UNIVERSITY of WASHINGTON

# BASA Model Sensitivity App

**Mindy Dai, Trinity Hinshaw, Veronica Lee, CL Roberts**

**December 11, 2024**

## Problem Statement

- Working with a BASA (Bayesian age-structured assessment) model that predicts herring stock
- Model requires an input parameter for mortality
- How sensitive is the model to a change in the mortality value?
- Goal: create an app where a variety of users can explore the BASA base model and its sensitivity

# Users and Use Cases

- Primary user and technician: CL Roberts
  - Will use the app to inform his dissertation work
  - Interested in generating residual values and statistics between the base model and mortality-adjusted model
- Secondary users: Ecosystem researchers
  - May want to extend the app's relevancy to their own population work
- Tertiary users: Board members for agency
  - Want to evaluate the BASA model's quality and advocate for fishing interests

# Technologies Used: Overview

- Built the app using Shiny for Python
- Created the UI components using Shiny
- Created technical components, along with corresponding tests, in Python modules
  - Run the BASA model using the set mortality value
  - Change the mortality parameter
  - Plot both base and adjusted model outputs for biomass
  - Calculate percentage error between the models' biomass outputs
  - Plot the percentage error values

# Technologies Used: Shiny App

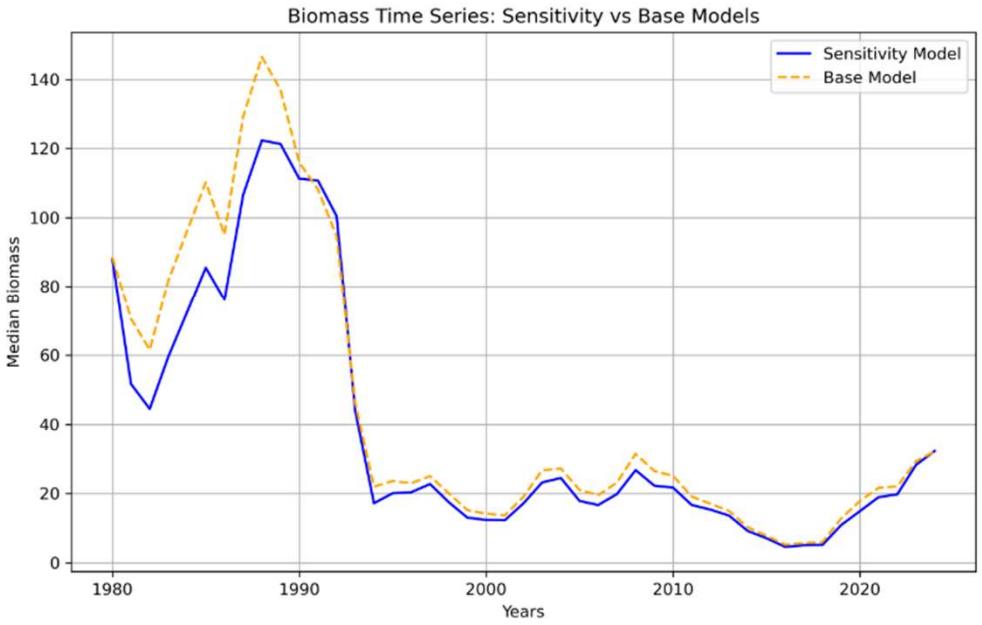
Natural Mortality

0 | 0.25 | 0.5

Run Model

Welcome

Comparison to base model biomass



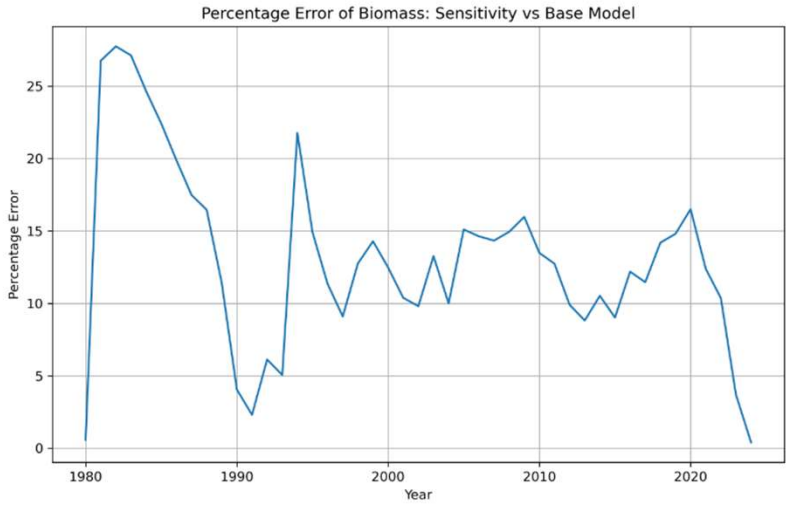
# Technologies Used: Shiny App

Natural Mortality

0 0.25 0.5

Run Model

Error analysis




| Mean Percent Error | Min Percent Error | Max Percent Error |
|---|---|---|
| 13.06% | 0.4% | 27.75% |

UI components

Tab j

Mortality slider

Other UI elements

Action Button

py modules

modify_mortality()

modify_other_params()

Writes `sensitivity-j` dir

run_basa()

read_biomass()

read_biomass()

sensitivity_comparison()

Download button

Writes analysis outputs to `csv`

plot_sensitivity_comparison()

plot_sensitivity()

Value boxes

Plots

Downloads folder

# Major Challenges

- Managing and resolving merge conflicts for Git
- Shiny would run on some computers but not others
  - Working with a relatively new package (“Shiny for Python”) with limited online help
- Figuring out how to connect component modules to each other
  - Optimizing the number of components needed
- Figuring out how to call components to the Shiny app
- Calling on functions from the component modules within our test modules *(prior to pyproject.toml)*

## Next Steps

- Expanding on the test modules to add more complexity, especially implementing tests for Shiny (using *shinytest*)
- Add different parameters for other sensitivity considerations (more sliders)
- Implement multiple tabs so that users can compare different sensitivity runs of the model
- Add feature to allow users to download plots and sensitivity run information to their local computer

# Where to Access

- GitHub Repository:
  - https://github.com/cl-roberts/pws-herring-basa/tree/sensitivity/sensitivity