Visualizing Spatial and Temporal Patterns of Coral Reef Stressors Surrounding Moorea, French Polynesia

A Capstone Project submitted in partial satisfaction of the requirements for the degree
of Master of Environmental Data Science
for the
Bren School of Environmental Science & Management

by

ALLIE COLE
FELICIA CRUZ
CHARLES HENDRICKSON
JAKE EISAGUIRRE

Committee in charge:
DERON BURKEPILE
ALLSION HORST

JUNE 2022

VISUALIZING SPATIAL AND TEMPORAL PATTERNS OF CORAL REEF STRESSORS SURROUNDING MOOREA, FRENCH POLYNESIA

As developers of this Capstone Project documentation, we archive this documentation on the Bren School's website such that the results of our research are available for all to read. Our signatures on the document signify our joint responsibility to fulfill the archiving standards set by the Bren School of Environmental Science & Management.

____

MEMBER NAME

____

MEMBER NAME

____

MEMBER NAME

____

MEMBER NAME

The Bren School of Environmental Science & Management produces professionals with unrivaled training in environmental science and management who will devote their unique skills to the diagnosis, assessment, mitigation, prevention, and remedy of the environmental problems of today and the future. A guiding principle of the School is that the analysis of environmental problems requires quantitative training in more than one discipline and an awareness of the physical, biological, social, political, and economic consequences that arise from scientific or technological decisions.

The Capstone Project is required of all students in the Master of Environmental Data Science (MEDS) Program. The project is a six-month-long activity in which small groups of students contribute to data science practices, products or analyses that address a challenge or need related to a specific environmental issue.

This Capstone Project Technical Documentation is authored by MEDS students and has been reviewed and approved by:

____

ADVISOR

____

ADVISOR

____

DATE

Table of Contents

Abstract

Coral reefs provide countless ecosystem services and economic benefits to communities around the world and support approximately 30% of all marine biodiversity. However, coral reefs are among the most vulnerable ecosystems to anthropogenic stressors, including rising ocean temperatures and land-based nutrient loading. Since 2005, the Moorea Coral Reef (MCR) Long Term Ecological Research project has been collecting data in Moorea, French Polynesia. Current work is being done to fill knowledge gaps in understanding how nutrients, coral bleaching, and benthic community dynamics affect coral reef health in the lagoons surrounding Moorea. MCR researchers collect data on a wide variety of biotic and abiotic variables at different spatial and temporal resolutions and there is no easy way for researchers to quickly visualize changes across space and time. To address this issue, an interactive web application has been developed for MCR LTER researchers to visually explore ecological changes, anthropogenic stressors, and community dynamics affecting the coral reefs surrounding Moorea. This web app allows users to hone in on certain locations or variables of interest in order to identify vulnerable coral reefs and explore coral reef ecosystem dynamics. Additionally, this app will act as a research and outreach tool to bring awareness to the status of coral reefs and allow for better management and protection of Moorea’s coral reefs.

Executive Summary

Coral reefs are one of the most diverse ecosystems on the planet and provide valuable ecosystem services for many coastal communities, however they are significantly threatened through a suite of environmental and anthropogenic stressors (Moberg & Folke, 1999, Woodhead et al. 2019). Understanding spatial and temporal patterns of both global and local drivers is critical for predicting future conditions and developing mitigation strategies.

The National Science Foundation (NSF) funded Moorea Coral Reef Long Term Ecological Research (MCR LTER) project has been collecting temporal and spatial data on the shallow reefs surrounding Moorea, French Polynesia since 2005. The study sites have undergone multiple ecosystem shocks since data collection began on the fore reef, including heat waves, coral-eating sea stars, and a major cyclone (Holbrook et al. 2018). The six long term research sites have responded differently to these shocks (Holbrook et al. 2018). The data utilized in this project focus on nutrient loading, coral bleaching, and benthic community dynamics in the lagoons surrounding Moorea (Adams et al. 2021, Donovan et al. 2020). There are knowledge gaps in understanding how these factors affect lagoon dynamics surrounding coral reef atolls and a platform where these variables can be compared was needed (Adams et al. 2021, Donovan et al. 2020). To address these knowledge gaps, an easy-to-use Shiny application has been created to visualize and analyze the geo-spatial ecological data collected in Moorea.

This app was created for use by MCR LTER researchers to easily visualize the extensive ecological data collected from the reefs surrounding Moorea and currently contains spatial visualizations for 2016 and temporal visualizations for 2006-2021. To build this app, ecological and environmental spatial layers were created in RStudio using the **leaflet** package, an array of analysis packages, and the **shiny** package to deploy the app. Results include eight spatial layers (Six layers: Percent Nitrogen and delta-N-15 by weight for *Turbinaria ornata* in January, May, and July of 2016; One layer: Percent coral bleaching; One layer: Predicted Sewage; One layer: bathymetry) that can easily be toggled on the app. In addition to the interactive spatial visualizations, the Shiny app includes time series visualizations for percent coral cover, percent algae cover, herbivore fish biomass density, and Crown-of-Thorns Sea Star density for the six LTER sites surrounding Moorea. The completed Shiny application as of June 2022 is currently hosted on the Bren School server and will continue being deployed here for six months, or until the UCSB GRIT team can host it on their server indefinitely. The complete Shiny application will be linked on the MCR LTER website. Accompanying this app is a User Guide outlining data and code management for MCR researchers to maintain the app and update the existing visualizations with future data. This includes details on GitHub usage, code access, and information for additional data/layer inclusion. Lastly, all code, data, and files pertinent to the creation and maintenance of the Shiny app can be found in a GitHub repository on the MCR LTER GitHub Organization.

Problem Statement

Coral reefs are vital ecosystems that protect coastlines from storms, provide natural resources to local communities, and support 30% of all marine biodiversity (Moberg & Folke, 1999, Reaka-Kudla, 2005). Increased nutrient input from human activities have caused coral reefs to decline at an alarming rate through increased bleaching, coral diseases, and macroalgae overgrowth (Vega et al. 2014). There is no concise, easy-to-use platform where these problems can be visualized. Spatially visualizing nutrient composition and coral bleaching can help determine high-risk locations on the reefs of Moorea, French Polynesia. The overall goal of this Capstone Project was to create an interactive web application to explicitly visualize the nutrient composition and coral bleaching status surrounding Moorea, French Polynesia in addition to benthic community dynamics through a Shiny app.

The goal of this Shiny app is to assist in determining how coral reef communities in Moorea have changed due to anthropogenic and environmental stressors, with the potential to add additional years of data as they arise.

Specific Objectives

Project Objectives:

1. User-friendly, aesthetically-pleasing Shiny application consolidating spatial and temporal visualizations and metadata
2. Written User Guide to accompany our Shiny app
3. Organized GitHub repository housed on the MCR LTER GitHub Organization

More specific information can be found in the Project Deliverables section.

Summary of Solution Design

Approach and Methods

Data for this project was accessed both through the MCR LTER data portal located on the MCR LTER website and from lead MCR LTER researchers. Data are stored in a main data folder with a static file path in the code for reading the data; all subsequent data must be added to this data folder. After reading the raw spatial data into RStudio, all necessary wrangling and cleaning was done to prepare the data for kriging and mapping through **leaflet**. Priority deliverables included producing spatial visualizations of percent nitrogen levels, percent coral bleaching, predicted sewage levels, and bathymetry. To create these visualizations, ordinary kriging was used on the geo-spatial data points of each layer for the surrounding reef boundary of Moorea. The interpolated spatial data was then rasterized for **leaflet** integration. Once the interactive map was created in RStudio through the **leaflet** package, RShiny was used to develop the web-based interactive application to satisfy the product deliverable for our client. All stretch goals were completed, which involved creating additional visualizations of long-term temporal data from 2006-2021 which included annual percent coral cover, percent macroalgae cover, herbivore fish biomass density, and Crown-of-Thorns Sea Star density for the six LTER sites. In addition to the Shiny app, an accompanying User Guide outlining all code and workflow has been shared with the client and can also be found on the MCR LTER GitHub organization repository with the rest of the files pertaining to this project. The Shiny app is temporarily hosted on the Bren School server where it will then be moved to the UCSB GRIT server within the next 6 months.

Data Management

The temporal data used for this project has been collected and managed by the MCR LTER and was downloaded directly from the MCR LTER website. The spatial data was collected and managed by the MCR LTER and was made available through lead MCR LTER researchers. The data was wrangled into Tidy format, where each observation is a row and each column is a variable. Each interpolated layer for the spatial data used for the interactive leaflet map was saved as a raster stack in a .nc file for future use. A User Guide has been developed to provide information on code and data management as well as app maintenance. This includes details on how to update data and code for future integration of new data sets.

Software and Tools

All programming for this Capstone Project was done using the R language, which is a free, open source software that executes the actual computation instructions from our code commands. Additionally, the team used RStudio, which is an Integrated Development Environment (IDE) that provides a helpful interface to R. RStudio is also a free, open source software application. Overall, R/Rstudio was leveraged to access the data, wrangle and clean the data, map the data using the **leaflet** package (Joe Cheng et al., 2021), and use the **shiny** package (Winston Chang et al., 2021) to develop an interactive web application for the client. Other packages that were used include, **testthat** (Hadley Wickham, 2011), **raster** (Robert J. Hijmans, 2021), **sp** (Pebesma, E.J., 2018), **gstat** (Pebesma, E.J., 2004), **here** (Kirill Müller, 2020), **tidyverse** (Wickham et al., 2019), **janitor** (Sam Firke et al., 2021), **viridis** (Simon Garnier, 2021), **automap** (Hiemstra et al., 2008), and **sf** (Pebesma, E., 2018). Furthermore, the team used

GitHub, which is a web-based Git repository hosting service that offers version control and source code management functionality of Git, as well as several collaboration tools like GitHub Issues. A project repository was created in GitHub to track the changes made to the code, create tasks, and store the work for the client and other collaborators to access.

Products and Deliverables

Project Deliverables for the Client

The project deliverables for the client included three main components: a Shiny app with spatial and temporal visualizations, a User Guide for code and data management, and an organized GitHub repository.

The Shiny app has multiple tabs to showcase spatial visualizations, temporal patterns, and data information associated with each. Through the spatial tab researchers can toggle between different time periods, and look at the corresponding levels of nitrogen, coral bleaching, and bathymetry. The temporal tab allows researchers to track trends of fish biomass, Crown-of-Thorns density, percent algae cover, and percent coral cover. With two options for organizing visualizations by variable and visualizations by site, the user has flexibility depending on what comparisons and trends they are most interested in. Both the spatial and temporal tabs have dropdown options for users to access information on the data used in the visualizations, such as where and how data was collected. There are also links to the MCR website where the raw data can be downloaded for free.

The second project deliverable for the client is the User Guide for code and data management. The code management section of the User Guide has a brief description of all .rmd files, and highlights details for incorporating future years' data. The majority of this comes in the form of commented code on the .rmd files, with explanations and commenting of code in between the chunks. The data management section of the guide outlines how researchers can upload new data without breaking the code. This includes where to put the data files, and what they should look like before uploading.

The GitHub repository for the Shiny app, the final project deliverable, is titled "MEDS" and lives on the MCR LTER GitHub organization. This is a public repository and includes all data and code files utilized for the creation of the app and also contains the User Guide. The file structure of this repository is outlined in both the User Guide and the User Documentation section of this document.

Summary of Testing

- Overview of reasons why our application might stop working

There are many possible reasons for the Shiny app to stop working, such as upgrades in R packages, modifications to the app or names of objects within the app, or external changes in the data format. Manual testing requires a user to interact with the app in order to detect any problems. However, automated testing is a much more efficient and effective way to alert the user to these problems and can be used to ensure that the Shiny app will continue to perform optimally into the future. We recommend utilizing two automated testing packages for Shiny apps in this Shiny app.

    - **Shinyloadtest** package makes it possible to load test deployed Shiny apps and helps developers and administrators estimate how many users their application can support. If an application requires tuning, load testing and load test result analysis can be used to identify performance bottlenecks and to guide changes to infrastructure, configuration, or code.

- The process for using the **shinyloadtest** package consists of three steps:
    - Record a typical user session for the app.
    - Replay the session in parallel, simulating many simultaneous users accessing the app.
    - Analyze the results of the load test and determine if the app performed well enough.
- The approach to using this package consists of the following steps:

1) Input the URL of the deployed app into the record_session() function.

```
shinyloadtest::record_session('Shiny_app_URL')
```

Ex: Using the URL from the Shiny app server, which the app is currently deployed on (URL will change when updated to the GRIT server)

```
shinyloadtest::record_session('https://shinyapps.bren.ucsb.edu/ShinyAppMooreaViz/')
```

2) Run the function to open a browser displaying the app.

3) Interact with the application as a typical user might then close the browser.

4) After closing the app, a file (`recording.log` by default) will be created that contains a recording of the session, which will serve as the basis for the load test.

5) Run the load test using the `shinycannon` command-line tool from the RStudio terminal tab. In this case we are simulating 5 users on the app.

```
shinycannon recording.log
https://shinyapps.bren.ucsb.edu/ShinyAppMooreaViz/
--workers 5 --loaded-duration-minutes 2
--output-dir run1
```

6) Run the load test using the `shinycannon` command-line tool from the RStudio terminal tab.

7) Analyze the results by reading the data into `shinyloadtest::load_runs()` and create a report with `shinyloadtest_report()`:

```
df <- shinyloadtest::load_runs("run1")
shinyloadtest::shinyloadtest_report(df,
"run1.html")
```

- Snapshot-based testing

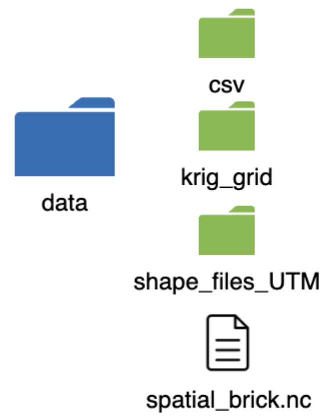
    Snapshot-based testing differs from other strategies in that Snapshot-based tests typically capture the state of the entire application, along with screenshots. This makes them very useful for testing the full user experience, but makes them very sensitive to any changes in the application, including any of its software dependencies (OS, R version, R package versions, etc).

    - **Shinytest** package provides tools for creating and running automated tests on Shiny applications. Shinytest uses a snapshot-based testing strategy. The first time it runs a set of tests for an application, it performs some scripted interactions with the app and takes one or more snapshots of the application's state. These snapshots are saved to disk so that future runs of the tests can compare their results to them.
        - The process for using the **shinytest** package consists of three steps:
            - Create a test recording
            - Interact with the application as normal and take snapshots
            - Finish recording
            - Run the test script again in the future
            - Decide if the app is running correctly by accepting or declining new snapshots.
        - Approach for using this package
            - Get the Shiny application to a state that looks good.
            - Create a test recording with shinytests's `` `recordTest()` `` function. This will open a web browser with your application running in an iframe, and a set of shinytest controls on the side.
            - Interact with the application as normal. These interactions will be recorded to a *test script*. Along the way, click the Take snapshot button to tell the test script to record snapshots of the application's state.
            - Once the recording is finished, it will play back the test script using a headless web browser and take snapshots where specified. These snapshots will be stored as the expected results.
            - When running the test script again in the future, it will take new snapshots. If the new snapshots are the same as the expected (old) ones, then the test passes. If the new snapshots differ from the expected ones, then the user will be prompted with a web interface whether to accept the new snapshots (and use them in the future as the expected snapshots), or to reject them

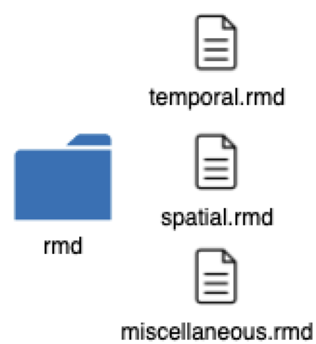
User Documentation

Data Management

Infrastructure/file structure



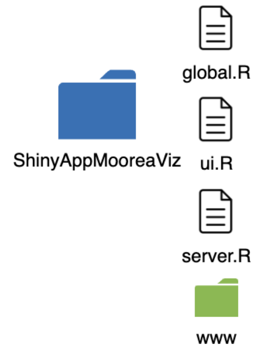
- data
    - csv: contains all raw datasets needed to produce the spatial layers, leaflet map, and temporal visualizations
    - krig_grid: contains the 5x5 meter gridded spatial pixel polygon of the reef boundary used for the kriging process
    - shape_files_UTM:
        - clipped_reef_bound: contains the shapefile for the reef boundary with the island clipped from the center
        - moorea: contains the shapefile for the island boundary
    - spatial_brick.nc: contains the 9 spatial layers as rasters



- rmd
    - temporal.rmd: includes all code to clean, tidy, subset, and join various temporal datasets in order to produce the plots for coral cover, macroalgae, fish biomass, and Crown-of-Thorns for the Shiny app. The final joined dataset is written to a csv file to be called into the Shiny app within the global.R file. This Rmd file is used to make sure the data are wrangled correctly and also includes code for exploratory plots that can be used in the app.rmd file to build the visualizations on the Shiny app.
    - spatial.rmd: Includes all code to clean, wrangle, and krige the spatial data to produce rasters. There is code to create the six LTER sites' bounding boxes and apply all visualizations to an exploratory leaflet map.
    - miscellaneous.rmd: Includes miscellaneous code useful for initial data exploration, plots, and visualization attempts. Includes code to make and clip the gridded box used for the kriging process, write and save shape files, and make explanatory ggplots with the island outline. Everything is in UTM.

- figs
    - leaf.html: contains the leaflet map
- ShinyAppMooreaViz
    - global.R: loads all packages and datasets used within the app

- ui.R: contains code for the user interface, including app layout and user inputs
- server.R: contains code for making reactive dataframes where necessary, plot outputs, and anything pertaining to producing the visualizations based on user inputs
- www: contains all photos used within the app and a .css script to do custom theming for the app. The www directory is the directory expected by a Shiny application to locally store the elements that will be rendered in the web browser and are not the output of the scripts. It must always have this name.



- island: contains the shape files for all of the surrounding islands in the French Polynesia island complex
- island_shape: contains the shape file for just the island of Moorea

Datasets and Metadata

The spatial data sets present in the Shiny application are derived from multiple survey methods. The bathymetry data for the lagoon was collected using airborne operated LiDAR and compiled into an XYZ data file. The predicted sewage data was calculated through modeling conducted by fellow researchers and stored as a spatial csv file. The percent nitrogen and isotopic data was collected through *Turbinaria ornata* samples approximately evenly distributed throughout the surrounding lagoon and stored as a spatial csv file. The coral bleaching data was collected through visual surveys conducted by divers. Divers would enter the water at anchor point locations where *Turbinaria ornata* was collected and swim in a compass heading visually estimating the percent of each coral bleached along the compass heading and stored as a spatial csv file. All spatial csv files were then krigged using ordinary kriging from the evenly distributed data points across the 4 spatial data sets.

The temporal data sets present in the Shiny application are surveyed on MCR LTER long-term survey sites. Across the six long-term sites there are permanent transects that are surveyed annually. The four temporal variables, percent coral cover, Crown-of-Thorns density, herbivore fish biomass density, and percent macroalgae cover have variable survey methods at the permanent transects. The four survey methods are conducted on or around the permanent transects in order to gather an accurate representation of the biota present on that portion of the reef during that survey period.

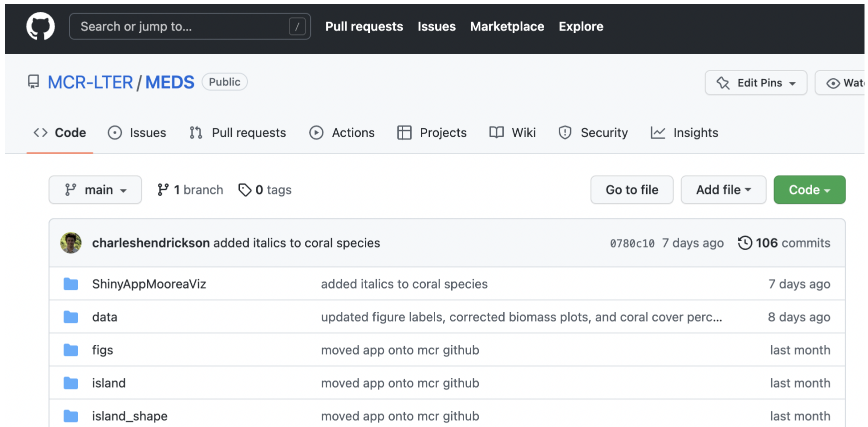
All associated metadata for the public data sets can be found on the MCR LTER website data repository. There is also metadata included within the Shiny app itself.

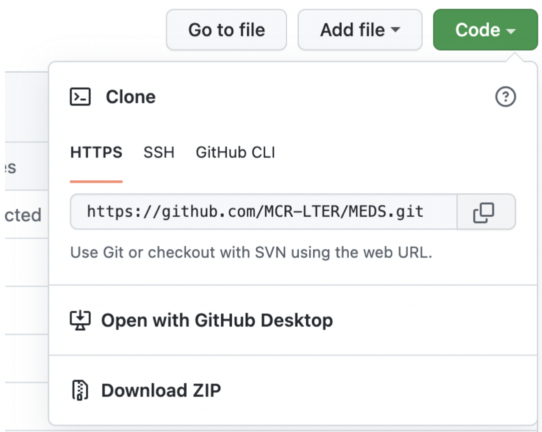
Code Management

App maintenance

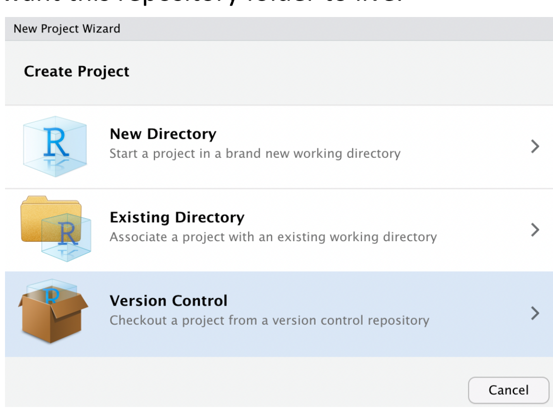
- All changes to the app should be pushed to GitHub. Instructions on how to push and pull to Git and how to create a GitHub account can be found below.
  - How to create a GitHub account
    - Great instructions can be found on the [GitHub website]()

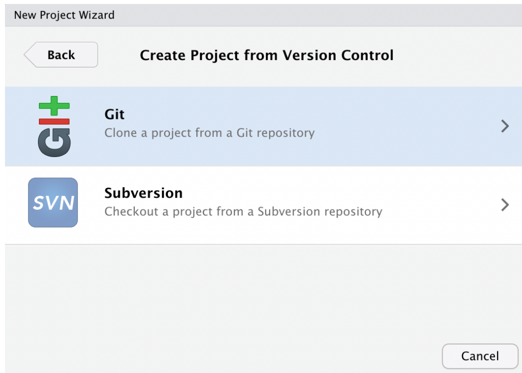
- Connecting your GitHub to your R and cloning the repository
  - To get started working in the repository you need to clone the MCR-LTER/MEDS repository and add it to your local machine. This is done by clicking the green code button on GitHub and copying the repository.

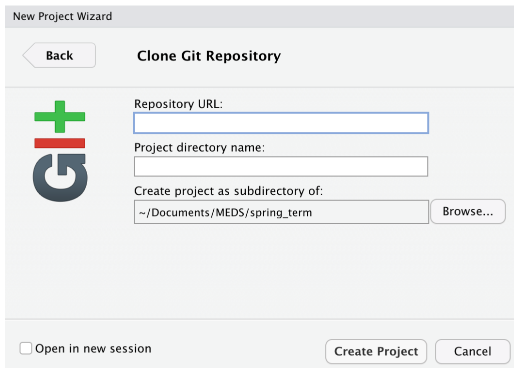




  - Once you have that copied you can go to RStudio, create a new repository, and choose the Version Control option. The page that pops up is where you can paste the repo link. Here you can also decide where on your local machine you want this repository folder to live.

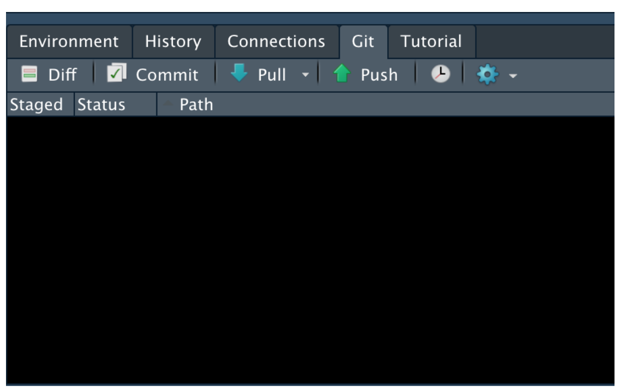




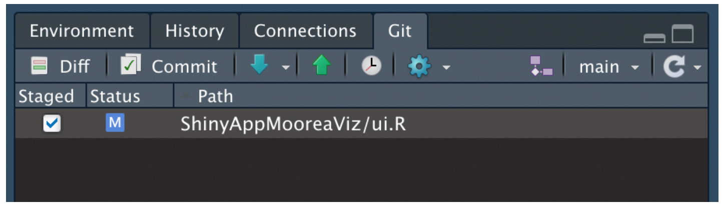


- Once everything is named and in the desired location, create the repository and you will now be connected to Git.

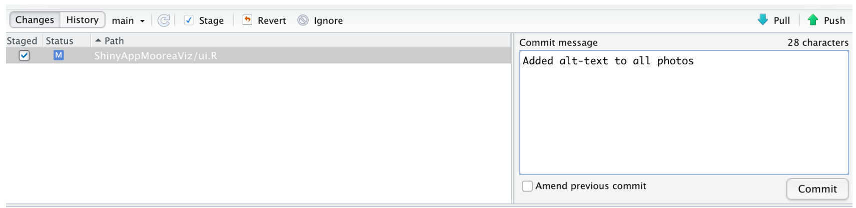
- How to push and pull from Git
  - From R
    - Once you have cloned the repo and created a version controlled repository in R (steps above) you should see a Git panel in the top right corner of your screen.



    - Any files that have changes you have made (save first!) will show up under that Git panel. Select the boxes of all the ones you want to push to GitHub.



- Then hit commit and type out a message to go along with your commit. This message will show up on GitHub explaining the changes you made.

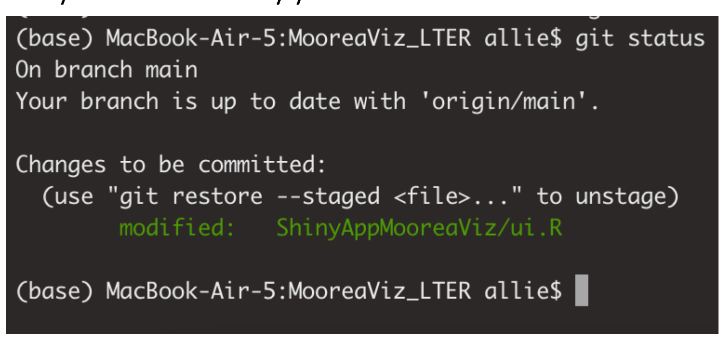


- Once you have committed, pull changes, and then push your changes, using the blue and green arrows in the photo above

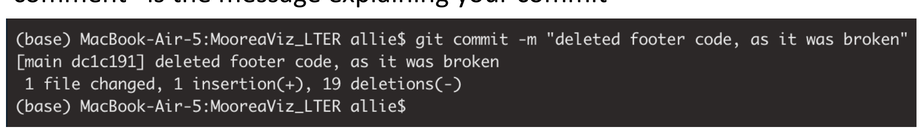
■ [From terminal]

- If you are more comfortable working in the terminal you can push and pull from there. This can be done from the Terminal in R or the terminal on your computer, the commands are the same
- The photos here are using the R studio terminal.
- Start by using ` git add .` this will select all the files that have changed (similar to selecting all the boxes in the workflow above) or you can type the name of the specific file you would like to add. IF done correctly it will look like nothing has happened.

- Next, verify that you have the files you need by typing `git status` If the files you want are ready you can move on.



- Now you are ready to commit. `git commit -m "comment"` where the “comment” is the message explaining your commit



- Now that everything is committed you can push by using `git push -u origin branch_name`

```
(base) MacBook-Air-5:MooreaViz_LTER allie$ git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 392 bytes | 392.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/MCR-LTER/MEDS.git
   acb7a2a..dc1c191  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

- Versioning and History with Git
  - If you need to access an older version of your code you can access all the previous commits through GitHub. Click the clock icon that tells you the number of commits.

  - Here you will see all the commits that have been made and you can click on one to access the code as it was at the time of that specific commit.

- Deploying an update with GRIT
  - Contact Arron Martin with GRIT to deploy an update: grit-zammad@ucsb.edu

Adding new data

Integrating new or updated data into the Shiny app requires downloading the new data into the data folder, reading in the data into the RStudio environment, and formatting and tidying the data before it can be visualized on the Shiny app.

New Spatial Data

- First, the new data must be downloaded into the R project.

    - Download the new data into the `csv` folder, which is within the `data` folder.
    - Spatial data should be .csv files, while LiDAR data should be .xyz files.
    - The new data should be labeled with a short, descriptive name and the year of collection (Ex: name_year.csv)
- Second, the new data must be read into the RStudio environment from the csv folder.
    - Navigate to the `ShinyAppMooreaViz` folder, which contains the `global.R` script.
    - Data will always be read in at the top of the `global.R` script.
    - The script's documentation will show which lines of code are used to read in specific data types, such as nitrogen, bleaching, or sewage data.

  Example:
        - To read in nitrogen data, navigate to the documentation called `nitrogen data`.
            - #nitrogen data
        - Then, input the complete file name into the file path after `data` and `csv`.
            - nitrogen_data <- read_csv(here("data", "csv", "name_year.csv")) %>% clean_names()
        - This code will create a data frame named `nitrogen_data` that will be called upon later in the Shiny app script.
    - Once all the files are set up to be read in, it is time to format and tidy the data.
- Third, the data must be formatted with the correct column names and row values. Additionally, the data frame must be formatted into Tidy format before it can be used in the Shiny app.
    - The columns within the new data frames must have the same names as the previous data.
    - The rows within the new data frames must have the same units as the previous data.
      Example:
        - Column names must be kept the same (percent_n_jan, percent_n_may, percent_n_july) . However, we use the `mutate()` function to modify the rows in these three columns so that they contain values in percentages, not decimals.
            - ```
              n_data <- nitrogen_data %>%
              mutate(percent_n_jan = percent_n_jan *100,
                     percent_n_may = percent_n_may *100,
                     percent_n_july = percent_n_july *100)
              ```
    - Untidy data must be wrangled into [Tidy format](), where each observation is a row and each column is a variable.
      Example:
        - The "nitrogen_data" data frame was not in tidy format so two columns named `type` and `percent_n` were created to contain the different nitrogen samples (percent_n_jan, percent_n_may, percent_n_july) and the percentages for those data types respectively.
            - ```
              pivot_longer(!1:5, names_to = "type", values_to = "percent_n") %>%
              separate(type, into = c("method","random", "date"), sep = "_") %>%
              dplyr::select(-random)
              ```
- Once the data frames are formatted correctly, the Shiny app can be run with the new data.

New Temporal Data

- To add new temporal data, the existing individual raw data csv files should be appended in a way that preserves the current format (i.e., all columns should have the same names and order).
    - The new files will then need to be placed in the `csv` folder within the `data` folder, replacing the outdated raw dataset using the *exact same name*.
- Rerun the entire Rmarkdown file (temporal.Rmd) to produce and export the updated combined temporal csv (temporal_data_joined.csv).
    - Note: Delete the existing temporal_data_joined.csv before running the entire Rmarkdown file to avoid issues with overwriting the pre-existing file.
- Since the temporal_data_joined.csv is already sourced in the global.R file within the `ShinyAppMooreaViz` folder, you do not need to edit anything within the Shiny app code files, aside from updating plot titles, labels, etc. that come with a new year's data (i.e., updating the plot subtitles to say Moorea French Polynesia, 2006-2022).
- Re-run the Shiny app and all temporal visualizations should be updated.
- Special notes
    - All reactive dataframes need to be followed by a set of parentheses when they are called into the renderPlot functions (i.e., `temporal_reactive_df()`)
    - All inputIds and plotOutputs used within the Shiny app need to have unique names

Archive Access

The Shiny application script, spatial script, temporal script, and User Guide will all be archived on the Moorea LTER organizational GitHub repository. All data used for this project can be found on the public GitHub repository MCR-LTER/MEDS. The Shiny app will be hosted on the Bren School server for up to six months until it can be moved over to the UCSB GRIT server.

References

Adam, T. C., Burkepile, D. E., Holbrook, S. J., Carpenter, R. C., Claudet, J., Loiseau, C., Thiault, L., Brooks, A. J., Washburn, L., and Schmitt, R. J. (2021). Landscape-scale patterns of nutrient enrichment in a coral reef ecosystem: implications for coral to algae phase shifts. *Ecological Applications* 31(1):e02227.

Donovan, M. K., Adam, T. C., Shantz, A. A., Speare, K. E., Munsterman, K. S., Rice, M. M., Schmitt, R. J., Holbrook, S. J., & Burkepile, D. E. (2020). Nitrogen pollution interacts with heat stress to increase coral bleaching across the seascape. *Proceedings of the National Academy of Sciences*, *117*(10), 5351. https://doi.org/10.1073/pnas.1915395117

Hadley Wickham (2011). testthat: Get Started with Testing. The R Journal, vol. 3, no. 1, pp. 5--10.

Hiemstra, P.H., Pebesma, E.J., Twenhofel, C.J.W. and G.B.M. Heuvelink, (2008). Real-time automatic interpolation of ambient gamma dose rates from the Dutch Radioactivity Monitoring Network. Computers & Geosciences, accepted for publication.

Holbrook, S.J., Adam, T.C., Edmunds, P.J. et al. (2018). Recruitment Drives Spatial Variation in Recovery Rates of Resilient Coral Reefs. Sci Rep 8, 7338 . https://doi.org/10.1038/s41598-018-25414-8

Joe Cheng, Bhaskar Karambelkar and Yihui Xie (2021). leaflet: Create Interactive Web Maps with the JavaScript `Leaflet` Library. R package version 2.0.4.1. https://CRAN.R-project.org/package=leaflet

Kirill Müller (2020). here: A Simpler Way to Find Your Files. R package version 1.0.1. https://CRAN.R-project.org/package=here

Moberg, F., Folke, C. (1999), Ecological goods and services of coral reef ecosystems. Ecological Economics. 29:215-233. https://doi.org/10.1016/S0921-8009(99)00009-9.

Reaka-Kudla, M. L. (2005). "Biodiversity of Caribbean coral reefs," in Caribbean Marine Biodiversity, eds P. Miloslavich and E. Klein (Lancaster, PA: DesTech Publishers), 259–276.

Robert J. Hijmans (2021). raster: Geographic Data Analysis and Modeling. R package version 3.5-2. https://CRAN.R-project.org/package=raster

Pebesma, E., (2018). Simple Features for R: Standardized Support for Spatial Vector Data. The R Journal 10 (1), 439-446, https://doi.org/10.32614/RJ-2018-009

Pebesma, E.J., (2004). Multivariable geostatistics in S: the gstat package. Computers & Geosciences, 30: 683-691.

Pebesma, E.J., R.S. Bivand, (2005). Classes and methods for spatial data in R. R News 5 (2), https://cran.r-project.org/doc/Rnews/.

Sam Firke (2021). janitor: Simple Tools for Examining and Cleaning Dirty Data. R package version 2.1.0. https://CRAN.R-project.org/package=janito

Simon Garnier, Noam Ross, Robert Rudis, Antônio P. Camargo, Marco Sciaini, and Cédric Scherer (2021). Rvision - Colorblind-Friendly Color Maps for R. R package version 0.6.2.r

Vega Thurber, R.L., Burkepile, D.E., Fuchs, C., Shantz, A.A., McMinds, R. and Zaneveld, J.R (2014). Chronic nutrient enrichment increases prevalence and severity of coral disease and bleaching. Glob Change Biol, 20: 544-554. https://doi.org/10.1111/gcb.12450

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

Winston Chang, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert and Barbara Borges (2021). shiny: Web Application Framework for R. R package version 1.7.1. https://CRAN.R-project.org/package=shiny

Woodhead, AJ, Hicks, CC, Norström, AV, Williams, GJ, Graham, NAJ. (2019). Coral reef ecosystem services in the Anthropocene. Funct Ecol.; 33: 1023– 1034. https://doi.org/10.1111/1365-2435.13331