Mapping Ocean Coral Bleaching

Jessica Farlow, Mikaela Gordon, Eli MacColl, Joseph Nisbet, Owen Spencer

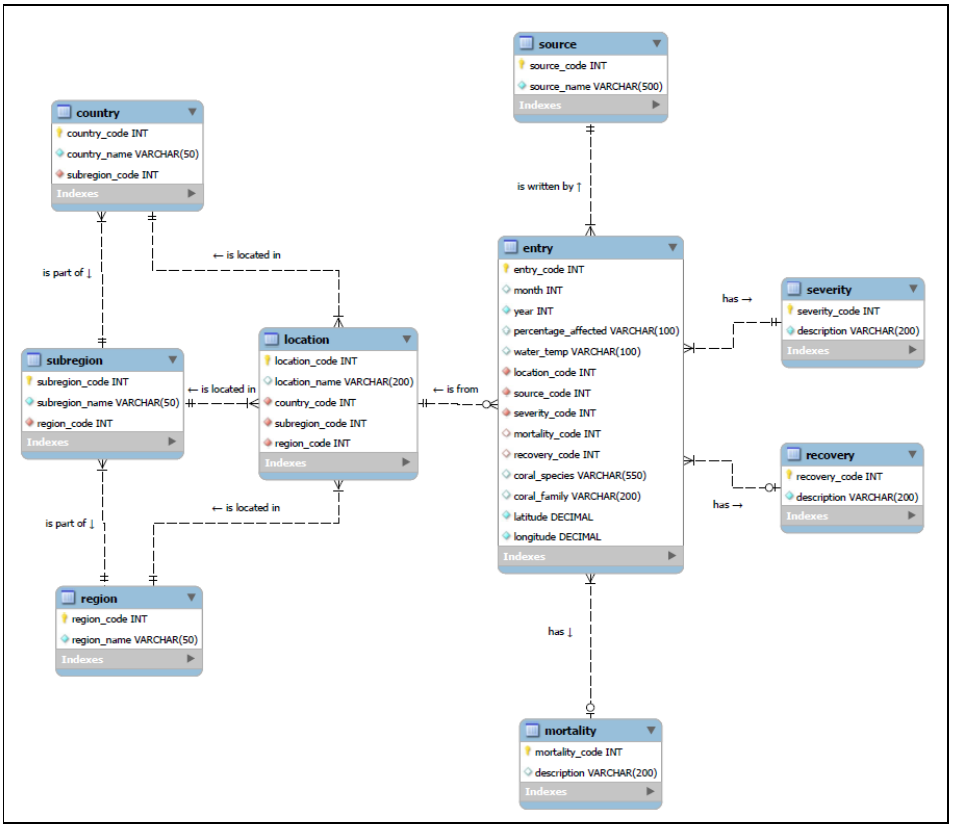
June 24th, 2021

Introduction

Coral reefs provide protection for sea life and coastal ecosystems all around the world. At every reef's core, hundreds of corals varying in species and size create complex ecosystems with the surrounding sea life. Unfortunately, rising sea temperatures along with increases in ocean acidification stress corals, causing them to expel an essential symbiotic partner, zooxanthellae. Zooxanthellae, microscopic photosynthesizing organisms, provide corals not only their color, but also a way to convert carbon dioxide waste into oxygen and glucose. Without zooxanthellae, corals become 'bleached', leaving corals vulnerable to disease and starvation. Coral bleaching events are occuring in increased frequency in a myriad of locations around the world. Coral bleaching destroys vital ocean ecosystems and leaves coastal environments vulnerable to severe storms. As such events have become larger and more dangerous, researchers around the world have begun tracking and analyzing the impacts of coral bleaching and death. By building a database to organize the information collected, researchers and environmentalists could observe and update ongoing surveys. Visualizations of coral bleaching events can also aid in global surveys, drawing attention to critical regions, as well as guide users to draw their own conclusions about the evidence provided.

In order to create this database, we requested access to an Excel spreadsheet from Harvard Dataverse. This spreadsheet contained entries from various sources on coral reef bleaching events worldwide, including information about coral reef locations, health, mortality, and more. After manipulating the spreadsheet into separate tables and creating an ER Diagram, we were able to use Python to write usable insert statements to import data into MySQL. There are a few noteworthy problems with the Excel spreadsheet that posed a challenge. For example, the spreadsheet was littered with inconsistencies in spelling and filing of information. The water temperature column included numbers along with descriptions such as "unusually warm," making it challenging to enter consistent information into our database. Some single entries for coral species included over 3000 characters. Despite these drawbacks, we were able to successfully obtain detailed maps to visualize our data and draw insightful conclusions about coral bleaching.

Database Design - ER Diagram



ER diagram for coral bleaching database

The diagram above outlines the design of our database. Our ER Diagram consists of nine tables, with the main table 'entry' storing every record of coral bleaching accumulated in the original dataset. Branching from this, 3 small tables contain information on the severity, recovery, and mortality of coral bleaching entries. Not every entry contained mortality or recovery rates, so making these attributes nullable in the entry table allowed us to include the data we had without compromising the number of usable records. Another small table contains sources for every entry in the database. Location data was the largest part of this project and the data itself. Not all entries had specific location data, but every entry did have a latitude and longitude. Because of this, we designed our database to locate coral reefs through optional region, subregion, and country tables, as well as a required latitude and longitude within the entry table. All of these tables are connected with one-to-many relationships to provide easy joins between locational data. This ER Diagram was forward engineered, and our data was added with insert statements created from the original data source.

Methods/Use Cases

The process of populating our database proved to be relatively tedious due to a combination of poorly-recorded data and difficulties with the SQL remote access database. We used Python scripts to format the raw data from Excel so that it could be copied and pasted directly into an SQL insert statement, creating a comprehensive setup file. A challenging aspect of using this method for cleaning the data was ensuring that corresponding primary key ids were properly assigned when used as foreign keys in linked tables.

Our solution to this was to store all unique combinations of linked data into their own respective arrays and use their indexes as primary key ids. For example, to create the location table, which requires foreign keys for country, subregion, and region, we first iterated through all of the data, storing all unique countries, subregions, and regions into separate corresponding arrays. We then iterated through the data again, this time finding the index of the country, subregion, and region in their respective arrays, added one to the id values so that our primary keys would start at one instead of zero in SQL, and then stored a string into an array that was formatted as an SQL insert statement. The location table then included the location name, country id, subregion id, and region id. This step provides an entry in the location table for every row of raw data we had, but we only required unique location entries, so we filtered our array of output statements into a new, final array with duplicates removed, and wrote the contents of this final array as a single line in a text file. This allowed us to then simply copy and paste the long string written to the text file into an SQL insert statement in the workbench.

Similar procedures were carried out for every table and resulted in a fully functional SQL setup file to create and populate our coral bleaching database. This was then distributed to our group members and was used to write all of our queries, draw insights, and visualize our findings. An example of the insert statements used is shown below.

```
INSERT INTO region (region_name) VALUES
(''),('Africa'),('Americas'),('Asia'),('Australia'),('Middle East'),('Pacific');
 INSERT INTO subregion (subregion_name, region_id) VALUES
('', 1), ('Eastern Africa', 2), ('Southwest Indian Ocean', 2), ('Eastern Caribbean
INSERT INTO country (country_name, subregion_id) VALUES
('France', 1), ('Macau', 1), ('Mauritania', 1), ('Mexico (Pacific)', 1), ('Panama
INSERT INTO location (location_name, country_id, subregion_id, region_id) VALUES
('Hermitage, Reunion', 1, 1, 1), ('Saint Gilles, Reunion', 1, 1, 1), ('Saint-Leu,
```

MySQL query used to import data

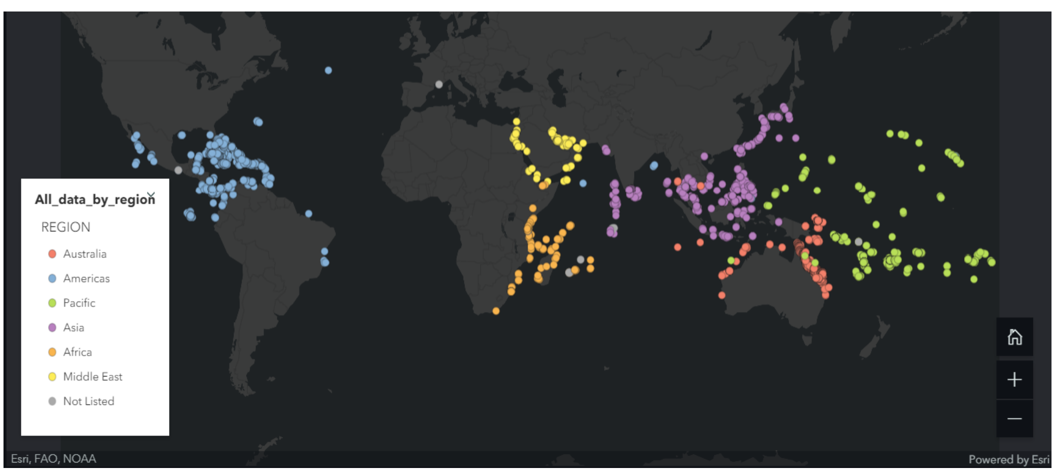
Analysis, Visualization, Insights

Using our database, we were able to write queries to obtain information and then export it to new .csv files. We used ESRI's ArcGIS Story Maps in order to map the data and put it into an interactive website, utilizing latitude and longitude data for the mapping, and other attributes for the symbols and differentiation of points. Our mapping queries were all very similar to each other with the exception of different 'where' clauses, filtering the data by region and year for a more readable map.

```sql
-- map 1: all data, most recent to least
select region_name, subregion_name, country_name, location_name,
       latitude, longitude, month, year, severity_id, severity_desc,
       coral_family, coral_species, mortality_code
from entry e
join severity using (severity_id)
left join source on (source_id)
left join location l using (location_id)
join country using (country_id)
join subregion s on (l.subregion_id = s.subregion_id)
join region r on (l.region_id = r.region_id)
order by year desc, month desc;
```

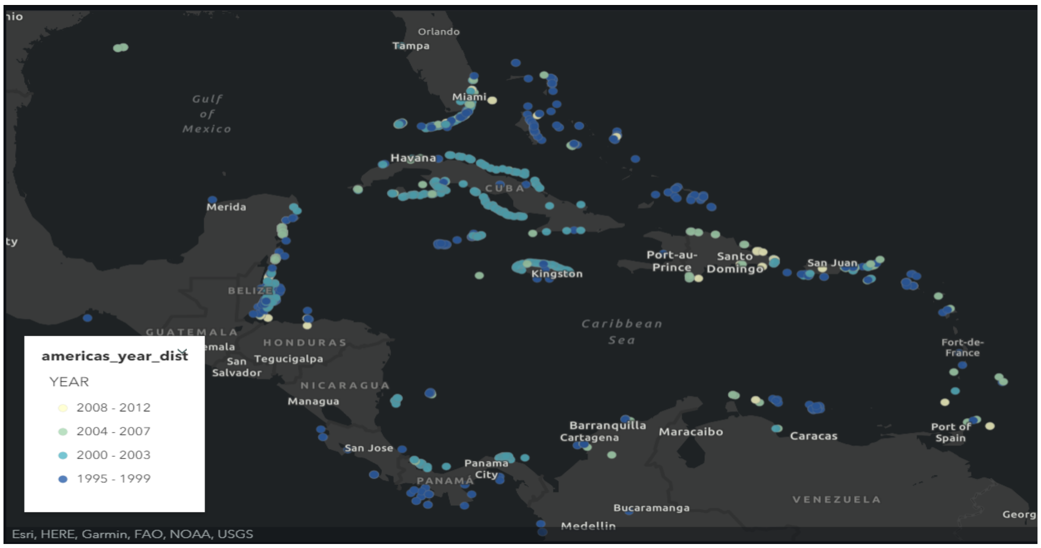
Base MySQL query used for visualizations, modified with 'where' clauses

The figure above shows the base query used for Map 1, which mapped all of our data points. All of the other map queries used this format with an additional 'where' clause to specify the region and years of interest. As well as including latitude, longitude, and specific attributes necessary for mapping, other columns are included because the maps are interactive. By doing this, the user can click on any point to see all of its corresponding data, not just the parts being used for a specific map. Map 1 below shows all of our data points, colored by region.



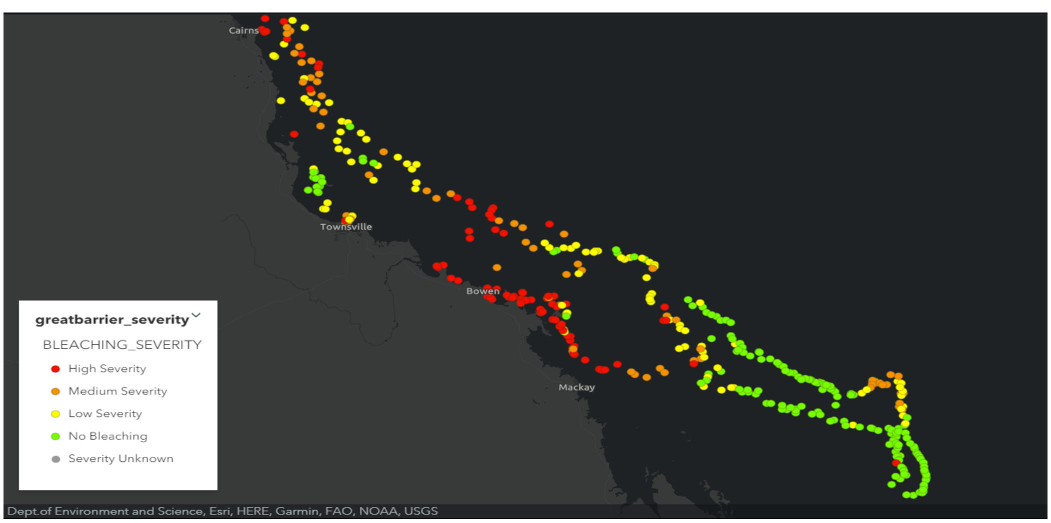
Map 1: All of the data points, grouped and colored by region

Map 2 illustrates the distribution of years vs location, focusing on the Americas region. Since our dataset is a compilation of many different data collections, it is important to consider the years of our data in order to better compare them. For example, data for Cuba and Jamaica are almost exclusively from the years 2000 - 2003, which is significant if you're trying to compare it to places like the Bahamas where most data is from 1995 - 1999, or the Florida Keys where data exists in many different time periods.



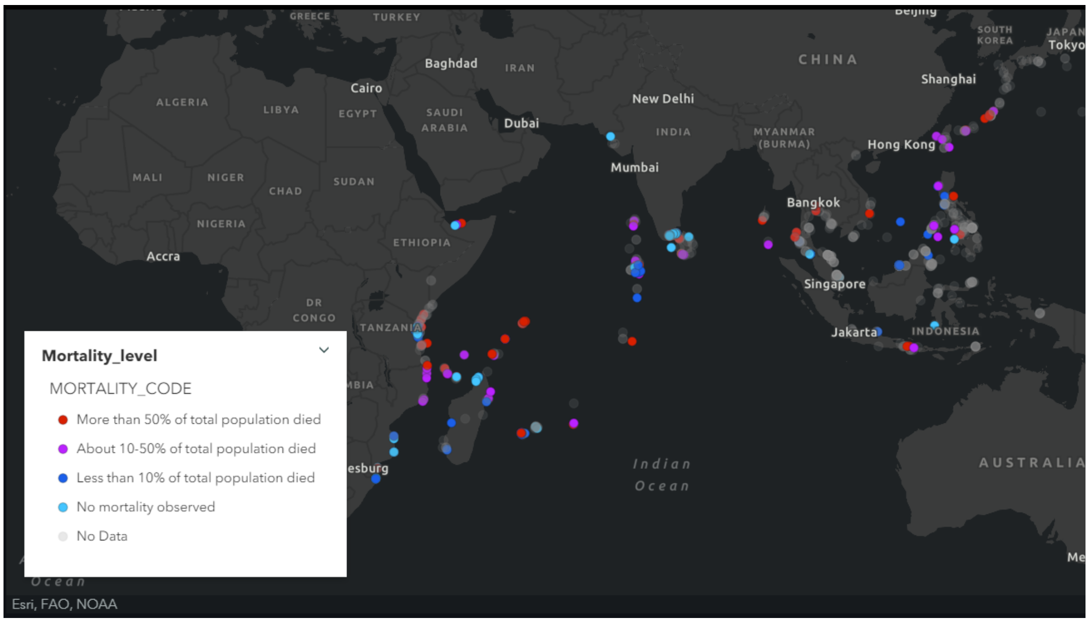
Map 2: Year Distribution of Entry Data for the America's region, 1995-2012.

Map 3 focuses on the severity of bleaching in Australia's Great Barrier Reef from 2000-2004. The most severe bleaching (shown in red) was found near the coast, while records with little or no bleaching (shown in yellow and green) are found farther out. This is to be expected, as shallow water near the coasts are vulnerable to warming and human pollution, two main stressors that cause coral bleaching.



Map 3: Great Barrier Reef Bleaching Severity from 2000 - 2004

Lastly, Map 4 shows the mortality codes and descriptions for the regions of Africa and Asia. Mortality code was one of the attributes that was more sparse in our data, shown by the light grey "no data" dots in the map. It is a common misconception that all bleached coral is dead, but this isn't the case. There are plenty of light blue, dark blue, and purple points, illustrating that it is possible for a majority of a coral population to survive even after bleaching. With that being said, bleaching still weakens the coral significantly and it is important to remember that the coral is still vulnerable.



Map 4: Mortality Levels for the Regions of Africa and Asia

All of the interactive maps, along with some more information and pictures, can be found at:

https://arcg.is/1Krfma0

Conclusion

Coral bleaching impacts ecosystems and coastal communities worldwide. As bleaching events increase in frequency and severity, these essential ecosystems face potential extinction, leaving coasts vulnerable to devastating disasters. To combat the looming threat, more resources and research are required to protect coastal environments. The database we created organizes thousands of coral bleaching surveys and visualizes each in easy-to-use ARCGIS maps. Not only do these maps and programs aid researchers, but they also raise awareness to public users about coral bleaching. With environmental conservation at its core, our database and visualizations could help research teams further populate a growing database and provide education opportunities for the public.

Looking ahead, many changes and efforts are needed to better our project. Although the Harvard Dataverse spreadsheet was useful, its organization was challenging to navigate. With many columns scarcely filled and thousands of entries, the spreadsheet and database itself would need a significant amount of manual entry and reorganization to better function in the database. Additionally, the sheer amount of inconsistencies in spelling, measurements, and column data made each entry vary in information. Many columns following severity, mortality, and recovery codes had spaces for describing the health of the reefs in each entry. These descriptions varied wildly, making these codes for each health characteristic difficult to understand. Overall, this spreadsheet could use a lot of reorganization, constraints, and normalization before our database becomes a publicly available program.

Despite the challenges with the data source, our maps and database proved useful in organizing and visualizing coral bleaching surveys worldwide. In the future, after much review of the data, it would be preferable to require users to enter data for almost every field, as well as add more columns for pH measurements and photographs.

Author Contributions

Owen Spencer initially set up the remote database using Heroku and organized the slideshow presentation. He also wrote the Introduction and Conclusion for the final report, as well as draft the Database Design section. Joe Nisbet created the first iteration of the conceptual/logical modeling of our database, which was then improved upon by Jessica. He helped reformat the excel spreadsheet to provide cleaner data and allow us to store location in separate tables. Jessica helped to clean the raw data as well as finalized the ER diagram of our database design in MySQL. She also contributed to the final writeup of the Database Design section of the report. Eli developed a series of python scripts to format the raw data into SQL syntax to create a setup file for our database. The setup file was the source of usable data for the project, and he was also responsible for the methods section in the final report. Mikaela wrote the map queries, created the visualizations, and embedded them into the final story map website with other information. She wrote the analysis, visualization and insights section for the report.