Background and context

Most common implemented ecological programs are designed to monitor animal populations, their habitats, and how people manipulate these populations and associated environments have experienced significant changes in data collection, management, and storage technology in recent years. Changes include new sensor technology, data collection methods, and data observing platforms. (Unsure about this NEON sentence, will probably remove). Both SECOORA and NEON have triggered rapid changes in the spatial and temporal scale of data collected. As an example, advancements in sensor technology have allowed for substantial changes in water quality monitoring, transitioning from discrete single location sample collections to real-time continuous observations at multiple locations. These monitoring programs are conceived, planned, and used by biologists, but these researchers often have limited training in basic data management, curation, and workflow of data generated from these platforms. Lowndes et al. (2017) highlights the results of a recent survey of program needs of NSF funded principal investigators in biological sciences. Of the 704 scientists who participated in the survey, these respondents identified data skills as the largest unmet need (Barone, Williams, and Micklos (2017); Lowndes et al. (2017)).

Continuous data, management and analyses

The US Gulf of Mexico 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 efforts, these projects will likely have data collection and evaluation efforts that occur frequently through the project lifetime.

Adaptive management

In some cases, data collected from these programs adaptively inform ongoing restoration actions to help improve their outcome. For these cases, data collection systems and related workflow must be flexible over time as data collection efforts vary, projects evolve, management actions are implemented, and/ or ecosystems change. At the same time, while ecosystems are changing, usually the people conducting the monitoring and technology used in the field may also change, potentially introducing unanticipated variability in data. These changes must also be considered along with ecosystem response to restoration and other management actions.

“Living data” are defined as data which are continuously collected and are critical to this type of adaptive learning to inform restoration and management actions. These informed adaptations 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. In a restoration or management context as these data are collected they must be processed, and analyses of these data 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). The adaptive management framework is widely discussed and considered in a variety of management and restoration projects (https://www2.usgs.gov/sdc/doc/DOI-%20Adaptive%20ManagementTechGuide.pdf ) but the success of these programs is highly variable for a variety of reasons (Walters 2007).

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 that are then monitored and evaluated to improve management actions. These programs face the challenge of ensuring that these continuous efforts meet quality assurance/quality control protocols to identify and correct inconsistencies and errors in field or sensor observations on a continuous basis. 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.

An example program that requires an adaptive management approach to restoration are projects funded by the National Fish and Wildlife Federation as part of the Gulf Environmental Benefit Fund (NFWF-GEBF). These projects explicitly require an adaptive management plan to guide the restoration process. Extensive data management plans are mandatory to capture data collected and analyzed as part of these programs with the overall purpose of creating opportunities to improve future restoration actions by maximizing learning from previous and ongoing restoration efforts.

The Lone Cabbage Reef (LCR) restoration project is a large restoration effort in the eastern Gulf of Mexico funded by NFWF-GEBF. The project’s primary goal is to restore historical oyster reefs so that they may be plastic to sea level rise, and fluctuations in river discharge. This project generates data from multiple sources including continuous autonomous water quality data from sensors and observations of oyster populations from 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. Because this is a long-term restoration project with numerous uncertainties in how the ecosystem will respond to restoration actions, developing a data management and workflow system that automates as many aspects of the workflow including QA/QC, measurement errors, and inconsistency in naming conventions is essential to allow for rapid analyses of data to inform decision making related to sensor deployment or modifying the reef restoration process through additional construction efforts.

Objective

In this chapter, I will document how the basic elements of the LCR restoration project water quality and biological data associated with oyster populations are managed. The objective is to develop and implement a data management workflow, which starts at the data collection point (i.e physical data sheet if required) and ends at the visualization/ interpretation of collected data from different data streams. I will document how these data are recorded, data quality assurance/quality control procedures, data checking (anomalous values), data visualization, and data releases for analyses using multiple software tools. This chapter provides an example of a living data project can function to inform an ongoing, long-term restoration project and serve as an example for other projects with data collection efforts.

Implementing a modern data workflow

Creating and automating a data management workflow for living data is an emerging skill for natural resource professionals. More than ever, data management is recognized as a core skill for biologists and ecologists (Hampton et al. 2017). Even though the design of my workflow will be specific to the LCR restoration project, the steps outlined can be broadly used for many conservation efforts. The tools used to implement the data management workflow, are also readily available online and most tools offer tutorials and workshops for more in-depth training. The approach for this workflow requires basic knowledge of computer coding and version control structure (to track changes in data and computer code). I will use freely available open source tools including program R (https://www.rstudio.com/) for data QA/QC, analysis, and visualizations, and GitHub (https://github.com/) for version control.

Field Collections

One of the goals of a successful data management plan is to minimize errors in data collected. Often, the first step in the data collection process is transcribing an observation in the field to paper or electronic datasheets for analyses back in the lab. This simple effort of recording the data in the field is the first opportunity to introduce errors in the data collection process. These errors can come from a variety of sources such as the wrong date or site name on a sheet or the person recording the data may be unfamiliar with terminology or protocols. To minimize these types of mistakes it is best to follow proven practices for data management such as those adopted by USGS recommend development of a standard set of data guidelines before field collections begin (Figure 3).

The guidelines define how basic data are recorded such as date and time formats, site naming conventions, and units of measure for specific observations. This type of predetermined information is a key first step in reducing the risk of this type of data error in the field. As an example, simple differences in how dates are recorded by different people such as YYMMDD or MMDDYY formats can create confusion as to when a sample was physically collected. Errors in site names can place the data observations in the wrong location spatially. To minimize this risk, when possible data sheets can be pre-populated with as much information as possible before going into the field.

Human collected data

For the Lone Cabbage Reef project (Figure 2, box A1), observational data collected in the field primarily consists of counts and size measurements of oysters from line transects among randomly selected oyster reefs delineated into strata based on specific research questions, which are then recorded on waterproof paper data sheets. To reduce chance of field errors and save time while in the field, I will work to develop and improve data workflow by providing guidance on pre-populating datasheets when possible with basic information including, date and location following data naming standards and field protocols (Figure 4).

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 observation 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 or wind observations from a NOAA weather buoy, which online access or software such as R can obtain.

The LCR project has a small array of sensors that track temperature and conductivity 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 450 observations. While the observations are collected automatically, there are still opportunities to introduce errors when these data are collected. This can include incorrect 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 will ensure a continuous sensor stream of interrupted measurements. The LCR project has developed existing protocols to minimize these errors. I will review these protocols and revise as necessary as part of my data workflow development.

QA/QC during data entry

Paper data sheets to electronic records

The process of transferring data from paper datasheets to electronic form that, which will make it compatible to a computer for data analyses, is a common source of potential errors. I will work to minimize this risk of errors as part of my workflow design. For data entered by hand, I will first use a system that reduces the likelihood of an errors being introduced into the data entry to start with. This will be done by using a data entry template that mirrors how the data are collected in the field on paper data sheets, with how the data are entered in the computer. 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. I will use 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 project, I will design an Excel workbook designed as a Data Template for easy and efficient data entry (Figure 5). This workbook will be 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 B2). 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 limiting 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 will be restricted from being entered at a size greater than 125-mm. While oysters greater than this size are observable, to enter a value above this level requires manual override from someone with supervisory control. This data entry system will also require a “double entry” system where each line of data will be entered into the workbook twice, typically by separate users, and then these data will be compared electronically. If the entered data do not match exactly, the original data sheets will be 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 will then 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.

Transfer 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, box B2). I will develop a three-step process where:

Step 1. Working with UF Library team, I will develop Python code that will distinguish 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 convention.

Step 2. Python code will then check for errors in these data including duplicate observations or data from a sensor that does not have an “identity” 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 will be reported.

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

Once imported, a second set of QA/QC protocols will be performed within the MySQL database, which will be examining observations for non-sense values based on expected temperature and conductivity values for the array location. While I will not develop the MySQL database as part of my thesis, I will work closely with UF Library staff to define database relationships, error checking routines, and workflow within the MySQL database. I will also develop basic Python skills to allow me to conduct routine maintenance on the database such as error checking and adding additional water quality stations as needed.

Data analysis, figures and tables

Once data are standardized and available for use in the computer, basic visualization of the data via graphs and figures is a key next step for data checking and the beginning of the analyses. I will develop a group of data visualization products to be used both to check data from field collections and water quality sensors. These figures will be integrated with the living data such that as data are entered into the database and after they pass initial QA/QC the figures will be automatically updated to allow visual assessments of the recorded data. I will focus my efforts on creating these visualization products for the water quality data collected by the LCR project.

I will also develop a set of summary tables as part of the data workflow to provide basic information on water quality variables at different time intervals. These summary tables and figures will follow data reproducibility guidelines from USGS where the tables will be created from the living data using standard code that reproduces the same table, adding newly updated data, when needed. By developing code for both tables, figures, and any analyses that are reproducible as the data are updated, this will reduce the time for the data feedback loop.

It is essential to use analytical methods to determine patterns, conceive generalizations, notice biological trends, and estimate data uncertainty. For funding agencies, having a clear data analysis workflow, allows the stakeholders to know that the data were structured and analyzed the same way, every time. Consistent data analysis workflow also enforces a standard of reliability for agencies conducting research.

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. The focus of version control is to confirm that changes in content are intended and planned. Version control is “a tool for managing changes to a set of files” (Huang and Gonzalez 2016,http://swcarpentry.github.io/git-novice/). Version control can be incorporated into a data workflow using software such as Github, (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 will allow the project data to be accessible and reproducible (https://www.usgs.gov/products/data-and-tools/data-management/repositories).

Version control can be critical to ensuring that data are not duplicated, lost, or time wasted by not working with the proper files. The Data Carpentries provide detailed reasons for using version control (Huang and Gonzalez 2016) that can be generalized as (1) a version control system saves all versions of a file. This allows you to go back in time to old versions if needed and see what person made changes or used a particular file, (2) version control records who made what changes to specific files and (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, (5) allows users to see when and how different files have been added or modified.

The LCR project will use a GitHub structure for version control. I will manage the Excel workbooks used for data entry and initial QA/QC in Git to allow each user to see when new data are available. I will also use GitHub to track changes in routine R files used for data summaries that are pushed to the web and included in standard reports to funding agencies.

Proper Storage

I propose that the data workflow for both data and code scripts be separated into two modes. The first mode is “development” mode, meaning that the data are currently undergoing a QA/QC process. The second mode is “production” mode, where the processed data are ready to be analyzed. Github repositories will only have publicly available production data and scripts (Figure 2, boxes D1 and D2). Raw sensor data files will not be found in these repositories.

For the data and scripts that are in development mode, the proper storage for these documents will be in our projects internal server, commonly referred to as the T:Drive. This server is only available to members of the LCR project is not publicly available. Raw sensor data files would be in this server.

Naming conventions for files

I will develop a naming structure which will require files start with the date of creation, in the format YYYYMMDD. Each file will have additional information, that will usually have a prior set of approved abbreviations, after the date. These naming structures automatically set all files chronologically, so there is very little confusion on when the files were created (Table 1). It is suggested that all files should be in all uppercase or all lowercase letters, instead of a combination of both. For the LCR project, I will propose as part of the naming convention standards that all files are 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.

Discussion

|  |  |
| --- | --- |
| Organization/ Person | Deliverables |
| ARCS (Advancing Science in America) | 1. Funding research that will:  * Encourage student knowledge sharing * Develop technologies   Inspire youth to pursue the challenge of STEM careers |
| LCR Project | 1. Executing Data Management and Access Plan:  * Observational data including autonomous stations and oyster reef sampling * Management of specific data products * Quality Assurance/Quality Control * Access to Data and Data Sharing Practices and Policies |
| Proposed research | 1. Investigate data management needs for LCR project 2. Design and create data management workflows as it pertains to the LCR project, for each data sampling type 3. Continually updating workflow for maximum efficiency 4. Ensure data management plan is compliant with USGS Data Standards |

Table 3- Deliverables comparison of ARCS, LCR Project and my proposed research

Using the Lone Cabbage Reef Restoration as an example, I will develop a data workflow that is adaptable to multiple types of data and meets best practices for data validation and reproducibility (Table 3). I will use the LCR restoration project as a case history to develop this data workflow. This data workflow will integrate living data from observations recorded on paper data sheets and electronically recorded from sensors that monitor water quality.

The main goal of this workflow is to make these data available for rapid analyses to adaptively assess the LCR restoration project to inform water quality and oyster monitoring efforts. This will help to meet 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). To do this, these data must be properly processed and managed to support reproducible analyses. My project will ensure that best practices are established and followed for data input, management, and basic summaries and visualization. This information will be useful for (1) increasing efficiency in the LCR project. The LCR project involves a large restoration project as well as integration of historical data from two other sampling epochs. Because a single data management workflow was not used across these epochs, significant effort has been required to standardize existing data. By establishing a data workflow at the beginning of the LCR restoration epoch, the data will be managed in a common structure over the life of the project. These productive data are used to make decisions in future conservation efforts. Having precise knowledge of biological data interpretations, will ensure both time and money are being used efficiently. (2) This data workflow will inform a variety of short-term decisions that must be 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 can be informed by rapidly processing existing data. This can prevent data gaps from occurring from events such as fouling of water quality sensors. By knowing the rate in which the water quality sensors are biofouling, we can increase our sensor servicing interval to clean the sensors before their measurement precision is compromised. Because of quick data interpretation turn around time, we were able to determine that fixed sensors in warmer summer months experience more fouling, and must be serviced more frequent than 14 days, usually about 9-10 days. Being able to keep sensors free of fouling, we are able to ensure proper and continuous data measurements of the sensor. We can also determine, whether or not a fixed sensor is in an appropriate location to measure changes in the parameters of interest such as changes in salinity following reef construction. (3) Long-term decisions as part of the adaptive management process of this project can also be informed by this data workflow. For example, this project is one of the first large oyster restoration projects funded in Florida by GEBF. Oyster reef restoration is a common topic for other possible projects and the LCR project can provide information on how funds could be allocated for sampling trips, surveys and equipment. 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.