Package: inseasonDashboard (via r-universe)

May 19, 2026

Title In-Season Fishery Monitoring Tools

Version 0.1.0

Description Tools for accessing AFSC and AKFIN databases and producing standardized in-season fishery summaries and diagnostics.

License file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports data.table, DBI, odbc, bslib, keyring, dplyr, ggplot2, scales, shiny, stats, rnaturalearth, rnaturalearthdata

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libsecret-1-dev libuv1-dev unixodbc-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev zlib1g-dev

Repository https://jimianelli.r-universe.dev

Date/Publication 2026-02-18 20:30:17 UTC

RemoteUrl https://github.com/afsc-assessments/inseasonDashR

RemoteRef HEAD

RemoteSha e98d4c32da6eeb40fd1ddbf4c58355c3b33aabd6

Contents

`comma` *function to format numbers to text with 1,000s comma*

Description

function to format numbers to text with 1,000s comma

Usage

```
comma(number)
```

Arguments

`number`	value to format

`db_connect` *Connect to AFSC and AKFIN databases*

Description

Uses keyring to securely retrieve credentials.

Usage

```
db_connect()
```

Value

A list with AFSC and AKFIN DBI connections

empty_message_plot *Create an empty plot with a centered message*

Description

This helper function returns a ggplot object containing a centered informational message. It is intended for use when no data are available for plotting (e.g., no length data for a selected species and time period), allowing Quarto / R Markdown workflows to continue without errors while clearly communicating the reason for the empty figure.

Usage

```
empty_message_plot(
  text = "No length data available for this time period for this species"
)
```

Arguments

text	Character string giving the message to display in the center of the plot.

Value

A ggplot object with no axes or background and centered text.

Examples

```
empty_message_plot()

empty_message_plot(
  "No length data available for this time period for this species"
)
```

get_catch_data_date *Pull Observer + EM catch data for a species and optional date range*

Description

Uses SQL templates with `-- insert species` and `-- insert year` placeholders. If `date_min` is provided, the year from `date_min` is used for the SQL `>= year` filter (first_year). Date filtering is also applied post-pull (inclusive).

Usage

```
get_catch_data_date(
  afsc,
  akfin,
  species = 202,
  date_min = NULL,
  date_max = NULL,
  tz = "UTC",
  return_sql = FALSE
)
```

Arguments

afsc	DBI connection to AFSC database.
akfin	DBI connection to AKFIN database.
date_min	Optional start date (inclusive) as "mm/dd/yyyy".
date_max	Optional end date (inclusive) as "mm/dd/yyyy".
tz	Timezone used when parsing datetime strings (default "UTC").
return_sql	If TRUE, return the final SQL text used (default TRUE).
fsh_sp_label	Numeric/integer species code(s) for the SQL IN clause.

Value

list(data_o=..., data_em=..., sql=list(...) if return_sql=TRUE)

---

get_codes *Retrieve code table rows for one or more species*

Description

Reads a SQL template file (GET_CODES.sql), injects one or more agency species codes, and runs the query against a provided database connection using the `sql_filter()` and `sql_run()` utilities.

Usage

```
get_codes(con, spec, sql_dir = "sql")
```

Arguments

con	A DBI connection object (e.g., your AFSC connection).
spec	Integer or numeric vector of agency species codes to filter on.
sql_dir	Character path to the directory containing GET_CODES.sql. Default is "sql".

Value

A data.frame containing the code table rows returned by the query.

Examples

```
## Not run:
afsc <- db_connect("afsc")
codes <- get_codes(afsc, spec = 202)
codes2 <- get_codes(afsc, spec = c(202, 203))

## End(Not run)
```

`get_council_catch_data` *Retrieve catch data by species and minimum year*

Description

Pulls catch data from the AFSC database using GET_ALL_CATCH.sql, filtering by agency species code and year >= input year.

Usage

```
get_council_catch_data(afsc, akfin, species, year_min)
```

Arguments

`afsc`	DBI connection object afsc
`akfin`	DBI connection object akfin
`species`	Integer observer program species code
`year_min`	Integer minimum year to include (inclusive)
`sql_dir`	Directory containing SQL files (default "sql")

Value

A data.frame with catch data

`get_length_freq_data_date` *Pull observer length-frequency data (AFSC) by species and date range*

Description

Uses the SQL template `sql/GET_CURRENT_LENGTH.sql`, inserting a species filter at the `-- insert species` flag and a year filter (>= first year from date_min) at the `-- insert year` flag, using the legacy `utils.r` helpers.

Usage

```
get_length_freq_data_date(
  afsc,
  species,
  date_min,
  date_max = NULL,
  gear = c("Trawl", "Pot", "Longline"),
  tz = "UTC",
  return_sql = TRUE
)
```

Arguments

afsc	DBI connection to the AFSC database.
species	One or more species codes used in OBSINT.CURRENT_SPCOMP.species.
date_min	Start date (inclusive) as "mm/dd/yyyy". Required.
date_max	End date (inclusive) as "mm/dd/yyyy". Optional.
gear	Optional gear filter: any of c("Trawl","Pot","Longline"). Default all.
tz	Timezone used when parsing datetime strings (default "UTC").
return_sql	If TRUE, return the final SQL text used (default TRUE).

Details

Length-frequency is summarized by SPECIES x GEAR x LENGTH.

Value

A list with:

raw data.table of pulled rows after date/gear filtering

lf data.frame length-frequency aggregated by SPECIES, GEAR, LENGTH

sql (optional) list(sql = , first_year_used =)

---

get_observer_cpue_data

Pull observer-based CPUE data and construct monthly CPUE indices

---

Description

Retrieves observer haul-level data for a given species, applies standard filtering rules (date, region, gear, species proportion), calculates CPUE, and aggregates results to monthly CPUE indices by gear and NMFS area. Optionally applies AKFIN catch-based blending weights to produce fleet-representative indices.

Usage

```
get_observer_cpue_data(
  con,
  species,
  prop_min = 0.3,
  date_min = NULL,
  date_max = NULL,
  region = NULL,
  gear = c("Trawl", "Pot", "Longline"),
  use_blend = TRUE
)
```

Arguments

con	A DBI connection to the AFSC observer database, or a named list with elements afsc and akfin. The AKFIN connection is required only when use_blend = TRUE.
species	Numeric vector of observer species codes.
prop_min	Minimum proportion of the species in the total catch required for a haul to be retained (default = 0.30).
date_min	Optional start date ("mm/dd/yyyy") for filtering hauls.
date_max	Optional end date ("mm/dd/yyyy") for filtering hauls.
region	Optional region filter. May be one or more of "AI", "BS", "GOA", "BSWGOA", or numeric NMFS area codes.
gear	Character vector specifying gears to include. Allowed values are "Trawl", "Pot", and "Longline".
use_blend	Logical. If TRUE, applies AKFIN-based catch weighting to monthly CPUE indices (default = TRUE).
year_min	Optional minimum year for filtering data. If date_min is supplied and year_min is not, the year is inferred from date_min.

Details

This function is designed for use in in-season and assessment-support workflows that rely on AFSC observer data, with optional integration of AKFIN catch data for blending. Effort is defined as haul duration (minutes) for trawl gear and number of hooks or pots for fixed gear.

Monthly indices are computed only when minimum sample size criteria are met (at least 2 vessels and more than 3 hauls per stratum).

Value

A named list with three elements:

data_obs Haul-level observer data with calculated CPUE and effort.

data_index_month Monthly CPUE indices by year, month, gear, and NMFS area, including standard errors.

meta List of metadata describing whether count data were present and which columns were used for haul and vessel identifiers.

References

Alaska Fisheries Information Network (AKFIN) NOAA Fisheries, Alaska Fisheries Science Center

See Also

sql_run, sql_filter

Examples

```
## Not run:
con <- list(
  afsc   = DBI::dbConnect(odbc::odbc(), "afsc"),
  akfin  = DBI::dbConnect(odbc::odbc(), "akfin")
)

res <- get_observer_cpue_data(
  con      = con,
  species  = 202,
  region   = "BS",
  gear     = c("Trawl", "Pot"),
  year_min = 2015
)

## End(Not run)
```

---

launch_inseason *Launch the inseason dashboard*

Description

Runs the Shiny application shipped with this package.

Usage

```
launch_inseason(..., quiet = TRUE)
```

Arguments

...	Passed to shiny::runApp(), e.g. launch.browser = TRUE.
quiet	Passed to shiny::runApp().

Value

(Invisibly) the result of shiny::runApp().

`plot_catch_depth_noaa_np_date` *Plot catch depth by latitude or longitude (NOAA North Pacific style)*

Description

Creates a depth–position scatter plot of catch observations using Observer and optional EM data. Depth is shown on a reversed y-axis, point size is scaled by catch weight (metric tons), and transparency is scaled by recency. The plot can be faceted by gear and includes optional annotations for total hauls and vessels. Longitude is automatically handled to avoid discontinuities at the dateline for Alaska-centric data.

Usage

```
plot_catch_depth_noaa_np_date(
  data_o,
  data_em = NULL,
  species_name,
  date_min = NULL,
  date_max = NULL,
  region = c("AI", "BS", "GOA"),
  gear = c("Trawl", "Pot", "Longline"),
  x_axis = c("Lat", "Lon"),
  facet_gear = FALSE,
  show_titles = TRUE,
  show_label = TRUE,
  show_counts = TRUE,
  size_range = c(0.1, 6)
)
```

Arguments

`data_o`	data.frame containing Observer catch data.
`data_em`	Optional data.frame containing EM catch data. If NULL or empty, the plot will be generated using Observer data only.
`species_name`	Character string giving the species name for the title.
`date_min`	Optional start date in `"mm/dd/yyyy"` format.
`date_max`	Optional end date in `"mm/dd/yyyy"` format.
`region`	Character vector specifying regions (e.g., `"AI"`, `"BS"`, `"GOA"`) or numeric NMFS area codes.
`gear`	Character vector of gear types to include (e.g., `"Trawl"`, `"Pot"`, `"Longline"`).
`x_axis`	Character string specifying the x-axis variable; one of `"Lat"` or `"Lon"`.
`facet_gear`	Logical; if TRUE, facet the plot by gear type.
`show_titles`	Logical; if TRUE, display the plot title.

show_label	Logical; if TRUE, display the upper-right label describing gear selection and x-axis variable.
show_counts	Logical; if TRUE, display total haul and vessel counts in the upper-left of each panel.
size_range	Numeric vector of length two giving the minimum and maximum point sizes used to scale catch weight.

Details

Depth is plotted in meters with a reversed y-axis. Catch weight is converted to metric tons where necessary. When longitude is selected for the x-axis, the function automatically chooses a continuous representation (either -180–180 or 0–360) to avoid artificial breaks near the dateline.

Value

A ggplot object.

Examples

```
## Not run:
plot_catch_depth_noaa_np_date(
  data_o = obs_data,
  data_em = em_data,
  species_name = "Pacific cod",
  date_min = "01/01/2020",
  date_max = "12/31/2023",
  region = c("BS"),
  gear = c("Trawl", "Longline"),
  x_axis = "Lon",
  facet_gear = TRUE
)

## End(Not run)
```

---

plot_catch_depth_noaa_np_grid_date

Plot gridded catch by depth vs latitude/longitude (date-range filter; dateline-safe)

Description

Aggregates observer catch into a 2D grid with depth binned in meters (default 10 m) and the x-axis (latitude or longitude) binned at a user-specified resolution in km. The plotted value is the sum of catch weight (metric tons) within each grid cell.

Usage

```
plot_catch_depth_noaa_np_grid_date(
  data_o,
  species_name,
  date_min = NULL,
  date_max = NULL,
  region = c("AI", "BS", "GOA"),
  gear = c("Trawl", "Pot", "Longline"),
  facet_gear = FALSE,
  x_axis = c("lat", "lon"),
  x_bin_km = 20,
  depth_bin_m = 10,
  show_titles = TRUE,
  show_label = TRUE
)
```

Arguments

data_o	Observer data.frame/data.table with required columns: GEAR_TYPE, LATDD_END, LONDD_END, RETRIEVAL_DATE, WEIGHT, NMFS_AREA, BOTTOM_DEPTH_FATHOMS.
species_name	Character scalar used in the plot title (if titles shown).
date_min	Optional start date (inclusive) as "mm/dd/yyyy".
date_max	Optional end date (inclusive) as "mm/dd/yyyy".
region	Vector: one or more of "AI","BS","GOA","BSWGOA", or numeric NMFS area code(s).
gear	Character vector: one or more of "Trawl","Pot","Longline". Default all.
facet_gear	If TRUE, facet by gear.
x_axis	Which horizontal axis to use: "lat" or "lon".
x_bin_km	Horizontal bin size in kilometers (default 20 km).
depth_bin_m	Depth bin size in meters (default 10 m).
show_titles	If FALSE, remove plot title/subtitle/caption.
show_label	If FALSE, remove the upper-right info label.

Details

This version is **dateline-safe** for Alaska data: if longitude spans 180°, the function automatically switches to a continuous longitude representation to avoid the large artificial break (wraps to 0–360 when that yields the smaller span).

Fill scale is aligned with your gridded map style: viridis + sqrt transform and legend title "Weight (mt)".

Value

A ggplot2 object.

`plot_catch_locations_noaa_np_date` *Plot catch locations on a NOAA-style North Pacific map (date-range filter)*

Description

Plots observer and EM catch locations with transparency scaled by recency and point size scaled by catch weight (metric tons). The map extent is automatically zoomed to the spatial extent of the data.

Usage

```
plot_catch_locations_noaa_np_date(
  data_o,
  data_em,
  species_name,
  date_min = NULL,
  date_max = NULL,
  region = c("AI", "BS", "GOA"),
  gear = c("Trawl", "Pot", "Longline"),
  size_range = c(1, 6),
  facet_gear = FALSE,
  show_titles = TRUE,
  show_label = TRUE,
  show_counts = TRUE,
  pad_frac = 0.08
)
```

Arguments

`data_o`	Observer data.frame (GET_CURRENT.sql output)
`data_em`	EM data.frame (GET_EM_CATCH.sql output)
`species_name`	Character string used in the plot title
`date_min`	Optional start date (inclusive) as "mm/dd/yyyy"
`date_max`	Optional end date (inclusive) as "mm/dd/yyyy"
`region`	Character vector: one or more of "AI","BS","GOA","BSWGOA"
`gear`	Character vector: one or more of "Trawl","Pot","Longline"
`size_range`	Numeric length-2 vector giving point size range (metric tons)
`facet_gear`	Logical; if TRUE, facet map by gear type
`show_titles`	Logical; if FALSE, suppress plot title
`show_label`	Logical; if FALSE, suppress upper-right gear label
`show_counts`	Logical; if TRUE, add haul/vessel counts upper-left
`pad_frac`	Fractional padding added around data extent (default 0.08)

Details

Observer points are always circles; EM points are always triangles. Robust to cases where one data source or one or more gear types are absent.

Adds upper-left annotations:

- Hauls: number of plotted rows (points)
- Vessels: number of unique vessel IDs across both sources (data_o$VESSEL and data_em$OBS_VESSEL_ID)

Value

A ggplot object

---

plot_catch_locations_noaa_np_grid_date

Plot gridded catch locations on a NOAA-style North Pacific map (date-range filter)

Description

Aggregates observer and EM catch data into a regular grid (default 20 km) in a North Pacific Lambert Azimuthal Equal-Area projection, then plots grid-cell totals on a NOAA-style basemap.

Usage

```
plot_catch_locations_noaa_np_grid_date(
  data_o,
  data_em,
  species_name,
  date_min = NULL,
  date_max = NULL,
  region = c("AI", "BS", "GOA"),
  gear = c("Trawl", "Pot", "Longline"),
  cell_km = 20,
  pad_frac = 0.08,
  facet_gear = FALSE,
  show_titles = TRUE,
  show_label = TRUE
)
```

Arguments

data_o	Observer data.frame
data_em	EM data.frame
species_name	Character string used in the plot title (if titles shown)
date_min	Optional start date (inclusive) as "mm/dd/yyyy"

date_max	Optional end date (inclusive) as "mm/dd/yyyy"
region	Vector: one or more of "AI","BS","GOA","BSWGOA", or NMFS area/s.
gear	Character vector: one or more of "Trawl","Pot","Longline". Default all.
cell_km	Grid cell size in kilometers (default 20)
pad_frac	Padding around data extent for grid creation (default 0.08)
facet_gear	If TRUE, facet the gridded map by gear (aggregates by GRID_ID + GEAR).
show_titles	If FALSE, remove plot title/subtitle/caption (clean figure).
show_label	If FALSE, remove the upper-right info label.

Details

Date filtering is applied (inclusive) using:

- Observer: RETRIEVAL_DATE
- EM: RETRIEVAL_END_DATE

Value

A ggplot object

---

plot_cumulative_catch_by_week

Plot cumulative catch by week with one line per year

Description

Uses WEEK_END_DATE (YYYY-MM-DD) to compute ISO week and year. Produces cumulative weekly catch curves with one line per year. The most recent year is always drawn in black and thicker.

Usage

```
plot_cumulative_catch_by_week(
  catch,
  region = NULL,
  areas = NULL,
  facet_gear = FALSE,
  gears = c("Trawl", "Pot", "Longline", "Jig"),
  show_titles = TRUE,
  title = NULL,
  y_units = "Cumulative catch (mt)",
  show_totals = TRUE
)
```

Arguments

catch	data.frame from GET_ALL_CATCH.sql
region	Optional region selector: "AI","BS","GOA","BSWGOA"
areas	Optional explicit area codes (overrides region)
facet_gear	Logical; if TRUE, facet by gear type
show_titles	Logical; if FALSE, remove title/subtitle
title	Optional plot title
y_units	Y-axis label
show_totals	Logical; if TRUE, annotate final-year catch in upper-left
gear	Character vector of gears to include after recoding

Details

Gear is taken from FMP_GEAR coded as: TRW = Trawl, POT = Pot, HAL = Longline, JIG = Jig

Weight is taken from WEIGHT_POSTED (metric tons). Area is taken from REPORTING_AREA_CODE.

Value

ggplot object

---

plot_length_frequency_noaa

Plot length-frequency composition (with region & gear options)

Description

Expects `lf` with columns: SPECIES, NMFS_AREA, GEAR, LENGTH, FREQUENCY

Usage

```
plot_length_frequency_noaa(
  lf,
  species_name = NULL,
  date_min = NULL,
  date_max = NULL,
  region = NULL,
  areas = NULL,
  gear = c("Trawl", "Pot", "Longline"),
  facet_gear = FALSE,
  show_titles = TRUE,
  show_label = TRUE,
  show_n = TRUE,
  y_free = TRUE
)
```

Arguments

lf	data.frame/data.table with length-frequency output
species_name	Optional species label for title
date_min	Optional start date (character "mm/dd/yyyy") for title only
date_max	Optional end date (character "mm/dd/yyyy") for title only
region	Optional region selector: "AI","BS","GOA","BSWGOA"
areas	Optional explicit NMFS area codes (overrides region)
facet_gear	If TRUE, facet by gear
show_titles	If FALSE, remove title entirely
show_label	If FALSE, remove upper-right label
show_n	If TRUE, add sample-size label(s) in the upper-left
y_free	If TRUE and faceting, allow free y-scales

Value

ggplot object

---

plot_observer_cpue *Plot Observer CPUE Time Series (Proportional Catch Method)*

Description

Generates CPUE indices by number and weight using observer data, where CPUE is weighted by the proportion of the selected species in total catch. Indices may be aggregated by year or by year × gear, with associated standard errors.

Usage

```
plot_observer_cpue(
  pulled,
  plot_type = c("MONTH", "GEAR", "YEAR"),
  region = c("BS", "GOA", "AI"),
  areas = NULL,
  gear = c("Trawl", "Pot", "Longline"),
  month_gear_facet = FALSE,
  base_size = 16
)
```

Arguments

plot_type	Character. Either "GEAR" (separate CPUE series by gear) or "Year" (aggregated across gear).
region	Character. Human-readable area label used in plot titles (e.g., "Bering Sea").
base_size	Numeric. Base font size passed to ggplot2::theme_bw().
data_o	data.frame or data.table Observer haul-level data containing species weight, effort, gear, year, and NMFS area.
species	Character or numeric. Species identifier used for labeling outputs.
code	data.frame. Lookup table containing observer program metadata (must include OBS_PROGRAM_NAME).

Details

This function is designed for use in in-season dashboards and assessment support tools. It assumes effort definitions differ by gear type (e.g., trawl duration vs hooks/pots).

CPUE indices are standardized by the mean CPUE within each gear and area combination prior to aggregation. Standard errors are propagated assuming independence.

Weight-based CPUE is in metric tons per unit effort.

Value

A named list with elements:

cpue data.frame containing CPUE indices and standard errors

plot_weight ggplot object for weight-based CPUE

plot_number ggplot object for number-based CPUE

See Also

get_observer_cpue_data, plot_cumulative_catch_by_week

---

run_inseason_dashboard

Run the inseason Shiny dashboard

Description

Launches the AFSC inseason dashboard bundled with this package. The app lives under inst/shiny/inseason/.

Usage

```
run_inseason_dashboard(launch.browser = TRUE, ...)
```

Arguments

launch.browser	Logical; open in a browser. Default TRUE.
...	Passed to shiny::runApp().

Value

Invisibly returns the Shiny app object.

Examples

```
## Not run:
run_inseason_dashboard()

## End(Not run)
```

Index