can live (5.0/2.0)

: Most fish can live (7.0/2.0)

Number of Rules : 34

Many of these rules make no sense, but some are worth entertaining. Nevertheless, the dataset information does not yet yield good rules, or trees.

### Association

Timeline

Description automatically generated with medium confidenceFor association I used Apriori, in order to use this tool, I had to reduce my dataset to nominals only. The attributes I used to generate the rules were monthNom, monthYear, tempNom, PHNom, doNom and season. Here the scales taken from USGS and other authorities was very helpful in classifying intervals for PH and dissolved Oxygen which are extremely important variables in this data and it represents the health of a water system as well as its biodiversity. I decided not to use all the scale denominations for both DO and PH but kept the most important so the models wouldn’t be overwhelmed by a large number of choices.

I learned a valuable lesson while preparing the data for association. I know that the closed world assumption is one of the most important academic tools to introduced complicated things and make them understood. It appears to me that the models that deal with generating rules, trees and association benefit a great deal from this view. The more complexity we add the least useful information we get from it.

A picture containing map

Description automatically generated

Graphical user interface, application

Description automatically generatedIn the first run I used the default settings, and this generated rules with good confidence but ultimately uninteresting results. The rules generated predominantly made association between season and temperature.

The high minimum confidence and small number of generated rules ensured only obvious rules were generated.

The rules generated like --if the moth and year is Sep 2017 and the season is Summer than the month nominal is September--, shows that my attempt to introduce redundancy keeping the same data in different formats ultimately generate duplicates that did not help in generating meaningful associations.

Regardless, as redundant and obvious these observations may be, they do ultimately show that our associations don’t misbehave showing wild unrealistic rules.

Graphical user interface, text, application

Description automatically generated

Here is the result of the rules created using the default settings for Apriori that we just discussed. It certainly shows that 10 rules in this case are not enough. Another solution would be to simplify the data to desensitize the model to trivial results.

Graphical user interface, application

Description automatically generatedI decided that it would be best to modify the settings of the algorithm to generate more interesting rules by setting the confidence to 0.8 and increasing the number of rules generated to 20, allowing some noise to extract more interesting insights instead of just the obvious.

This generated rules that at a glance were more intelligent in the sense that it deviated from the intuitive obvious.

Despite the wealth of new information, the model still maintains a certain regularity, perhaps showing how sensitive is the environment surrounding the data.

Another important lesson that I learned was that sometimes we want to see something interesting when in reality is best the data behaves in the most boring ways as possible. This is true for this dataset and is reassuring that very few irregularities were found.

Nevertheless, the rules generated by these settings generated information worth future investigation.

Text

Description automatically generated

Some of the interesting rules generated by the model were rules 17, 18, 19, 20 which indicates a bad PH quality in September more specifically in the summer of 2017. This result is worth noting.

## Conclusion

The temperature of the water has dropped over the years, but this can be due to bias during data collection. Regardless, the data spans over 4 years and the more we collect the surer we will be of our insight.

In the future I believe the Patch Research would benefit from partnering with the CS Department and Data Mining classes. A discussion on which and how the datum can be collected can improve computer modeling and interpretation to ensure that the future insights are as useful and interesting as possible. Good questions to ask would be if there is a trend in PH and dissolved oxygen as suggested by the data. Right now, the answer remains inconclusive, but future studies could either confirm or refute the argument.

Personally, I found that collecting discrete nominal data is extremely important for running some of the models and that part of the success in maintaining projects like this is time and experience. This opened my eyes to the fact that nothing can substitute a good hands-on project and those results can always get better, so it is imperative to know when to stop or at least when to take a brake to show the findings because the ideal is that such endeavors remain continuous.

# References

ArcGis Pro, n.d. s.l.: s.n.

Coutinho, J. B., n.d. *gitlab.com/jeffersonbourguignon/data-mining-project-patch-reservoir.git.* [Online]   
Available at: https://gitlab.com/jeffersonbourguignon/data-mining-project-patch-reservoir.git

GitLab, n.d. s.l.: s.n.

*Patch Reservoir Research* (2017-2021) Worcester State University GIS Department.

Python, n.d. s.l.: s.n.

R Studio, n.d. s.l.: s.n.

USGS, n.d. *usgs.gov/media/images/ph-scale.* [Online]   
Available at: https://www.usgs.gov/media/images/ph-scale

Visual Studio Code, n.d. s.l.: s.n.

Weka, n.d. s.l.: s.n.