Development of Application Software for Watershed Data Management, Visualization, and Analysis with Shiny Framework

Nick Zinck, University of Massachusetts

March 2018

Contents

1 Introduction

A comprehensive watershed protection plan involves the collection of water quality, meteorological, and hydrological data. Large amounts of data can be strenous to manage if proper systems are not put in place for data management. Poor data management can be detrimental and may result in data loss, poor quality data, or underutilizion of data due to the timely process of querying, visualizing, and analyzing poorly-managed data. Database software is a great solution to store and organize large datasets, yet database software often lack data visualization and analysis tools. Commonly, databases are paired with an outside application specialized for data querying, visualization, and analysis. Two applications were developed with the free and open source application development framework, Shiny, for facilitated watershed data management by the the Massachusetts Department of Conservation and Recreation (DCR). One application is designed for facilitated watershed data visualization and analysis and the other for facilitated watershed database data entry. This project is a product of the collaboration between UMass Amherst and the DCR.

1.1 Objective

This project works to facilitate the DCR's data entry, querying, visualization, and analysis process through an R-based application creation tool called Shiny. An application can automate many of the tedious day to day processes of managing a watershed as well as allow for powerful and expeditious visualization and analysis. Developing an application, rather than using propritery software, allows for full customization by the developer to better target specific needs of the agenecy responsible for watershed management and protection. A well fitting application can greatly increase the timeliness and ability to explain data and generate insights which can direct decision making for these agencies. Increased water quality data insight can also greatly influence an agency's sampling plan to better represent the watershed and focus on certain areas of high interest. The overall objective of this project is

to maximize the efficiency of the DCR's ability to manage, visualize, and analyize data to inform decision making. The application developed in this project can be used as an example for other watershed management agencies.

1.2 Scope of Work

Two applications were created to facilitate with watershed data management at the DCR. These applications will be used across the three watersheds: Quabbin, Ware River, and Wachusett, which are under the management of two seperate DCR offices, the Quabbin and Wachusett Offices. The applications are designed to meet the needs of both offices which require neccesary variations in the application features to address the variation between the two offices needs including differing sampling plans. Although some variations are inevitable, efforts have been made to make the data management of the two offices more congruent. A large piece of this congruency has been in the form of database alterations to make the organizization and formatting as similiar as possible between the two offices. This effort has also included moving data that exists outisde of a database into a common database where data observations all share a similiar tidy data regimen. Naming conventions were also examined and modified to simplify the application creation and decrease potential for mistakes.

Watershed data Importer Tool (WIT) is the smaller of the two applications which is designed an interface for facilitated raw data import of watershed field and laboratory data. Raw data from a number of predetermined sources can be imported through a simple user interface. WIT transforms raw data into the desired database format, provides quality control checks, and imports the data to the right location in the database. WIT will help ensure in the future that all data remains stored in databases as this tool makes database storage timely and efficient.

Watershed data Analysis and Visualization Environment (WAVE) is the larger of the two applications and as its name suggests it is designed to facilitate data querying, visualization,

and analysis of water quality. The application opens in a web browser and allows the user to query data by user selection inputs including locations, parameters, and dates. More advanced filters for data querying are also offered including filtering data based on meteoroligical events and excluding flagged data. The corresponding queried dataset will be presented in an interactive table which the user will be able to output as a csv file. Discrepencies between watersheds and datasets exist (and are desired) which are reflected in variations of query selections and filters in WAVE.

WAVE consists of numerous visualization and analysis tools to give more insight on a queried dataset. Visualization tools for tributary and reservoir data include time-series plots, correlation plots, and distribution charts (histograms, density curves, and box-and-whisker plots) to visualize trends and patterns on selected water quality parameters. Geospatial data visualization and analysis allows one to spatitally view data statistics on an interactive map. Heatmap and profile line plot tools are also available for reservoir profile data. Statistics can be quickly generated with WAVE including min, max, average for user selected temporal and spatial groupings. More advanced statistical analysis includes Mann-kendall statistics and pearson correlation matrixes. WAVE also makes information related to the sampling history of a specific site or a specific parameters easily accesible.

Both WAVE and WIT are shared openly through Github and are ran locally on a computer with minimal setup, although hosting the application online is also a possibility. The application is organized in a modular manner which eases future updates to the code as well as minimizes code repitition. Future additions to WAVE can be added as a seperate and independent module. A developer manual was also created to help future developers of WAVE as well as user setup guides for WAVE and WIT. As the needs of the DCR changes, both applications can dynamically change with it.

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 sources of the water supply are the Quabbin Reservoir and the Wachusett Reservoir which are managed in partnership with the Department of Conservation and Recreation (DCR). The system also has an emergency water supply source, the Sudbury and Foss Reservoirs [DCR, 2013]. The Massachusetts Department of Conservation and Recreation (DCR), Division of Water Supply Protection, Office of Watershed Management (DWSP) manages and protects the drinking water supply watersheds that provide water for approximately 2.5 million Massachusetts residents [DCR, 2013].the total watershed area of the active Reservoirs is over 200,000 acres the area of each watershed and reservoir can be seen in table 2.1. The Quabbin and Wachusett Reservoirs are protected and over 85% of the watershed lands that surround the reservoirs are covered in forest and wetlands [MWRA]. Table 2.2 shows the land use break down of each watershed. 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 transfers water from the Quabbin reservoir intermittingly to the Wachusett Reservoir to maintain the water level and water quality of the Wachusett Reservoir [DCR, 2007] which most always makes up the majority of the total inflow to the Wachusett Reservoir. The MWRA can also divert water from the Ware River, located between the two reservoirs, to either the Quabbin Reservoir or the Wachusett Reservoir through the Quabbin Aqueduct, though it is DWSP and MWRA operating policy to divert only to Quabbin [DCR, 2013]. Transfers generally occurs 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 decision

Table 2.1: Watershed Acreage

Watershed	Land area (acres)	Reservoir area (acres)	Total watershed area (acres)
Quabbin Reservoir	95,466	24,469	119,935
Ware River	61,737	0	61,737
Wachusett Reservoir	70,678	4,122	74,800

Table 2.2: Watershed Land Use Percentage

Watershed	Forestry	Wetland	Agriculture	Residential	Other
Quabbin Reservoir	88.2	5.6	2.2	1.5	2.5
Ware River	75.6	11.4	3.2	4.2	5.7
Wachusett Reservoir	67.3	7.7	5.7	10.8	8.4

making in reservoir management that can alter water quality of source water for the MWRA Supply System. It is essential to monitor the water quality in the whole watershed to best understand reservoir processes which can allow for more informed actions when water quality issues occur.

2.1.2 Water Quality Sampling Plan

DCR conducts extensive water quality monitoring of the surface waters (tributaries and reservoirs) in the water supply watersheds that are used to assess current water quality conditions and to establish ranges of values for parameters considered normal or typical [DCR, 2013]. Data collected in routine sampling over several years are used to assess watershed trends while shorter term studies may be conducted to evaluate specific issues [DCR, 2013]. The task of water quality sampling of the three watersheds in the water supply system are divided between the Quabbin and Wachusett office of the DCR Water Supply Protection Division. The Quabbin Office is responsible for the sampling of the tributaries in the Quabbin and Ware River watersheds and sampling in the Quabbin Reservoir while the Wachuchesett

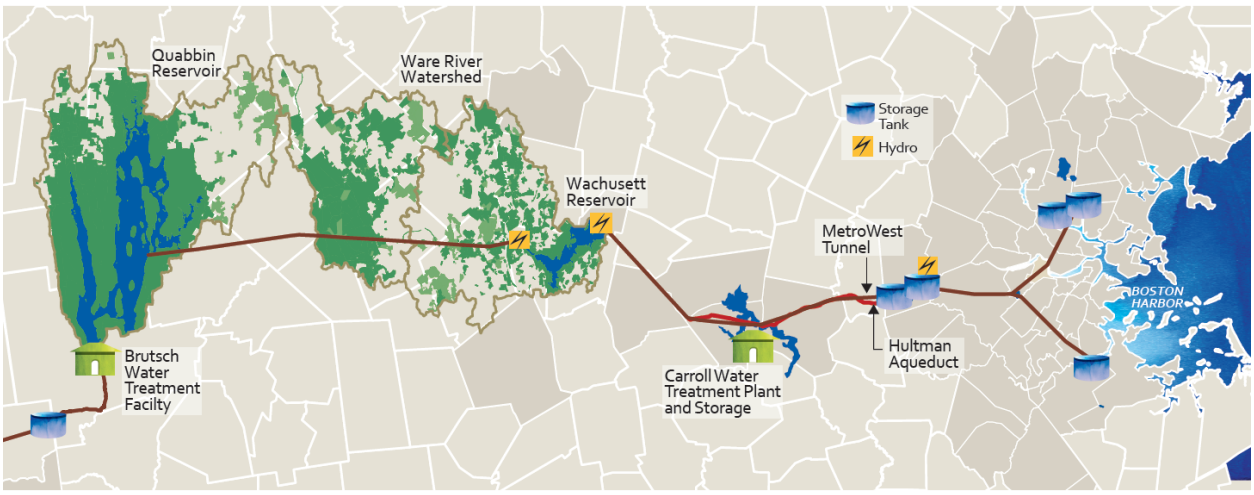

Figure 2.1: MWRA Water Supply System

Office is responsible for the sampling of the tributaries in the Wachusett watershed and the Wachusett Reservoir. Each office creates their own sampling plan for the reservoirs and tributaries in their respective watersheds which consists of a schedule of routine sampling and storm sampling at various sites. Each watershed is divided into sanitary districts [DCR, 2013]. Figure 2.1 shows all of the historical and active documented sampling locations of the DCR. More detailed sampling maps of each reservoir developed by the DCR are appended to this document.

The Quabbin Office's tributary water quality monitoring program of has included up to 14 sampling stations in the Quabbin Reservoir watershed, up to 10 sampling stations in the Ware River watershed [DCR, 2013]. Core sites are long-term monitoring stations, while EQA sites support ongoing evaluations of threats to water quality by sanitary district [DCR, 2013]. Reservoir and are monitored monthly by boat during the months of April through December, weather-permitting, with samples collected from several depths at each location. Tributary temperature, dissolved oxygen, pH, and specific conductance levels are measured biweekly with a multiprobe meter. Samples from core sites are collected by hand biweekly for turbidity, bacteria, and calcium analyses, while samples for nutrient analysis are collected quarterly. Samples from EQA Sites are collected biweekly for alkalinity, turbidity, bacteria,

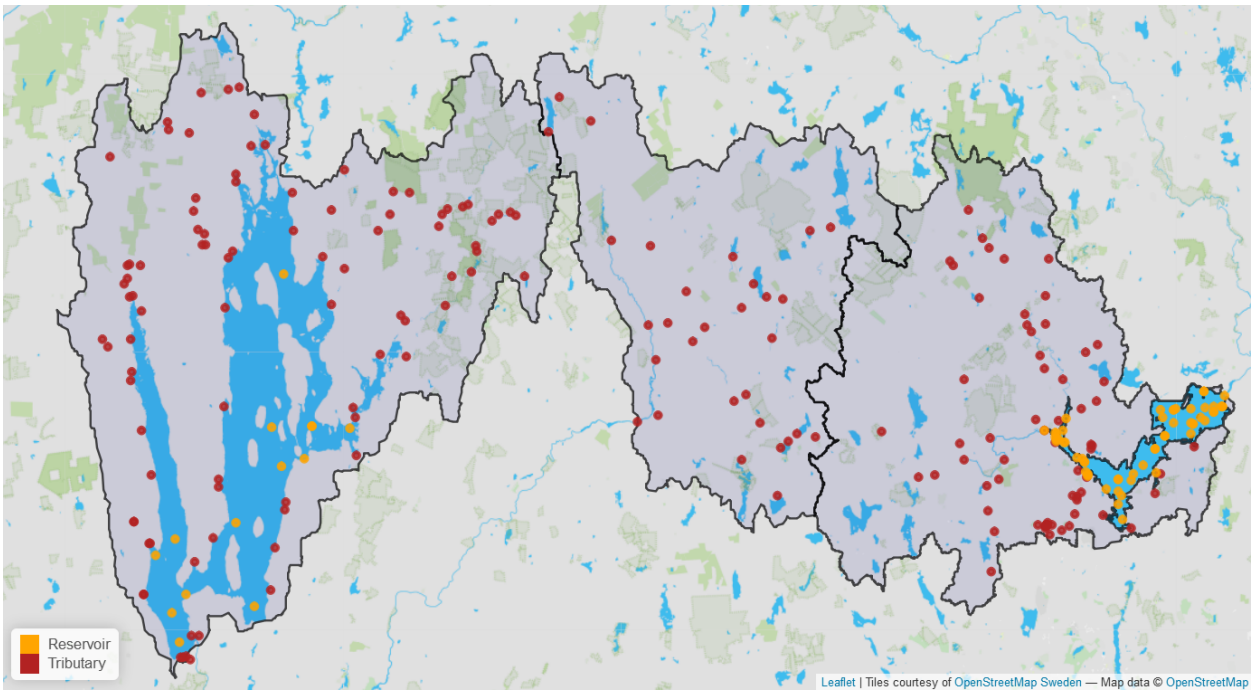

Figure 2.2: Watershed Sampling Locations

nutrients, calcium, and UV254 [DCR, 2013]. Water quality parameters that are measaured in tributaries are shown in figure 2.2.

The Wachusett Office samples each tributary station weekly or biweekly throughout the entire year, except for during inadequate flow conditions. Temperature and specific conductance are measured in the field and samples are collected for analysis of E. coli and measurement of turbidity. Nutrient samples are collected monthly for total phosphorus, ammonia, nitrate-nitrogen, nitrite-nitrogen, total Kjeldahl nitrogen, total organic carbon, total suspended solids, and UV-254. Reactive phosphorus (orthophosphate) samples were collected monthly and UV-254 samples were collected weekly from the Stillwater and Quinapoxet Rivers. Depth was recorded manually or using automated depth sensors at seven of the nutrient stations and flow calculated using rating curves developed and updated by DWSP Environmental Quality staff. Water quality parameters that are measaured in tributaries are shown in figure 2.2.

Reservoir Water column profile data for temperature, specific conductance, chlorophyll a,

dissolved oxygen concentration and percent saturation, are collected in both the Quabbin and Wachusett Reservoirs. Nutrient samples are collected at three depths from the epilimnion, metalimnion, and hypolimnion and analyzed for water quality parameters including nitrate-nitrogen, ammonia-nitrogen, total Kjeldahl nitrogen, total phosphorus, silica, UV-254, and alkalinity. Phytoplankton data is also collected at both Reservoirs. Surface grab samples are collected in the Wachusett Reservoir for E. coli and fecal coliform. Water quality parameters that are measaured in both reservoirs are shown in figure 2.2.

DCR DWSP and MWRA cooperate with U.S. Geological Survey to maintain continuous, real time recording gages at a total of ten sites nicluding sites at the Stillwater River, Quinapoxet River, West Branch Swift River, and East Branch Swift River [DCR, 2013]. Precipitation and other metereological data is obtained from NOAA weather stations. Additional sampling related to forestry practices is also conducted to compare water quality parameters between managed and unmanaged forests.

2.1.3 Previous Watershed Studies

Many studies have been conducted under the partnership of University of Massachusetts and the DCR. Data analysis and model calibration and validation in many of these studies have benefitted from the vast amount of water quality and hydrological data that is collected in the Quabbin, Ware River, and Wachusett watersheds. The water quality impacts from extreme precipitation events have been examined through statistical analysis 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 [Jeznach, 2013; Devonis 2011; Sojkowski]. Other water quality modeling topics have included the effects of climate change on the Wachusett reservoir [Jeznach,] and modeling the fate of Natural Organic Matter and fecal pollution [ ]. Presumably, having a thorough sampling plan and well maintained data can allow for more informed and facilitated research studies.

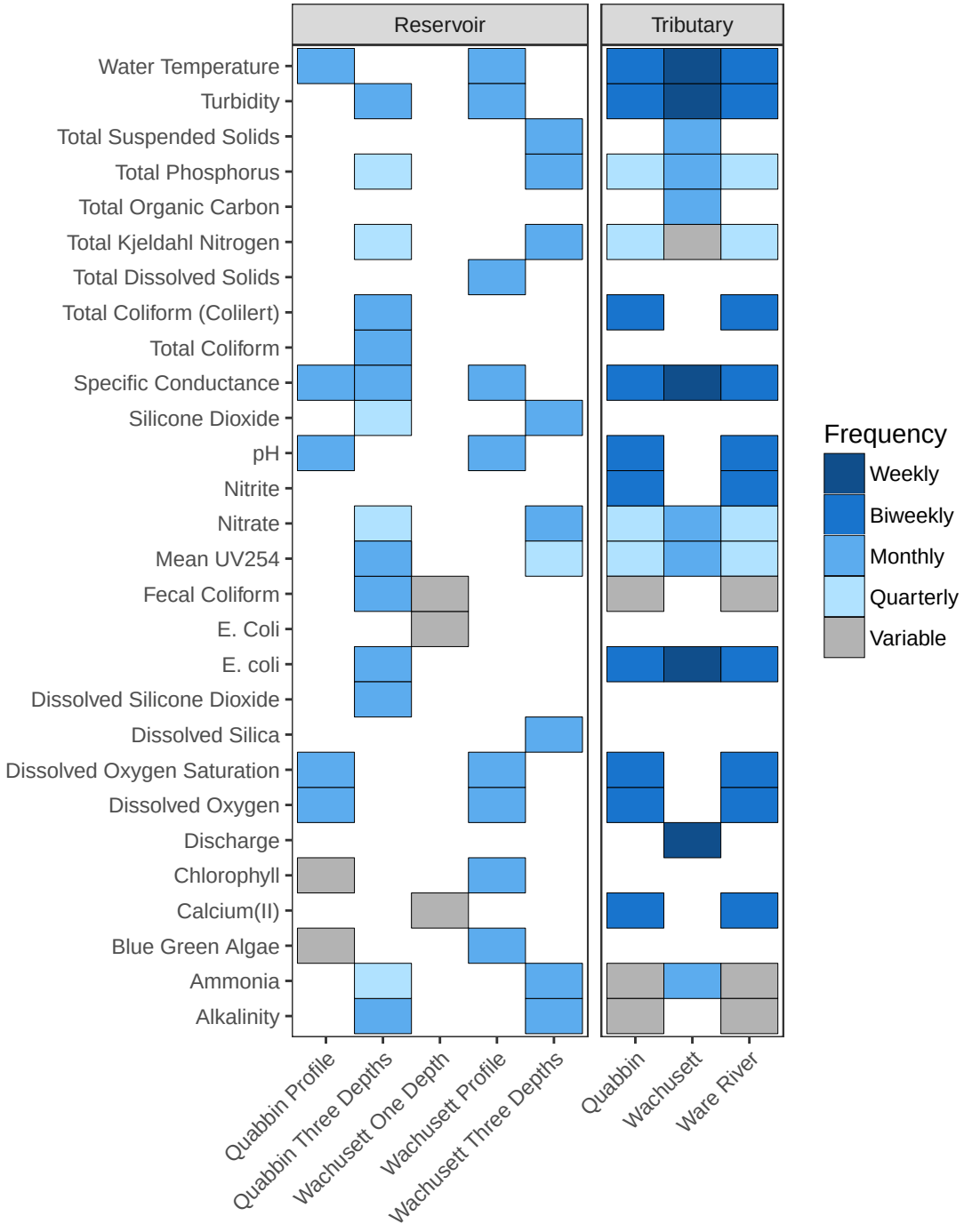

Figure 2.3: Needs to be changed!! (incorrect data!)

2.2 Watershed Data Management

The process of managing watershed data can be simplified into three parts: data collection, data storage, and data utilization. Data collection includes the act measuring data observations in the field or a lab as well as importing data measurements into a database. Data storage is the location in which the data lives and is ever changing as more data is collected. Data utilization includes the querying, visualization, analysis, or any other process from which conclusions can be drawn from the data to benefit decision making.

Proprietary software exist that can greatly aid in the watershed data management process. AQUARIUS, created by Aquatic Informatics, is a collection of leading application software for water supply management. Two of the five AQUARIUS applications are AQUARIUS Samples and AQUARIUS Time-Series. AQUARIUS Samples streamlines the production and management of environmental lab and field sample data [aquaticinformatics]. Aquarius Time-Series is a powerful platform for managing water resources and is used by the USGS as well as many other agencies and is notorious for building and maintaining rating curves [aquaticinformatics]. This paper will focus on water quality data management, thus AQUARIUS Samples better resembles the watershed data management system that this paper describes. Proprietary software like AQUARIUS Samples attempt to supply a versatile product that can be widely used on any watershed. This can be a difficult task due to the variations between different agency's needs and processes. Proprietary software is often limited in data visualization and analysisand cannot be customized within the software. Propietary software is also costly.

An alternative to proprietary software for watershed data management is to develope a personalized application using an open source application development framework paired with a database. With the advancement of programming languages and increased number of development frameworks, it is getting easier to build a custom dashboard or application. Creating an application, rather than using a proprietary software allows for more flexibility and customization. Developing an application is less costly than using a proprietary software,

yet it is likely much more timely. Oregon Department of Environmental Quality has develpped an application which provides an interactive means for users to query from multipe databases and evaluate status and trend at individual sampling stations [Byant, 2016]. This application is built with the Shiny application development framework in the statistical programming language R.

2.2.1 Data Storage

Due to the vast amount of water quality data collected by the DCR, it is essential to use a relational database for efficient data storage. Spreadsheet software is not an effective way to store large datasets. A relational database is a means of storing information in tables with rows and columns in such a way that information can be retrieved from it [oracle]. A table is referred to as a relation in the sense that it is a collection of objects of the same type (rows) [oracle]. In a database for watershed data, an object (row) can be a field observation, lab measurement, site location information, or another object. Data in a table can be related according to common keys or concepts to another table allowing for the ability to retrieve related data. A Relational Database Management System handles the way data is stored, maintained, and retrieved. Oracle, MYSQL, Microsoft SQL Server, PostgreSQL Microsoft Access are among the most popular relational database management systems [DB-engines]. Many of database management systems are free and open source. Applications can be designed to connect with a database to retrieve data from a database or write new data to a database.

2.2.2 Application development Frameworks

Application development frameworks assist with application creation and are essentially a reusable, “semi-complete” template application that can be specialized to produce custom applications [ ]. Selection of a proper application development framework can greatly decrease the amount of work one must do and knowledge one must possess to create an application.

Most application development frameworks can be considered either a front-end framework or a back-end framework which create a front-end server and a back-end server, respectively. The front-end server and the back-end server communicate through a common API, usually JSON, which is considered the universal binary [ ]. A front-end server is responsible for the construction and layout of the user interface, which is what the user sees and interacts with (often in a web browser). The user does not see the back-end server but this does the bulk of the work. The seperation of front-end server and back-end server can allow for increased customization as various front-end and back-end frameworks can be paired. Some frameworks serve as both a front-end and back-end framework which usually have a benefit of simplicity in the development process. This paper focuses on this latter type of framework that function as both a front-end and back-end due to the simplicity of the desired application. Likely, a mimimilistic user interface will suffice and most of the focus should be on the data science potential of the framework, which will likely be determined from the programming language that the framework uses.

R and Python are common programming languages for data science and both have their own collection of data science application framework libraries. Python is more widely used than R [ ], though R specializes in statistics and data visualization [ ]. Shiny is a development framework library in R created by RStudio which allows a relatively unexperienced developer to build an application in the R language and is very well documented. Application framework libraries in Python tailored toward data science include Bokeh, Spyre, and Dash [ ]. All of these libraries are free to use for developing applications and dashboards. These application frameworks leverage JavaScript and HTML to render the user interface which can be opened in a web browser. Each framework has its differences and some may work better than others for a particular purpose. Familiarity of a particular programming language can also influence the decision of which application development framwork to choose.

3 System Development

3.1 System Architecture

A system architecture is the conceptual model that defines the high level structure for the many working parts of an application (wikipedia). WAVE and WIT have a relatively simple system architecture relative to other web applications. Shiny is both a front-end and back-end framework which allows one unified server component, rather than a seperate front-end server and a back-end server. The Shiny server executes R code for data computations and creates the user interface through rendering javascript and HTML. The application also ran locally at the DCR which means that the client and server are both being ran on the users computer. An application that is hosted on the web has an additional process of sending the requests of a user through a proxy to help balance the amount of traffic that each server takes on. The user (client) sends information to the server through interactions with the user interface and the server will update the user interface accordingly. The server can fetch and retrieve data from the database (or intermediary data storage).

The WIT and WAVE system architecture can be seen in Figure 3.2 and figure 3.3, respectively. The common components that both systems have include the Server, User (Client), Github Code Script, Configuration File, and stored data (either in a database or RDS files). Prior to the start of a session for either WAVE and WIT, a configuration file with user settings is read and the latest application code is pulled from Github. After this step, the application system process differs as is seen in Figures 3.2 and 3.3. Each component of the system architecture are discussed in more detail in the following sectins.

3.2 Configuration File

Prior to the start of a session, a configuration file is read into R which gives information about the desired configuration settings like the location of database, Github repository, the office the user is in, and other settings. The configuration file is used for the purpose of

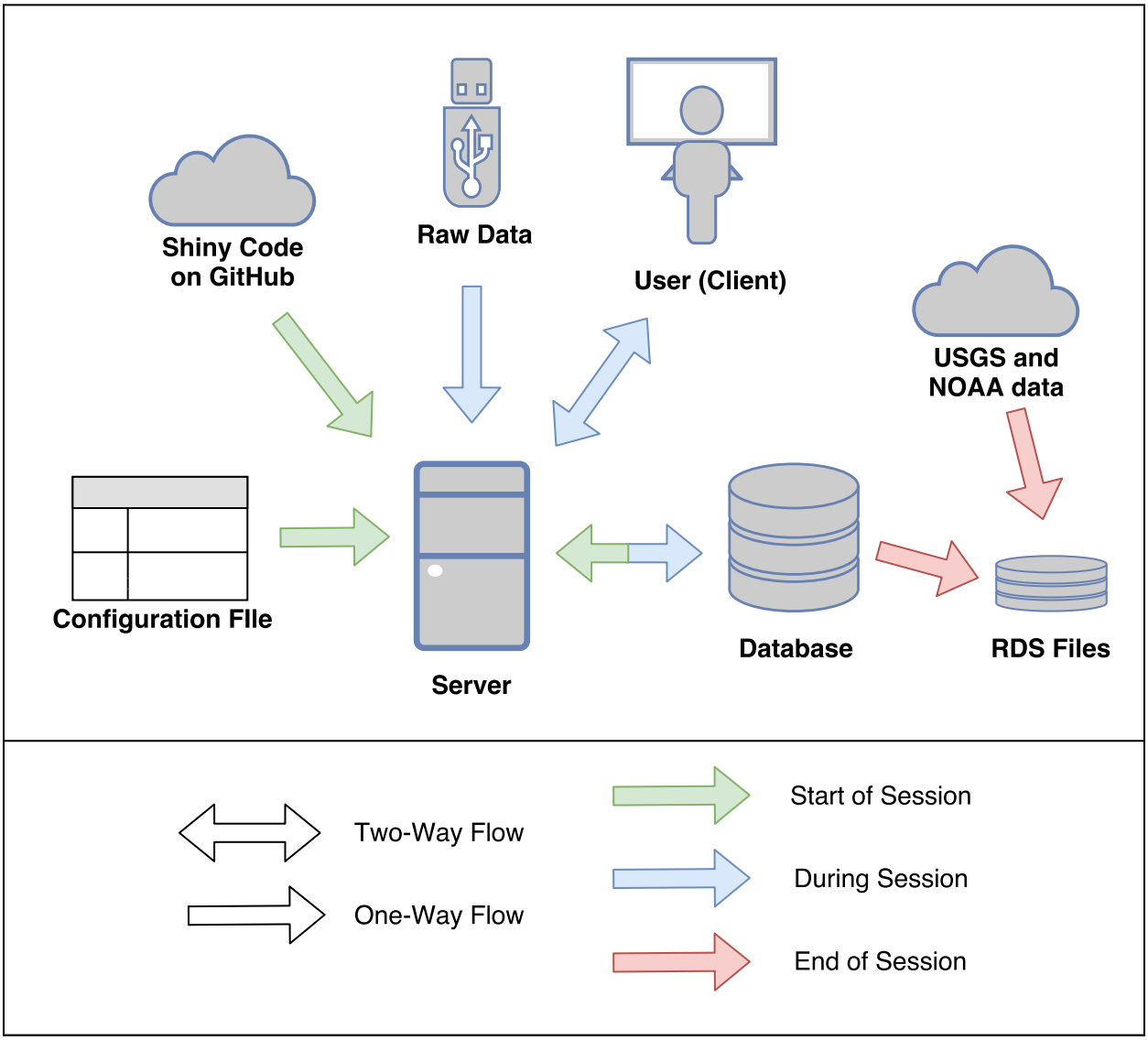

Figure 3.1: WIT Architecture - update remove green from db

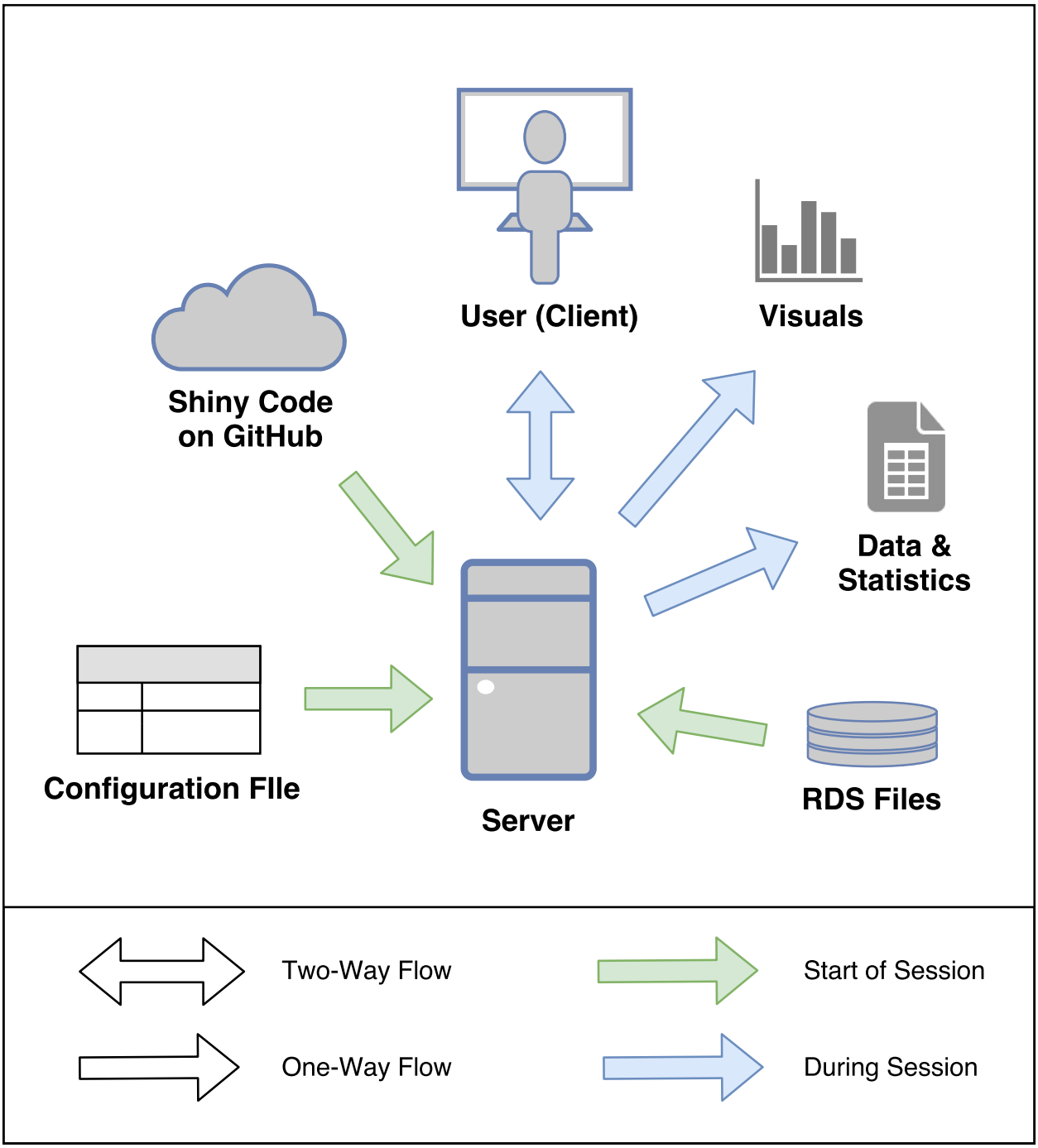

Figure 3.2: WAVE Architecture

allowing variability in application per user preferences. The Wachusett and Quabbin Office have different raw data and thus they will need different settings. They also have different databases and different directories where they store the raw data files. The configuration file can be the place of any future settings. The configuration file differs for WAVE and WIT. WIT has the more detailed elaborate configuration file due to difference in data sources. Once the configuration file is read, the application files are fetched from the specified GitHub repository and ran which opens the application and starts at a session. ## Github Code Files

3.3 Github

The code lives on Github. Github is a great tool for web developement and code collobaration. GitHub has version control and ...

3.4 Database development

Water quality data at the DCR has been collected since as early as 1984(8). Computer data storage was not yet common, and records were filed on pieces of paper. As computers became more relevant a mix of spreadsheet storage and database storage was used for data storage. Today, an effort is being made to move all data from historical paper records and those in spreadsheets to the database. Database are a much more efficient method of storing large amounts of data than either paper records or spreadsheets. The DCR currently uses the Microsoft Access relational database management system for their data storage needs. The Wachusett Office has runs three databases, one for water quality data, one for aquatic biological data and profile water quality data, and one for hydrlogical data. The Quabbin Office has one database for all of its water quality data. As a database can be seen as a collections of tables, it is not very important if all tables are stored in one database or if three seperate databases are used, though it makes sense to keep all tables that are related (i.e. have common keys that link them) in one database.

Table 3.1: Tidy Sample Data

Location	Date	Parameter	Result
Riverbend	01/01/2020	Turbidity	12
Riverbend	01/01/2020	Nitrate	13
Riverbend	01/01/2020	Ammonium	14
Riverbend	01/01/2020	Spec. Cond	15

Table 3.2: non-Tidy Sample Data

Location	Date	Turbidity	Nitrate	Ammonium	Specific Conductance
Riverbend	01/01/2020	12	13	14	15

There are many types of tables in each database. Within a particular table, similiar objects are contained in rows (oracle). Observation data is stored in a table, which in each row is a unique measurements of a parameter from the field or laboratory. Each observation is unique in its combined location, date, and parameter and gets its own row. This format is considered "tidy" and facilitates linking tables together in a relational database as well as for visualization and analysis in programming languages designed for data science. An example of tidy data and non-tidy data can be seen in figure 3.2 where columns named "Parameter"" and "Result" are used in tidy data instead of including a column for each parameter. The tidy data format allows for additional columns like "Units" and other information specific to that one observation to be recorded in the database. The tidy data also allows for the observation to be linked by foreign key field, like parameter and location, to another table with more information on a that parameter or location, respectively.

Due to the focus on data which represents field or laboratory measurements, tables for information other than this are considered metadata tables in this report and in the application. Metadata is a term used for a set of data that describes and gives information about other data [Google]. Metadata tables in the database include tables with information on parameters, locations, flags, sample flag index, and more. Location Tables gives location information like . Parameter tabels gives . Flag tables give informatin on what a flag ID .

Sample flag index table is used to relate the sample tables with the flag tables which is needed due to the one-to many relationship of samples to flags. Becuase a sample measurement can have zero, one, or many flags associated with it, an intermediate table is used to relate a sample ID with a falg ID, which are both foreign keys. An example of these database tables are shown in the appendix.

Both WAVE and WIT need to connect and access data from the four Access Databases. Connections are made with an open Database Connectivity (ODBC) application program interface to externally fetch data from a database and to write data to a database. WIT, wich is used to process and import data to the database, both writes adn reads data to and from the database. WAVE only needs to read in data from the database. There are commonly two schemes of reading data with an application, either read all data in at first and close the connection or connect each time the user asks for specific information and only return that information. Although the latter approach is more common, both applications use the prior approach to load in all data to the R session at the start of a user session. This is beneficial in our case for less opening and closing of connections since creating and removing a connection in Microsoft Access can be timely as well as lock the database fromother use if a proper system is not put in place to avoid this. In many cases, there may be too much data to be feasible to load in all at once. See figure 3.3 (one for WIT and one for WAVE)

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.

3.5 RDS files

Reading Data into the R session during a WAVE session was actually modified further make start up time faster. The time that it takes to fetch all the related data from R and make neccesary data transformations was about 15 seconds which is undesirable. There likely exists

other alternative database maanagemment sytems which are faster at this process, though Microsoft Access use has famiarity at the DCR and keeping this database management system provides some stability in an otherwise quickly changing data environment. In R, RDS files are the fastest format to be read and saved [ ]. Based on this knowlege, a decision was made to have the App read in directly from data in RDS files that have already been transformed to the desired format of WAVE. Reading in data from RDS files upon starting a WAVE session takes about 2 seconds compared to 15 seconds of the database fetching. There needs to be a means of how and when the RDS files become updated. Since WIT is used for data import, every time the user closes a WIT session, this cuases the action to run a batch file which runs a R file behind the scenes which will update the RDS files. The RDS files can also be updated by directly clicking on and running the batch file, to ensure that the data in WAVE is the most up to date, although this will likely not be neccesary. The date of the RDS files were lst updated is also saved and displayed in WAVE. Each dataframe is stored in its own RDS file with a descriptive name (e.g. df_trib_quab).

As previously discussed the data that is read in is in the form of RDS files for mainly speed considerations. The RDS files also already have the dataframes (a data storage ____ in R) in organized and seperated in the way that the Application uses them which simplifies the application code.

3.6 USGS and NOAA data

This architecture differs than the more basic previously described architecture with the added components of raw data, RDS files, and USGS NOAA data. WIT allows raw data on a users computer to be selected, processed, and imported into a database. When the session terminates, an R script is ran which fetches data from a database and transforms it and saves in as RDS files. As previously mentioned, this essentially works to keep the RDS files updated as new data is added to the database. The RDS files containing USGS and NOAA data is also updated be fetching USGS and NOAA data from _____ on the portal.

3.7 Visuals and Statistics

The WAVE application runs as peviously discussed with the added features of outputs in the form of visualizations and data and statistic tables. It is not desired to have a user of WAVE alter the database or data, so there is a one-way flow of data into the aplication. The data is read in to R prior to the application session from RDS files.

3.8 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

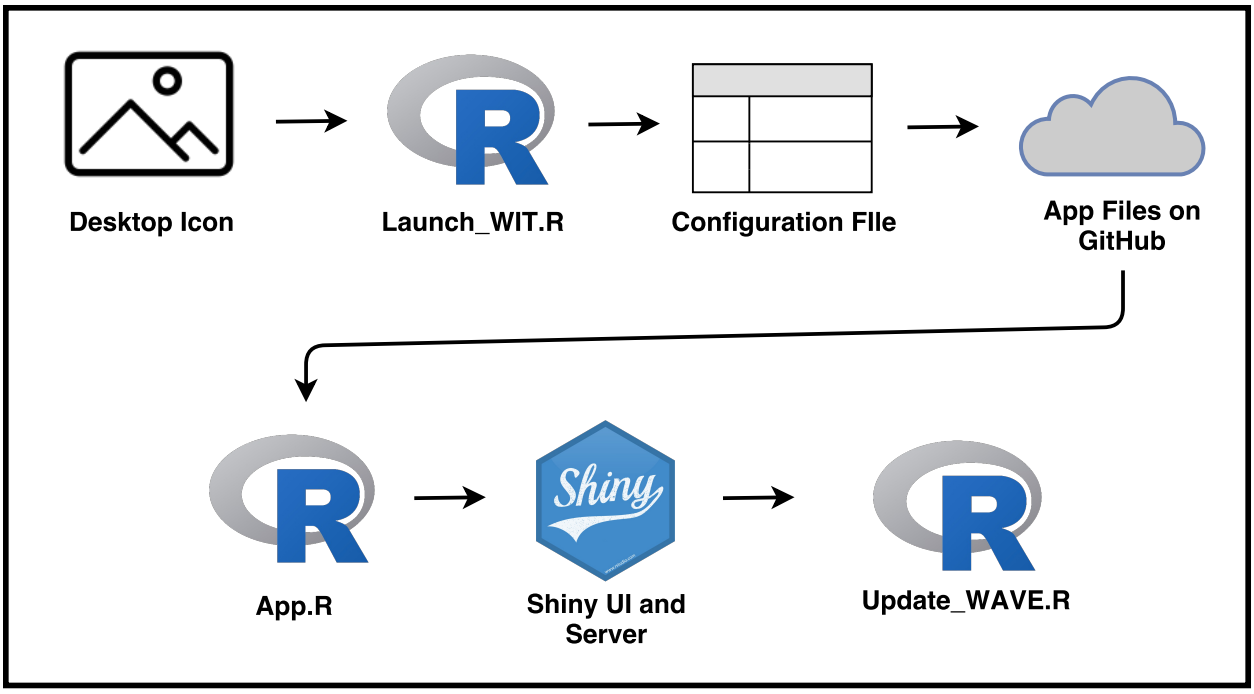

Figure 3.3: WIT Launch

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

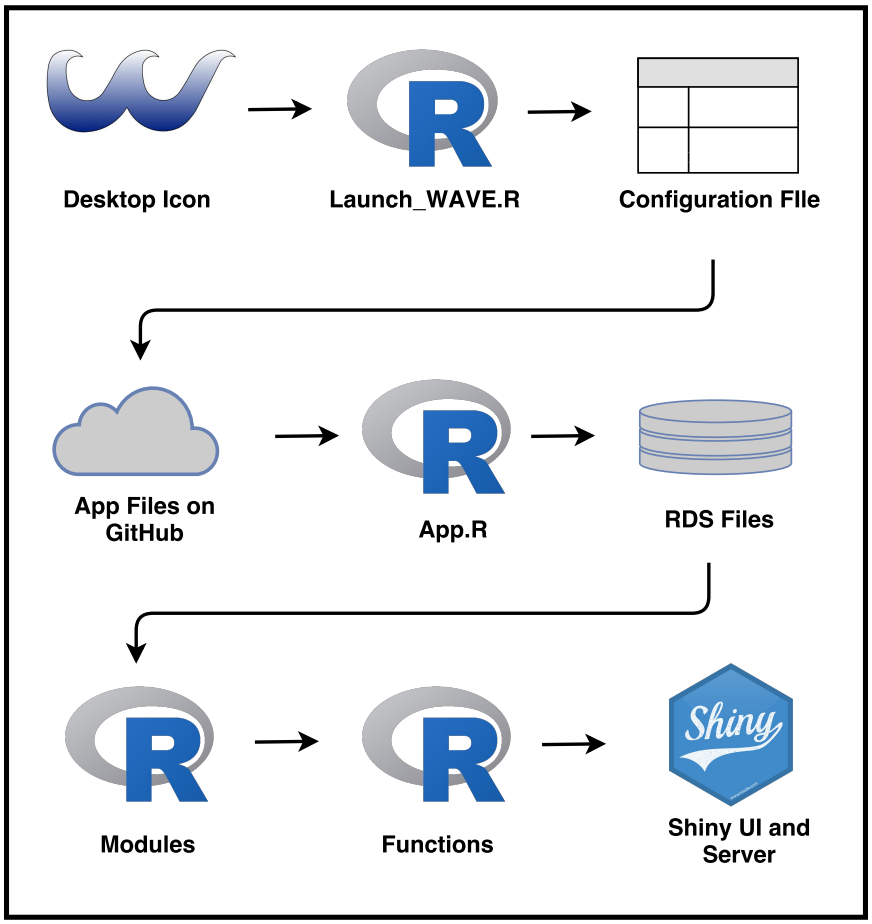

Figure 3.4: WAVE Launch

4 Code Development

4.1 Overview

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.

Dataframes are a concept used in R and also the Pandaas package in Python that are built for efficient data science. A dataframe is technically a list of named vectors. This differs

from a matrix which is a two dimensional vector. In a datafrmae, each list item is a named vector and relates to the column. rows are observations

Shiny is

Other libraries that the code uses include Dplyr, ggplot, . These packages are also created by RStudio and most of them are included in the tidyverse

4.2 WIT

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.

4.3 WAVE

The features WAVE offers results in a large code (base?) (there is roughly 10,000 lines of code with minimal repitition). Thus, organization of the code is crucial for efficient fixes, updates, and additions. The application as a whole can be viewed as a collection of individual dashboards organized in various tabs in the user interface. The user only evers sees one tab and thus one dashboard at a time. Each dashboard tab is essentially its own smaller

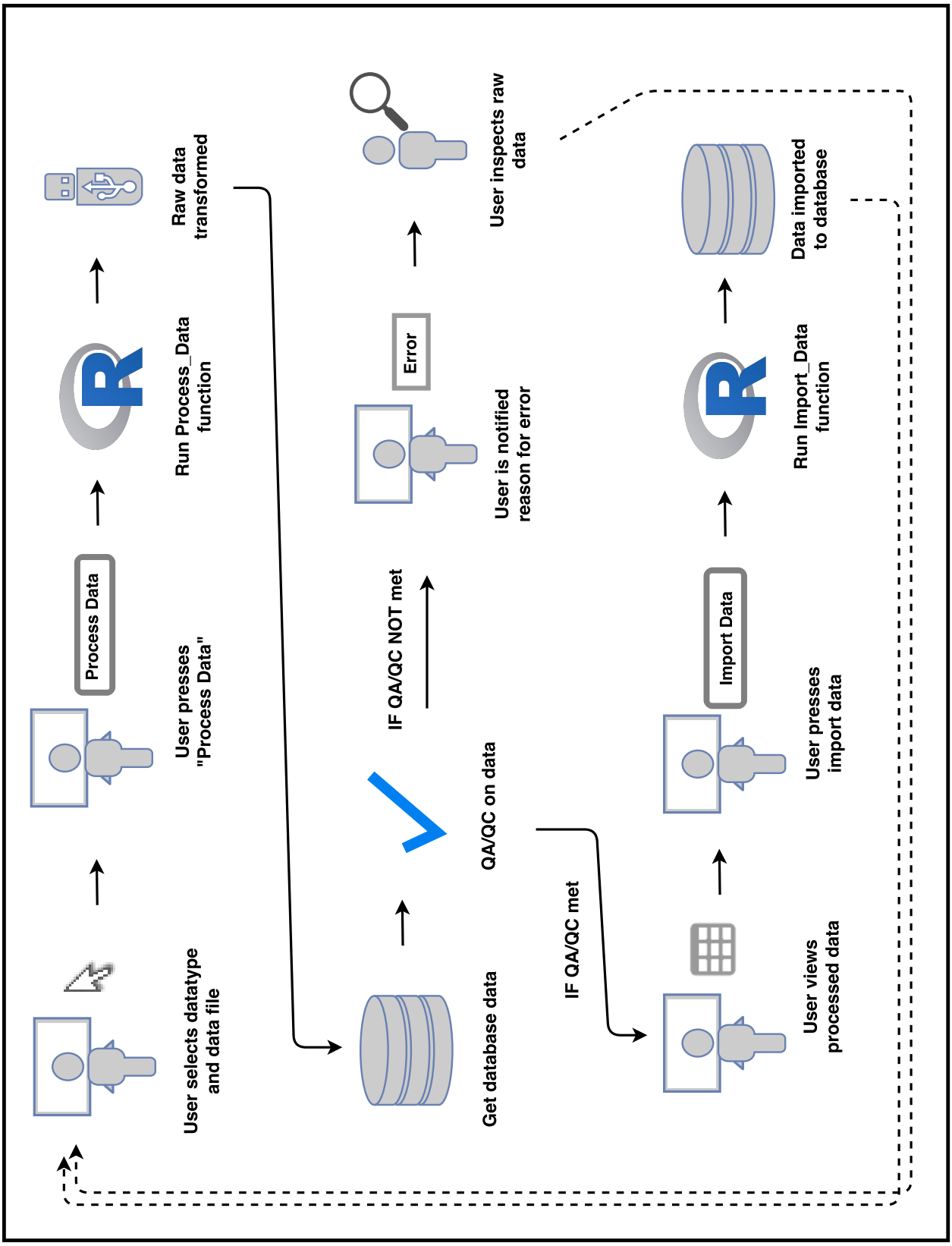

Figure 4.1: WIT Session Flow

application as it contains its own user interface components (inputs and outputs) as well as a server script which describes how each output is created. In the Shiny framework, modules are a way of defining these independent, self-containing pieces of Shiny code, seperate than in the main application user interface or server code. A module consists of a user interface function and a server function. Modules also reduce code repition in the same way as functions reduce code repitition. A module can be defined once, yet used in multiple locations. This is very beneficial becuase many features in the application are used for All three watersheds or both reservoirs. Just like functions, the modules can take arguments which allow for variations in their use. For organization purposes, the code for each modules user interface and server function are stored together in a seperate script file which is named based on the name of the module name.

The modules in WAVE are grouped in different categories and stored in folders accordingly. These categories includes Filter, Plots, Statistics, Metadata, and other. The Module layout can be seen in Figure . A module connected to the right of another module indicates that the module on the right is used within the module on the left. The breaking down modules into more modules is done for similiar reasons as why the application is broken down into modules. Having isoloatd specific code allows for more easier understanding of code. If one is working on a specific feature contained in a module, he/she will not be able unintenitionally alter other pieces of the code.

Diagram of Modules

4.3.1 Filter Modules

There is a group of module scripts in the application which are the part of the code which is responsible for the user selecting and filtering data. The largest filter module is the main water quality filter which is used for both tributary and all reservoirs selecting and filtering data. Arguments are used to create variations of the filter tab, for example adding a depth select range for profile data selection. The filter module also has the option to select all

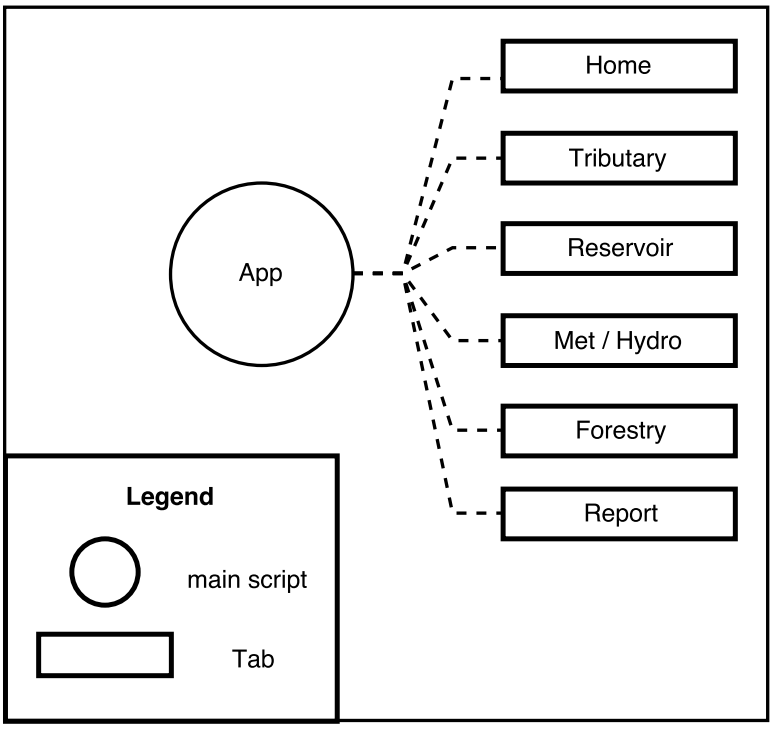

Figure 4.2: App Highest Level Tab Layout in the Top Navigation Bar

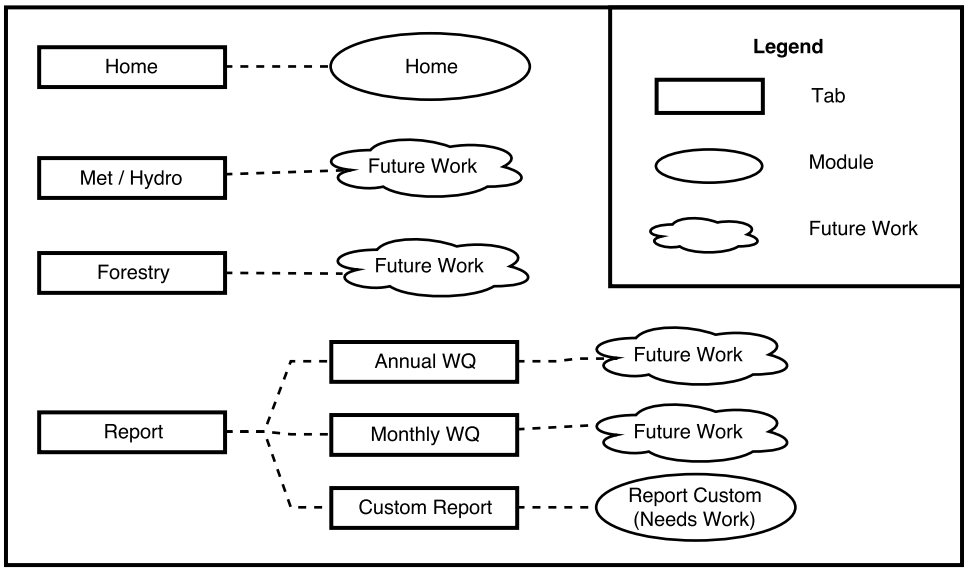

Figure 4.3: Tab Layout Structure for Home, Met / Hydro, Forestry, and Report Tabs

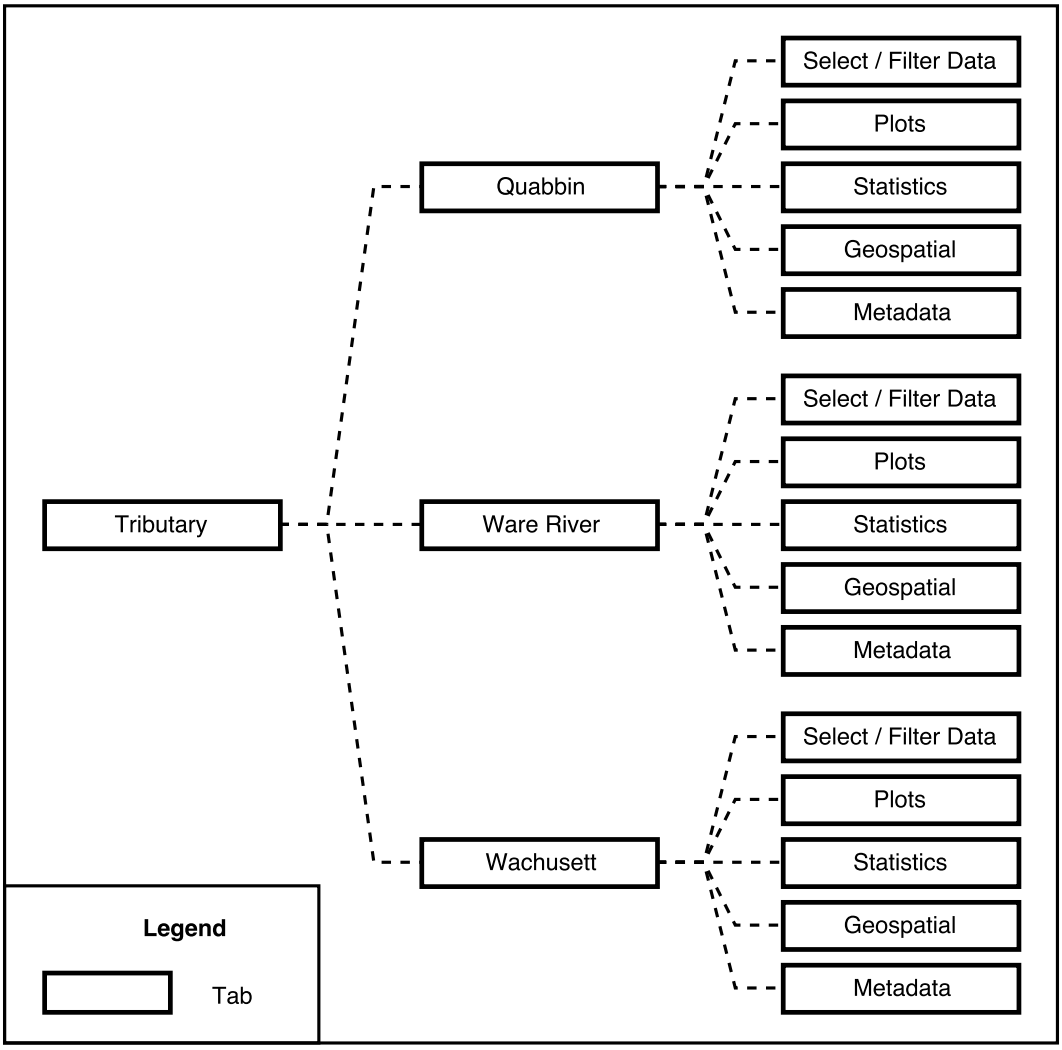

Figure 4.4: Tab Layout Structure of Tributary Tab. To be continued in Figure 4.6 and Figure 4.7

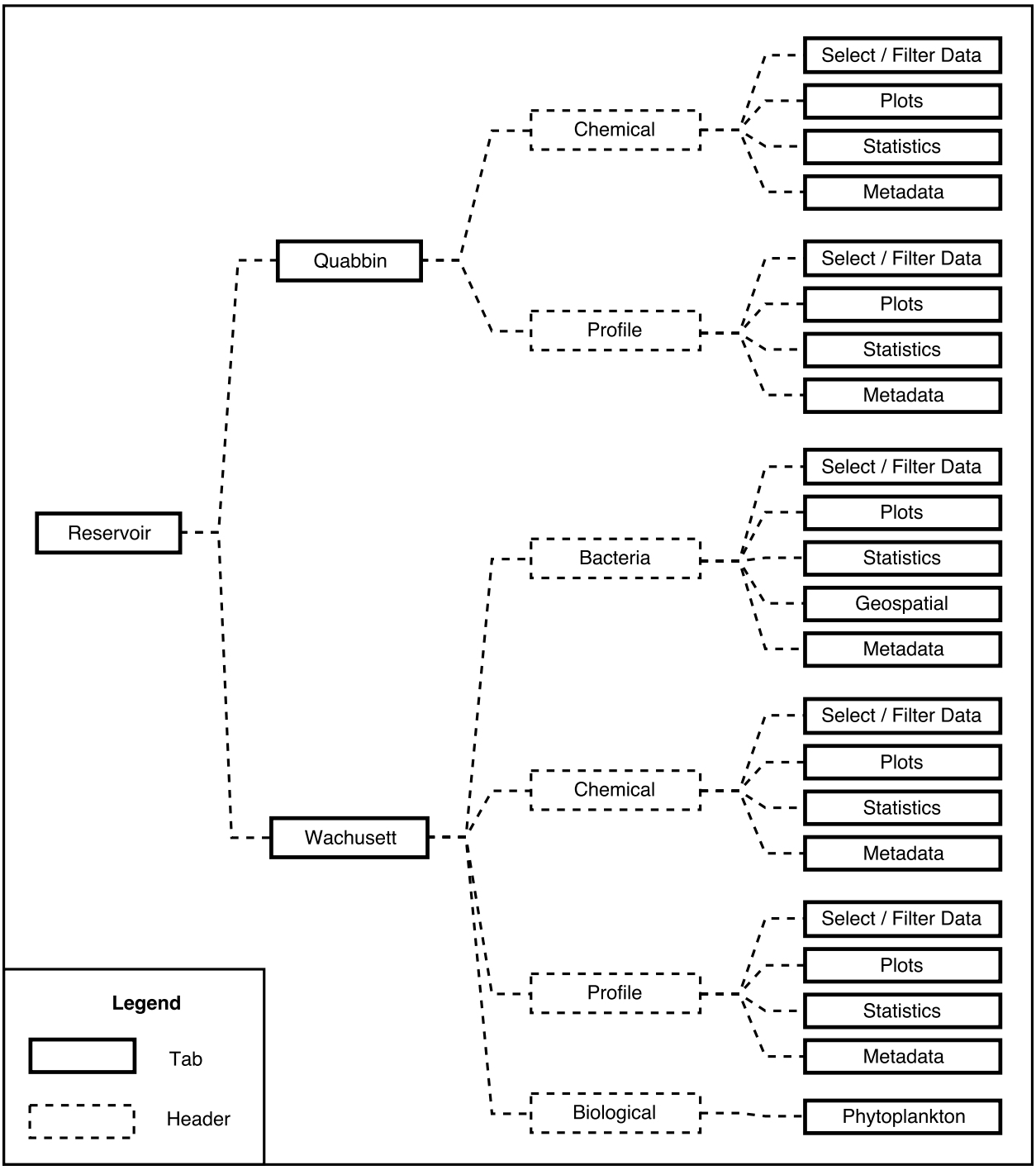

Figure 4.5: Tab Layout Structure of Reservior Tab. To be continued in Figure 4.6 and Figure 4.7

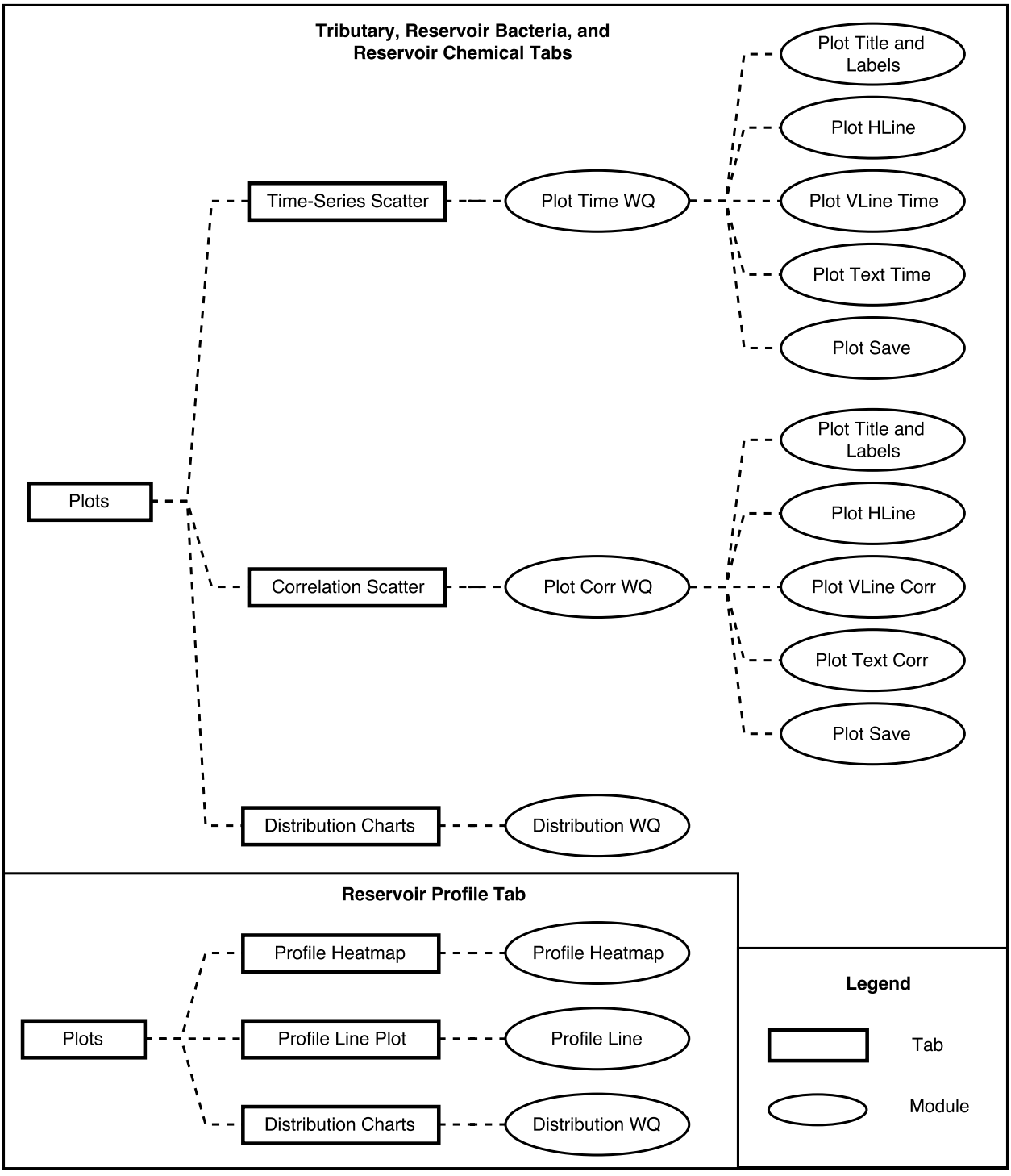

Figure 4.6: Tab Layout and Module Structure of Plots Tab. The Plots Tab appears in multiple locations in the Tributary and Reservoir Tabs. Plots for Reservoir Profile data differ compared with the Plots for Tributary and other Reservoir data, and this is reflected in a difference in layout

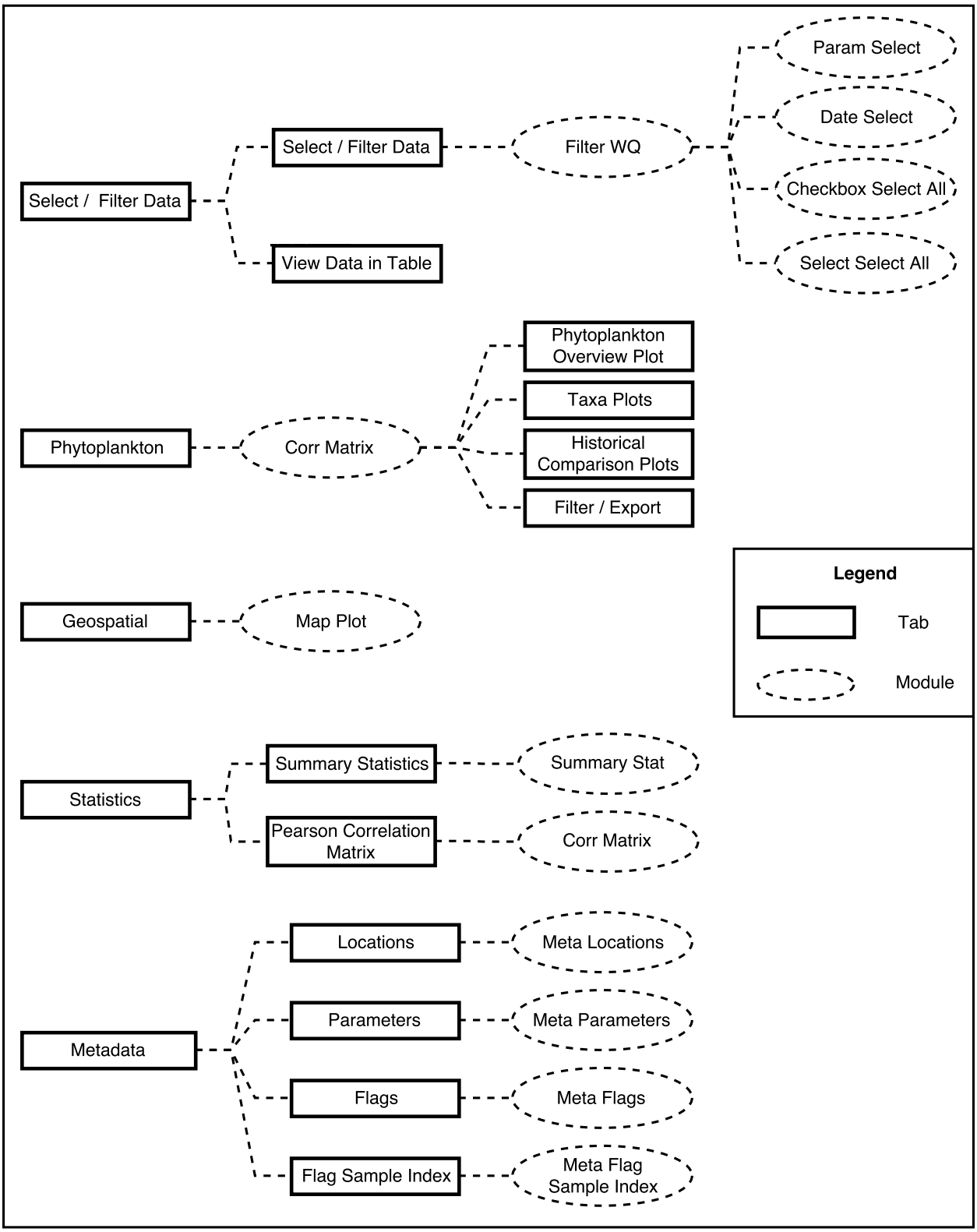

Figure 4.7: Tab Layout and Module Structure of Select / Filter, Phytoplankton, Geospatial, Statistics, and Metadata Tabs

data, which if the user selects this preference, none of the filters will be used and the whole dataframe will be selected. The selected/filtered data that the user selects in the filter tab is also used in the plot and statistics tabs. Sending the filtered dataframe to the plot and statistics tabs is preffered rather than each of the plot and statistics tabs having there own data filters due to code repetition

Filter modules also include smaller modules which are used inside the main water quality filter module (Modules on Modules!). These include modules that are essentially customized / souped up versions of the avaiable shiny widgets. They include site selections, parameter selection, data selection, a select input with select all capability, and check box input with select all capability.

4.3.2 Plot Modules

There are a number of modules that are considered plotting modules. These modules include the time-series plot, correlation plot, distribution, and profile plots. These plot modules use the data that the user selects in the filter tab. Depeding on the desired data that each plot uses, additional selections/filters will be promted for the user to narrow down the selected data. For example the time-series plot can only handle one or two parameters at a time, on its primary y-axis and secondary y-axis, respectively. The user is thus given an additional parameter seelction specific for the plot, which will narrow down the list of selected parameters to jsut one or two. The plots in the applicaiton are generated with ggplot (the ggplot2 library in R) which is Hadley or RStudiod?. ggplot allows for easy grouping by aesthetics (x, y, color, shape, size, facets).

Including in the plot modules, are also plot input widgets which have been seperated out to reduce the clutter in the main plots. These widgets inluded title and label widget, plot save widget, . These widgets are called within the larger plot functions and serve to adjust the plot via user inputs. For example, the title and lable widget contains user interface components to allow the user to choose their title and axis labels and on the server side it will

take an intermediate plot object and add the indicated titles and labels to the intermediate plot object. The save plot, acts differently as it does not adjust the plot, yet creates the UI and server for outputting the plot.

4.3.3 Statistic Modules

There are a number of statistical modules that are used. R was built for statistical programming. The combination of a group_by and summarze funcitons in Dyplyr allow for an easy way to generate statistics from a dataframe groupedwith various user selected inputs. The pearson correlation coefficient is also calculated using the ___ library. The Mann-Kendall library is also used to calculate the Mann-Kendall Statistic of set of data of a certain parameter at a site.

4.3.4 Metadata Modules.

The Metadata Hey for

4.3.5 Other Modules

Other modules, that do not fall into the precious categories include the geospatial plot, met

4.3.6 Functions

4.3.7 Naming Conventions

Naming conventions have been set to allow for an easier understanding of the code. Due to the large amounts of objects, it can be

4.3.8 Developer Manual

Table 4.1: Naming Conventions

Input Object	input object name
Output Object	output object name
function	FunctionName
Reactive Expression	Reactive Expression Name
Module Server Function	MODULE NAME
Module UI Function	MODULE NAME UI
file	file name.R

5 Application Features

5.1 WIT

5.2 Raw Data File Lookup

WIT will provide an easy selection of the raw data files on ones computer. The user will first select the dataset cetegory that he/she desires to import. Since WIT is configured to read from specified directories on ones computer, the user will then select a raw data file from a dropdown list. The file list in the application will consist of the raw datafiles that are saved in the appropriate directory on a users computer. After the user imports the data, the raw data file is moved from the ____ folder to the processed folder.

5.3 Data Processing

The data is processed to convert raw data into the format of the same type of data in the database. Each raw data source varies in the format or data. The application will automatically process a raw data file into the diesired format, by one of a number of processing script files for that specific raw dataset type. If the processing is succesful, the data will appear in a table for the user to examine.

5.4 Quality Control

There are many quality controls that are placed in the processing scripts. Some are general ot all scripts and some ore specific for each.

5.5 Data Importation

Data is imported into the correct location of the database with a click from of a button.

5.6 WAVE Features

5.6.1 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.

5.6.2 Data Visualization

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

5.6.2.1 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?)

5.6.2.2 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.

5.6.3 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.

5.6.3.1 Heatmap (Interpolated Color Profile Plot)

The

5.6.3.2 Profile Line Plot

5.6.3.3 Phytoplankton

5.6.4 Statistics

5.6.4.1 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.

5.6.4.2 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"

5.6.5 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.

5.6.6 MetaData

5.6.7 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.

6 Discussion and Reccomendations

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

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)

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

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.)

6.1 Pros and Cons of Application

Custombility

Upkeep

6.2 Future Work

6.2.1 Meteorological and Hydrological Data

6.2.2 Forestry

6.2.3 Reports

7 Appendix

7.1 WAVE Developer Manual

7.2 WIT Developer Manual

8 Extras and Trash Bin (Not a real Section)

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 development 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.

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.

9 References