Thesis

Nick Zinck

March 2018

Contents

1 Introduction

1.1 Problem Statement

A Reservoirs water quality effects the type and extent of neccesary treatment process of the water supply prior to distribution. Generally, less treatment is required for a water utility who's source is a remote, healthy reservoir than a reservoir which has been degraded by anthroppologic or other means. It is often infeasible to obtain water from completely pristine areas, especially for water utilities who supply water to urban areas. Watershed management can help ensure the water quality of a reservoir and as well as predict, lessen, or prevent reservoir water quality degredation. from occuring. (Quantity as well.)

Water Quality data collection and analysis can assist on decision making for utilities. It is not feasible to know every, yet an adequate sampling plan can bring insight about the condition of the reservoir as well as the tributaries entering the reservoir. It is not just enough to collect this data, the task fo water quality monitoring does not stop after the field or after the lab. Water Qulaity data analysi scan help answer questions and suggest solutions once problems arise. Even in the case of non problemtic situations water qualit data can help one understand the unique processes of the reservoir in question, calibrate water quality models, and suggest the updated sampling plans. Questions must be asked like is the current sampling plan adequate to get as good of an undestanding of the water quality processes that are happeing in the reservoir and watershed, within reasonable time and cost constraints.

1.2 Objective

A comprehensive watershed monitoring program includes collecting much water quality, meteorological, and hydrological data. Although data collection is always important, much of this data is underutilized due to a timely processes of searching, displaying, and analyzing data. Spreadsheet applications are likely not an effective way to store large datasets and visualization and analysis tools are limited. Relational database applications serve as a better home for these large datasets, yet visualization and analysis tools are commonly even more limited. A logical solution is to store large

datasets within a relational database and pair this database with an outside application specialized for data visualization, analysis, and automation.

This project specifically is working to facilitate the DCR's (Department of Conservation and Recreation's) data entry, searching, visualization, and analysis process through an R-based application creation tool called Shiny. R is an open source programming language used largely for data statistics and visualization. Shiny is an open source application that allows one to create applications which have a friendly user interface component as well as a server component which uses R to do all the work. A Shiny application can be fully customized by the designer for unique tasks. Features of the DCR Shiny application include easy data searching, water quality time series plotting and analysis, water quality regression analysis, geospatial data visualization and analysis, and more. Through the insight that this application brings, future data collection needs can be better assessed which will direct changes to the current watershed monitoring program.

Automation is a useful, yet underutilized tool that can save an organization much time day to day on decreasing the number of repeated tasks that come along with searching, displaying, and analyzing water quality data. A custom application with dashboard toallow a user to easily perform data science with the power of R, with little knowlege of R programming language is highly desired. This application should include but not limited to facilitated data entry, querying, visualization, and analysis of water quality data and other related watershed data. Save Tiem an dMoney , inrease quality control

1.3 Scope of Work (Application Purpose)

The scope of this project is to create a Watershed Data Management System for the Quabbin and Wachusett Watersheds. This project worked to develope a custom dashboard-like application software called Watershed data Analysis and Visualization Environment (WAVE). As its name indicates, this application allows for data querying, visualization, and data analysis. The application opens in a web browser and allows the user to select desired locations, parameters, dates, as well as more advanced filters and the corresponding queried dataset will be presented in a table and the user will be able to output this data. This data query as well as all components of the application are seperated by water quality type (e.g. Tributary, Reservoir bacteria, Reservoir Profile

data) and by watershed (e.g. Quabbin and Wachusett). Customboility and variablity between the watersheds is desired and the application design reflects this.

Visualization dashboards were created to view data in a speciofic customized fashion based on a user selected inputs. Statitical Analysis of parameters is useful to understand watershed and reservoir water quality in regards to temporal and spatial trends of water quality parameters and also correlations between water qualitiy parameters.

Temporal Analysis and spatial analysis (do more (some) reading)

Raw Data from various sources can be imported by a common importer tool that transforms raw data into the desired database format, provide quality control checks, and add the data to the right location in the database.

The Application is shared openly through Github and can be ran locally on any computer with two lines of code. Hosting the application online at a URL is also a possibility.

2 Background

2.1 Watershed Overview

2.1.1 Quabbin and Wachusett Reservoir

The metropolitan area of Boston, Massachusetts, recieves its drinking water from the Massachusetts Water Resource Association (MWRA) water supply system. The water supply consists the Quabbin Reservoir and the Wachusett Reservoir which is managed in partnership with the Department of Conservation and Recreation (DCR). Both the reservoirs combined supply about 200 million gallons per day to consumers (MWRA website). The Quabbin and Wachusett Reservoirs are protected and over 85% of the watershed lands that surround the reservoirs are covered in forest and wetlands. Because they are well-protected, the water in the Quabbin and Wachusett Reservoirs is considered to be of very high quality (MWRA website). The water supply system is rather unique in that the Quabbin Aqueduct, a 24.6 mile long tunnel, connects the Quabbin Reservoir to the Wachusett Reservoir. The MWRA transfes water (i.e. Quabbin Transfer) from the Quabbin reservoir intermittingly to the Wachusett Reservoir to maintain the water level and water quality of the Wachusett Reservoir [DCR, 2007]. In a given year, the Quabbin Transfer most always makes up the majority of the total inflow to the Wachusett River. The MWRA can also divert water from the Ware River watershed, located between the two reservoir watersheds, to either the Quabbin Reservoir (preferable) or the Wachusett Reservoir through the Quabbin Aqueduct, but these diversions are rare. The Ware River usually has poorer water quality which makes an input location with a long hydraulic retention time desirable, thus the Ware River is most always diverted to a locaiton in the Quabbin Reservoir that is hydraulicly far from the Quabbin Aquaduct. Transfers generally occur from June through November and can last for weeks at a time to meet higher water demands, maintain the water level, and mitigate water quality concerns in the Wachusett Reservoir (DCR, 2007).

The complex nature of the system allows for decisions to be made that can alter the water quality that ends up in the Boston Supply System. Although all water enters from one locaiton in the Wachusett Reservoir, it is essential to monitor the water quality in the whole watershed to best understand why the processes that might create a water quality issue and to address the source or cuase of the problem. The streams and

the reservoirs water quality are frequently tested by the Massachusetts Department of Conservation and Recreation. Stream guages are insstalled to estimate the flow of major tributaries to hte reservoirs, both by the United States Geological Survey (USGS) and by the DCR. The Quinapoxet and the Stillwater River are gauged by the USGS which were installed in the 1990's. Some meteroligcal data is also collected at various locations in the watersheds and additional data can be attained from the NOAA / NCDC.

2.1.2 Water Quality Sampling Plan

The water quality sampling plan consists of routine sampling at various sites. The task of water quality sampling plan of the watersheds in the water supply system are divided between the Quabbin and Wachusett office of the DCR Water Supply Protection Division. Each office creates their own sampling plan for the reservoirs and tributaries in their watersheds each are responisble for. The Quabbin Office is responsible for the sampling of the tributaries in the Quabbin and Ware River watersheds and sampling in the Quabbin Reservoir itself. The Wachuchesett Office is responsible for the sampling of the tributaries in the Wachusett watershed and the Wachusett Reservoir itself. Each Office has sites that they sample consistantly every year which are considered core sites. Other Site locations are temporaraily selected for various reasons including to target areas of concerns and to target areas that are usually underrepresented in the sampling plan.

The tributary Water quality parameters sampled in both offices are Turbidity, Water Temperature, E. coli, Mean UV254, Ammonia, Nitrate, Nitrite, Specific Conductance, Total Kjeldahl Nitrogen, Total Phosphorus. Most of these parameters are sampled on a biweekly basis, although ( ) are sampled ( ). The sampling frequencys of water quality parameters in each watershed is displayed in image 1. The Sampling frequency is for the Core Sites at the sampling frequency that had the longest duration between the five years of 2013 to 2017. In most cases the frequencies of the core samples are the same as the frequencies of the other non-core sites? Water quality data related to forestry practices is also collected to compare water quality parameters between managed and unmanaged forests.

Do not need this text: replace with graphic: The QUabbin Office additionally samples

Dissolved Oxygen, Dissolved Oxygen Saturation, pH, Alkalinity, Calcium(II), Fecal Coliform, Total Coliform (Colilert). The Wachusett Reservoir additionally samples phosphate-phosphorous, total organic carbon, and total suspended solids as well as operates.

Reservoir water quality samples are conducted in by various means depending on the parameter. All reservoir samples are taken by boat during the time of the year with no ice freeze. ( ) are collected at reservoir locations, at three depths: surface, mid, and deep. The parameters cllected at both reservoirs include . Profile Data is collected in both reservoirs at about a meter depth. Bacteria Samples are collected in the Wachusett at the surface only which include (). Phytoplankton data is also collected at both Reservoirs. Do I include Bacteria? brief? The sampling frequencys of water quality parameters sampled in within each reservoir is displayed in image 2 for five consecutive years, 2013 to 2017. If the sampling frequency parameter happened to change during this time period, the sampling frequency that had the longest duration between the five years was chosen. In most cases the frequencies of the core samples are the same as the frequencies of the other non-core sites?

Trib Plot geom_tile Parameters on y. Location on X. Quabbin, Ware, Wachusett. Color Grid - COlor discrete by frequency, make kind of scale-ey.

Res Plot Geom_tile Parameters on y. Locaiton on X. Quabbin, Wach. Color by frequency. Facet a geom_tile for Type: Profile, a

Checks marks papers for other graphic ideas.

Turbidity, Water Temperature, E. coli, Mean UV254, Ammonia, Nitrate, Nitrite, Specific Conductance, Total Kjeldahl Nitrogen, Total Phosphorus,

Quabbin Dissolved Oxygen, Dissolved Oxygen Saturation, pH, Alkalinity, Calcium(II), Fecal Coliform, Total Coliform (Colilert),

Wach Discharge,

Core Sites and EQA Sites

Questions should be asked the sampling plato make informed decisions to update the sampling plan on a continuous basis. The reservoir and watershed is always changing, as well should the strategies to maintain adequate water quality. It is likely that as more information is disscovered about the reservoir, more quesitons can be asked. Are

there locations that we should be samplin gmore? Is the sampling frequent enough to see trends? Is the sampling more frequent than neccesary to see trends? Are there parameters that should be added to the sampling plan? Are there any parameters that should be removed from the sampling plan? This Application will give insight to data collection needs

2.1.3 Previous Watershed Studies

Lily and marks paper. Look at Lit review. Many studies have been conducted under the partnership of University of Massachusetts and the DCR. Many of these studies have benefitted from the vast amount of water quality and hydroogical data that is collected in the Quabbin, Ware River, and Wachusett Watersheds. These studies have included many hydraulic and water quality models and analysis on the Wachusett Reservoir. The water quality impacts from extreme precipitation events have been studied through statistical forcasting? of potential loads coupled with a reservoir hydraluic and water quality model (Jeznach HageMan). The fate of a contaminant spill in the Wachusett Reservoirs have also been modeled (Lily 2013, 2011 Devonis) (Sojkowski) (more). Other water quality modeling topics have included the effects of climate change on the Wachusett reservoir (Loly) and modeling the fate of Natural Organic Matter and fecal pollution ( ). Having a propper sampling plan can allow for more informed research studies. Water Quality (profile) data is essential for the calibration and validation of hydraluic and water quality models of a Reservoir. Well maintained data further facilitates this process.

2.2 Watershed Data Managament System.

For the respect of simplicity, a watershed data management system, or really any data management system, can be broken down into three aspects: 1. data importation, 2. data storage, and 3. data utilization. Data importation includes the act of getting data measurements from the field or a lab into a database. This data can span from manually read data to data that is transmitted automatically SCADA. Data storage is the location in which the data lives and is ever changing as more data is collected. Data usage is includes the querying, visualization, analysis, or any other process from which conclusions can be drawn from the data to benefit decision making.

Proprietary software exists that accomplish all of three of these features. Aquarius, created by Aquatic Informatics, is a leading software service for water supply management. Aquarius Time-Series software is used for rating curve creation. The USGS uses Aquarius Time-Series for their rating curve generation. Aquarius-Samples is used also used specifically storage of a variety of field and laboratory data. An internal watershed management system is also a possibility to meet. An internal watershed management system could be developed using multiple components, (software, databases, programming languages) to create watershed management system.

Watershed data collection varies within each watershed. Proprietary software attempt to supply versatile a product that can be widely used on any watershed. This can be a very difficult task due to the discrepancies in water quality data collected as in the types of parameters, the instruments used. The analysis and visualization for more complex analysis can also differ. These software programs seem to be a good solution to data management alone, but lack on data visualization. Also, it is useful to have data in house if analysis wants to be done on it. Data in some of these systems can be stored on a cloud and harder to access than if they are stored internally. Aquarius costs money, which can be an economic , although the benefit of a data management system can save money due to time saved and a quality control.

With the development of coding languages, it is getting easier to build a custom dashboard or application with a application development framework. Creating an application, rather than using a proprietary software allows for more flexibility. It is time consuming creating an application. the pay off of customizability and . Also, a proprietary software like Aquatic Informatics software will be responsible for bug fixes and adding new features. A custom App requires the water management facility to add more features as well as keep in touch with updating and changing software libraries that could, but shouldn't, effect the application's performance (If this happens the application can always fall back on a previous version of a library).

2.2.1 Data Storage

There are various types of data collected in the watershed. Due to the vast amount of data from various sources the raw data can be in very different formats. Storing data in a central database in similar format is crucial to set up a platform for efficient data

analysis. The application provides a connection to this data in the database with an easy-to-use user interface for a human user to interact with the data.

Relational databases are a common way to store data, I think that data

many databases have the ability to be a back-end server. This means that the database has ability for queries. Most of these databases use a form of SQL language , although there are non-SQL databases like MongoDB. Microsoft Access is an example of a SQL database and one can write. A front end is even possible with visual basic language for a sense of a user interface. Visual Basic is slow and is not well developed in comparison to other more.

There are many open source and free sources of

Aquarius stores data in their own database in the cloud. It is unknown to the author of this paper what type of database that Aquarius uses, although it is likely a relational database.

2.2.2 Data Importation and Data Utilization

There are many different effective ways to go about data importation but importing data by hand is no longer one of them. Most always data that one collects is not in the same format as how the data is store. Data transformations must be made including but not limited to column name changes, adding or removing columns, data type, and spreading or gathering columns. Spreadsheets also can be used to import data, though manually typing data and making format changes can be timely and prone to errors. Programming languages can be used as a tool to transform data to a particular format. R and Python are just two of the many programming languages that can be used for this process. A code script written with a programming languages can additional perform set quality control measures including checking for duplicate data or alerting a user when their is an usual data value. Even further, a user interface can be created for a user to be able to run this code without being familiar with the programming language or integrated development environment (IDE) is note required. A common user interface can be built to input data of all types.

2.2.3 Application Developement Frameworks

Applications can be created to access the data from the database and allow for more customizability and data use than what is offered in most database software. The same programming languages used to transform and import the data can likely be used for the the more popular programming languages to facilitate query, export, visualization, and analysis of the data. Spreadsheets can also be used for data analysis and visualization, though this is slower. An application with a friendly user interface can be built from writing code in one of the programming languages to allow for user to more easily create visualizations, perform data analysis. This user does not need to have any program experience.

Selection of a proper application development framework can greatly decrease the amount of knowledge one must have to create an application. Most application development frameworks can be considered either a front-end framework or a back-end framework which communicate through a common API, usually JSON, which is considered the universal Binary. The frontend framework is responsible for the construction and layout of the user interface, which is essentially what the user sees and interacts with. The backend is responsible for actual computational operations of the application and accepts information from the front end when the user requests it. The seperation of front end and backend can allow for full costumization and allow for front end and back end specialists to cleanly on each part seperately. In the case of relatively simple data science applications, this paper explores application developement frameworks that function as both a front end and back end framework (or "full stack" framework). This approach is deemed appropriate becuase only a mimimilistic user interface is needed and most of the focus will be on the data science potential of the backend framework which is likely restricted by the programming language that the framework uses.

The application that this paper is focused on is Shiny, a package in R. Data Science Application lobraries in Python include Bokeh, Spyre, Dash, Pyxley, IPython notebook, Bowtie. D3 for javascript is also. All of these libraries are free and open source. These all differ in certain ways but all are capable of producting web applications focused on data science. many of these application frameworks leverage javascript to render the user interface inputs and outputs in a way that little knowlege of Javascript is neccesary.

Python is a much more widely used, more developed language than R. Python is used for many purposes outside of data science. Python has a total of____ packages compared to . Pandas package is debatedly the leading data scienceframework within Python. Pandas Package in python is meant for data science. It is similiar o R DPLYR. same users?

Shiny is web application framework for R data projects. A user can create an application in the form of a website, an html document, or a dashboard. Shiny offers hosting services which cost some money in which a user can easily launch an application without the knowlege or hastle of hosting their own application as well as recieve customer support. Shiny is designed for people with people who have experience with R but do not have nay application developement experience. No web development skills are required. Shiny is well ducumented including many tutorial documents and videos as well as extra webinars and Github and Stackoverflow help.

Bokeh is a web application framework for Python Projects and is similiar to Shiny. Bokeh seems to offer more in depth interactive ability of plots and other graphics, yet Shiny does offer basic interactive graphic features as well. "Bokeh is a Python interactive visualization library for large datasets that natively uses the latest web technologies. Its goal is to provide elegant, concise construction of novel graphics in the style of Protovis/D3, while delivering high-performance interactivity over large data to thin clients." - Bokeh. A benefit of Bokeh is that the visualizations can be connected to almost any web tool, widget, or framework, outside of Bokeh itself (Bokeh). Python frameworks like Django, Pyramid, or Flask have greater customobility than Shiny and can be used for things outside of data science.

Spyre is another web application framework for providing a simple user interface for Python data projects. Dash created by Plotly is another alternative to build dashboards using Python which utilizes plotly.js, a leading web chart library, without the use of Javascript. Pyxley python package makes it easier to deploy Flask-powered dashboards using a collection of common JavaScript charting libraries. UI components are powered by PyxleyJS. Bowtie is also an interactive dashboard toolkit in python which can be used to create web applications for data science. D3 is a javascript library which combines powerful visualization and interaction techniques.

For the reson of simplicity, documentation, and previous familiarity in R, decided that Shiny is the best option. This decision should be made on a case to case basis depending

on desired use and previous familiarity with a language. Resources outside of this paper should be used to determine which is the best fit for one's data needs. Little porgramming knowlege

Aquarius Time-Series and Aquarius Sample have a data import feature. Aquarius has the ablility to import data from common more main stream instruments which basically has an internally built code to correctly transform this data into the neccesary format for the database. A user can also import their create custom transformation " " to transform a particular dataset. Auquarius has the ability to perform queries and see basic plots of data including scatter plots and box plots.

R was developed for statistics and specializes in visualization.

3 Application Developement process

3.1 Database Development

Access database.

4 databases

Spreadsheets to databases. Now i can have opinions slightly.

A tidy data format is not only better for data science, it is also essential for proper database storage. This is the best way to link diferent data tables and relate data tables together via keys to allow for joins to be possible. Storing data in multiple data tables of related information is much more efficient than. Metadata is considered. An example of tidy data is a “Parameter”" and “Result” column rather than a column for each parameter. This also allows for a single “Units” column which would not have been possible with the wide format.

In the future it may makes sense to move towards a different database software due to the the outdating of Microsoft Access or whether be the DCR terminating the Microsoft Access license. If so this will likley to a relativiely smooth process in converting to anohter relational database since the all have a common API, SQL.

3.2 Code Development

R is the primary language used to write the code, although there are also direct uses of CSS and one or two instances where direct javascript is usesd. Other languages are used indirectly if a function in R is written in another language.

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, .) and graphical techniques, and is highly extensible. One of R’s strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. R is available as Free Software under the terms of the Free

Software Foundation's GNU General Public License in source code form. It compiles and runs on UNIX, Windows and MacOS. (**???**)

Many users think of R as a statistics system. We prefer to think of it of an environment within which statistical techniques are implemented. R can be extended (easily) via packages. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics. R has its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both on-line in a number of formats and in hardcopy.

RStudio is the leading (IDE) for the R language. RStudio creates a number of packages that have driven R to be highly used. Among some of the more highly used packages in R are ggplot, dplyr, tidyr, RMArkdown, and Shiny. Many of RStudio's packages are contained under the " " called tidyverse which can savetime and be loaded all together.

Shiny is

Github

3.2.1 WAVE

Modules and Functions

Naming Conventions

Developer Manual

3.2.2 WIT

3.3 Application Deployment

There were many possibilities of ways to launch hte application. A large driver of this decision is based on the answer to the question: who do we want to be able to have access to the app. Becuase this application is to be used primarily by the DCR internally, it was not neccesary to host the application on the worled wide web. Since there was a discrete small set of computers that need to have access to the application, it is feasible to install this application on each of these computers to run locally. This

would not be possible if it was desired for an application with access from users on any computer connected to teh internet. If it was neccesary to do so we would have had to set uop our own server, pay for the hosting service taht Shiny drectly offers, or pay for another outside cloud hosting sevice. This also felt like the safer bet for any data security purposes since not hosting the data on the web allows for the data to stay on the DCR's internal network.

There are two primary launching options to launch a shiny App locally. The first is by executing the code directly, or using RSTudio Runapp button, which requires the application script to be stored locally. The second launching option is made possible by a built in function in Shiny called RunGitHub which will fetch the most updated code stored in a Github Repository. Github also happens to be, not coincidentaly, the leading code sharing, storage . . . ..

A desktop shortcut can be created to allow the user. figure _ shows the picture of the desktop icon used for WAVE which is a modified version of Department of Conservation logo. The desktop shortcut when clicked executes batch file to run the R script to intall/load packages and the Rungithub function. A configuration file was created to allow for customized computer settings including personalized user settings. The configuration file includes information

Although a user does not need a copy of the code on the user's computer, using RunGitHub command in Shiny still requires R to be installed on a computer. To avoid this requirement of downloading R, Shiny can be packaged with a porta. Also, the computer will need a web browser installed. This is most definitely already available on a working computer, yet there can be descrepencies between how various web browser's interpret HTML and Javascript, so it is safer to use have a consistant browser. (Reference for packaged) .

The Application can also be packaged. Include link to this details. A portable Chrome and portable R

4 Application Features and Findings (Results)

4.1 Data Query and Export

It is advantageous, if not essential, to have fast access to water quality and other watershed data. It is beneficial for any scientist or engineer to be able to acess this data with ease. In practice, this is often not the case due to all data not being stored in a location known by all scientists and engineers. Lack of experience with certain technologies can also inhibit a person from being able to access timely data. Although spreadsheets are commonplace among most workplace settings, being able to query this data for the exact data that a perrson is looking for can be timely and troublesome. Queries in a relational database is a better approach to this common task.

The application queries the data based on the user selected input. The user is prompted to first select one or more Site Locations with a map that indicates which sites are selected. The user is then pompted to select the parameters and date range in which data is avaiable for the selected sites. Additional filters can also be applied to the data which include selection of seasons, months, years, flags, storm event, and other eventually meteorlogical and hydrological conditions. Queried data can be exported to a csv file with the click of a button.

4.2 Data Visualization

This appliaction allows for water quality visualization of temporal trends and temporal statistical analysis.

4.2.1 Time Series Scatter Plot

A scatter plot is avaiable to see a water quality parameters trends over a specified duration of time. The data is queried in a similiar manner as discussed previously and a user is able to plot data from multiple Locations and one or two parameters. The user can chooose to group or facet by Location or Flags. A Facet creates multiple plots, all with similiar mapping techniques, to allow for easy comparison across plots.

The user is given many options to customize how the plot displays the data. the user can choose a log-scale for the Y-axis as well as to start at zero. These options are not applicable for the x-axis becuase the X-axis in date. The user can adjust point size and point color (if the user has not already specified that the group by color). The user can choose form many display themes that are offered in ggplot2. The user can add a horizontal line, vertical line, or floating text anywhere on the plot. The plot automatically creates logical axis labels and a descriptive plot title, which the user can override with custom text or choose to have no labels or title. The user can save a plot with in multiple formats to a specified plot width and plot height. Figures " " are examples of these saved plots. The user can choose to turn on the interactive plot features that are allowed by Plotly which allows the user to hover over a data point for info an also toggle the plot in various ways like zooming in and out.

Temporal Trend lines can be added to the plots to help visualize if there are any temporal trends, if it is not clear when just looking at data points. The trendlines will automatically group in a fashion identical to the points. If the data points on the scatter plots are grouped by Locations as idicated by colors, than the trendlines will also appear grouped by these same colors. Grouping by shape and faceting works similiarly.

Three methods for trendlines are available in this application. The first method is a linear trendline which is the linear line that minimizes the residuals of the fit. The second method is the Loess method which is a . The third method is a Generalized Additive method which . The user can choose to show a confidence ribbon with choices of confidence intervals of 0.90, 0.95, and 0.99. If the user selects a confidence ribbon with 90% confidence, a shaded region will appear on the plot where the data is 95% likely to . This should be used with caution becuase the linear confidence interval does not take into account the seasonal variation, and assumes that all variation is " “.

The user can choose to add a secondary parameter to the plot by introducting a secondary y-axis. This feature allows the user to compare temporal trends of two water quality parameters. A plot comparig of two parameters, on the x-axis and y-axis can usually display a clearer picture of the relationship between two parameters which will be discussed in the Correlation Section. A benefit of the two parameter temporal plot is to keep more of the temporal information and can allow one to visuallly see a more complex trend like a delayed response trend. (Is there a scientific word for this?)

4.2.2 Correlation Scatter Plot

The applcation has a scatter plot feature designed for a scatter plot between two water quality variables. Water Quality data of two parameters are paired based on location and day of sampling. The two water quality observations are thus converted into one data point on the plot with the x location determined by the value of the first water quality parameter and the y location determined by the value of the second water quality parameter.

Trendlines can be added to. Similiar to the time-series plots, the methods for the trendlines are linear, Loess, and Generative Additive.

This anaylsis will be extended for water quality parameters to be correlated with metereological data and hydrological data.

4.2.3 Distribution

The distribution of Results of a particlular water quality parameter can be visualized in the application. Based on the user selected Locations, Parameter, and Date Range, the user can create a histogram, Density Plot, or box plots.

4.2.4 Heatmap (Interpolated Color Profile Plot)

The

4.2.5 Profile Line Plot

4.2.6 Phytoplankton

4.3 Statistics

4.3.1 Spatial and Temporal Statistics

Temporal Statistics can be computed with minimal effort in the application. Based on the user selected query of Locations, Parameters, and Date Range, the following statistics will be calcluated for each parameter: number of samples, average result,

minimum result, maximum result, 1st quartile (25 percentile), median, 3rd quartile (75 percentile), variance, standard deviation, geometric mean, and Mann-Kendall statistic. Any blank data (represented by NA in R) is ignored for these statistic calculations. The geometric mean is " ". The Mann-Kendall statistic

Before Statistical calculation, the data can be grouped by Location as well as various types of temporal schemes. The data can be grouped by year, season (independent of year), month (independent of year), season and year, and month and year.

4.3.2 Pearson Correlation Matrix

A pearson correlation matrix can be created in this application. Based on the user selected query information, a correlation matrix is generated to show the corrlation of parameters across all of the parameters that the user has selected (the user must select more than one parameter). Positive correlation statistics, R values, are shown in red and negative R values are shown in blue.

Confidence intervals are calcualted to determine the significance of the pearson correlation coefficient. This is crucial in case the user was to fasley interprate the correlation matrix as significant, when not.

"Find more out about significance. Add to App"

4.4 Geospatial Data Mapping

Home Tab

Spatial trend analysis is incorporated in many locations in the app but primarily lives on the map plot tab. Geospatial plots allows the user to easily compare a parameter statistic across all sites in a visual of plots on their choice of map. Spatial analysis also exists in any tab when multiple sites are chosen for analysis.

4.5 MetaData

4.6 Data Import

The Watershed data Importer Tool (WIT) facilitates importing raw data from multiple sources into the database. Each Data Type has a formatting function script that is written in R to format the data. As more data sources are added or data sources are changed, these can be uploaded into the database. The user is prompted to select dataset type and the user will be shown a list of raw data files in the appropriate dataset type location on their computer. The user then selects a file from their computer and then press a button to format the data. A Warning message will be sent to the user if their was a problem with the data or if the data already exists in the database. After a successful formatting, the user will be able to see the formatted data in a table on the screen and an import button will appear that the user can press to import the data if they are satisfied with how the data looks. Once the data is imported, the raw data file is moved from the unprocessed folder on their computer to the processed folder.

5 Discussion and Reccomendations

5.1 Pros and Cons of Application

Custombility

Upkeep

5.2 Future Work

5.2.1 Meteorological and Hydrological Data

5.2.2 Forestry

5.2.3 Reports

6 Appendix

6.1 WAVE Developer Manual

6.2 WIT Developer Manual

References

"Shiny - Tutorial." 2018. Accessed January 7. https://shiny.rstudio.com/tutorial/.

"Shiny - Widget Gallery." 2018. Accessed January 7. https://shiny.rstudio.com/gallery/widget-gallery.html.