This document outlines the GenEst application.

**GenEst App**

The GenEst app is structured as a classic shiny app, with two scripts: one covering the user interface (ui.R) and the other covering the server-side operations (server.R).

Shiny apps are structured to allow reactive programming (think of it as an active conversation) between the UI and the server via inputs (user-generated objects) and outputs (server-generated objects). Reactive programming is also supported by the use of reactive values objects within the server.

*Server-Side*

The server.R script contains the code that takes the inputs provided by the user, runs code, and returns outputs to the user.

The server code is held within the generic function returned by the server.R function. Within the generic function, the majority of the code constrained to only be executed once the user performs certain actions in the UI (which updates a specific input), which prevents code from running prematurely. The only code within the function that is not held back is the code that creates the disclaimer and the code that initializes the reactive values object.

The following outline of the server-side code is structured as

Section

User action

Server executions

Data Input

Search Efficiency Data File is uploaded

SE data file is read into R

Column names from the SE data are pulled in and updated in model input drop downs

SE data table is rendered for output

Carcass Persistence Data File is uploaded

CP data file is read into R

Column names from the CP data are pulled in and updated in model input drop downs

CP data table is rendered for output

Search Schedule Data File is uploaded

SS data file is read into R

Column names from the SS data are pulled in and updated in model input drop downs

SS data table is rendered for output

SS translation table is rendered for output

Carcass Observations Data File is uploaded

CO data file is read into R

Column names from the CO data are pulled in and updated in model input drop downs

CO data table is rendered for output

Meta Data File is uploaded

MD data file is read into R

MD data table is rendered for output

Search Efficiency Analysis

At least one SE observation column is selected

SE data table for analysis is rendered for output

Run SE Model button is pushed

The SE models are run using the inputted values

The SE theta is created using the model output and the inputted values

AIC tables are generated for each size class

The model and size class options are updated in the output drop downs

The specific AIC table and figure outputs are cleared (render NULL)

Generate AIC Table button is pushed

The AIC table for the user-inputted size class is rendered for output

Generate SE Figure button is pushed

The SE figure for the user-inputted size class and model is rendered for output

Populate Options button is pushed

As long as the SE model has been run, drop downs for model selection are provided for each size class

Carcass Persistence Analysis

The last time observed and first time absent CP observation columns are selected

CP data table for analysis is rendered for output

Run CP Model button is pushed

The CP models are run using the inputted values

The CP theta is created using the model output and the inputted values

AIC tables are generated for each size class

The model and size class options are updated in the output drop downs

The specific AIC table and figure outputs are cleared (render NULL)

Generate AIC Table button is pushed

The AIC table for the user-inputted size class is rendered for output

Generate CP Figure button is pushed

The CP figure for the user-inputted size class and model is rendered for output

Populate Options button is pushed

As long as the CP model has been run, drop downs for model selection are provided for each size class

Detection Probability Estimation

Estimate Detection Probability button is pushed

Detection probability is estimated for each size class

Detection probability table is rendered for output

Fatality Estimation

Estimate Total Carcasses button is pushed

Total number of carcasses is estimated for each user-inputted split category

Total carcass table and figure are rendered for output

*User Interface*

The ui.R script contains the code that generates the web document (written in HTML) that creates the user interface. The ui.R script consists of a single call to the navbarPage function, which creates the UI definition used by shiny to create the app. navbarPage creates a page with a top level navigation bar that allows toggling among tabs. Subsequent embedded calls to tabsetPanel create further articulation of the UI by allowing tabbed pages to occur on tabs of higher level pages.

The following outline of the UI code is structured according to the tabs from the main page on downward.

Home

A simple splash page, contains only the GenEst logo.

Data Input

Sidebar: file inputs

Main panel: tabset panel with further tabbing

Search Efficiency

Shows SE data

Carcass Persistence

Shows CP data

Search Schedule

Shows SS data

Carcass Observation

Shows CO data

Meta Data

Shows MD

Analyses

A tab page broken into tabs

General Inputs

Sidebar: general model inputs

Main panel: empty

Search Efficiency

Sidebar: model inputs

Main panel: tabset panel with further tabbing

Data

Shows data being modeled

Model Table

Shows AIC table for a given size class

Figure

Produces a figure for a given size class and selected model

Model Selection

Used to select a model for each size class

Options can only be populated once the model has been run

Carcass Persistence

Sidebar: model inputs

Main panel: tabset panel with further tabbing

Data

Shows data being modeled

Model Table

Shows AIC table for a given size class

Figure

Produces a figure for a given size class and selected model

Model Selection

Used to select a model for each size class

Options can only be populated once the model has been run

Detection Probability

Sidebar: model run button, search schedule translation table

Main panel: where the detection probability table is output

Fatality Estimation

Sidebar: model inputs

Main panel: tabset panel with further tabbing

Data

Shows data being modeled

Table

Produces the fatality table

Figure

Produces the fatality figure

About

GenEst logo and additional details about the app.