Read me file for SOAR monitoring data analysis

Jordanna Barley

June 12, 2023

**Repository configuration:**

<https://github.com/jmbarley1/SOAR_monitoring>

The repository that I have created for this project was meant to be self-explanatory, but I will briefly explain here. There are four folders that I have stored all the data, R code, markdowns, etc. The R code that I have written are in two scripts. First, I start with an ‘environmental variables’ script which loads in all the data and makes changes to it based on what I want to do with it. At the beginning of the analysis script, I tell R to run the environmental variables script so that every session you have the same variables and objects available to you. I also started an R project, which someone working in R later should be able to use. To use this repository, you will have to clone it to your computer. Here is a [helpful link](https://docs.github.com/en/repositories/creating-and-managing-repositories/cloning-a-repository) about cloning GitHub repositories. Also, here is a [link](https://happygitwithr.com/rstudio-git-github.html) on how to use GitHub within R studio, which is what I have done here. Also, I can transfer the ownership of this repository to someone else working with/on the SOAR project.

**Data configuration:**

I configured a master datasheet that I used to extract the data from each site. The second sheet on the spreadsheet is a list of all the columns and what data they include. Some of the columns were not used but I kept them because they might be helpful in the future. For the most part, in this data sheet, each row is one oyster. There are a couple of sites that did not provide individual oyster- level data, and in those cases, average size and proportional data are provided instead.

**Analysis:**

I have annotated the R script describing what I did for the analysis. Overall, I tried to visualize the data in many ways to search for outliers and patterns. I also ran some generalized linear models, which we ended up not really reporting on because of how variable the data are. One of the main findings of the analysis is that many wild oysters (or those deployed through other restoration projects) were misidentified as SOAR oysters. However, we did see high survival in all but a few sites. In addition, there does seem to be a correlation between SOAR oyster deployment and increased recruitment at a couple of sites.

**Proportional survival:** I calculated the proportional survival across all the sites. This was done by dividing the number of live oysters collected in each replicate sampling effort (ie. Quadrat) and dividing by the total number collected in that replicate. Then, I averaged the replicate data to get one proportional survival for each site.

**Live oyster density:** I calculated the density of live oysters by multiplying the average number of live oysters found in each quadrat by the quadrat multiplier. For example, if the quadrat used was ¼ m2, I multiplied the average number of live oysters by 4 to get the density measured in m2.

**Population Estimate:** I multiplied the density of live oysters by the reef area to calculate the population estimate. The population estimates were generally larger than the number of oysters deployed at each site. This indicates that oysters that were either wild oysters or deployed as part of another restoration program were counted at SOAR oysters.

**Recruitment:** For the sites that provided recruitment data, I calculated recruitment density by the same method as above. For the New Hampshire site, I was not able to calculate density because they did not collect their data using quadrats and we just had a total count of recruits.

**Average size and growth:** I averaged the size of oysters within each replicate. To calculate growth, I subtracted the mean size of the oysters at deployment. This metric ended up not being useful because many of the growth data were negative, likely due to a significant amount of non-SOAR oysters being counted during monitoring of the SOAR program.

Once I calculated these metrics, I visualized them in various configurations of graphs. In the script, I have annotated what metric is visualized with each code chuck.