#### Case Study: Establishing a progressive data management workflow for biological data to inform adaptive management decisions

Abstract

Data management and data collection techniques have been advancing at a rapid pace over the past decade. A constant struggle in biology is to keep up with these advancements and the lack of reproducibility and accessibility to these data make these challenges apparent. These challenges to analyze data in a timely fashion hinders the ability to make rapid and informed decision making in ecological efforts. We customized a modern workflow for continuous and discrete long-term ecological data to assist in adaptive decision making that focuses on tackling many of the data management concerns with these types of data. We accomplish this by 1) standardizing field data management methods; 2) perform quality assurance and control (QA/QC); 3) create scripts to analyze data and inform decision making; 4) and 5) use a workflow using version control to track changes make updates to scripts and documents as necessary. The workflow uses open source software and tools to create a modern-day data management structure, which can be implemented in many efforts.

#### Introduction and Background

Traditional field biology programs, many of which are designed to monitor animal populations and their environments, have experienced a substantial evolution in data collection, management, and storage technology in recent years. Changes include new sensor technology, data collection methods, and data observing platforms that are being used in large-scale monitoring programs including SECOORA (Southeast Coastal Ocean Observing Regional Association) and NEON (National Ecological Observing Network). As an example, advancements in sensor technology have allowed for significant changes in water quality monitoring such as transitioning from discrete single location and single point in time sample collections to real-time continuous observations at multiple locations. While the scale and technological capacity of many monitoring programs has increased these monitoring programs are still most often conceived, planned, and used by personnel trained as biologists and not data scientists. The lack of training in basic data management, curation, and workflow of data generated from these types of data collection platforms was demonstrated in a recent NSF (National Science Foundation) survey (Lowndes et al., 2017) which highlighted that of the 704 scientists who participated in the survey, “data skills” (e.g. multi-step workflows, ability to store, share and publish data) was identified as the largest unmet need (Barone, Williams, and Micklos, 2017; Lowndes et al., 2017).

The US Gulf of Mexico region is undergoing a large restoration effort to reverse observed declines in key ecosystem components including seagrass, fish communities, and oyster reefs using funding from the consolidated Deepwater Horizon settlements (see https://www.nfwf.org/gulf/Pages/home.aspx as an example). These restoration projects vary in spatial scale and funding, but, like other restoration efforts, these projects will have data collection and evaluation efforts that occur frequently throughout the project. These projects explicitly require an adaptive management plan to guide the restoration process. Extensive data management plans are mandated with the overall purpose of creating opportunities to improve future restoration actions by maximizing learning from previous and ongoing restoration efforts.

Our project is a large-scale restoration effort in the eastern part of the Gulf of Mexico funded by NFWF-GEBF on Lone Cabbage Reef (LCR). Our project’s primary goal is to restore specific historical oyster reefs so that they may be resilient to changing sea level and river discharge. This project generates data from multiple sources including continuous autonomous water quality data via sensors and observations of oyster populations by field biologists. These data are generated at different time frequencies with sensor data obtained at hourly time intervals from multiple spatial locations and biological data collected at discrete time intervals from multiple spatial locations. For both cases, there is a need to prepare data, meet data quality standards, and complete routine analyses of data to ensure data collected are useful for project objectives and questions. For adaptive management to work efficiently in our project, we developed a system which captures data as it is collected, guides the data to analyses, version control and data storage. Our data management system is essential to improve data quality by reducing the likelihood of data collection and analysis errors.

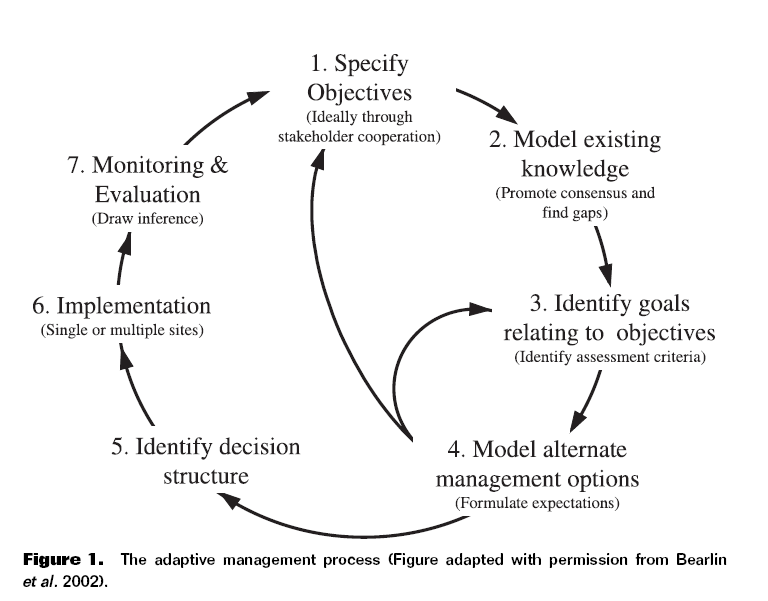
#### Box 1. Terminology

**“Living data”**

“Living data” are defined as data which are continuously collected and updated. These types of data are critical to this type of adaptive learning to inform restoration and management actions (Yenni et al., 2018). These informed adaptations during a restoration project can be small such as shifting the location of an autonomous sensor, to larger changes including restoration practices or revamping of sampling programs because of low statistical power. Living data are challenging to work with from a data management perspective because the data (by design) change as new data are collected and usually frequently. In a restoration or management context as these data are being collected, they must be processed, and analyses of these data to be completed to help draw inferences on how the system of interest is responding to the restoration action. This idea of iteratively integrating new data, analyses, and comparing these outcomes with previously stated objectives is not new and is a central aspect of the “adaptive management” process for natural resources first described in the 1970’s (Holling 1978; Walters 1986).

**Adaptive management**

By design, an adaptive management program requires rapid feedback between data collection, analyses, and interpretation to drive the process of updating knowledge, examining management and restoration options, making decisions and implementing actions. This process is repeated (Figure 1) to improve management actions such as identifying the best restoration approach. To carry out a restoration project adaptively, data used in these continuous efforts meet quality assurance/quality control (QA/QC) protocols to identify and correct inconsistencies and errors in field or sensor observations before these data are used in an analysis. Errors in these data, or delays in producing the data in a usable framework, can quickly lead to a breakdown in the adaptive learning process either in terms of slowing the analyses limiting their utility for timely decision making, or worse, erroneously informing the decision-making process because of errors in data management or analyses.

Figure 1- The adaptive management process (Schreiber et al., 2004, adapted from Bearlin et al., 2002).

#### End of Box 1

Though there are only a few team members collecting and analyzing data, we have tackled some challenges to create and implement our current data management workflow. Our study involves water quality observations collected every hour every day and humanly observed oyster counts, and measurements surveyed during summer and winter. We created a data management workflow to efficiently process and analyze data to actively inform decision-making on efforts such as the amount of sampling trips needed to optimize oyster density estimates. We use software and tools that are open sourced, widely available and familiar to most field biologists such as R and Microsoft Excel. This paper explains our approach with the purpose of creating a guide for conservation efforts and emphasizing the necessity for establishing a data management workflow.

**#### Box 2. LCR data type generation**

The LCR project generates data from autonomous sensors, and humanly observed counts/measurements. The project is funded by NFWF-GEBF, over an 8-year span, to reconstruct the Lone Cabbage oyster reef and monitor the surrounding water quality. Several types of data are collected at various frequencies (seasonally, bi-monthly) and each data type requires individualized attention.

*Highest-frequency, autonomous water quality observations*

We collect hourly water quality observations from 11 different sites around Lone Cabbage reef. These observations are downloaded from the field bi-monthly. Checking on these sensors, manufactures Diver and Star-Oddi, also allows for regular maintenance of the sensor and its protective housing to ensure continuous functionality. These data are the most intensive and require strict data management protocols (Box 3).

*Medium-frequency, oyster counts and measurements*

Every year we count oysters along transects and measure their lengths, at randomized locations, during the sampling periods of summer and winter. These data are recorded in the field on datasheets and translated into a computer through a dual-entry system. These data can be intensive as they require dual-entry and checking prior to being analyzed. Our data management workflow reduces the chance for human introduced errors (Box 4).

*Low-frequency, water quality YSI measurements*

During water quality service trips, we also collect YSI measurements to provide a supplemental check on our autonomous sensor observations. These measurements are recorded once during the water quality service trip for each site location. These observations are the least intensive as their frequency is low, and they are manually entered in the MySQL database (Box 3).

Though these data types may seem relatively similar, they are collected and handled differently to ensure data integrity. Addressing the variety of concerns that have been discovered through regularly updating these data types can also address many data management challenges which researchers may confront.

**#### End of Box 2**

### Establishing a modern data workflow

Prior to the start of the LCR project there was a need to create a workflow to manage and store water quality observations. This need required the expertise of the University of Florida Academic Research Consulting & Services (ARCS) to provide guidance on how to implement this workflow. The justificaiton to create a streamlined workflow was to track multiple autonomus sensors through space and time regardless of where they are located. The workflow we have developed [Figure 1] for water quality management requires some level of konwldge of computational tools (e.g., MySQL and R) and verstion control (e.g., GitHub, git). More than ever, data management is recognized as a core skill for biologists and ecologists (Hampton et al., 2017). This manuscript illustrates our project’s data management workflow for the purpose of providing guidance where others might be able to implement this system into their own conservation efforts. Boxes 3 and 4 describe our workflow implementation.

**#### Box 3. Water quality workflow**

For more information about the MySQL import process go to (zenodo link for MySQL).

1. Prepare physical data sheets with pre-populated fields, such as location and date.

2. Download hourly water quality observations and record water quality service trip summary and YSI one-time measurements.

3.A. Import all hourly water quality observations, using a Python import process, into our custom-built MySQL database for permanent unprocessed storage.

3.B. Use R programming to clean, analyze, and visualize observations.

3.C. Use a standardized GitHub workflow (zenodo link for Github workflow) to update and store processed data, scripts and documents.

Figure 2- Data workflow for water quality observations

1. Datasheets include pre-populated fields including the location, date to minimize error. 2. Water quality hourly sensor observations are downloaded in the field, unto a field laptop, and the technicians physically write the observation counts and any notes pertaining to the download. YSI measurements are also recorded at each location and physically written into the same datasheet. 3.A. Files are uploaded into a secure University of Florida internal server and a trigger starts the Python import process into our MySQL database, permanent storage. YSI measurements are manually entered into our MySQL database. The Python import process includes QA/QC procedures such as duplicate observations flagging. 3.B. QA/QC R scripts pull and process the water quality observations to check for flatlined or out of bound measurements. 3.C. Processed data, scripts, and documents are then stored in GitHub. Standardized GitHub workflows are used during collaborative projects to ensure proper version control utility.

**### End of Box 3**

**Adding water quality measurements to our permanent MySQL Database and version control**

Our MySQL database permanently stores water quality observations and can be access by any team member through a username and password. We store water quality data in specific tables where the sensor serial number and location must be pre-defined prior to importing the sensor observations (zenodo link for mysql). These pre-definitions allow us to track which sensors are in which location at a specific time, which addresses our most challenging concern. We use R scripts to pull these unedited observations and process additional QA/QC procedures. These processed observations and their accompanying scripts are then updated using version control in the project GitHub (<https://github.com/LCRoysterproject>) in our master data repository. This repository includes an up to date `master` branch which is protected from any unintended or incorrect updates using GitHub repository restrictions. To submit any changes to the `master` branch (referred to as a “pull request”), it is mandatory to have the changes/edits reviewed by another member of our team to ensure data integrity (zenodo GitHub workflow link). Every “pull request” requires a thorough message describing each change, in the event an update to the `master` branch has to be investigated. Version control allows for team members to view a previous iteration of the `master data` branch and go back to that iteration if needed. This workflow protects the `master branch` from possibly merging accidental or incorrect changes, giving a layer of needed security to the data.

**Automated data checks through Python and scripts**

Water quality observations are imported into our MySQL database through custom Python scripting. The Python import process provides QA/QC procedures such as flagging duplicate water quality observations. If observations are flagged through the Python import process an inspection of these observations take place to find out if they are truly duplicating. All unique observations are imported into our MySQL database, where they will be additionally reviewed via R programming scripts. The R scripts check for out of range measurements and additional scripts remove flatlined water quality measurements (usually due to ocean fouling). These R scripts are manually ran and

**#### Box 4. Oyster observation workflow**

For more information about our dual entry system visit here (zenodo link for data packet). Note, many of the lab processes are similar to the water quality workflow.

1. Prepare physical data sheets with pre-populated fields, such as location/site and date.

2. Team members walking along transects will count and measure oysters and record their exact GPS coordinates, oyster heights, and oyster counts, which they will physically write these measurements on data sheets.

3.A. Dual entry system with data validation, and reconciliation (zenodo link data packet).

3.B. Internal reporting and power analysis through scripts.

3.C. Use a standardized GitHub workflow (zenodo link for Github workflow) to update and store processed data, scripts and documents. This is the same process as the water quality workflow (Box 3).

Figure 3- Data workflow for water quality observations

1. Datasheets include pre-populated fields including the location, date to minimize error. 2. In the lab the team member who went out into the field, usually but not always, will enter their observations in a data packet that includes a dual entry system and data validation tools to ensure that the data entered into the packet is within range and standardized (e.g., site location, capitalization, appropriate oyster height range, etc) . 3.A. Use R programming to create internal reporting to estimate densities and power analysis which influence the number sampling trips needed for the season. 3.B. QA/QC R scripts pull and process the water quality observations to check for flatlined or out of bound measurements. 3.C. Processed data, scripts, and documents are then stored in GitHub. Standardized GitHub workflows are used during collaborative projects to ensure proper version control utility.

**#### End of Box 4**

**Data validation on newly added oyster data**

Oyster counts and measurements are subject to data validation feature through Microsoft Excel. Data validation ensures that every new observation manually entered is restricted and limited to what is applicable for that column. Such restrictions include oyster height measurement ranges, site location names, and acceptable dates for surveys. Two people, usually the technicians who surveyed the oysters, separately enter oyster observations under with the data validation restriction, in two separate Microsoft Excel tabs (Box 4 3.A). An additional Microsoft Excel tab will conclude with the two separately entered versions are identical. If the dual entry versions are not the same a “check” notification will appear on the Excel cells that do not match. The flagged cells will then have to be reconciled by a third team member, who will investigate the discrepancy using the original data sheets.

**Adding oyster observations to a central storage and version control**

Similarly, reconciled oyster observations are ultimately stored in our master data repository on GitHub and team members are required to follow the same workflow as previous mentioned (Box 4 3.C). The workflow

The only difference is that oyster counts, and measurements are not stored in MySQL since our MySQL database was created specifically for water quality observations. Oyster measurements are also stored on a University of Florida protected server as back- up.

#### Paper data sheets to electronic records guidelines

The process of transferring data from paper data sheets to electronic form, which makes it compatible with a computer for data analyses, is the most common source of potential errors. Minimizing the risk of errors is the main characteristic of the workflow design. For data entered by hand a data entry system that reduces the likelihood of introducing errors via data entry is used. This is done by using a standardized template, so data sheets and digital spreadsheets are input similar ways. This follows USGS Data Management guidelines, which suggests that the most effective way to ensure data quality, is to prevent the creation of defective data. The LCR restoration project uses a Data Template structure based on USGS Data Management Standards (<https://www.usgs.gov/products/data-and-tools/data-management/quality-design-recommended-practices?qt-science_support_page_related_con=0#qt-science_support_page_related_con>).

For the LCR restoration project, a designed Excel workbook is used and is intended as a Data Template for easy and efficient data entry (Figure 5). This workbook is modified for data entry using “Data Validation” features in Excel that restrict the types of data that can be typed into each predefined column (Figure 2, Box B1). These restrictions include the use of “drop down” style menus that require the user entering data to choose a value for entry based on a pre-populated list of values. These pre-populated lists of values, such as site name abbreviations, are based on the terms defined by the data abbreviations guide for the project. Other types of restrictions include specific formatting for date or time values, as well as “limits” on observational data entered in each cell. By restricting the choice of the user when selecting locations, dates, units, and measurement ranges this limits the potential for data entry errors such as capitalization or use of zeros instead of the letter “O”. To simplify entry, each data column matches an entry on the physical data sheet used in the field.

As an example of the capability of “Data Validation” features, oyster length measurements are restricted from being entered at a size greater than 125-mm and give an error message to the user if done so. While oysters greater than this size are observable in nature, to enter a value above this level requires manual override from someone with supervisory control. This data entry system also requires a “double entry” system where each line of data are entered into the workbook twice, typically by separate users, and then these data are compared electronically. If the double entered data do not match exactly, the original data sheets are examined to determine why discrepancies exist. Using different people for each round of data entry is preferred because different people may interpret the handwriting on the field data sheets differently. A third tab is be used in Excel to compare the two user data entry tabs for discrepancies. Any identified errors are then reconciled against the field data sheets and by a project supervisor.

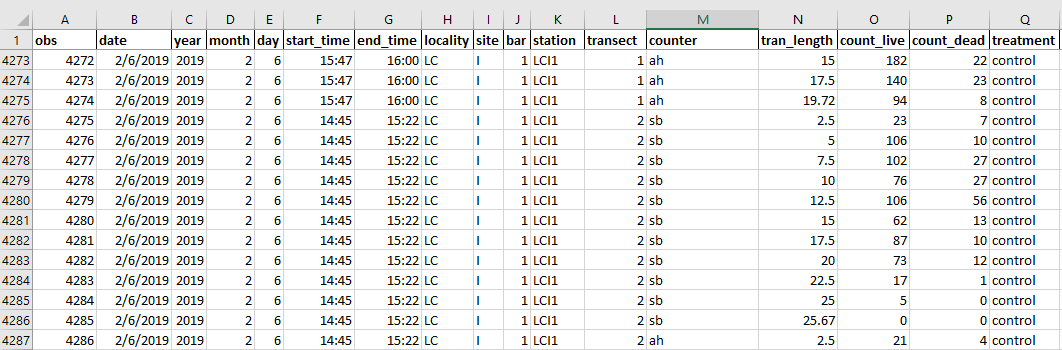


Figure 5- Data entry workbook in Excel to enter in data collected from the field. Each column is restricted on what information can be entered into it.

#### Sensor collected data

Sensor collected data differs from human collected data, in that sensor data are measurements recorded by an instrument automatically. These types of data are a common component of many large-scale observational platforms that may record environmental or biological data continuously, and then make these observations available for use at set time intervals or through “live” feeds. Examples of these types of data include river discharge information provided by USGS (https://github.com/USGS-R/waterData) or wind observations from a National Oceanic and Atmospheric Administration (NOAAA) weather buoy (https://github.com/ropensci/rnoaa), which can be accessed online by APIs or software such as R.

The LCR restoration project has a small array of sensors (N=10) that track the temperature (°C) and conductivity (μS/m) of water near the oyster reef restoration site. To retrieve the data from these sensors, the sensor must be physically removed from the water and the associated data files are downloaded from the receiver (Figure 2, Box A2). Sensors are serviced bi-weekly to ensure functionality.

An individual sensor data file with 14-days’ worth of observations contains about 900 lines of data and a total of about 400 observations. While the observations are collected automatically, there are still opportunities to introduce errors when these data are collected. This can include incorrectly naming of files once downloaded to a laptop in the field, copying over files on the laptop erroneously, or failing to “start” the sensor once redeployed. Reducing these error opportunities ensures a continuous sensor stream of interrupted measurements.

#### Transfer of electronic records from sensor to database

When individually collected sensor data files are transported back to the lab, these files must be checked for errors and the data amended to an existing database to provide a continuous record of the water quality observations of interest (Figure 2, Box B2). A three-step process has been developed where:

Step 1. Working with University of Florida Library team, Python code has been developed that distinguishes files from each of the two types of sensors that make up the water quality sensor array (Star-Oddi or Diver), based on proper file naming conventions (i.e 20200219\_wq6\_star.dat).

Step 2. Python code checks data for errors including duplicate observations or checks that the sensor is identified properly in our database. As an example, all active and functioning sensors, which are deployed in the field, are stored in a data table in our MySQL database, where the start day, time, and location are recorded. If the data file list of sensors does not match the list of active sensors known in the database, then an error message is reported.

Step 3. MySQL imports all checked and correct observations in their appropriate tables.

Once imported, a second set of QA/QC protocols is performed within the MySQL database, which examines observations for non-sense values based on expected temperature and conductivity values for the array location. While I have not directly developed the MySQL database, I have worked closely with University of Florida Library staff to define database relationships, error checking routines, and workflow within the MySQL database.

Developing a database to store the water quality datasets, prior to data collection, has given the project a security in knowing that every possible instance of a mistake or error has be thoroughly thought through. Some mistakes that have been taken into consideration in the database are the possible locations of the sensors. Sensors can move from one location to another if need be, and the database has a “check-in” and “check-out” procedure to ensure that only the data from active sensors can be imported. Thinking through the possibilities of how data collection can possibly be entered or imported incorrectly, is a major advantage in creating a workflow design.

### Data analysis, figures and tables

Once data are standardized and available computational use, basic visualization of the data via graphs and figures the next step for data checking and the beginning of the analyses (Figure 2, Boxes C1 and C2). A group of data visualizations has been developed to produce products to be used both to check data from field collections and water quality sensor data. These figures are integrated with the living data such that as data are entered into the database, and after they pass initial QA/QC, the figures are automatically updated to allow visual assessments of the recorded data.

A set of summary tables has been developed as part of the data workflow to provide basic information on water quality variables at different time intervals. These summary tables and figures follow data reproducibility guidelines from USGS where the tables are created from the living data using standard code that reproduces the same table and adding newly updated data when needed. Many functions from the R package `tidyverse` (<https://www.tidyverse.org/>) are maintained and updated regularly to ensure compatibility with other existing functions, are be used to create these tables and figures. By developing code for tables, figures, and any other reproducible analyses as the data are updated, total time for data feedback loop is reduced.

### Version control

Version control is defined as a software that allows for the saving and management of changes in content, documents, and other developmental information over time. The key purpose of using version control software is to document and confirm that changes in content are intended and planned. Version control can be incorporated into a data workflow using software such as Github (https://github.com), (Figure 2, Boxes D1 and D2). The USGS Data Management Guidelines encourage the use of version control software and repositories for data and code used for projects, which allows the project data analysis to be accessible and reproducible (https://www.usgs.gov/products/data-and-tools/data-management/repositories).

Version control is critical to ensuring that files are not duplicated, lost, or time is not wasted by not working with the proper files. The Data Carpentries (https://datacarpentry.org/) provide detailed reasons for using version control (<http://swcarpentry.github.io/git-novice/>) that can be generalized as (1) a version control system saves all versions of a file, (2) version control records who made what changes to specific files and makes the user write detailed notes about what they changed (3) allows these changes to be undone if needed, (4) version control software notifies each user when there is a conflict between different people’s work such as code.

The LCR restoration project uses use a GitHub structure for version control (https://github.com/LCRoysterproject). Excel data validation workbook are used for data entry and initial QA/QC in Git to allow each user to see when new data are available through GitHub versioning. GitHub is used to track changes in routine R files used for data summaries and standard reports to funding agencies. GitHub versioning gives the LCR restoration project security in that any version of a file can be viewed at any time so there is never a complete loss of data or code.

##### Naming conventions for sampling files

The LCR restoration project uses a naming structure which requires that all sampling files start with the date of creation, in the format YYYYMMDD. Each file has additional information which will usually have a prior set of approved abbreviations, after the date. One advantage of this naming structure is that all files are ordered chronologically when sorted by name, so there is very little confusion on when the files were created (Table 1). Following the guidelines from USGS Data Standards suggest that file names should be in all uppercase or all lowercase letters, instead of a combination of both. For the LCR restoration project, naming convention standards require that all files are in lowercase, and the context of the file names are separated with an underscore. If files are not named correctly, they will be renamed to follow our guidelines. Files that are not named correctly, also have the risk of being overlooked, or re-organized in an incorrect folder. Correct naming conventions are critical to create the correct interface between the field collected water quality sensor data and the Python code that reads and stores these data.

|  |  |
| --- | --- |
| description\_of\_file | naming\_convention |
| Oyster quadrat sampling on Feb.16, 2020 in Lone Cabbage Reef in a .csv file type | 20200216\_lcr\_oys\_quadrat.csv |
| Sensor 1 (manufacturer Diver) serviced on March 2, 2020 in a .MON file type | 20200302\_wq1\_diver.MON |

Table 1- Example of file naming structure, starting with date, species type, location, and sampling or general attribute of the file

As per USGS Data Standards, naming conventions are necessary to make data easier to use, to integrate and to share. This is especially true because data that are represented are in a format that has already been established and planned (<https://www.usgs.gov/products/data-and-tools/data-management/data-standards#examples>). For the LCR restoration project a column definition table was created beforehand, on how each data type will be named, formatted, and its description. These naming convention standards add data integrity and accuracy to keep track on what data are being added and when how they might be used for analyses (Table 2).

|  |  |  |  |
| --- | --- | --- | --- |
| definition\_of\_file | column\_name | format | unit |
| sampling or survey date | date | YYYYMMDD |  |
| starting x utm transect location, latitude | start\_xutm | numerical | UTM |
| ending x utm transect location, latitude | end\_xutm | numerical | UTM |
| starting time of sampling | start\_time | HH:MM | UTC |

Table 2- Example of column naming conventions and descriptions.

#### Data back up

Using GitHub for version control and data storage is both convenient, reliable, and easily shareable. However, there should always be a backup workflow in place for data, code, and documentation. The LCR restoration project continues through its data management workflow by backing up repositories to an internal University of Florida server. This server is also private to only authorized users at the University of Florida. Backing up routinely creates an additional level of protection for the LCR restoration project data and files.

#### Discussion

The main goal of implementing a consistent workflow is to make data available for rapid analyses to adaptively assess the LCR restoration project and ongoing water quality and oyster monitoring efforts. This workflow meets the adaptive management requirements for this project by providing the data in a structure that allows rapid assessment and evaluation to inform decision making related to the ongoing monitoring efforts (Figure 2, Box E). These data are processed and managed to support reproducible analyses. Much of the project data and analysis can be reproduced at a later time by any member of the project team or any collaborator consulting with the LCR restoration project. These methods ensure that the best practices are established and followed for data input, management, and basic summaries and visualization. This workflow is useful for (1) increasing efficiency in the LCR restoration project. The LCR restoration project involves a large restoration project as well as integration of historical data from two previous sampling epochs. Because a single data management workflow was not used across these epochs, significant effort has been required to standardize existing data. Since a data workflow was established at the beginning of the LCR restoration epoch, the data are managed in a common structure over the life of the project. Having precise knowledge of biological data interpretations, will ensure both time and money are used efficiently. (2) This data workflow informs a variety of short-term decisions that are made to adaptively improve the ongoing LCR monitoring efforts. As an example, sampling frequency, sampling locations, and sampling times of both the oyster populations and water quality are routinely informed by rapidly processing existing data. This has prevented data gaps from occurring from events such as biofouling of water quality sensors. (3) Long-term decisions as part of the adaptive management process of this project are also informed by this data workflow. Overall well-designed data workflow programs are critical to meeting basic requirements of an adaptive management plan. When combined this approach can be highly effective in maximizing the effectiveness of conservation actions such as the LCR restoration in a cost-effective manner.