Cape Cod Commercial Fishermen's Alliance

Data Portal Developer Guide

Danielle Quinn | quinn.dea@gmail.com | 30 December 2021

DATA PORTAL DEVELOPER GUIDE

Raw Data Requirements

Description Of Dummy Data

Samples of anonymized data were used to build a data portal prototype, including three primary datasets:

- Electronic Vessel Trip Reporting (eVTR) Data (orig-data/FY2019-eVTRdata-GARFO.csv)
- Electronic Monitoring (EM) Data (orig-data/Example of an FY19EM Summary File.xlsx)
- Dealer Data (orig-data/FY19 Dealer Data SIMM.xlsx)

Detailed descriptions of these data sets, including variable classes, summary statistics, and categorical levels can be found in Appendices A – C. The data portal prototype was built using these specific data structures and characteristics; if real values fall outside of the range of values supplied in the dummy data, there may be instances where the code and/or design will need to be updated. Numeric variables are less likely to be problematic in this regard than categorical variables. If additional levels are added to categorical values, scale assignments, input controls, and data processing steps may / will need to be adjusted. Note that when these data sets are imported into R, any spaces in the variable names are replaced by periods.

Data Portal Requirements

Species Reference Table

A reference table (data/species.csv) containing aliases of 104 common names was accessed from https://github.com/gamaynard/ElectronicMonitoring, and includes six variables:

- PEBKAC: character string of full and abbreviated forms of the name found in the eVTR, EM, and dealer data
- ITIS: five to six digit Integrated Taxonomic Information System species code
- AFS: character string of standardized name
- GROUNDFISH: logical value indicating whether the species is considered a groundfish
- NESPP3: three digit Northeast Species Code (market category excluded)
- NESPP4: four digit Northeast Species Code (market category included)

This table is used to standardize the reporting of species across data sources to those names found in the AFS variable. Note that this reference table does not include all aliases of species

names found in the dummy data set. Where AFS names are assigned based on similarity scores, some errors will occur. For instance, the eVTR data included a value of species_id SROB, representing Sea Robin; because this abbreviation was not included in the PEBKAC variable of the species reference table, the record with the highest similarity score is LOB, which represents American Lobster and these eVTR records were subsequently mislabeled.

Data Pre-Processing

Dealer Data

The dealer data (orig-data/FY19 Dealer Data SIMM.xlsx) was processed in preparation for integration into the data portal using the script pre-processing/pre-processing.R. This script accomplished the following:

- Permit numbers (Vessel.Permit.No) and VTR numbers (Vtr.Serial.No) were converted to character strings to maintain any leading zeros and were stored in new variables called PERMIT and VTR, respectively
- Vessel names (Vessel.Name) were converted to uppercase letters and stored in a new variable called VESSEL
- Dates (Date.Sold) were converted to POSIX datetime values and stored in a new variable called DATE
- Species ITIS codes (Species.Itis) were referenced against the species reference table (see *Species Reference Table*, above) to join species names (AFS) as a new variable called species
- Variable names were converted to lowercase letters
- The processed mri, vessel.reg.no, state.land, port.land, species.itis, landed.weight, live.weight, date.sold, permit, vessel, vtr, date, and species variables were exported to a new file data/simm_processed.csv

Electronic Monitoring (EM) Data

The EM data (orig-data/Example of an FY19EM Summary File.xlsx) was processed in preparation for integration into the data portal using the script pre-processing/pre-processing.R. This script accomplished the following:

- The Location variable, which contained latitude and longitude information in the form "*latitude, longitude*" was separated into two columns representing latitude (LAT) and longitude (LON)
- Any values of LAT and LON with absolute values less than five were replaced with NA
- The Start.Timestamp and End.Timestamp variables were converted to POSIX datetime values and stored in new variables called STARTTIME and ENDTIME, respectively
- Using the stringdist::simstring() function, a similarity score (optimal string alignment method) was calculated between each values of the Species variable and the vector of species names and abbreviations (PEBKAC) found in the species reference table (see *Species Reference Table*, above); the species name (AFS) corresponding to the highest similarity score was used to populate a new column in the EM data called SPECIES; the original Species variable was removed
- Variable names were converted to lowercase letters
- The processed EM data was exported to a new file data/em_processed.csv

Electronic Vessel Trip Reporting (eVTR) Data

The eVTR data (orig-data/FY2019-eVTRdata-GARFO.csv) was processed in preparation for integration into the data portal using the script pre-processing/pre-processing.R. This script accomplished the following:

- All variable names were converted to lowercase letters
- Any records with a serial number (serial_num) that did not contain 16 digits were omitted; these represented paper trip reports which were not included herein
- The date_sail, date_land, datetime_haul_start and datetime_haul_end variables were converted to POSIX datetime values and stored in new variables called DATE.SAIL, DATE.LAND, HAULSTART, and HAULEND, respectively
- Permit numbers (vessel_permit_num) were converted to character strings to maintain any leading zeros and were stored in a new variable called PERMIT
- Missing values in the lat_second and lon_second variables were replaced with 0
- Any positive longitude values (lon_degree) were multiplied by -1 to ensure that they represent locations west of the Prime Meridian

- Degrees (lat_degree, lon_degree), minutes (lat_minute, lon_minute), and seconds (lat_second, lon_second) were combined to generate decimal degrees for both latitude (LAT) and longitude (LON) using the formula degree + minute/60 + second/(60^2) for latitude and degree – minute/60 – second/(60^2) for longitude
- Gear codes (gearcode) were replaced with human-readable values
    - GNS was replaced with GILLNET
    - HND was replaced with JIG
    - LLB was replaced with LONGLINE
    - OTF was replaced with TRAWL
    - PTL was replaced with LOBSTER POT
- Where possible, the first 14 digits of 16 digit serial numbers (serial_num) were used to generate VTR numbers, stored in a new variable called VTR; where serial numbers were not 16 digits in length, VTR was populated with NA
- Permit numbers (PERMIT) were referenced against the dealer data to join vessel names (VESSEL) as a new variable; any records with missing vessel names were dropped
- Using the stringdist::simstring() function, a similarity score (optimal string alignment method) was calculated between each values of the species_id variable and the vector of species names and abbreviations (PEBKAC) found in the species reference table (see *Species Reference Table*, above); the species name (AFS) corresponding to the highest similarity score was used to populate a new column in the eVTR data called SPECIES
- Variable names were converted to lowercase letters
- The processed serial_num, gearcode, gearqty, gearsize, lat_degree, lat_minute, lat_second, lon_degree, lon_minute, lon_second, ntows, species_id, port_landed, date.sail, date.land, haulstart, haulend, permit, vessel, lat, lon, vtr, and species variables were exported to a new file data/evtr_garfo_processed.csv

R Packages

The required R packages and their dependencies can be installed directly from CRAN using install.packages().

- shiny (version 1.7.1)
- shinydashboard (version 0.7.2)

- shinyjs (version 2.1.0)
- shinyBS (version 0.61)
- dplyr (version 1.0.7)
- ggplot2 (version 3.3.5)
- lubridate (version 1.8.0)
- forcats (version 0.5.1)
- tibble (version 3.1.6)
- tidyr (version 1.1.4)
- stringr (version 1.4.0)
- readr (version 2.1.1)
- purrr (version 0.3.4)
- DT (version 0.20)
- leaflet (version 2.0.4.1)
- leaflet.extras (version 1.0.0)
- mapview (version 2.10.0)

Environmental Data

A sample of environmental data was manually downloaded from the Environmental Research Division's Data Access Program (ERDDAP) via their online data portal (https://coastwatch.pfeg.noaa.gov/erddap/index.html). This included water temperature, salinity, and conductivity data (intended to be used as general placeholders rather than important environmental metrics for users) collected between 1 Jan 2019 and 31 Dec 2020 from four Northeastern Regional Association of Coastal Ocean Observing Systems (NERACOOS) stations:

- A01 Massachusetts Bay (www.neracoos.org/erddap/tabledap/A01_sbe37_all.html)
- B01 Western Maine Shelf (www.neracoos.org/erddap/tabledap/B01_sbe37_all.html)
- E01 Central Maine Shelf (www.neracoos.org/erddap/tabledap/E01_sbe37_all.html)
- F01 Penobscot Bay (www.neracoos.org/erddap/tabledap/F01_sbe37_all.html)

Recommendations on accessing and integrating more robust environmental data from a variety of sources are provided in the *Next Steps* sections of this document.

Species Colour Assignments

The species colour scale reference table (species_hex.csv) contains the variables species (title case of standardized AFS species names) and hex (a corresponding hexadecimal code) and was used to define the colour used in data visualizations associated with that species (see *Workspace and Data Preparation*, below). Hexadecimal colours were manually assigned to each species based on customized groupings (Table 4).

Data Portal Structure & Processes: Workspace and Data Preparation (global.R)

The overall structure of the data portal, including the relationship between the various data sets and R scripts, is demonstrated in Figure 1. The design of the directory structure allows a developer to quickly identify and find the code used to accomplish specific tasks (Table 5)

R Packages

The required R packages are loaded from the local library of installed packages.

Processed Data Import and Cleaning (global-components/data-import-prep.R)

Minor data cleaning processes were applied to the incoming data.

Dealer Data

- The start.timestamp variable was converted from character to POSIX date
- A new POSIX date variable called dummy.date was created from date.sold to be used to compare daily and monthly values across years
- A new numeric variable called year was created from start.timestamp
- The species variable was converted to title case
- White spaces were trimmed from the port.land variable
- A spelling error in port.land was fixed; values of Portlane were changed to Portland

Electronic Monitoring (EM) Data

- The start.timestamp variable was converted from character to POSIX datetime
- A new POSIX date variable called start.date was created from start.timestamp

- The evtr variable was converted from numeric to character
- The species variable was converted to title case

Electronic Vessel Trip Reporting (eVTR) Data

- Underscores in variable names were replaced with periods
- The variables date.sail, date.land, haulstart, and haulend were converted from POSIX datetime to POSIX date; the original POSIX datetime values were used to populate new variables datetime.sail, datetime.land, haulstarttime, and haulendtime
- A new POSIX date variable called usedate was created; this was populated with the haulstart date where possible, and date.sail where haulstart was not available. New numeric variables week and month were created from usedate
- The port.land variable was converted to title case
- A spelling error in port.land was fixed; values of Harwichport were changed to Harwich Port
- The species variable was converted to title case
- A new numeric variable called haul.id was created to be used as an identifier of individual hauls based on unique combinations of vtr, haulstarttime, and haulendtime

Environmental Data

- Data from the four stations were merged into a single data frame which summarised the daily minimum, mean, and maximum values of temperature, conductivity, and salinity, by station; these values were stored in variables with names formatted *metric_statistic* (i.e. temperature_max, salinity_min)

Colour and Fill Scales **(**global-components/import-scales.R**)**

Customized colour and fill scales were defined for categorical variables:

- Species (fillScaleSpecies and colScaleSpecies, n levels = 89, n colours = 33)
    - based on the hexadecimal codes found in the species colour scale reference table (species_hex.csv)
- Electronic monitoring category (fillScaleCategory, n = 5)
- Electronic monitoring subject (fillScaleSubject, n = 17)

    - linked to parent category (Table 6)
- Catch (fillScaleCatchType, n = 2)
    - levels represent Kept and Discarded catch
- Gear type (fillScaleGearType, n = 6)
- Port landed (fillScalePort, n = 9)
- Sales metric (fillScaleMetric, n = 2)
    - levels represent live weight sold (dealer data), landed weight sold (dealer data), and weight reported (eVTR data)
- Station (colScaleStation, n = 4)
    - levels represent NERACOOS stations with available environmental data

Data Portal Structure & Processes: User Interface (UI) (ui.R)

The data portal uses a standard dashboard page layout from shinydashboard, with a header, toggled sidebar panel and main body panel.

Sidebar Panel (ui-sidebar.R)

The sidebar panel contains menu and submenu items used for navigating around the data portal:

- Welcome
- eVTR Data
    - Catch
    - Effort
    - Environmental
- Dealer Data
    - Sales
    - Comparison to eVTR
- EM Feedback
    - Catch Reviews
    - Other Events
- Export Data
- About the Data

The sidebar panel also contains two input controls: a select input control (inputId = vessel), used to select one vessel, and a date range input control (inputId = date), used to select a minimum and maximum date value.

Main Panel (ui-body.R)

The layout and contents of the main body of the data portal is dependent on the submenu item selected by the user in the sidebar panel. Each menu subitem has a corresponding script (i.e. ui-components/ui-body-catch.R) defining the layout and contents of that item using an HTML based UI definition, which are sourced by ui-body.R. These scripts contain static input controls, reactive output controls, and text sourced from text-files/.

Welcome (ui-components/ui-body-welcome.R)

The welcome tab contains general instructions sourced from text-files/ and two inactive input controls, representing a possible approach for users to enter a secure username and password before accessing the raw data.

Data Portal Structure & Processes: Reactive Server (server.R)

The server function contains the reactive components of the data portal. Each menu subitem has a corresponding script (i.e. server-components/server-catch.R) which are sourced by server.R.

User Defined Data Queries (server-components/server-dataprep.R)

outputId	inputID	Description
sidebar_vessel_input	vessel	Select one vessel; populated using vessel values available in evtr data
sidebar_date_input	date	Select date range; populated using usedate values available in evtr_full(), em_full(), and simm_full()

In addition to the server scripts for each menu subitem, the server-components/ directory contains the server-dataprep.R script. The dealer, eVTR, and EM data sets are filtered based on the vessel selected by the user in the sidebar panel (inputId = vessel) and stored in the reactive functions evtr_full(), em_full(), and simm_full(), respectively. Using this filtered data, the upper and lower range of the date range input control (inputId = date) in the side bar is limited to the available date range, and the default values are set as the last 90 days of the available range. The

eVTR, EM, dealer, and environmental data sets are then filtered based on this default date range or a customized date range selected by the user and stored in the reactive functions evtr_(), em_(), simm_(), and enviro_data_() respectively.

eVTR Data

Catch Data (server-components/server-catch.R)

Input Controls

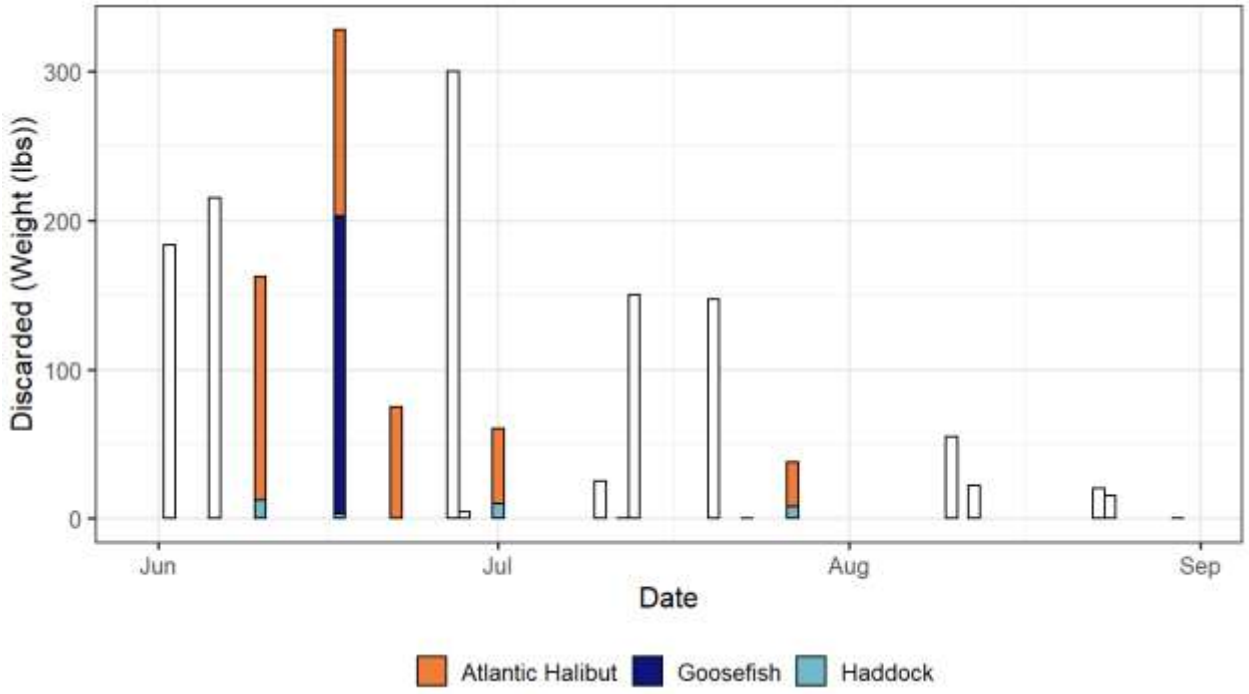
The Catch submenu item uses the following input controls for user interaction:

outputId	inputID	Description
catch_catchtype_input	catch_catchtype	Select one catch method; Kept, Discarded, or All
catch_gear_input	catch_gear	Select one or more gear types; populated using gear values available in evtr_()
catch_species_input	catch_species	Select one or more species; populated using species values available in evtr_()
catch_evtr_input	catch_evtr	Select one or more eVTRs; populated using vtr values available in catch_data()
catch_fill_input	catch_fill	Select one variable used to define fill scale; Catch Type, Gear Type, or Species
catch_click	catch_click$x	Click on the catch_plot to select a bar
catch_table_res_input	catch_table_res	Select one resolution for summarising data; Overall, By Species, or By Gear

Reactive Data

A reactive working data set catch_data() is generated based on the reactive evtr_() data. The data are grouped by vtr, usedate, gear, species, lat, and lon; for each group the total value of kept and discarded are calculated, respectively, as well as the combined total (total). When user input is provided by the catch_gear or catch_species input controls, the appropriate filters are applied. The value provided by the catch_catchtype input control is used to determined which metric (kept, discarded, or total) is used in the plots, maps, and tables and stored as a new variable called use_col.

Reactive Figure



A reactive function `catch_plot_prep()` defines the process for rendering `catch_plot()` in two layers, allowing users to highlight specific eVTR values selected with the `catch_evtr` input control:

(1) Baseplot: Using `catch_data()`, an unfilled bar plot is generated with `usedate` on the x axis and the sum of the metric selected by the user (stored in `use_col`) on the y axis.

(2) Overlay: Using `catch_data()`, the `catch_evtr` input control is used to filter the data appropriately; if no eVTRs have been selected, no filter is applied. The value of the `catch_fill` input control determines the variables used to group the data: (a) if `Gear Type`, the data are grouped by `usedate` and `gear`; (b) if `Species`, the data are grouped by `usedate` and `species`; (c) if `Catch Type`, the data are grouped by `usedate`.

- If the `catch_fill` input control is `Gear Type` or `Species`, a bar plot is generated with `usedate` on the x axis and the sum of the metric selected by the user (stored in `use_col`) on the y axis. The fill colour is based on `Gear Type` or `Species`, as defined in `fillScaleGearType` or `fillScaleSpecies`, respectively (see *Colour and Fill Scales*).

- If the catch_fill input control is Catch Type, the total kept and total discarded catch are summed for each value of usedate. The data is reshaped into long format, with a variable called catchtype indicating whether the corresponding value represents the total kept or total discarded catch. A bar plot is generated with usedate on the x axis and the total values of kept and discarded on the y axis. The fill colour is based on catchtype, as defined in fillScaleCatchType (see *Colour and Fill Scales*).

Clicking on the plot stores the associated x and y values, stored as catch_click$x, which is converted to a date and the vtr values of records in the catch_data() with matching dates are printed to the screen. If no data is available to generate catch_plot(), the renderPlot(catch_plot_prep()) function will display a "No data available" message.

Reactive Map

A reactive function catch_map_prep() defines the process for generating the catch_map() as a Leaflet heatmap with overlaid points based on the catch_data(). To structure the data for the heatmap, catch_data() is grouped by location (lat and lon), and the sum of the metric selected by the user (stored in use_col) is calculated for each location. This data is expanded so that each unique location has one row per catch recorded, and the heatmap is rendered using a Spectral gradient. The catch_data() is then structured for the point map; it is grouped by vtr, location (lat and lon) and species, and the sum of the metric selected by the user (stored in use_col) is calculated for each group. A character string is created for each point with the eVTR and location information, which is used to integrate popups that appear when the user clicks on the map points. If the catch_species input control is used to select ten or fewer species, the character string used in the map popup is modified to list the catch for each species at that location. If the catch_evtr input control is used to select specific eVTR values, the point data for those trips will be highlighted in gold on the map. If no data is available to generate catch_map(), the renderLeaflet(catch_map_prep()) function will display a "No data available" message.

Reactive Table

eVTR	Date Fished	Total Kept	Total Discarded	Total
24169919060207	2020-06-02	764	637	1401
24169919060607	2020-06-06	1191	2123	3314
24169919061007	2020-06-10	685	1656	2341
24169919061707	2020-06-17	788	802	1590

A reactive function catch_table_prep() defines the process for generating the catch_table() from the catch_data() data. Using catch_data(), the catch_evtr input control is used to filter the data appropriately; if no eVTRs have been selected, no filter is applied. The value of the catch_table_res input control determines the variables used to group the data: (a) if By Species, the data are grouped by vtr, usedate, and species; (b) if By Gear, the data are grouped by vtr, usedate, and gear; (c) if Overall, the data are grouped by vtr and usedate. For each group, the total kept, total discarded, and overall total catch are calculated. Before rendering, the total of each metric are calculated and those values appended to the bottom of the table, and the variable names are cleaned.

Download Buttons

The following download buttons are included:

- export_catch_data
    - Starts with: catch_data()
    - Modifications:
        - "eVTR" prefix added to evtr values to retain value as character string
        - usedate renamed to date.general
        - columns reordered; unnecessary columns dropped
    - Produces: catch_data.csv
- export_catch_plot
    - Saves catch_plot_prep() as catch_figure.png
- export_catch_table
    - Starts with: catch_table_prep()
    - Modifications:
        - "eVTR" prefix added to evtr values to retain value as character string

- Produces: catch_table.csv

Effort Data (server-components/server-effort.R)

Input Controls

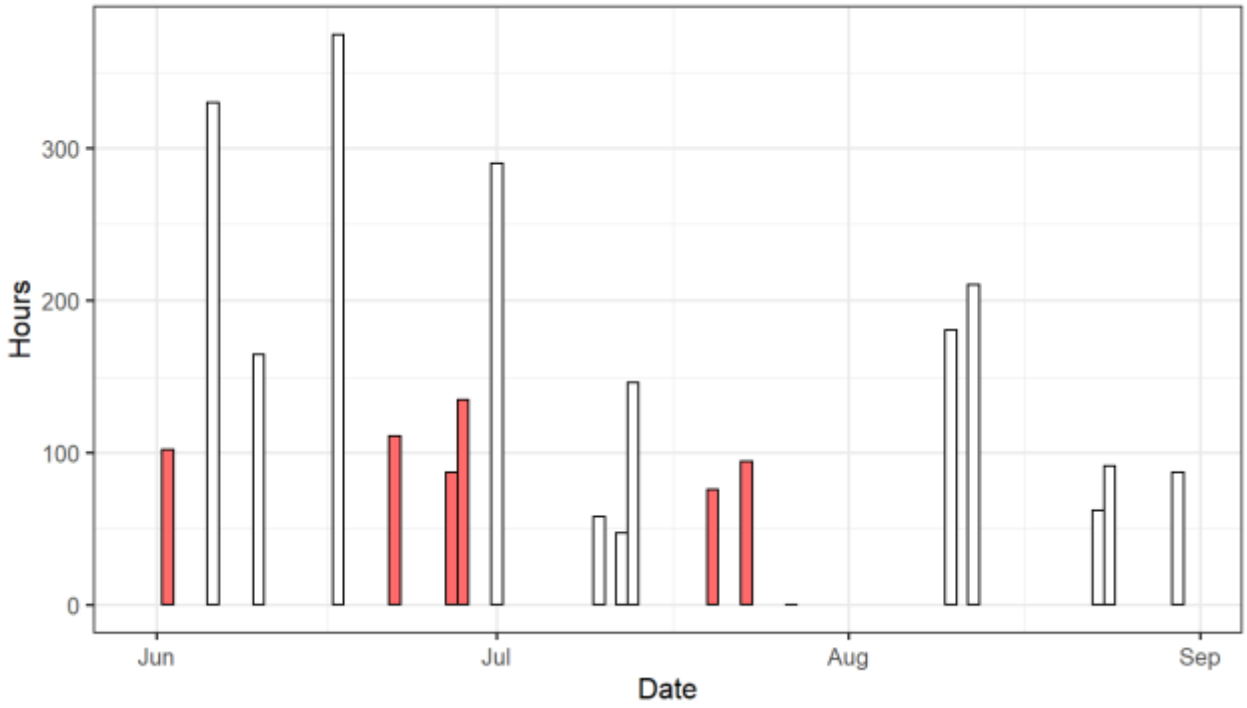
The Effort submenu item uses the following input controls for user interaction:

outputId	inputID	Description
effort_gear_input	effort_gear	Select one or more gear types; populated using gear values available in evtr_()
effort_evtr_input	effort_evtr	Select one or more eVTRs; populated using vtr values available in effort_data()
effort_click	effort_click$x	Click on the effort_plot to select a bar

Reactive Data

A reactive working data set effort_data() is generated based on the reactive evtr_() data. The data are filtered to only include records matching the gear type provided by effort_gear input control. An effort metric is calculated based on the gear type selected: (a) for Jig, Trawl, and Gillnet, effort (haultime) is the time difference between the haulstarttime and haulendtime; (b) for Lobster Pot and Longline, effort is the existing gear quantity (gearqty) variable.

Reactive Figure



A reactive function effort_plot_prep() defines the process for rendering effort_plot() in two layers, allowing users to highlight specific eVTR values selected with the effort_evtr input control:

(1) Baseplot: Using effort_data(), an unfilled bar plot is generated with usedate on the x axis and the sum of the effort metric (haultime or gearqty) on the y axis.
(2) Overlay: Using effort_data(), the effort_evtr input control is used to filter the data appropriately; if no eVTRs have been selected, no filter is applied. A bar plot is generated with usedate on the x axis and the sum of the effort metric (haultime or gearqty) on the y axis. The fill colour is based on Gear Type, as defined in fillScaleGearType (see *Colour and Fill Scales*).

Clicking on the plot stores the associated x and y values, stored as effort_click$x, which is converted to a date and the vtr values of records in the effort_data() with matching dates are printed to the screen. If no data is available to generate effort_plot(), the renderPlot(effort_plot_prep()) function will display a "No data available" message.

Reactive Map

A reactive function effort_map_prep() defines the process for generating the effort_map() as a Leaflet heatmap with overlaid points based on the effort_data(). To structure the data for the heatmap, effort_data() is grouped by location (lat and lon), and the sum of the appropriate effort metric based on gear is calculated for each location; the effort metric is haultime if the effort_gear input control is Jig, Trawl, or Gillnet, and gearqty if the effort_gear input control is Lobster Pot or Longline. This data is expanded so that each unique location has one row per unit of effort recorded, and the heatmap is rendered using a Spectral gradient. The effort_data() is then structured for the point map; it is grouped by vtr, location (lat and lon) and gear, and the sum of the effort metric is calculated for each group. A character string is created for each point with the eVTR, location, and gear information, which is used to integrate popups that appear when the user clicks on the map points. If the effort_evtr input control is used to select specific eVTR values, the point data for those trips will be highlighted in gold on the map. If no data is available to generate effort_map(), the renderLeaflet(effort_map_prep()) function will display a "No data available" message.

Reactive Table

eVTR	Date Fished	Gillnet	Lobster Pot	Longline	Jig	Trawl
13930719092506	2020-09-25				112	
13930719100313	2020-10-03				86053	
13930719100808	2020-10-08				710	
13930719101310	2020-10-13				0	

A reactive function effort_table_prep() defines the process for rendering the effort_table(). The evtr_() data is filtered to omit records where the gear is Other, and effort calculated for each record. For records associated with Lobster Pot and Longline gear, effort is measured as gearqty; for records associated with Trawl, Jig, and Gillnet gear, effort is measured as haultime calculated as the difference between the haultimestart and haultimeend. The data is then grouped based on vtr, gear, and usedate, and the total effort was calculated by group. The resulting data frame is converted to wide format and empty gear columns added as needed. Finally, the total effort based on each gear is calculated and those values appended to the bottom of the table.

Download Buttons

The following download buttons are included:

- export_effort_data
  - Starts with: effort_data()
  - Modifications:
    - "eVTR" prefix added to evtr values to retain value as character string
    - date.sail and date.land variables from evtr data set appended based on matching evtr values
    - usedate renamed to general.date
    - columns reordered; unnecessary columns dropped
  - Produces: effort_data.csv
- export_effort_plot
  - Saves effort_plot_prep() as effort_figure.png
- export_effort _table
  - Starts with: effort_table_prep()
  - Modifications:
    - "eVTR" prefix added to evtr values to retain value as character string
  - Produces: effort_table.csv

Environmental Data (server-components/server-enviro.R)

Input Controls

The Environmental submenu item uses the following input controls for user interaction:

outputId	inputID	Description
enviro_category_input	enviro_category	Select one category of environmental data; Temperature (alias temperature), Salinity (alias salinity), or Conductivity (alias conductivity)
enviro_type_input	enviro_type	Select one value type of environmental data; Minimum (alias min), Average (alias

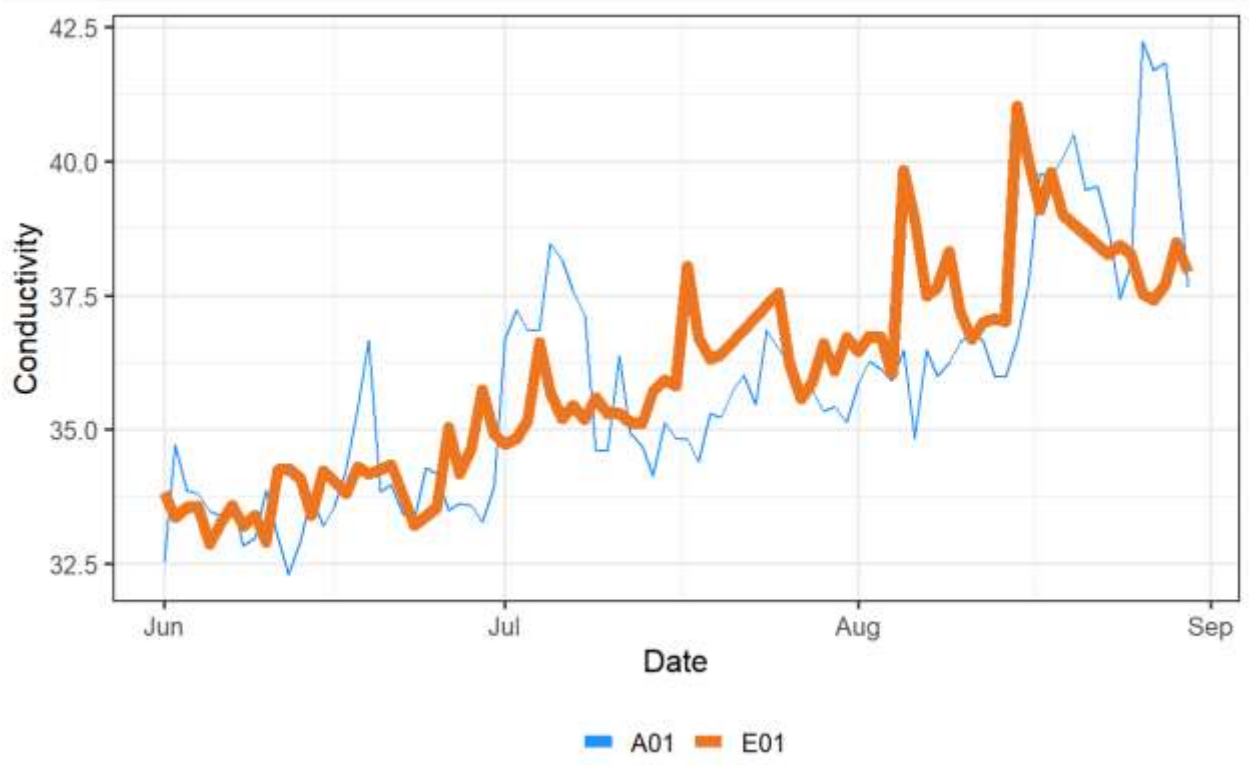
		mean), or Maximum (alias max)
enviro_depth_input	enviro_depth	Select one depth; populated using depth values available in enviro_data_()
station_map_click	station_map_marker_click$id	Click on a marker on the station_map to select one station
enviro_catchtype_input	enviro_catchtype	Select one catch method; Kept (alias kept), Discarded (alias discarded), or All (alias all)
enviro_species_input	enviro_species	Select one or more species; populated using species values available in evtr_()
enviro_resolution_input	enviro_resolution	Select the resolution at which to summarise the data; Day, Week, or Month

Reactive Map

A reactive function station_map_prep() defines the process for generating the station_map() as a Leaflet point map based on the enviro_data(). The data is filtered to only include environmental data at the depth specified by the enviro_depth input control, and one point is added to the map per station. A character string is created for each point with the station name and station id value, which is used to integrate popups that appear when the user clicks on the map points. Clicking on a point on the map will highlight it and stores the associated id values, representing station as station_map_marker_click$ids. This information is used to query the station_details() data, print the station id and name to the screen, and provide the user with the link to the associated NERACOOS webpage.

A reactive reference table station_details() is generated based on the enviro_data_() data, simplified to only include the unique station, name, and link variables of records from the station provided by the station_map_marker_click$id map click input. A reactive working data set station_data() is generated based on the enviro_data_() data. The values provided by the enviro_category and enviro_type input controls are pasted together to identify which metric (i.e. temperature_max) is used in the plots and tables and stored as a new variable called yval.

Reactive Figure 1



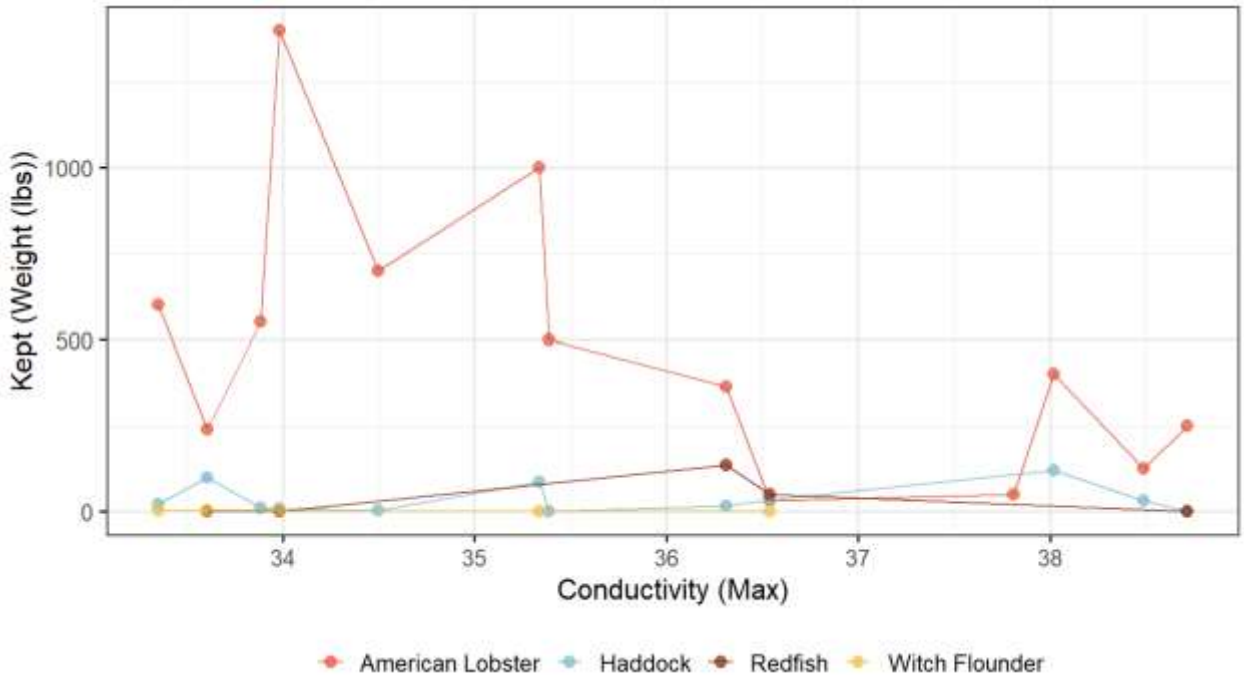
A reactive function enviro_plot_prep() defines the process for rendering enviro_plot(). The station_data() is used to generate a line plot with usedate on the x axis and yval on the y axis, with each station having a separate line. When the user clicks on a blue point on the station_map(), a layer is added to the enviro_plot() to highlight the line representing the selected station as defined by station_map_marker_click$id.

Reactive Data

A reactive working data set evtr_enviro_data() is generated based on the reactive evtr_() and station_data() data. The data are filtered to only include records from the station provided by the station_map_marker_click$id map click input, and at the depth provided by the enviro_depth input control. The value of the enviro_resolution input control determines the variables used to group the data: (a) if Day, the data are grouped by species and usedate; (b) if Week, the data are grouped by species and week; (c) if Month, the data are grouped by species and month. Based on these groups, the evtr_() data is summarised, with the total kept (kept),

total discarded (discarded), and total overall (total) catches calculated. When user input is provided by the enviro_species input controls, the appropriate filter is applied.

Reactive Figure 2



A reactive function enviro_catch_plot_prep() defines the process for rendering enviro_catch_plot() using the evtr_enviro_data(). The value of enviro_catchtype determines which metric (kept, discarded, or total) is used on the y axis. The values provided by the enviro_category and enviro_type input controls are pasted together to identify which metric (i.e. temperature_max) is used on the x axis.

Reactive Table

Date Fished	Station	Environmental Metric	Species	Kept	Discarded	Total	Metric
2020-06-02	E01	10.53	American Lobster	240	100	340	
2020-06-02	E01	10.53	American Plaice	2	0	2	
2020-06-02	E01	10.53	Atlantic Cod	150	140	290	
2020-06-02	E01	10.53	Atlantic Halibut	0	23	23	

A reactive function enviro_table_prep() generates the enviro_table() directly from the evtr_enviro_data() data after modifying and capitalizing the names of the variables. Before rendering, four additional values are calculated: (a) the average value of the environmental

variable selected by enviro_category and enviro_type, (b) the total kept catch, (c) the total discarded catch, and (d) the total catch; these summary values are appended to the bottom of the table.

Download Buttons

The following download buttons are included:

- export_enviro_data
    - Starts with: enviro_data_()
    - Modifications:
        - "eVTR" prefix added to evtr values to retain value as character string
        - variable names cleaned
    - Produces: enviro_data.csv
- export_enviro_plot
    - Saves enviro_plot_prep() as enviro_figure.png
- export_catch_enviro_data
    - Starts with: evtr_enviro_data_()
    - Modifications:
        - columns renamed; unnecessary columns dropped
    - Produces: catch_enviro_data.csv
- export_catch_enviro_plot
    - Saves catch_enviro_plot_prep() as catch_enviro_figure.png
- export_catch_enviro_table
    - Starts with: catch_enviro_table_prep()
    - Produces: catch_enviro_table.csv

Dealer Data

Sales (server-components/server-sales.R)

Input Controls

The Sales submenu item uses the following input controls for user interaction:

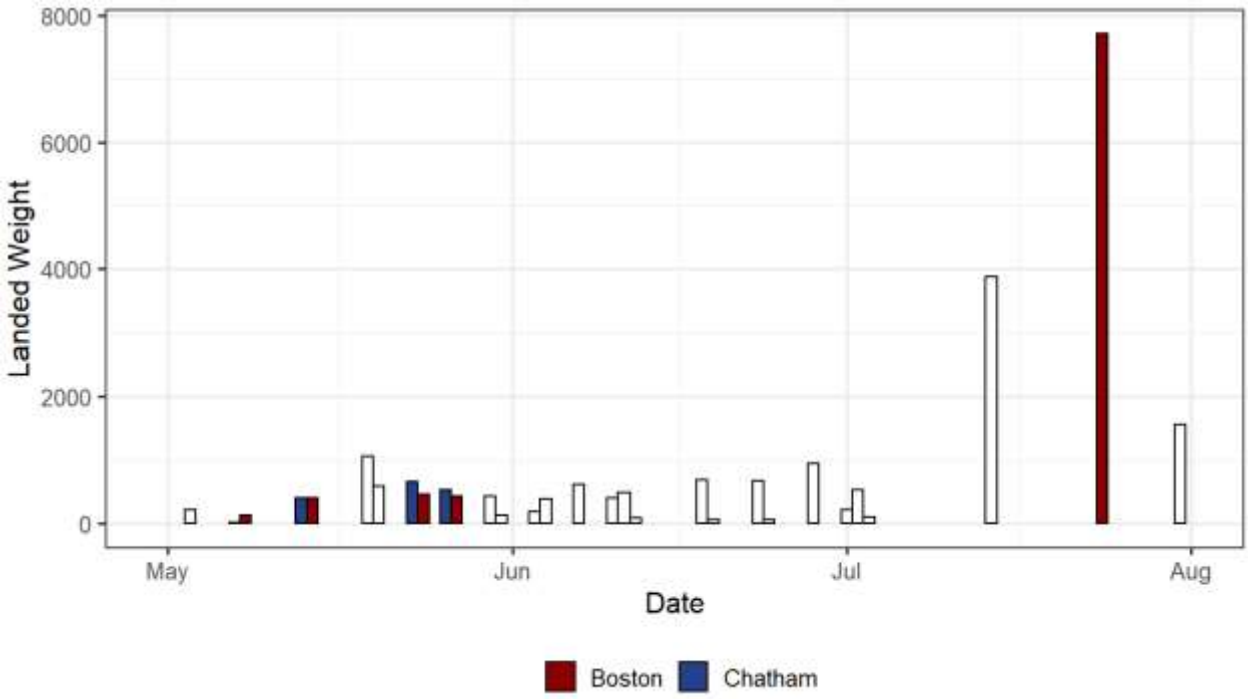
outputId	inputID	Description

sales_metric_input	sales_metric	Select one sales metric; Landed Weight, or Live Weight
sales_port_input	sales_port	Select one or more ports; populated using port.land values available in simm_()
sales_species_input	sales_species	Select one or more species; populated using species values available in simm_()
sales_evtr_input	sales_evtr	Select one or more eVTRs; populated using vtr values available in sales_data()
sales_fill_input	sales_fill	Select one variable used to define fill scale; Port or Species
sales_click	sales_click$x	Click on the sales_plot to select one bar

Reactive Data

A reactive working data set sales_data() is generated based on the reactive simm_() data. The data are grouped by vtr, date.sold, species, and port.land; for each group the total value of landed.weight and live.weight are calculated, respectively. When user input is provided by the sales_port or sales_species input controls, the appropriate filters are applied.

Reactive Figure



A reactive function sales_plot_prep() defines the process for rendering sales_plot() in two layers, allowing users to highlight specific eVTR values selected with the sales_evtr input control:

(1) Baseplot: Using sales_data(), an unfilled bar plot is generated with date.sold on the x axis and the sum of the metric selected by the user using the sales_metric input control (landed.weight or live.weight) on the y axis.

(2) Overlay: Using sales_data(), the sales_evtr input control is used to filter the data appropriately; if no eVTRs have been selected, no filter is applied. The value of the sales_fill input control determines the variables used to group the data: (a) if Port, the data are grouped by date.sold and port; (b) if Species, the data are grouped by date.sold and species. The total landed.weight and live.weight is calculated for each group. A bar plot is generated with date.sold on the x axis and the sum of the sales_metric input control metric (landed.weight or live.weight) on the y axis. The value of the sales_fill input control determines the fill colour of the bar plot, based on port.land or species, as defined in fillScalePort or fillScaleSpecies, respectively (see *Colour and Fill Scales*).

Clicking on the plot stores the associated x and y values, stored as sales_click$x, which is converted to a date and the evtr values of records in the sales_data() with matching dates are printed to the screen. If no data is available to generate effort_plot(), the renderPlot(effort_plot_prep()) function will display a "No data available" message.

Reactive Table

eVTR	Port	Date Sold	Species	Weight Sold (Landed)	Weight Sold (Live)
24385919082714	Chatham	2019-08-27	Spiny Dogfish	6000	6000
13930719082620	Gloucester	2019-08-27	Atlantic Mackerel	700	700
24385919082810	Chatham	2019-08-28	Spiny Dogfish	6000	6000
24385919082913	Chatham	2019-08-29	Spiny Dogfish	5634	5634

A reactive function sales_table_prep() defines the process for generating the sales_table() from the sales_data() data, which only includes modifying and capitalizing the names of the variables. Before rendering, the total weight reported by all eVTR records and the total weight reported by all dealer records are calculated and those values appended to the bottom of the table.

Download Buttons

The following download buttons are included:

- export_sales_data
  - Starts with: sales_data()
  - Modifications:
    - "eVTR" prefix added to evtr values to retain value as character string
    - variable names cleaned
  - Produces: compare_data.csv
- export_sales_plot
  - Saves sales_plot_prep() as sales_figure.png
- export_sales_table
  - Starts with: sales_table_prep()
  - Modifications:
    - "eVTR" prefix added to evtr values to retain value as character string
  - Produces: sales_table.csv

Comparison to eVTR Data (server-components/server-compare.R)

Input Controls

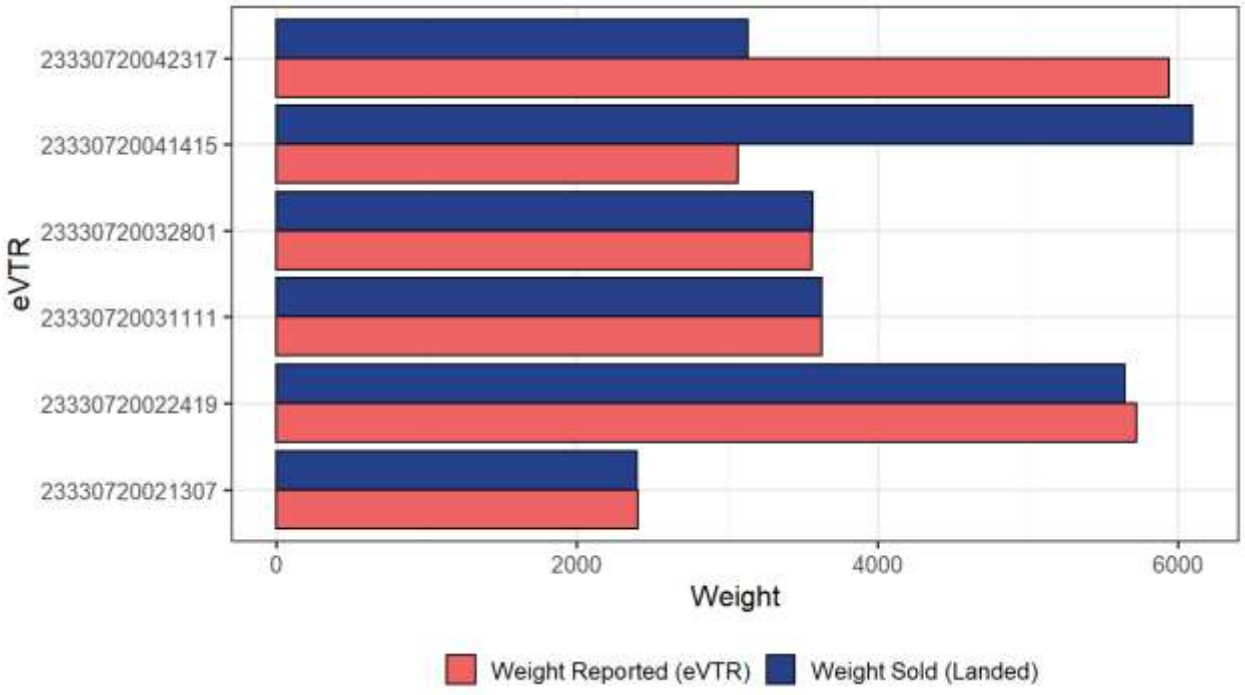
The Comparison to eVTR submenu item uses the following input controls for user interaction:

outputId	inputID	Description
compare_port_input	compare_port	Select one or more ports; populated using port.land values available in simm_() and evtr_()
compare_species_input	compare_species	Select one or more species; populated using species values available in simm_() and evtr_()
compare_match_input	compare_match	Select the set of variables to be used for matching eVTR data and dealer landings; eVTR or eVTR and Port
compare_table_res_input	compare_table_res	Select one resolution for summarising data; Overall, By Species, By Port, or By Port and Species

Reactive Data

A reactive working data set compare_data() is generated based on the reactive evtr_() and simm_() data. Unused variables are dropped and remaining variable names are updated to better reflect which data set they belong to: (a) the eVTR data includes the variables eVTR, Species, eVTR Date, and eVTR Port; (b) the dealer data includes the variables eVTR, Species, Dealer Date, and Dealer Port. Both data sets are grouped by these variables; for each group the total weight reported by the eVTR data and the total landed weight reported by the dealer data is calculated as eVTR Weight and Dealer Weight, respectively. The remaining summary tables are joined based on matching eVTR and Species values. When user input is provided by the compare_port or compare_species input controls, the appropriate filters are applied.

Reactive Figure



A reactive function compare_plot_prep() defines the process for rendering compare_plot(). If the value of the compare_match input control is eVTR and Port, a filter is applied to the data, keeping only records where both the port.land and evtr values in the eVTR and dealer data match. A bar plot is generated with total reported weight (Reported) on the x axis and evtr on the y axis. Two bars are drawn for each value of evtr; one representing the total weight reported by the eVTR data, and one representing the total weight reported by the dealer data. The fill

colours of these bars are defined by fillScaleMetric (see *Colour and Fill Scales*). If no data is available to generate compare_plot(), the renderPlot(compare_plot_prep()) function will display a "No matches have been found between the eVTR data and the dealer data" message.

Reactive Table

eVTR	Species	eVTR Port	Dealer Port	eVTR Weight	Dealer Weight
24385920042909	Atlantic Cod	Gloucester	Gloucester	140	140
24385920042909	Atlantic Halibut			0	0
24385920042909	Atlantic Wolffish			0	0
24385920042909	Cusk	Gloucester	Gloucester	286	572

A reactive function compare_table_prep() defines the process for generating the compare_table() from the compare_data() data. The value of the compare_table_res input control determines the variables used to group the data: (a) if By Species, the data are grouped by eVTR and Species; (b) if By Port, the data are grouped by eVTR, eVTR Port, and Dealer Port; (c) if By Port and Species, the data are grouped by eVTR, eVTR Port, Dealer Port, and Species; (d) if Overall, the data are grouped by eVTR. For each group, the total weight reported by the eVTR data and the total weight reported by the dealer data are calculated, and the table rows are arranged in order of descending eVTR.

Download Buttons

The following download buttons are included:

- export_compare_data
  - Starts with: compare_data()
  - Produces: compare_data.csv
- export_compare_plot
  - Saves compare_plot_prep() as compare_figure.png
- export_compare_table
  - Starts with: compare_table_prep()
  - Modifications:
    - "eVTR" prefix added to evtr values to retain value as character string
  - Produces: compare_table.csv

EM Data

Catch Reviews (server-components/server-catchreviews.R)

Input Controls

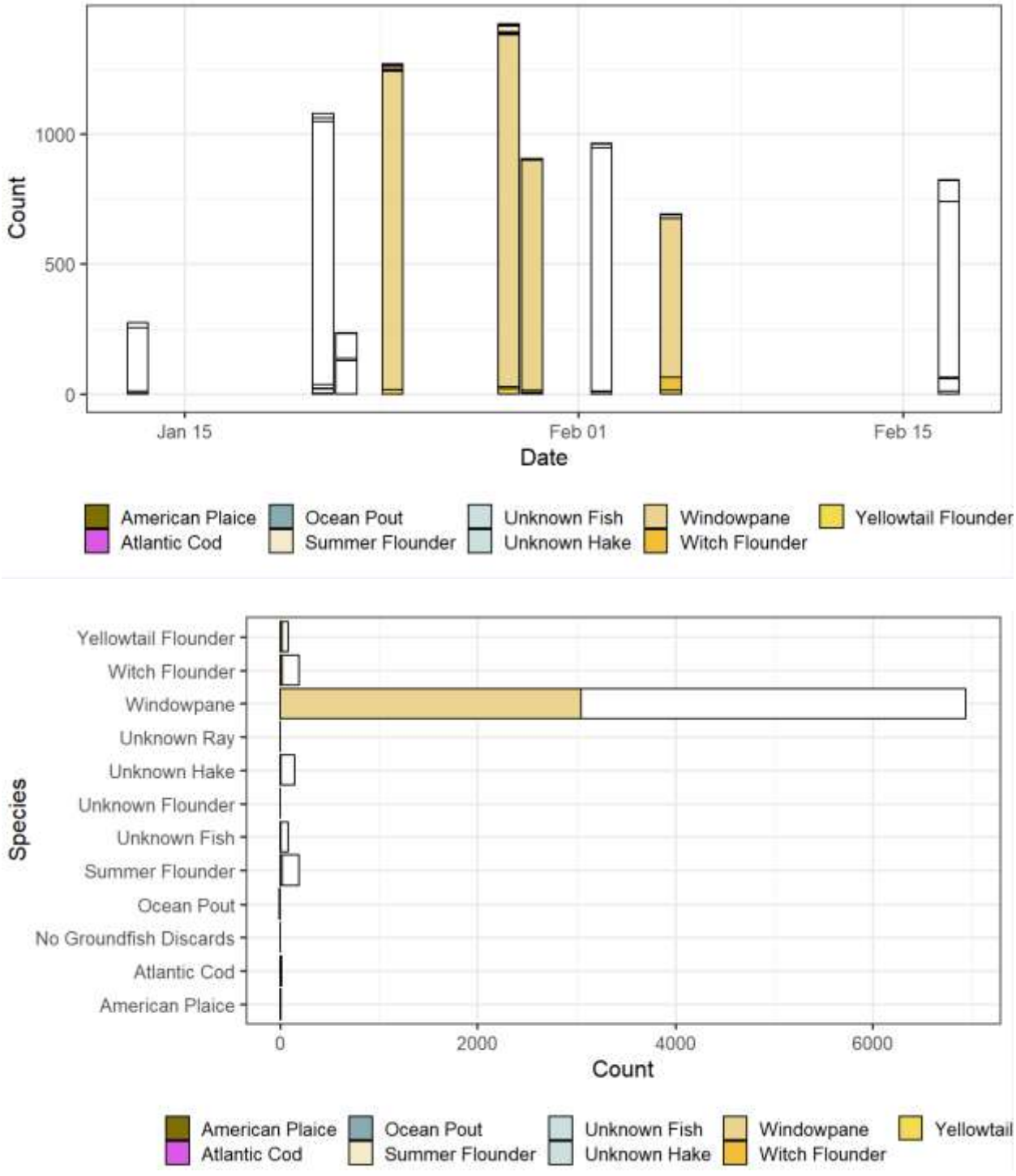
The Catch Reviews submenu item uses the following input controls for user interaction:

outputId	inputID	Description
catchreview_species_input	catchreview_species	Select one or more species; populated using species values available in em_()
catchreview_evtr_input	catchreview_evtr	Select one or more eVTRs; populated using vtr values available in catchreview_data()
catchreview_detail_input	catchreview_detail	Select one review metric; Individual Fish Reviewed, or Reviews
catchreview_plottype_input	catchreview_plottype	Select one view of the plot; by Date or Species
catchreview_click	catchreview_click$x catchreview_click$y	Click on the catchreview_plot to select one bar; x represents date in Date view, y represents species in Species view

Reactive Data

A reactive working data set catchreview_data() is generated based on the reactive em_() data, filtered to only include records where the category is Catch Event. The value of the catchreview_plottype input control determines the variables used to group the data: (a) if Date, the data are grouped by evtr, species, and start.date; (b) if Species, the data are grouped by evtr and species. For each group. a count variable is calculated based on the catchreview_detail input control: (a) if Individual Fish Reviewed is selected, count represents the total number of individuals included in catch events, found as the sum of quantity; (b) if Reviews is selected, count represents the number of catch events. When user input is provided by the catchreview_species input control, the appropriate filter is applied.

Reactive Figures



A reactive function `catchreview_plot_prep()` defines the process for rendering `catchreview_plot()` in two layers, allowing users to highlight specific eVTR values selected with the `catchreview_evtr` input control. The variables plotted on the x and y axes are dependent on

the value of the catchreview_plottype input control: (a) if Date is selected, start.date is plotted on the x axis, and count is plotted on the y axis; (b) if Species is selected, count is plotted on the x axis and species is plotted on the y axis. The value of the catchreview_plottype input control also determines the variables used to group the data: (a) if Date, the data are grouped by start.date and species; (b) if Species, the data are grouped by species. The total count is calculated for each group.

(1) Baseplot: Using catchreview_data(), an unfilled bar plot is generated following the conditions described above.
(2) Overlay: Using catchreview_data(), a bar plot is generated following the conditions described above, including an additional grouping level using evtr. The catchreview_evtr input control used to filter the data appropriately; if no eVTRs have been selected, no filter is applied. Once any unselected eVTRs have been removed, the data is aggregated based on the groupings describe above and the total count recalculated. The fill colours of these bars are defined by fillScaleSpecies (see *Colour and Fill Scales*).

Clicking on the plot stores the associated x and y values, stored as catchreview_click$x and catchreview_click$y. When catchreview_plottype is Date, the x value is converted to a date and the evtr values of records in the catchreview_data() with matching dates are printed to the screen. When catchreview_plottype is Species, the y value is rounded to the nearest integer and the corresponding value of species identified; the evtr values of records in the catchreview_data() with matching species are printed to the screen. If no data is available to generate catchreview_plot(), the renderPlot(catchreview_plot_prep()) function will display a "No data available" message.

Reactive Table

eVTR	Date Range	Species	Catch Reviews	Total Individuals
13930719090906	2019-09-09 to 09-10	Atlantic Cod	2	2
13930719090906	2019-09-09 to 09-10	Haddock	5	5
13930719090906	2019-09-09 to 09-10	Pollock	4	4
13930719090906	2019-09-09 to 09-10	Redfish	1	1

A reactive function catchreview_table_prep() defines the process for rendering the catchreview_table() from the em_() data, filtered to only include records where the category is

Catch Review. The data is grouped by evtr and species and the total number of individuals and the total number of catch review events in each group is calculated. Based on evtr, information about the minimum and maximum start.date associated with that trip are appended to the data. When user input is provided by the catchreview_species input control, the appropriate filter is applied. Finally, the variable names are updated before the table is rendered.

Download Buttons

The following download buttons are included:

- export_catchreview_data
  - Starts with: em_()
  - Modifications:
    - filter to only include records where category is Catch Review
    - "eVTR" prefix added to evtr values to retain value as character string
    - columns renamed; unnecessary columns dropped
  - Produces: catchreview_data.csv
- export_catchreview_plot
  - Saves catchreview_plot_prep() as catchreview_figure.png
- export_catchreview_table
  - Starts with: catchreview_table()
  - Modifications:
    - "eVTR" prefix added to evtr values to retain value as character string
  - Produces: catchreview_table.csv

Other Events (server-components/server-otherevents.R)

Input Controls

The Other Events submenu item uses the following input controls for user interaction:

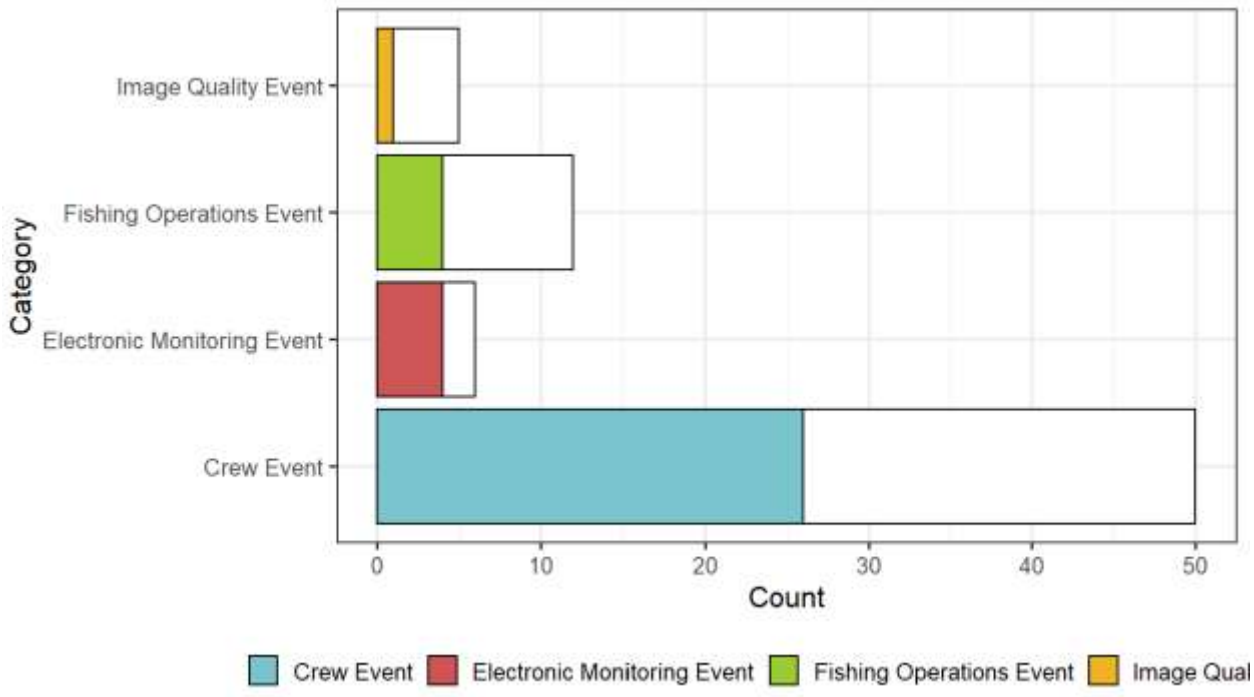
outputId	inputID	Description
otherem_evtr_input	otherem_evtr	Select one or more eVTRs; populated using vtr values available in otherem_data()
otherem_plotview1_input	otherem_plotview1	Select one view of the plot; by Date or Category

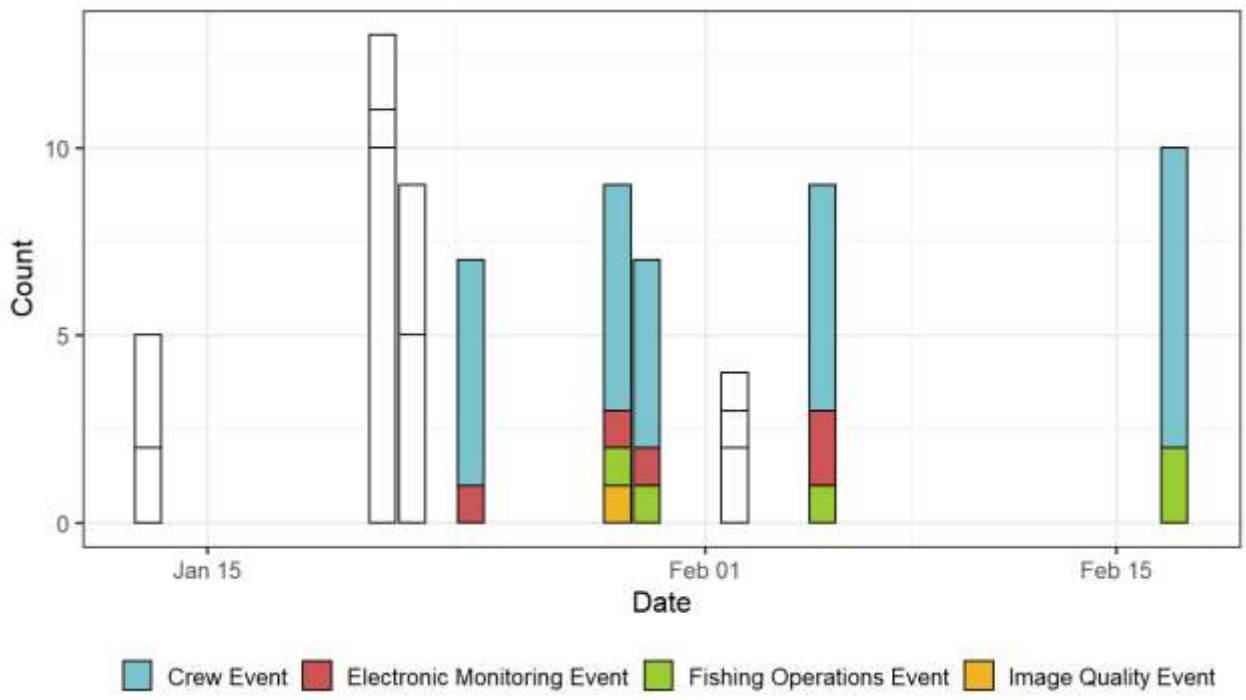
otherem_click1	otherem_click1$x otherem_click1$y	Click on the otherem_category_plot to select one bar; x represents date in Date view, y represents category in Category view
otherem_category_input	otherem_category	Select one category; populated using category values available in otherem_data()
otherem_plotview2_input	otherem_plotview2	Select one view of the plot; by Date or Subject
otherem_click2	otherem_click2$x otherem_click2$y	Click on the otherem_subject_plot to select one bar; x represents date in Date view, y represents category in Subject view

Reactive Data

A reactive working data set otherem_data() is generated based on the reactive em_() data, filtered to only include records where the category is not Catch Event or Optional Event and the subject is not High, Medium, or Low. These subject values refer to image quality and are not relevant herein.

Reactive Figures



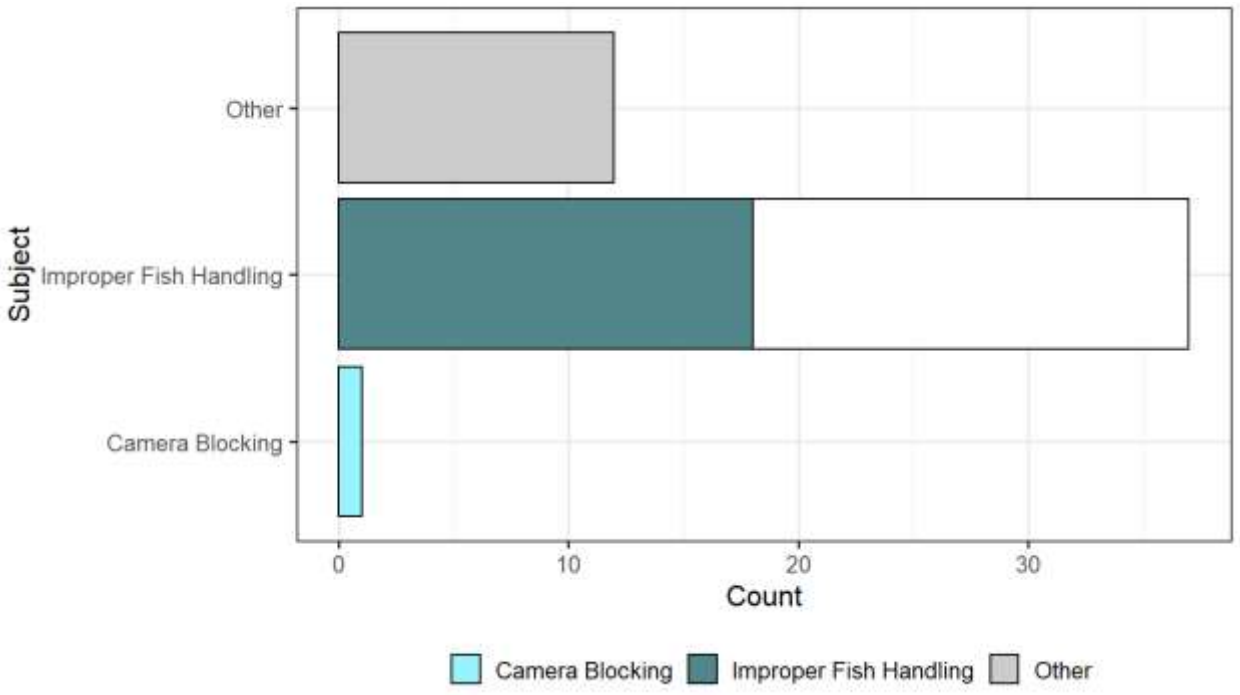


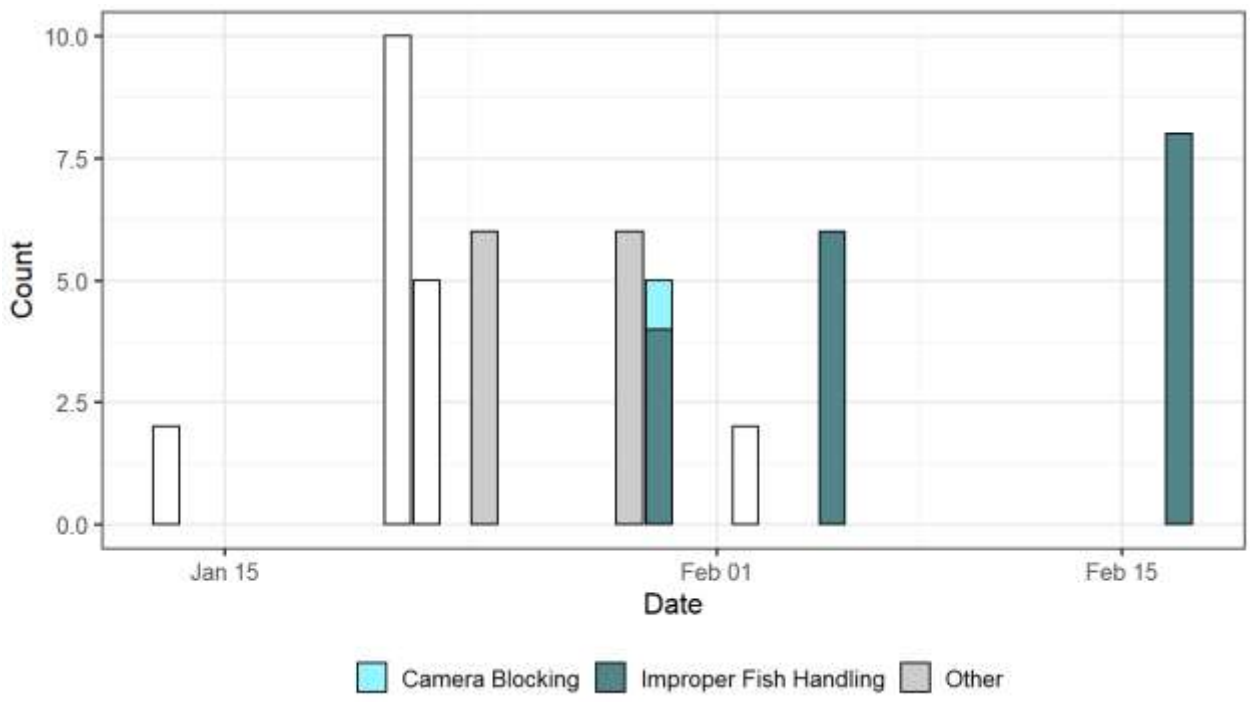
A reactive function otherem_plot_category_prep() defines the process for rendering otherem_category_plot() in two layers, allowing users to highlight specific eVTR values selected with the otherem_evtr input control. The variables plotted on the x and y axes are dependent on the value of the otherem_plotview1 input control: (a) if Date is selected, start.date is plotted on the x axis, and count is plotted on the y axis; (b) if Category is selected, count is plotted on the x axis and category is plotted on the y axis. The value of the otherem_plotview1 input control also determines the variables used to group the data: (a) if Date, the data are grouped by start.date and category; (b) if Category, the data are grouped by category. The total count is calculated for each group.

(1) Baseplot: Using otherem_data(), an unfilled bar plot is generated following the conditions described above.

(2) Overlay: Using otherem_data(), a bar plot is generated following the conditions described above, including evtr as an additional grouping level. The otherem_evtr input control used to filter the data appropriately; if no eVTRs have been selected, no filter is applied. Once any unselected eVTRs have been removed, the data is aggregated based on the groupings describe above and the total count recalculated. The fill colours of these bars are defined by fillScaleCategory (see *Colour and Fill Scales*).

Clicking on the plot stores the associated x and y values, stored as otherem_click1$x and otherem_click1$y. When otherem_plotview1 is Date, the x value is converted to a date and the evtr values of records in the otherem_data() with matching dates are printed to the screen. When otherem_plotview1 is Category, the y value is rounded to the nearest integer and the corresponding value of category identified; the evtr values of records in the otherem_data() with matching category are printed to the screen.





A reactive function otherem_plot_subject_prep() defines the process for rendering otherem_subject_plot() in two layers, allowing users to highlight specific eVTR values selected with the otherem_evtr input control. The data are filtered to only include records where the category is the same value selected by the otherem_category input control. The variables plotted on the x and y axes are dependent on the value of the otherem_plotview2 input control: (a) if Date is selected, start.date is plotted on the x axis, and count is plotted on the y axis; (b) if Subject is selected, count is plotted on the x axis and subject is plotted on the y axis. The value of the otherem_plotview2 input control also determines the variables used to group the data: (a) if Date, the data are grouped by start.date and subject; (b) if Subject, the data are grouped by subject. The total count is calculated for each group.

(1) Baseplot: Using otherem_data(), an unfilled bar plot is generated following the conditions described above.

(2) Overlay: Using otherem_data(), a bar plot is generated following the conditions described above, including evtr as an additional grouping level. The otherem_evtr input control used to filter the data appropriately; if no eVTRs have been selected, no filter is applied. Once any unselected eVTRs have been removed, the data is aggregated based on the groupings

describe above and the total count recalculated. The fill colours of these bars are defined by fillScaleSubject (see *Colour and Fill Scales*).

Clicking on the plot stores the associated x and y values, stored as otherem_click2$x and otherem_click2$y. When otherem_plotview2 is Date, the x value is converted to a date and the evtr values of records in the otherem_data() with matching dates are printed to the screen. When otherem_plotview2 is Subject, the y value is rounded to the nearest integer and the corresponding value of subject identified; the evtr values of records in the otherem_data() with matching subject are printed to the screen.

If no data is available to generate otherem_category_plot() or otherem_subject_plot(), the renderPlot(otherem_category_plot_prep()) and renderPlot(otherem_subject_plot_prep()) functions will display a "No data available" message.

Reactive Table

eVTR	Date Range	Crew Event	Electronic Monitoring Event	Fishing Operations Event	Image Quality Event	Vessel Event
15228020011312	2020-01-13	2			3	
15228020012112	2020-01-21 to 01-22	15	1	6		
15228020012411	2020-01-24	6	1			
15228020012909	2020-01-29	6	1	1	1	

A reactive function otherem_table_prep() defines the process for rendering the otherem_table(). Using the otherem_data(), a reference table date_info is generated, containing the minimum and maximum start.date associated with each evtr. The otherem_data() is grouped by evtr and category and the total number of events in each category is calculated. These results are converted to wide format and the date range information from date_info is added. Finally, the total number of events in each category is calculated and those values appended to the bottom of the table.

Download Buttons

The following download buttons are included:

- export_otherem_data
    - Starts with: otherem_data()
    - Modifications:

        - "eVTR" prefix added to evtr values to retain value as character string
        - date.sail and date.land variables from evtr data set appended based on matching evtr values
        - start.date renamed to date.em.start
        - columns reordered; unnecessary columns dropped
    - Produces: otherem_data.csv
- export_otherem_plot_category
    - Saves otherem_plot_category_prep() as otherem_category_figure.png
- export_otherem_plot_subject
    - Saves otherem_plot_subject_prep() as otherem_subject_figure.png
- export_otherem_table
    - Starts with: otherem_table()
    - Modifications:
        - "eVTR" prefix added to evtr values to retain value as character string
    - Produces: otherem_table.csv

Export Data (server-components/server-export.R)

Currently inactive, the intention of this tab is to provide users with a single button to download all of the datasets, plots, and tables offered in the data portal. It may be appropriate to consider a form of automated reporting to implement here rather than having users downloading a large number of individual files.

About the Data

Currently blank, the intention of this tab is to contain general text information about the scope and sources of the data included in the data portal. This can be created as a simple text file or as a markdown document to be sourced into the code as it becomes available.

User Testing and Feedback

Feedback: Existing Features

User testing, formal questionnaires, and informal discussions with users provided valuable insight that influenced the iterative development of the data portal prototype. A key piece of feedback from one user was that while they found the data portal simple and easy to use, they cautioned against adding too many options or interactive components for fear that it could quickly become overwhelming or confusing. Two users had the opportunity to review the data portal in more depth, with their own real world data incorporated into the existing data portal template, where possible (Appendix F). This effort provided specific, high-quality feedback that should be carefully considered in future iterations of this tool.

Data Sharing

An initial focus of this project was to scope out an approach for sharing data among fishermen and/or sector managers. However, after initial conversations with users, it was clear that the was not wide interest in sharing data and the overall user perception was that it would be difficult to manage privacy concerns and ultimately was not particularly useful. A survey among industry stakeholders reported 69% responded that they would not want to share their data and only 54% said they would be interested in seeing data from others. In the same survey, 69% of respondents said that they do not have additional data that they would be interested in integrating into the data portal. the nature of the data portal prototype changes to a working product and users can start integrating for their own purposes, this may be an important topic to revisit. In terms of data accessibility broadly, sector managers were specifically excited about the potential for individual fishermen to be able to easily access and explore their own data without the need for managers to facilitate this process.

Data Resolution

The broad temporal scale of the data led to challenges in generating readable visualizations and summary statistics and interpreting outputs. To combat this, a single date range input control allows users to define a time range globally (i.e. across all tabs) on the sidebar. A reasonable default time frame of three months was selected. However, because the data portal prototype does not access real time data, the current iteration of this tool simply defaults to the most recent three months that eVTR data are available for the selected vessel. This change improved the overall performance of the data portal by ensuring that the visualizations and tables were not immediately overwhelming to users while providing them with the option of modifying the date range being examined.

Flexibility and Utility

Broadly, users agreed that the data portal would be useful if made available with their own data, and the ease of use and accessibility was improved over multiple iterations. Users noted that the

portal would be most useful if it incorporated real-time or near-real-time data that could be explored and acted upon within the fishing season instead of being limited to data from previous seasons. While accomplishing this would require more coordination between those submitting data, those storing and managing the data, and those managing the data portal, it is likely a worthwhile endeavor that would significantly increase the long-term engagement of users.

Feedback from an early survey of industry stakeholders highlighted several key areas of high value, and iterations of the data portal subsequently incorporated these perspectives.

- Comparing eVTR data to Dealer data (54% indicated “most useful”)
- A map of catch data (54% indicated “most useful”)
- Comparing fishing data to water temperature; both surface (54% indicated “most useful”) and bottom (62% indicated “most useful”)

In the same survey, elements that were rated as “less useful” included depth data, days at sea, and protected species interactions.

In early iterations, users selected which eVTR records to highlight by selecting the corresponding row from the tables found on each tab. However, this process became arduous when many eVTR records were available, or when the resolution of the table was detailed enough to require multiple rows for each eVTR value. Instead, it was suggested that users are provided with a separate optional input control to manually select eVTRs of interest from a drop down menu. Providing users with both options for testing received mixed feedback about preferred methods. To better help users identify eVTRs of interest, visualizations were created with interactive click options that resulted in eVTRs being printed to the screen. While users generally provided positive feedback about this feature, some asked if additional details about the clicked points could also be shared. This has not been implemented but may be a good addition for future iterations of this product. Similarly, features were added to the catch, effort, and environment tab maps, allowing users to click on points and access the underlying data from that location / record.

Aesthetics

Plot and map aesthetics were iteratively modified many times during the life of the project, with in-depth discussions around colour, layout, and labelling. One suggestion that came up on multiple occasions was to restructure some of the plots (i.e. catch vs water temperature) to incorporate two y axes so that relative values can be more easily viewed and interpreted. The ability to hover over a particular point or bar on a plot and have a popup display the actual value would be helpful for users but was not a high enough priority to invest time in at this point. A significant amount of time was spent selecting colours to represent the various levels of categorical values, but these efforts generally received positive feedback from users, who appreciated that the method of aggregating species represented by a single colour agreed with how they might choose to group species from a fishing perspective.

Clarity

In response to both internal and external feedback, the overall clarity of the data portal was improved using the following approaches:

- Standardizing instructions on each tab
- Standardizing font sizes, formatting, plot aesthetics, terminology, and wording across all elements
- Modifying heading to be dynamic and include vessel name on each tab
- Clearly labelling plot axes and providing interactive click options to help users focus in on specific values in complex visualizations.

Environmental Data

The environmental data tab underwent a significant amount of redesigning, restructuring, and rethinking. The data portal prototype certainly does not include all the ways in which such data might be utilized and is a source of ongoing discussion (see F*eedback: Wishlist > Environmental Data*, below). Some features that were suggested and have been incorporated include:

- Aggregating data based on reasonable depth bins so users can see the full range of available data from multiple stations.
- Incorporating several environmental variables, such as water temperature and salinity, providing users with the option to cycle through these metrics when examining catch comparisons.
- Including a link directly to the source of the environmental data based on the station selected by the user and ensuring that this link opens in a new browser window.
- Adding options for how the aggregated environmental data should be summarized before comparisons with catch occur (i.e. minimum, average, maximum values of the environmental parameter).
- Allowing users to define the resolution at which environmental data should be summarise before comparisons with catch occur (i.e. daily, weekly, monthly); feedback from users indicates that aggregating environmental data by day is preferred and/or appropriate.

Feedback: Wishlist

Users reported that they would be interested in incorporating user-specific custom settings that would allow them to more quickly explore the data that is of most interest to them. At the prototype stage this is not something that has been explored, but future iterations of this product that include user-specific profiles or accounts could have these features added. A general note about future iterations of the data portal: elements that highlight important values should be prioritized in the context of streamlining the overall user experience (i.e. the most important information should also be the easiest information to access). To identify which elements these are, more user testing may be required, with directed questioning about actionable information found within the data portal.

Data Types

Surveys of potential users were used to scope out the relative importance of data types and features that could potentially be included in future iterations of the data portal. Users reported high interest levels in incorporating additional environmental data metrics, including expanding water temperature to incorporate bottom temperature, lunar cycles, currents, tides, and moderate interest levels in incorporating quota availability and auction pricing, and other market information. Of these, the environmental data is more readily available but often at a resolution that is too detailed for efficient use within the shiny platform. In initial discussion, there is uncertainty about where standardized quota and auction data could be sourced from in a consistent, automated (or at least semi-automated) way. If the data are made available, it could be relatively simple to expand the sidebar menu of the data portal to include a modularized tab focused on this context. More robust measures of effort and catch-per-unit-effort would improve the usefulness of the tool but will require more in-depth discussions and strategies for how to calculate relative and accurate measures of effort given the complexity of the data; it will almost certainly require that the data are considered at a higher resolution to account for variable gear, fishing areas, and trip characteristics.

A survey of industry stakeholders also highlighted the value of making electronic monitoring audit reports available through the data portal. The primary challenge here will be in determining how that information is stored and the most effective means of sharing general and detailed information with users.

Data Resolution

In the context of big-picture approaches used in the data portal, we specifically asked for feedback from users about the resolution of the trip data and whether delivering information at a subtrip level was an appropriate baseline. One user replied, “Leave the multiple subtrips per trip (so multiple lines of data per trip) since that makes sense for fishermen if they switched stat area or gear.” It would be valuable to be able to have the ability to scale the data from the individual haul to aggregated trip level; however, this will require a significant amount of work to decide how each visualization and analysis should react to changes in the resolution of the data and requires a more in-depth look at the data.

Similar feedback noted the value of integrating more detailed gear options. For instance, including metrics like gear size and configuration (i.e. mesh size, square or diamond cod end, etc.) rather than overall gear type. This will no doubt have its challenges as there are so many configurations that are being used and no standardized way of reporting them. Further discussions with fishermen may facilitate a better understanding of how configuration classifications could be recoded into the existing data structure.

Users also reported that a useful feature of a future iteration of the data portal would be the ability to quickly select a single date and quickly access broad summary statistics without needing to navigate through the multiple tabs. A *Quick View* tab could be added containing key

values in simple value boxes or an option to download a one page report with these most valuable metrics.

Aesthetics

While improvements in colour and fill aesthetics were certainly made across iterations, the issue of having so many levels in some categorical variables (i.e. species) remains an issue. One suggestion is to use a combination of fill colour and fill pattern to better communicate these levels. While the current visualization package does not support fill patterns, other open source package extensions do, so this may be a useful approach in the future. This will also help address concerns about the accessibility of visualizations to those with various types of colour blindness.

Environmental Data

Existing environmental data has the highest potential for more robust integration into the data portal. For the demonstration purpose of the data portal prototype, several small sample datasets were manually downloaded from the Environmental Research Division's Data Access Program (ERDDAP) via their online data portal (https://coastwatch.pfeg.noaa.gov/erddap/index.html). A working version of the data portal should include automated downloads or access to the data via an API. Considerations will need to be made about how much flexibility the user has in selecting which data to access and download in real time, the resolution of the data, and which, if any, data should be held in memory. Because such data may be very large, careful planning will be required to prevent the data portal and R from slowing down or crashing. One method may be to update a standardized environmental database at a pre-determined frequency (i.e. daily) and build the data portal capabilities to match the resolution and content of that data. Storing it in a database format facilitates the use of R packages such as `dbplyr` that allows data to be accessed and queried without needing to import all of it into the working environment, thus reducing memory pressure.

In discussions with users with various perspectives (sector managers, fishermen, resource managers) bottom temperature was often discussed, with the eMOLT data having high potential for more robust analyses. Fishermen who are involved in collecting eMOLT data were particularly interested in being able to easily access and use this information. There are challenges with managing this data, which is both high resolution and spatially and temporally sparse. More discussions are needed to determine the best way to aggregate and simplify this data; the bottom temperature data indirectly associated with a fishing event could be based on fishing location in the context of distance (i.e. distance from fishing location), matching depth contours (i.e. similar characteristics as fishing location), or both (i.e. similar characteristics as fishing location, within a particular geographic region). Temporally, both the resolution of the data (i.e. summarise by hour, day, week) and the generalization of the data (i.e. match data within hours, days, or weeks of fishing event) need to be considered.

Other potential environmental metrics were suggested in surveys and in directed feedback:

- Sea surface temperature
- Other sources of bottom temperature

- Currents
- Lunar / tidal cycles
- Wave height
- Wind

Analyses

Comparisons

The comparison of eVTR catch records to dealer sales records demonstrated a high degree of mismatching between the two sources. It is recommended that more detailed exploration be done to determine how the resolution of trips could be changed to better reflect units of sale. As one user explained, “When you are doing the dealer to evtr comparison, you need to aggregate all the subtrips to match the trip evtr number. Otherwise, you are comparing entire trip landings from dealer to just one subtrip.” Modifications were made in response to this point, but it is still unclear how well this comparison reflects actual discrepancies in the reported landings. Among users the usefulness of this comparison, in general, was unclear. However, some users noted that a more useful comparison would be between the reported eVTR discards to the electronic monitoring discard records. This is certainly possible and could be added as a separate module to the EM Feedback menu item. However, it was noted that this may be difficult to accomplish or interpret because the electronic monitoring data is processed before the actual audit occurs.

Integrating Real Data

Two instances of the data portal were created to handle real world data as a means of gathering user feedback. Unfortunately, due to a lack of standardization in the data sets shared with us from users, there were significant challenges in integrating the data with the existing data portal structure, some of which were not possible to overcome. This exercise demonstrated the added value of creating the data portal with standardized input data structures already well defined and enforced. Some issues such as deviations in variable names were manually fixed or customized code developed to accomplish these tasks. However, these processes were time consuming and created some uncertainty around how the data should be interpreted.

Using these efforts as a case study, two major sources of data issues were identified, briefly described below:

(1) Inconsistencies in the formatting and organization of the data files

The data files submitted by the two test users for integration into the data portal were inconsistent both in comparison to each other and, in the case of multi-year data, in comparison across years. The naming conventions of the files and the saved file format varied, as did the way that data were aggregated or split up into multiple data files across years. Within the data files, significant differences were noted in which variables were available. Beyond inconsistency in variable names, there were examples of data sets missing key variables required for integration into the data portal; for instance, an electronic

monitoring (EM) data set that did not contain information about the subject of each event. In cases such as this one there was no possible solution, so the user was not able to test that portion of the data portal.

(2) Inconsistencies in the values of categorial variables

After some manual processing that improved the structure of the data files enough to integrate into some or most of the data portal, it was noted that certain categorical variables had levels outside of the scope of those provided in the dummy data. Upon further examination, there were some instances that we could manually recode the values to fit within the expected scope; for instance, some electronic monitoring (EM) data included values of category that could be logically matched to known category levels (i.e. “Crew Event” vs “Crew”). This highlights the need for additional data cleaning or more standardized data collection/storage methods prior to data integration to the data portal.

Moving forward, these two issues should be more fully scoped out and solutions for data consistency prioritized and established.

Tables and Figures

Table 1. Hexadecimal colour codes based on species groupings.

Hexadecimal Code	Species	Hexadecimal Code	Species
#000000	Winter Flounder	#CCE1DF	Albacore
#10157D	Anchovies		Alewife
	Goosefish		American Shad
#105BEB	Bluefish		Anchovies
#2A6514	Longfin Squid		Atlantic Herring
	Shortfin Squid		Blueback Herring
#379B90	Black Sea Bass		Bonito
#4D8F34	Sculpin		Cetacean
#4D96E4	Atlantic Mackerel		Clearnose Skate
	Whiting		John Dory
#4DE5D4	Bluefin Tuna		Menhaden
	Sea Scallop		No Groundfish Discards
#506291	Pollock		Non-Fish
#5F5F67	Smooth Dogfish		Pinniped
	Spiny Dogfish		Rock Crab
#6355B8	Unknown Ray		Sea Raven
	Unknown Skate		Spider Crab
	Winter Skate		Unknown Bird
#6B1748	Red Hake		Unknown Fish
#71BACC	Haddock		Unknown Flounder
#75DC4E	Tautog		Unknown Hake
	Weakfish		Unknown Roundfish
#7F6E05	American Plaice	#D40421	Jonah Crab
	Fourspot Flounder	#DA5AE6	Atlantic Cod
#822004	Redfish	#E44D30	American Lobster
#8BABB3	Atlantic Wolffish	#EBD48F	Windowpane
	Ocean Pout	#EFD4DD	Butterfish
#95AAE0	White Hake		Cunner
#984CEB	Barndoor Skate		Cusk
#A76690	Silver Hake		Golden Tilefish
#B2AADE	Thorny Skate	#F27C3A	Atlantic Halibut
#B7C5D5	Scup	#F2BF3A	Witch Flounder
#C3C6C6	Atlantic Sturgeon	#F2DC54	Yellowtail Flounder
	Basking Shark	#F6EBCB	Summer Flounder
	Bigeye Thresher Shark		
	Porbeagle		
	Sand Tiger Shark		
	Thresher Shark		
	Tiger Shark		

	Unknown Shark

Table 2. Electronic monitoring (EM) categories and associated subjects.

Category	Subject
Catch Review	Catchreview
Crew Event	Camera Blocking
	Camera Dirty
	Discarding Out Of Camera View
	Improper Fish Handling
	Lump Discarding
	Other
Electronic Monitoring Event	Camera Failure
	Measuring Surface Visibility
	Other System Issues
	Video Gaps
Fishing Operations Event	Gear Conflict
	Mechanical Failure
	Other
	Other Gear Issues
Image Quality Event	High
	Low
	Medium
	Unusable
Optional Event	Optional
Vessel Event	Haulbegin
	Haulend
	Slipped or Tripped Bag
	Sortend

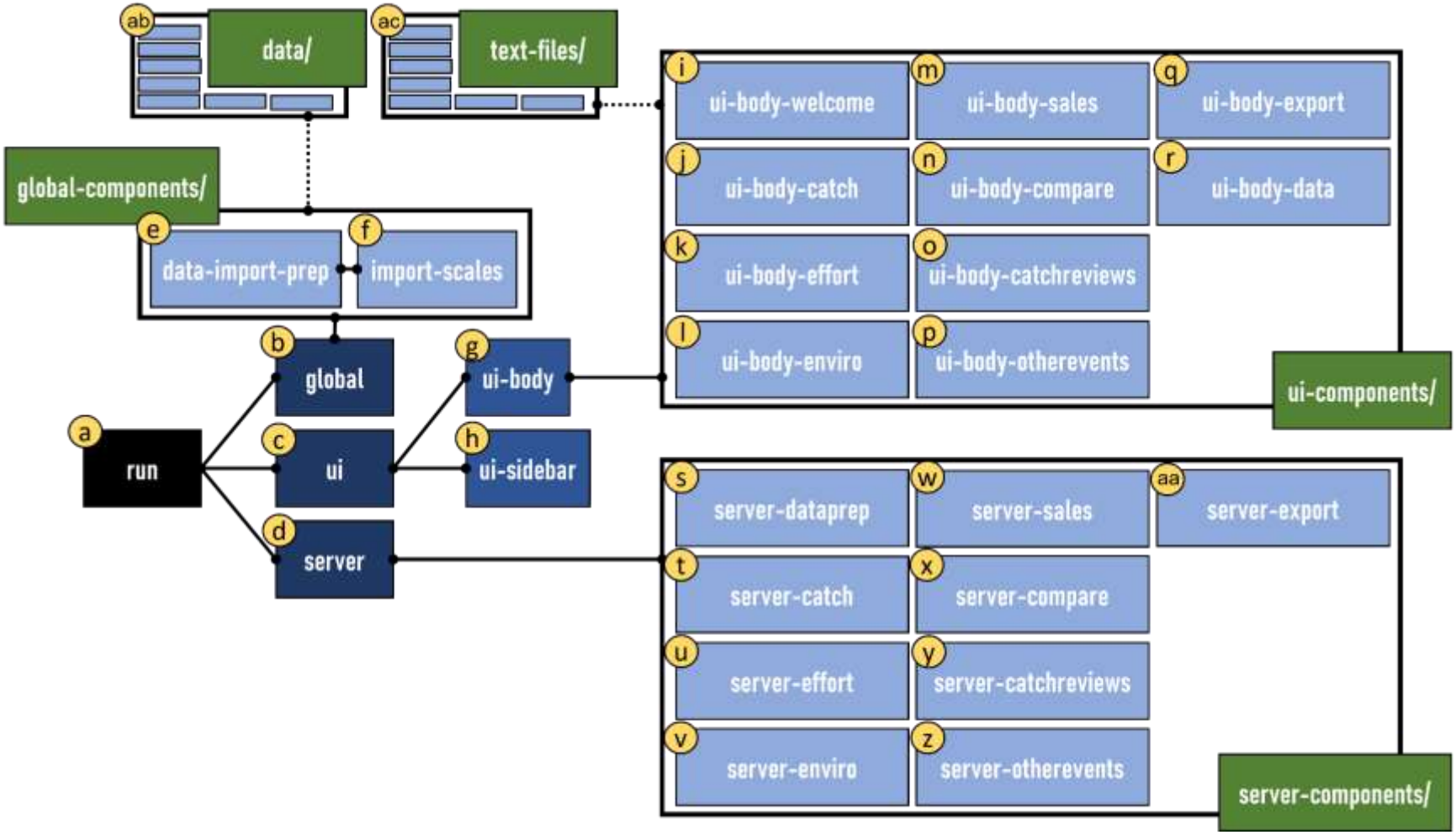


Figure 1. Relationship of data portal components, including data files, text files, and R scripts.

Appendix A. Structure and summary of raw dummy environmental monitoring (EM) data.

File Name Example of an FY19EM Summary File
File Type Excel (.xlsx)
Observations 30342
Variables 18

Variable Name	Data Class	Description
ID	double*	*unique values:* 17663 *missing values:* 0 *format:* 4 to 6 digits
Vessel	character	*unique values:* 9 *missing values:* 0 *levels:* [1] V1 [2] V10 [3] V12 [4] V15 [5] V2 [6] V3 [7] V5 [8] V7 [9] V9
Haul No	double	*range:* 0 – 17 *missing values:* 0
eVTR	double*	*unique values:* 173 *missing values:* 0 *format:* 12 to 14 digits
Start Timestamp	character+	*range:* 1 May 2019 10:01:12 – 29 Apr 2020 16:48:27 *unique values:* 29480 *missing values:* 0 *format:* 'yyyy-mm-dd hh:mm:ss EDT'
End Timestamp	logical	*missing values:* 30342 (100%)
Retained	character	*unique values:* 2 *missing values:* 0 *levels:* [1] false [2] true
Species	character	*unique values:* 57 *missing values:* 5171 (17%) *levels:* [1] American Plaice(American plaice) [2] American Plaice(FLDAB) [3] Atlantic Cod(Atlantic cod) [4] Atlantic Cod(COD) [5] Atlantic Halibut(Atlantic halibut) [6] Atlantic Halibut(HAL) [7] Atlantic Wolffish(Atlantic wolffish) [8] Atlantic Wolffish(CAT) [9] Bird(BRD)

		[10] Cetacean(CETA) [11] Fish unknown [12] Fish unknown(FSHNK) [13] Fish unknown(null) [14] Groundfish Flounder Unknown [15] Groundfish Flounder Unknown(GFFLDR) [16] Groundfish Flounder Unknown(null) [17] Groundfish Round Fish Unknown [18] Groundfish Round Fish Unknown(GFRD) [19] Groundfish Round Fish Unknown(null) [20] Haddock(HADD) [21] Haddock(Haddock) [22] Hake, NK [23] Hake, NK(HAKNS) [24] Hake, NK(null) [25] Marine Mammal(null) [26] No Groundfish Discards(NGD) [27] No Groundfish Discards(null) [28] Non-fish(NONFSH) [29] Non-fish(null) [30] Non-Groundfish(NGF) [31] Non-Groundfish(null) [32] Ocean pout(Ocean pout) [33] Ocean pout(POUT) [34] Pinniped(null) [35] Pinniped(PINNI) [36] Pollock(Atlantic pollock) [37] Pollock(POLL) [38] Ray(null) [39] Ray(RAY) [40] Redfish NK(Acadian redfish) [41] Redfish NK(RED) [42] Shark(null) [43] Shark(SHNS) [44] Sturgeon(STNS) [45] Summer Flounder(FLUKE) [46] Summer Flounder(Summer flounder) [47] White hake(WHAK) [48] White hake(White hake) [49] Windowpane flounder(FLSD) [50] Windowpane flounder(Windowpane flounder) [51] Winter Flounder(FLBB) [52] Winter Flounder(Winter flounder) [53] Witch Flounder(FLGS) [54] Witch Flounder(Witch flounder) [55] Yellowtail flounder(FLYT) [56] Yellowtail flounder(Yellowtail flounder) [57] NA
Location	character	*unique values:* 15111 *missing values:* 0 *format:* 'latitude, longitude'
Weight Estimated by	character	*unique values:* 7 *missing values:* 5171 (17%) *levels:* [1] LENGTH

		[2] SCALE [3] SUB-SAMPLE [4] UNDETERMINED [5] VISUAL ESTIMATE [6] VOLUMETRIC [7] NA
Discard Condition	character	*unique values:* 4 *missing values:* 5171 (17%) *levels:* [1] INTACT [2] LUMF [3] NOT BROUGHT ONBOARD [4] NA
Length	double	*range:* 1 - 104 *missing values:* 13442 (44%)
Estimated weight	double	*range:* 0.0007 – 50.0000 *missing values:* 10089 (33%)
Quantity	double	*range:* 1 - 486 *missing values:* 5173 (17%)
INC life status	character	*unique values:* 4 *missing values:* 30298 (> 99%) *levels:* [1] Alive [2] Dead_fresh [3] Dead_severely [4] NA
Category	character	*unique values:* 7 *missing values:* 0 *levels:* [1] Catch Review [2] Crew Event [3] Electronic Monitoring Event [4] Fishing Operations Event [5] Image Quality Event [6] Optional Event [7] Vessel Event
Subject	character	*unique values:* 23 *missing values:* 0 *levels:* [1] Camera Blocking [2] Camera Dirty [3] Camera Failure [4] Catchreview [5] Discarding Out Of Camera View [6] Gear Conflict [7] Haulbegin [8] Haulend [9] High [10] Improper Fish Handling [11] Low [12] Lump Discarding [13] Measuring Surface Visibility

		[14] Mechanical Failure [15] Medium [16] Optional [17] Other [18] Other Gear Issues [19] Other System Issues [20] Slipped Or Tripped Bag [21] Sortend [22] Unusable [23] Video Gaps
Comments	character	

[*] considered categorial
[+] considered datetime

Appendix B. Structure and summary of raw dummy electronic vessel trip reporting (eVTR) data.

File Name FY2019-eVTRdata-GARFO
File Type comma separated values (.csv)
Observations 13056
Variables 56

Variable Name	Data Class	Description
DOCID	double*	*unique values:* 1014 *missing values:* 0 *format:* 7 to 14 digits
NRPAGES	integer	*range:* 1 – 19 *missing values:* 0
DATE_SAIL	character+	*range:* 29 Apr 2019 23:58:00 – 29 Apr 2020 03:59:00 *unique values:* 987 *missing values:* 0 *format:* 'mm/dd/yyyy hh:mm'
DATE_LAND	character+	*range:* 1 May 2019 09:00:00 – 30 Apr 2020 10:21:00 *unique values:* 992 *missing values:* 0 *format:* 'mm/dd/yyyy hh:mm'
TRIPCATG	integer	*value:* 1 *missing values:* 0
CREW	integer	*range:* 1 – 5 *missing values:* 0
NANGLERS	logical	*missing values:* 13056 (100%)
OPERATOR_NUM	integer*	*unique values:* 15 *missing values:* 0 *format:* 8 digits
OPERATOR_NAME	character	*unique values:* 9 *missing values:* 0 *levels:* [1] O1 [2] O1_ [3] O10 [4] O12 [5] O15 [6] O2 [7] O3 [8] O7 [9] O8
DATE_SIGNED	character+	*range:* 2 Jan 2019 – 4 Oct 2020 *unique values:* 282 *missing values:* 0 *format:* 'dd-monthabb-yy'
FISHED	integer	*value:* 0

		missing values: 0
VESSEL_PERMIT_NUM	integer*	*unique values:* 12 *missing values:* 0 *format:* 6 digits
SERIAL_NUM	double*	*unique values:* 1021 *missing values:* 0 *format:* 6 digits
TRIP_ACTIVITY_TYPE	integer*	*unique values:* 4 *missing values:* 1704 (13%) *levels:* [1] 0 [2] 2 [3] 3 [4] NA
IMGID	double	*range:* 5.104e+06 – 3.110e+15 *missing values:* 0
PAGENO	integer	*range:* 0 – 15 *missing values:* 0
DATE_RECEIVED	character+	*range:* 1 May 2019 – 4 Oct 2020 *unique values:* 188 *missing values:* 0 *format:* 'dd-monthabb-yy'
GEARCODE	character	*unique values:* 7 *missing values:* 3^ (< 1 %) *levels:* [1] [2] GNS [3] HND [4] LLB [5] OTF [6] OTH [7] PTL
GEARQTY	integer	*range:* 1 – 2100 *missing values:* 5 (< 1%)
GEARSIZE	double	*range:* 0 – 800 *missing values:* 255 (2%)
AREA	integer	*range:* 121 – 611 *missing values:* 5 (< 1%)
DEPTH	integer	*range:* 0 – 410 *missing values:* 5 (< 1%)
LAT_DEGREE	integer	*range:* 40 – 42 *missing values:* 164 (1%)
LAT_MINUTE	integer	*range:* 0 – 59 *missing values:* 164 (1%)
LAT_SECOND	integer	*range:* 0 – 61 *missing values:* 3593 (28%)
LON_DEGREE	integer	*range:* -71 – 71 *missing values:* 164 (1%)

LON_MINUTE	integer	*range:* 0 – 59 *missing values:* 164 (1%)
LON_SECOND	integer	*range:* 0 – 61 *missing values:* 3593 (28%)
LORAN1	integer	*range:* 13612 – 13860 *missing values:* 12985 (> 99%)
LORAN2	integer	*range:* 25814 – 44414 *missing values:* 12985 (> 99%)
CLATDEG	integer	*range:* 40 – 42 *missing values:* 6558 (50%)
CLATMIN	integer	*range:* 0 – 59 *missing values:* 6558 (50%)
CLATSEC	integer	*range:* 0 – 60 *missing values:* 6558 (50%)
CLONDEG	integer	*range:* 69 – 71 *missing values:* 6558 (50%)
CLONMIN	integer	*range:* 0 – 59 *missing values:* 6558 (50%)
CLONSEC	integer	*range:* 0 – 60 *missing values:* 6558 (50%)
CLORAN1	logical	*missing values:* 13056 (100%)
CLORAN2	logical	*missing values:* 13056 (100%)
NTOWS	integer	*range:* 0 – 30 *missing values:* 5 (< 1%)
TOWHRS	integer	*range:* 0 – 792 *missing values:* 2 (< 1%)
TOWMIN	integer	*range:* 0 – 60 *missing values:* 2 (< 1%)
DATETIME_HAUL_START	character[+]	*range:* 30 Apr 2019 20:25:00 – 29 Apr 2020 23:57:00 *unique values:* 521 *missing values:* 8511[^] (65%) *format:* 'mm/dd/yyyy hh:mm'
DATETIME_HAUL_END	character[+]	*range:* 30 Apr 2019 21:26:00 – 30 Apr 2020 01:00:00 *unique values:* 521 *missing values:* 8511[^] (65%) *format:* 'mm/dd/yyyy hh:mm'
CAREA	integer	*range:* 121 – 611 *missing values:* 4655 (36%)
MESH	double	*range:* 0 – 12 *missing values:* 205 (1%)
CATCH_ID	double[*]	*unique values:* 6631 *missing values:* 0 *format:* 8 to 18 digits
SPECIES_ID	character	*unique values:* 83 *missing values:* 0 *levels:*

		[1] ALWF [2] BFT [3] BLU [4] BON [5] BSB [6] BUT [7] CAT [8] CATN [9] COD [10] CRHS [11] CRJ [12] CRNS [13] CRRK [14] CRSP [15] CUN [16] CUSK [17] DCP [18] DGCH [19] DGSM [20] DGSP [21] FLBB [22] FLDAB [23] FLFSP [24] FLGS [25] FLSD [26] FLUKE [27] FLYT [28] HADD [29] HAL [30] HERR [31] HRBB [32] ILX [33] JDO [34] LOB [35] LOL [36] MACK [37] MEN [38] MONK [39] MONKH [40] MONKL [41] MONKT [42] NC [43] OFF [44] POLL [45] POUT [46] RAY [47] RED [48] RHAK [49] SCAL [50] SCALS [51] SCUL [52] SCUP [53] SHAD [54] SHAK [55] SHBA [56] SHMNS [57] SHPB [58] SHST [59] SHTB [60] SHTH [61] SHTI [62] SKATE [63] SKATW [64] SKBARN [65] SKBARNW [66] SKCL [67] SKL [68] SKLWIN [69] SKLWINW [70] SKTHOR [71] SKWIN [72] SKWINW [73] SPHAK [74] SRAV [75] SROB [76] STB [77] STUR [78] TAU [79] TILEG [80] WEAK [81] WHAK [82] WHK [83] WKSQ
KEPT	integer	*range:* 0 – 12000 *missing values:* 5792 (44%)
DISCARDED	integer	*range:* 0 – 15000 *missing values:* 3519 (27%)
SPECIES_COUNT	logical	*missing values:* 13056 (100%)
DEALER_NUM	integer*	*unique values:* 34 *missing values:* 6474 (50%) *format:* 1 to 5 digits
DATE_SOLD	character+	*range:* 31 Dec 1969 – 1 May 2020 *unique values:* 312 *missing values:* 6474 (50%) *format:* ‘dd-monthabb-yy’
PORT_LANDED	character	*unique values:* 17 *missing values:* 8221^ (63%) *levels:* [1] [2] BOSTON [3] CAPE MAY [4] CHATHAM [5] CUBITT CREEK [6] CUNDYS HARBOR [7] DENNIS [8] GLOUCESTER [9] HARWICHPORT [10] MENEMSHA [11] NEW BEDFORD [12] NEWBURYPORT [13] POINT JUDITH [14] PORT CLYDE [15] PORTLAND [16] PROVINCETOWN [17] WESTPORT

STATE_LANDED	character	*unique values:* 6 *missing values:* 8221^ (63%) *levels:* [1] [2] MA [3] ME [4] NJ [5] VA
VTR_RESOLUTION	character	*unique values:* 2 *missing values:* 0 *levels:* [1] HAUL [2] SUB-TRIP
FW55_EXEMPTION	integer	*range:* 0 – 1 *missing values:* 0

* considered categorial
+ considered datetime or date
^ missing values stored as ‘ ’ rather than NA

Appendix C. Structure and summary of raw dummy dealer data.

File Name FY19 Dealer Data SIMM
File Type Excel (.xlsx)
Observations 3204
Variables 21

Variable Name	Data Class	Description
Sector Id	double*	*unique values:* 4 *missing values:* 0 *format:* 2 to 4 digits
Sector Name	character	*unique values:* 4 *missing values:* 0 *levels:* [1] Sector A [2] Sector A 1 [3] Sector B [4] Sector C
Mri	double*	*unique values:* 11 *missing values:* 0 *format:* 2 to 4 digits
Vessel Permit No	double*	*unique values:* 11 *missing values:* 0 *format:* 6 digits
Vessel Name	character	*unique values:* 8 *missing values:* 0 *levels:* [1] V1 [2] V10 [3] V12 [4] V15 [5] V2 [6] V5 [7] V7 [8] V8
Vessel Reg No	character	*unique values:* 11 *missing values:* 0 *format:* 6 to 8 digits/characters
Vtr Serial No	double*	*unique values:* 263 *missing values:* 565 (18%)
State Land	character	*unique values:* 2 *missing values:* 0 *levels:* [1] MA [2] ME
Port Land	character	*unique values:* 7 *missing values:* 0 *levels:* [1] Boston

		[2] Chatham [3] Gloucester [4] Harwich Port [5] Plymouth [6] Portland [7] Portlane
Port Code	double*	*unique values:* 17 *missing values:* 0 *format:* 10 digits
Dealer Name	character	*value:* DEALER ABXXYZ *missing values:* 0
Dealer Permit No	double	*value:* 1234 *missing values:* 0
Date Sold	datetime	*range:* 2 May 2019 – 30 April 2020 *unique values:* 247 *missing values:* 0 *format:* 'yyyy-mm-dd UTC'
Market Category Code	character	*unique values:* 14 *missing values:* 0 *levels:* [1] HA [2] JB [3] KN [4] LG [5] LS [6] MD [7] MK [8] MX [9] QT [10] SK [11] SQ [12] SR [13] UN [14] XG
Grade Code	double	*range:* 0 – 64 *missing values:* 0
Nespp3 Code	double*	*unique values:* 39 *missing values:* 0 *format:* 2 to 3 digits
Nespp4 Code	double*	*unique values:* 93 *missing values:* 0 *format:* 3 to 4 digits
Species It is	double*	*unique values:* 37 *missing values:* 0 *format:* 5 to 6 digits
Species Name	character	*unique values:* 37 *missing values:* 0 *levels:* [1] ANGLER [2] BASSSTRIPED

		[3] BLUEFISH [4] BUTTERFISH [5] COD [6] CRABJONAH [7] CUNNER [8] CUSK [9] DOGFISH SPINY [10] FLOUNDERAM. PLAICE [11] FLOUNDERSUMMER [12] FLOUNDERWINTER [13] FLOUNDERWITCH [14] FLOUNDERYELLOWTAIL [15] HADDOCK [16] HAKE MIX RED & WHITE [17] HAKERED [18] HAKESILVER [19] HAKEWHITE [20] HALIBUTATLANTIC [21] JOHN DORY [22] LOBSTER [23] MACKERELATLANTIC [24] POLLOCK [25] REDFISH [26] SCUP [27] SEA BASSBLACK [28] SKATEBARNDOOR [29] SKATELITTLE [30] SKATESMOOTH [31] SKATEWINTER(BIG) [32] SQUID (LOLIGO) [33] TAUTOG [34] TILEFISHGOLDEN [35] TUNABLUEFIN [36] WEAKFISHSQUETEAGUE [37] WHITINGKING
Landed Weight	double	*range:* 1 – 9560 *missing values:* 0
Live Weight	double	*range:* 1 – 10898 *missing values:* 10 (< 1%)

* considered categorial

Appendix D. Description of data portal components and general task list. The pre-processing script (pre-processing/pre-processing.R) must be applied to the original data files found in orig/ before the data portal can be executed.

	File Name	Description	General Task List
a	run.R	Runs application from project directory	
b	global.R	Prepares application environment	- loads packages - sets scientific digit parameter - imports prepared data[e] - imports customized scales for visualizations[f]
c	ui.R	Defines data portal layout	- defines body layout content[g] - defines sidebar layout content[h]
d	server.R	Generates reactive content	- prepares data subsets based on sidebar input controls[s] - generates reactive content for each tab[t-aa]
e	data-import-prep.R	Import and prepare processed data files	- import and prepare Electronic Monitoring (EM) data from **data/em_processed.csv** - import and prepare Dealer data from data/simm_processed.csv - import and prepare eVTR data from data/evtr_processed.csv - import and prepare Environmental data from data/A01_sbe37_all_dab4_6226_5825.csv, data/B01_sbe37_all_dab4_6226_5825.csv, data/E01_sbe37_all_dab4_6226_5825.csv, and data/F01_sbe37_all_dab4_6226_5825.csv
f	import-scales.R	Defines customized scales for visualizations	- defines species colour and fill scale - defines EM subject fill scale - defines EM category fill scale - defines catch type fill scale - defines gear type fill scale - defines port fill scale - defines sales metric fill scale - defines environmental data station fill scale
g	ui-body.R	Defines data portal body layout	- defines body layout for each tab[i-r]
h	ui-sidebar.R	Defines data portal sidebar layout	- generates vessel selection selectInput control - generates date range selection dateRangeInput control

	File Name	Description	General Task List
i	ui-body-welcome.R	Defines welcome tab body layout	- accesses descriptive / instructional text[ac] - defines placement of text and input controls - generates inactive textInput controls as placeholders for user and password options
j	ui-body-catch.R	Defines catch tab body layout	- accesses descriptive / instructional text[ac] - defines placement of text, input controls, figure, map, table, and download buttons
k	ui-body-effort.R	Defines effort tab body layout	- accesses descriptive / instructional text[ac] - defines placement of text, input controls, figure, map, table, and download buttons
l	ui-body-enviro.R	Defines enviro tab body layout	- accesses descriptive / instructional text[ac] - defines placement of text, input controls, figures, maps, table, and download buttons
m	ui-body-sales.R	Defines sales tab body layout	- accesses descriptive / instructional text[ac] - defines placement of text, input controls, figure, table, and download buttons
n	ui-body-compare.R	Defines compare tab body layout	- accesses descriptive / instructional text[ac] - defines placement of text, input controls, figure, table, and download buttons
o	ui-body-catchreviews.R	Defines catchreviews tab body layout	- accesses descriptive / instructional text[ac] - defines placement of text, input controls, figure, table, and download buttons
p	ui-body-otherevents.R	Defines otherevents tab body layout	- accesses descriptive / instructional text[ac] - defines placement of text, input controls, figures, table, and download buttons
q	ui-body-export.R	Defines export tab body layout	- accesses descriptive / instructional text[ac] - defines placement of text and download buttons
r	ui-body-data.R	Defines data tab body layout	- accesses descriptive text[ac]
s	server-dataprep.R	Prepares data subsets based on sidebar input controls	- prepares data subsets based on vessel and date range selected with sidebar input controls
t	server-catch.R	Generates catch tab body reactive content	- prepares data based on tab-specific input controls - generates reactive figure, map, table, and download buttons

	File Name	Description	General Task List
u	server-effort.R	Generates effort tab body reactive content	- prepares data based on tab-specific input controls - generates reactive figure, map, table, and download buttons
v	server-enviro.R	Generates enviro tab body reactive content	- prepares data based on tab-specific input controls - generates reactive figures, map, table, and download buttons
w	server-sales.R	Generates sales tab body reactive content	- prepares data based on tab-specific input controls - generates reactive figure, table, and download buttons
x	server-compare.R	Generates compare tab body reactive content	- prepares data based on tab-specific input controls - generates reactive figure, table, and download buttons
y	server-catchreviews.R	Generates catchreviews tab body reactive content	- prepares data based on tab-specific input controls - generates reactive figures, table, and download buttons
z	server-otherevents.R	Generates otherevents tab body reactive content	- prepares data based on tab-specific input controls - generates reactive figures, table, and download buttons
aa	server-export.R	Generates export tab body reactive content	- generates download buttons

Appendix E. Questions to guide user testing and feedback on the existing data portal prototype with real world data incorporated.

Thank you for providing in-depth feedback to the Groundfish Data Portal. We appreciate your thoughtful responses and will compensate you for your effort. Please read through the questions before beginning your review of the Data Portal. If you prefer, after you play with the Portal a bit, Mel will sit with you and you can talk through your answers instead of writing them.

You have FY2019 and FY2020 data loaded into the portal; use the date range to show the desired trips. Please remember that the data is coming from your sector manager, so it will include all trips with a submitted eVTR or dealer reported data. However, if you had dealer data but not an eVTR for a given trip (i.e., HMS tuna trip), you won't see tuna in the eVTR page but it will show up on the dealer page.

Questions have been broken down into categories to simplify feedback.

1. Instructions and Ease of Use
    a. Are the instructions easy to understand? If not, how can we improve them?
    b. Would you have preferred to watch a demonstration video prior to use?
    c. Was the portal easy to navigate and use? If not, how can we make it better?
    d. Is the side panel menu easy to navigate?
    e. Is it intuitive that to select something you click on it and to deselect, you click again? (like in the Table Detail)

2. Aesthetics
    a. How can we make the overall layout better?
    b. What do you like about the figures, tables, and maps? How can we improve them?
    c. When specifying a specific eVTR to display in the plots, do you prefer entering it into the text box (e.g. in the eVTR Data>> Catch tab) or clicking on it in the summary table (e.g. in the EMFeedback>>Other Events tab)?
    d. How can we improve the colours used in the figures?
    e. Can you easily match the colors in the legend to the colors on the graph?
    f. There are more species than we have easily distinguishable colors, so some species or species market categories have been lumped together under the same color. See the table below.

Groundfish, Allocated (Quota)	
	Cod
	Haddock
	Redfish
	Pollock
	White Hake
	Flounder, Dab/American Plaice
	Flounder, Yellowtail

	Flounder, Gray Sole/Witch
	Flounder, Winter/BlackBack
	Halibut
	Wolffish
	Ocean Pout
	Flounder, Windowpane/Sand Dab

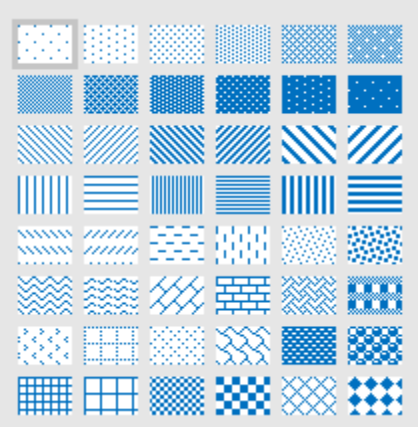
High Frequency Non Groundfish	
	Black Sea Bass
	Bluefin Tuna
	Bluefish
	Butterfish
	Cunner
	Cusk
	Golden Tilefish
	Dogfish, Smooth
	Dogfish, Spiny
	Flounder, Summer
	Fluke
	Founder, Four Spt
	Hake, Red
	Hake, Silver
	Jonah Crab
	Lobster
	Monkfish
	Monkfish, Heads
	Monkfish, Liver
	Monkfish, tails
	Sculpin
	Scup

	Sea Robin
	Sea Scallop
	Skate Wings, Little Winter Mix
	Skate Wings, Unclassified
	Skate Winter, Wing
	Skate, Little
	Skate, Little Winter Mix
	Skate, Winter
	Skate, Barndoor
	Skate, Barndoor Wing
	Skate, Thorny
	Skate, unclassified
	Squid, Illex
	Squid, Loligo
	Striped Bass
	Tautog
	Weakfish
	Whiting/Kingfish
	Mackerel

Lower Frequency or No Value Non Groundfish	
	Crab, Horseshoe
	Crab, Spider
	Crab, Rock
	Herring
	Herring, Blue Back

	Menhaden
	Shad, American
	Alewife
	Dogfish, Chain
	Hake, Spotted
	John Dory
	Ray
	Sea Raven
	Skate, Clearnose
	Bonito
	Shark, Basking
	Shark, Porbeagle
	Shark, Sand Tiger
	Shark, Thresher
	Shark, Thresher Big Eye
	Shark, Tiger
	Sturgeon
n/a	Scallop Shells

- Are there any species that we lumped that should be its own color? For instance, can all the skates be the same color?
- Are there species with their own color that could be lumped with others?
- Would you prefer if we used patterns instead of or in combination with solid colors (examples below)?
- Do we need to treat the colors for bait skate and skate/monk/groundfish trips differently?



3. Data Available
    a. IF this Data Portal provided you with your in season (updated once a week) and historic data, which data fields would you use most often?
    b. How important is it to have in-season data available? If only historic data (through the end of last fishing year) was available, would you still use the Data Portal?
    c. Which data elements are NOT useful?
    d. Which data elements would you use regularly?
    e. What other data would you like included?
    f. Is having the highlight data from a specific trip option (the box in the upper right) a feature we should keep?
    g. The eMOLT data is complicated and we need your help thinking about how to best use/display it. Currently you have to click on an empty spot on the map in hopes of eMOLT data existing there and the buffer around the actual data is fairly large (since there is not eMOLT data everywhere).
        i. How far away could the eMOLT data be from your fishing location and still be useful? 1 mile? 5 miles? 10 miles? 20 miles?
        ii. How long ago could the eMOLT data be from the date of your catch and still be useful? 1 day? 2 days? 7 days? Same month?
        iii. iThe eMOLT data set is very large, which makes it difficult and slow to display all of it on the chart. What elements are most important to you?
4. Analyses
    a. Will you use the export feature? Which data or charts or plots are you most likely to export? What will you do with the export?
    b. We have both catch and effort data in the Portal, which means we could display Catch Per Unit Effort. Would that be a good addition?
        i. How would you like CPUE displayed? As a table or plot? Proposed display: On the eVTR Data > Catch tab, a fill drop down could be the CPUE along the vertical axis. Effort would be number of hauls.
        ii. How do you want CPUE measured? The number of hauls or the quantity of gear (number of hooks) and/or duration of tow?
    c. Are there other analyses (charts, plots) that you would like to see added?
5. Your Business
    a. What business decisions do you make that requires reviewing your data, in any form (paper, logbooks, digital, database)? Examples might be: deciding how much quota to lease, buying/selling permits, revenue and tax projections, where/when to fish, when to switch fisheries.

b. Of those decisions, where do you currently get the data you need? Would you turn to this Data Portal to access necessary data? If yes, which data are you using?
c. Thinking of those business decisions, please list any other data sets or analyses that you would like to see included in the Data Portal.
d. How often do you think you would use the Data Portal?
e. Would you access it from a home computer or would you prefer to access it from the boat's computer (assuming you have internet)? On your phone?

Now we're going to ask specific questions for each page:

6. eVTR Data
    a. The table at the bottom of each of the eVTR pages (Catch, Effort, and Environmental) updates in response to your filter selections (species, gear). For example, in the Catch tab you can see the number of pounds of cod by picking cod from the species list. Is this feature worth keeping?
    b. In these tables, each subtrip gets its own entry. This means that there may be multiple entries with the same VTR number. Should we combine subtrips into one single trip entry or keep it at the subtrip level? When considering this, remember that a subtrip reflects changes in gear or stat area.
        i. Is "subtrip" a well used term among fishermen? If we use the term to explain why the same VTR number may show up on multiple dates, will that make sense?
        ii. Do you have suggestions for how to handle the "subtrip" vs. "whole trip" issue?
    c. In the upper right corner of each of the eVTR pages you can input one or more specific eVTR numbers into a text box. Those data points will be highlighted on the figures and maps and the table will only include data from those eVTRs. Is this feature worth keeping?
    d. Please test the export buttons and look at the exported file. Is this what you expected to see? If not, what else should be included in the export?
7. eVTR Data > Catch
    a. The table at the bottom of the Catch tab has an additional feature, where you can choose to see the overall values or break them down into either species or gear. Is this feature useful? Is this feature easy to understand?
    b. Are there other "Fill Based On" options you would like to see added?
    c. Please test the export buttons and look at the exported file. Is this what you expected to see? If not, what else should be included in the export?
8. eVTR Data > Effort
    a. Is the quantity of gear the correct unit for the vertical axis?
    b. Please test the export buttons and look at the exported file. Is this what you expected to see? If not, what else should be included in the export?
9. eVTR Data > Environment
    a. Assuming that the final Portal has more environmental data loaded (more buoys and/or other variables like salinity, tides, etc), are the plots and tables easy to interpret and use?
    b. What other environmental variables should be a high priority to add to the Portal?
    c. Please test the export buttons and look at the exported file. Is this what you expected to see? If not, what else should be included in the export?
10. Dealer Data
    a. You'll notice in the Dealer Data tabs (Sales and Comparison to eVTR) you are not able to highlight a specific eVTR in the upper right corner like you could in the eVTR tabs (Catch and Effort). Instead, you can click on a record in the table at the bottom of the page to only show data points associated with that eVTR in the figure(s). Is this a feature worth keeping? Do you prefer the method used in the eVTR tabs or the Dealer tabs?
11. Dealer Data > Sales

a. Are the filter options all useful? Do you care about seeing the data broken down by port?
b. Are there other filter options we should add?
c. Please test the export buttons and look at the exported file. Is this what you expected to see? If not, what else should be included in the export?

12. Dealer Data > Comparison to eVTR
    a. NOTE: there is something glitchy going on with the data (we are working on it) where if you reported at the haul level, it appears to be duplicating the catch for each reported haul.
    b. Please test the export buttons and look at the exported file. Is this what you expected to see? If not, what else should be included in the export?
13. EM Feedback
    a. Right now this section just shows the EM review data and the various "events" that the reviewer records (like dirty cameras, improper fish handling, and issues with the EM system). This was the data we had access to. Are there other EM data elements that you would like to have available in the Portal? Trip selected for review? Audit status? Copy of your AMP report from GARFO? Your EM feedback letter from the EM Service Provider?
14. EM Feedback > Catch Reviews
    a. Is it helpful to see the number of individual fish reviewed by EM? Is there something else in the EM data you would rather see?
        i. We tried to do a comparison of eVTR and EM data, but there is a lot of post-processing done by NOAA, which means it is not a straightforward comparison. If this is a high priority item, we can work on getting the equations they are using.
    b. Please test the export buttons and look at the exported file. Is this what you expected to see? If not, what else should be included in the export?
15. EM Feedback > Other Events
    a. Is it helpful to see your progress over time related to crew events? What would you do with this data?
    b. Please test the export buttons and look at the exported file. Is this what you expected to see? If not, what else should be included in the export?

Appendix F. Directed feedback from user testing, incorporating real world data into the existing data portal prototype template, in December 2021.

User 1

Already looks at paper logbooks for remembering past years' catch, looking for trends (high catch rates, times/places to avoid, planning fishing trips), planning quota leases.

Yes, would use portal at home on computer to review for historic trends and planning quota leases and identifying where/when he should avoid fishing to reduce cod catch and other limited species.

Despite not being computer savvy, he would use this portal if it was available. Once he learns the tech, he can be comfortable with it. Would prefer in-season data and would use more often then but would also use it sporadically if only past fishing years available, to supplement his paper logbooks.

eMOLT data high priority- he is an eMOLT vessel and would like to see his own eMOLT data in the portal.

- Would like the temp and catch graphs to overlay on same graph (double Y axis, with X= dates/time) so he can see the trends himself.
- More useful to classify by depth contours, not miles away. Stay on 100fathom ledge and five miles away could be close enough.
- Weekly is okay. Monthly could be okay at deeper depths(minus big storms)

CPUE should be measured by hours of towing. If not hours, would be happy with catch per day or catch per tow.

Other environmental data of interest is tides.

Showing depth of fishing activity is important – this data exists in the EVTR dataset, although not in the files we've been working with. Recommend adding it as a fill option for catch (eVTR data) (bin in 10 fathom increments)

Temp units: C or F is okay but F would be better

Definitely need to have a fill option in the dealer data for market categories. Critically important for the Max Retention EM boats, since they land their sublegal discards as a new market category.

For Max Retention/Dana, it was not useful to see EM feedback in chart form (he just read written reports)

Would you download data/charts? Not usually, but could see doing so to send the details to government/sector manager to dispute / question a problem with the data.

Instructions were easy to understand and to navigate portal. Might use a video demo.

Colors were fine- no problem with existing scheme and the groups we've binned. All monk can be one color, etc. Not a fan of the pattern option.

Would like to have price data from dealers. If not his own data (which NOAA should be able to provide), he would also find weekly regional averages helpful.

Subtrip as a term was fine to use. Can leave the multiple subtrips per trip (so multiple lines of data per trip), since that makes sense for fishermen if they switched stat area or gear.

Would like to search for a specific trip by date (as opposed to by VTR number)

Showing Effort as "hours towed" is correct.

Seeing data broken down by port is helpful.

Was less interested in VTR/Dealer comparison… he feels that his dealer is trustworthy.

User 2

I am a small-boat, seasonal groundfish fisherman. I looked at the prototype data portal primarily from the perspective of an EM user tasked with review of the output coming from the system to determine if it matches both eVTR data and personal logbooks and experience. This is critical for quota monitoring and cost control and compliance. Evaluating the potential for using the data as a fishing planning tool was a secondary consideration. I have attempted to answer the evaluation queries in narrative form by section. I called the data prototype up an an IPad Pro while answering questions on an HP laptop. I later used the laptop to export data as this was more expedient.

I. Instructions and Ease of Use I was able to navigate to the pertinent information in order to look at my catch data using the instructions provided. There was no clear instruction to a fisherman interested specifically in what his permit was being charged for quota usage by the fisheries service. This would appear to be found in the Dealer Data under "Sales" and/or "Comparison to EVTR". It may be useful to consider the different perspectives of potential users and provide more specific instructions and example templates based on that. For instance, a chart showing the quota usage by species, cumulative and by date or eVTR would be helpful. A video demonstration would probably be helpful, or at least knowing that this was an option would be. I did find the side panel self-explanatory and straightforward, and it was nice to have all options present in one space. As to clicking on options, I found it intuitive but more difficult on the iPad (and phone) than on a laptop due to the size of the screen.

II. Aesthetics In its prototype form, the site is not especially compelling to look at. Many marine software vendors allow users to personalize their experience by adding a background picture of their boat, which could help here. In keeping with the theme of user purpose-based interaction, adding sections to the sidebar for instructional videos and example templates for quota monitoring and other business-based data representation would be useful. I enjoyed seeing the data represented on bar charts. While I understood that my vessel basically had two types of trips (on the fish, not on the fish), it was interesting to see them depicted this way. When we are not on the fish, we catch a background level of target that is similar throughout our fishing area. On the fish, we see a doubling or more of catch rate for target species. This really jumps out at me and could prove to be very useful. The charts of dealer data showed just how thoroughly inaccurate it can be and may save a lot of money on quota in the longer term. I did not find the map to be immediately useful at all, since I already know where I

fish and the road map shown appears to be opposed to showing more than one trip on the exact same location. The environmental data could be very useful if it could be used to find areas with specific bottom temperature. I absolutely prefer clicking on the summary table to highlight a specific trip. The less data entry the better! So helpful to have all the trips listed there below the chart for the specific date range. The colors used in the figures and graphs were easy to match to the legend. As a bottom longline vessel, there are some species (most flounders, many sharks) we almost never catch. The groupings of species under all headings made sense to me for my fishery and I don't think they need to change. If there are anomalies in the data for one of the grouped species the user can always zoom in to a greater level of detail to sort it. The solid colors work well for me but if one were to highlight certain species within the grouped ones (I suggest porbeagle and blue sharks – which didn't appear on the list) the hatching could be used for them. I am agnostic on the question of bait skate vs skate/monk trips – ask a gillnetter! Seems like it would make sense though.

III. Data Available One of the greatest strengths of the portal from my perspective is the potential to see near real-time reporting and quota usage. We are now more than 10 years into a quota-based management system and still have a convoluted way of calculating quota used, involving eVTR, dealer data, and now – feedback from EM trips. In-season data fields would always be most useful but being able to make year over year comparisons would also be very helpful. If only historical data were available, the portal would be useless for quota (cost) monitoring and would only be useful for a superficial type of business planning. One of the aspects of the presentation I found particularly annoying was the server time-out. Resetting meant also resetting date ranges every time and it seems like once the user has cleared the password threshold they should be able to keep the page up as long as needed! I found all the data presented to be useful; and having a way to summarize catch by species for a given data range is very helpful. Perhaps adding a summary table by species instead of just a total weight to the eVTR and dealer data would accomplish this. Having the ability to highlight a specific trip could be useful and should therefore be kept. The chart showing EM feedback events would be more helpful if one could look at the specific report by the reviewer (more on this later!). They are not self-explanatory and without reference or more detail are simply indicators of poor performance. The eMOLT data is very important to bottom longline vessels as we can use bottom temperature to avoid spiny dogfish at certain times of the year. Almost as important as distance (5 miles?) is depth of observation, since the profile is never static. Historical observations are important as are recent ones, to help discern patterns. Data from within a week or two are most useful. If eMOLT could display observations taken at similar depths (+/- 5 fathoms – nobody uses meters) within 10 miles of the vessels fishing location it would be most helpful. Finally, regarding the presentation, showing vessel location on a nautical chart would be more meaningful than a road map!

IV. Analyses I would download data pertinent to clarifying the record of quota usage for my vessel, including data, charts and plots showing dealer-reported information. I would also save data surrounding EM review events to learn more about them in order to improve performance. I can also see the potential for creating a year-end summary table of the data for my records. Catch per unit effort is a key metric in my fishery. I look at CPUE currently in terms of catch pounds of target and nontarget species per hook, per hour, per day and per trip. Using trip returns I can then convert these to dollars per unit effort. I also overlay catch area, bottom water temperature and other factors. Facilitating any of these views would be a huge help since I have not currently found a consistent way to enter all the data required and generally do it by hand. Other data that is pertinent to CPUE for my

business includes the price of fuel, the amount of my target species landed domestically and imported, and fixed costs like insurance and rent. Being able to add these would create a baseline for measuring the success of a trip.

V. My business I am at a point in my business where I do not intend to borrow money to invest in more gear or permits but review of the data and compilation of reports for financial help would be useful if I were. This includes revenue history and projections (backed by pounds landed). Quota tracking and purchasing is the most likely use for me, as currently I am unaware of another way to see dealer data for my landings. I see the portal as the primary feedback mechanism for EM, and thus critically important to the long term success of the program. So the review data is also important. More information regarding dealer data – how it is generated, how to follow up misreporting, etc would be helpful. I estimate I would use the portal on a bimonthly basis to track quota and reports. I would likely use it less frequently for business or trip planning. However, if I were a fleet owner, I can see using it daily. Having worked with the portal to some extent I have found it easier to dig into and navigate with my laptop, although I would use the iPad to check reported landings. I do not anticipate using a phone for this. VI. Specific Questions – eVTR Data The table summarizing the eVTR catch data is one of the most critical for quota tracking comparison and is essential information. So- worth keeping! Again, adding a table summarizing totals of quota-managed species here and in the dealer data section would be helpful. Subtrip-level reporting is important to me for trip area planning, but it could make for a cleaner presentation to combine them into one and leave some indicator that subtrip data is available to break out if desired (i.e. an asterisk or letter indicator). Much of the lingo surrounding EM is not used by fishermen currently but there will be a next generation SOMEday and accountability and its nomenclature will be a part of their fishery. Fishermen will get better at understanding the named components of a trip as the program evolves. Our trips tend to be short and simple with limited amounts of bycatch. Breaking out individual trips for clarity is not super-important but I can understand why a trip boat out for a week would want to scrutinize the data on that basis. Exporting data presupposes a familiarity with Microsoft Excel or other .csv format, which may be a stretch for some fishermen. Aside from stretching column width, the data transferred pretty much as expected. I did notice some inconsistencies between the chart of dealer vs. eVTR and the data as transferred. Catch: One of the first things I did when looking at the table was to break it down by species using the "species" option above the chart to the right of "catch type". This is the most useful feature of the data for quota tracking. It also led to the aforementioned insight regarding types of trip. Looking at bycatch in this way is also helpful – seeing dogfish drop off seasonally or encountering different species by area can help with planning. I did not find it as useful to break out the table by species using only the "table detail" option above the table since the resulting table was not easily coaxed into totals by species. I did notice that using "species" at the top of the page to select a specific species (e.g.cod) and then specifying "by species" above the table meant the name of the species was exported with the data, while choosing "overall" in the table detail meant it wasn't. Again, this is an area where a report template for those interested in quota tracking could be useful. Effort: For a hook trip, number of hooks is the proper quantity of gear. Each set is a different length and number of hooks is the critical factor for CPUE. The exported data is not at all what I expected. When I used the "export data" feature at the top of the page I got a table of values that did not organize the data coherently – different dates appeared out of order and trip effort was only partially listed in most cases. When I used the "export current table feature" the data was organized by eVTR number (and hence, sail date) but was only partially represented. On a trip where I fished 6000 hooks (1/7/20) the table indicated 2100. We fished

7200 hooks on a trip from 1/21/20-1/22-/20 and the table indicated 3700. I expected to see the reported quantities of hooks! Environment: There is a lot of information to digest here. I was able to easily look at a plot that showed target species dropping off and bycatch of spiny dogfish ticking upward as water temperature at depth trended upward in the spring. I chose "51" meters as the depth designation but the actual depth of the buoy measurement wasn't indicated on the plot or table. This could be a tremendously useful table for scientifically-minded fishermen to use, though understanding the spatial and temporal relations to one's own effort will take some trial and error. I am not aware of the effects of other variables such as salinity or other measured/known factors on my fishery and it would be interesting to explore. I tested each of the export options and was able to see a .png file of the "catch/temperature plot" but got a "server error" message when I tried to export any of the data.

VII. Dealer Data I found the method of highlighting a specific trip used here was preferable to that used in the eVTR tabs. I didn't have to copy and paste anything. It is a feature worth keeping. The only thing is, it wasn't immediately clear how to revert back to the entire table without highlighting all the trips. The filter options are useful to clearly demonstrate how little the dealers care for accuracy in reporting, and what a nightmare it must be for the fisheries service to tease out the pertinent information from this disaster of a database. The port landed information for my vessel was hopelessly incorrect. The landed and live weight data listed in the landings data looked correct for the most part and showed up as expected when exported. This seems to be the correct way to generate a quota tracking report! The comparison data seems way out of whack and was alarming in that what looked like massive overreporting of catch by the dealer could result in extra quota charges. Since I have been reporting at the haul level I understand this is a work in progress. It appeared that some of the data was correct and some was double-counted or inaccurate when I exported the file, and it did not match the plot.

VIII. EM Feedback If there are reports available regarding the compliance status of my vessel, I absolutely want to see all of them. If and when NMFS has the spine to actually require accountability, understanding the requirements of the systems will be an essential component of operations. What is currently listed on the portal is a summary of the failings of each trip and many of the fish reviews. Including more data would formalize a feedback mechanism that now only occurs via random email from the reviewer. While important to me this would seem to be even more critical to a fleet operator or non-owner operated vessel. Some fishermen look at enforcement measures as suggestions and examine any new regulation in light of its loopholes. Not accurately estimating catch seems to be the loophole in this system the way EM is currently envisioned. I am aware that reviewers look closely at measured bycatch and that is important to record. However, industry has leveraged the target monitoring rate of 100% to do away with dockside monitoring. This means that a vessel landing large quantities of one species can potentially continue to report it as another. It would be interesting to see not only the number of bycatch fish examined, but also the estimate of landed weight and number of legal-sized catch generated by the reviewer. Without creating that estimate, there will continue to be a truck-sized loophole in the accountability area. I did not realize comments were available on an individual fish-reviewed basis until I downloaded the data using the "export data" tab. This could have been made clearer on the summary page. We had to work through subsampling protocols in the course of piloting the audit model for longline boats and the reviewer comments and frustration recalls that! This helps to shed light on the trend of the fisherman and reviewer to close the loop on issues that affect the accuracy of EM and I was glad to see it in the data table although it made for a large file. It

seems like this data and the “other event” information could be combined in some way to facilitate examination of the whole trip from the perspective of EM compliance, but this is probably not the right forum to suggest it. The questions suggest that it should be helpful to see progress over time but one message that comes through the data indicates that each gear type will have inherent problems implementing EM and it is up the NMFS to decide what the tolerance should be. I was frankly shocked at the number of “other events”, and I was piloting the gear, with a view toward having it accepted as a lower-cost alternative to live observers. I can imagine that a fisherman without the same attitude toward EM may find even more comments and some indicator from the service on what is acceptable would help to put this portion of the portal into context. Thank you for the opportunity to review and comment on the Data Portal.