DSAF Shiny Prototype Directory

Jake Lawlor
December 10, 2025

Summary

The attached DSAF Shiny Application allows users to view the status of a hypothetical fish population – separated between multiple jurisdictional zones in Canada – assessed by six indicators of species' change. To balance computation and interactivity, the app relies are precomputed visualizations, which are uploaded on load and displayed according to user inputs.

Here, I outline the contents of the application folder, including scripts to preprocess data, scripts to create interactive visualizations, intermediate data objects, and application scripts.

Directory overview

```
.
├── Data/
│   ├── Data_SHinyApp_Proof_of_Concept/        Various input data from DSAF
│   └── eez_can/                               Shapefile of Canadian EEZ
└── shiny/
    ├── preprocessing_scripts/                 Scripts for making interactive plots
    ├── shiny_processed_data/                  Intermediate objects for plots
    ├── www/                                   Final objects to load into App
    ├── ui.R                                   User interface - overall
    ├── ui_Tab1.R                              Module for Page 1 (about)
    ├── ui_Tab2.R                              Module for Page 2 (data page)
    ├── ui_Tab3.R                              Module for Page 3 (science)
    ├── ui_Tab4.R                              Module for Page 4 (download)
    ├── app_text.R                             All application text, saved as list
    ├── server.R                               Server logic
    └── global.R                               Global objects to upload in app
```

Making / Editing Plots

All plots in the application are pre-rendered from scripts in the "preprocessing_scripts/" folder, and saved into the "www/" folder for use in the Shiny app. In "preprocessing_scripts", the file "00_plot_opts.R" controls global options (colors to represent each region) that apply to all following plots. The script, "01_preprocess_all.R" sources all other scripts to create all interactive plots for the application. For example, if the variables in "00_plot_opts.R" are changed, running "01_preprocess_all.R" will override existing plots with the new settings.

Editing Application Text

For organization, all text from the shiny app is stored in the script, "app_text.R". In this script, all text objects are organized in a hierarchical list object, with sublist for text in the header, tab 1, tab 2, and so on. This script it sourced by the application in each UI script. Changes here will be reflected when the application is run. Note that some of the elements are character strings, but some are wrapped in html functions (e.g., h2("text")), or written with html formatting (e.g., HTML("some words are bolded")). New text can either follow this same structure, or be entered directly into UI pages of the app.

Application Logic

The application logic is simple. The "global.R" script uploads all pre-made plots for a given species (here, "sppx") into a list, and the server pulls elements from these list to render plots according to user inputs. When the user selects "Species X", the server pulls from the "sppx_plots" list, and when the user queries for "p_abundance", the server renders an output from sppx_plots$p_abundance. This structure results in fast rendering and stability since the computation for plots is done in advance, but it is likely to face limits scaling up to many species since the application would need to store every plot in advance.

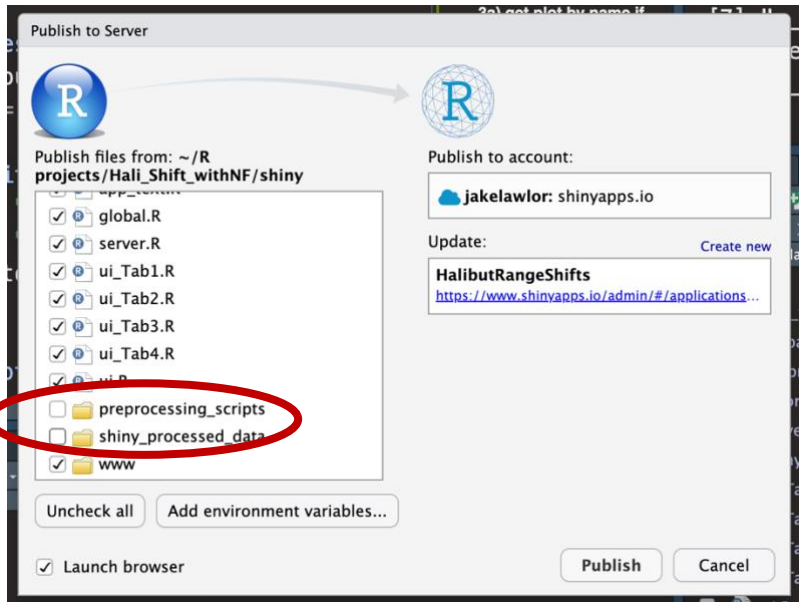
Placeholder Objects

This application also relies on a few pre-rendered objects that are placeholders, not made directly from the provided data. These objects are meant for demonstration purposes only, but could be further developed in the future. For example, the "download" button on UI Tab 4 is meant to show how custom summary outputs could be programmatically written from application data: when a user selects a focal species and study area, a summary of the species' status in that region could be generated. Here, we use a stored static pdf

("shiny/www/example_summary.pdf") to demonstrate what this might look like. Replace this file to change the downloaded pdf output.

Hosting the Application

Not all files are necessary for hosting the application on a Shiny server, and excluding unnecessary files will save space and improve efficiency on the server. Inside the "shiny/" folder, "preprocessing_scripts/" and "shiny_preprocessed_data/" do not need to be included. When hosting from RStudio, these folders can be un-selected on publish (see below). For publishing, all app scripts (UI, server, global, and text) are necessary, and the "www/" folder.



Contact

For questions or follow-up, please feel free to contact me at jake.lawlor@mail.mcgil.ca.