Package: ebswp (via r-universe)

May 19, 2026

Title Facilitates Model Running for EBS Pollock

Version 0.0.1.0000

Author Jim Ianelli [aut, cre]

Maintainer Jim Ianelli <`Jim.Ianelli@noaa.gov`>

Description Facilitates model running for EBS Pollock.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

URL `http://afsc-assessments.github.io/EBSpollock/`,
`http://afsc-assessments.github.io/EBSwp/`,
`http://afsc-assessments.github.io/ebswp/`,
`https://afsc-assessments.github.io/ebs_pollock/`

VignetteBuilder knitr

Depends R (>= 4.1.0), ggplot2, dplyr, xtable, ggthemes, here

Imports stringr

Config/pak/sysreqs libicu-dev

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

Date/Publication 2026-02-18 06:08:31 UTC

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

RemoteRef HEAD

RemoteSha 0420e77369383d7c929b164a234fb75e064c4fd2

Contents

Index **41**

`.get_ats_df` *Extract Acoustic Trawl Survey Data*

Description

Extracts data for either biomass or some other metric based on the given models in the list `M`. The function formats the data into a consistent structure for further processing.

Usage

```
.get_ats_df(M, biomass = TRUE)
```

Arguments

`M`	A list of model outputs. Each item should contain at least the elements `yr_ats`, `ob_ats`, `eb_ats`, `sd_ob_ats`, `ot_ats`, `et_ats`, and `sd_ot_ats`.
`biomass`	A logical flag. If `TRUE`, data extracted is for biomass; if `FALSE`, data for another metric is extracted. Default is `TRUE`.

Value

A dataframe with columns: year, Model, obs, pre, lb, and ub.

`.get_avo_df` *Get avo data into frame*

Description

This function iterates through a list of models and extracts year and associated measurements.

Usage

```
.get_avo_df(M)
```

Arguments

`M`	A list of models where each model has the needed attributes.

Value

A data frame with columns for year, model, observed values, predicted values, lower bounds, and upper bounds.

`.get_bts_df` *Extract bottom trawl survey data from a list of models*

Description

This function iterates through a list of models and extracts year and associated measurements.

Usage

```
.get_bts_df(M, biomass = TRUE)
```

Arguments

`M`	A list of models where each model has the needed attributes.
`biomass`	Logical indicating if biomass data should be used. If FALSE, other measurements are used.

Value

A data frame with columns for year, model, observed values, predicted values, lower bounds, and upper bounds.

`.get_cope_df` *Extract copE data*

Description

Extract copE data

Usage

```
.get_cope_df(M)
```

Arguments

`M`	list object created by read_admb function

Value

dataframe of spawning biomass

`.get_cpue_df` *Extract CPUE data*

Description

Extract CPUE data

Usage

```
.get_cpue_df(M)
```

Arguments

`M`	list object created by read_admb function

Value

dataframe of observed and predicted CPUE

`.get_like_df` *Extract Likelihood Data Frame*

Description

This function extracts likelihood values from a list of models and formats them into a structured data frame.

Usage

```
.get_like_df(M)
```

Arguments

`M`	A list of models where each model contains likelihood components.

Value

A data frame containing the likelihood components, model name, natural mortality value (M), and negative log likelihood (NLL) for each model in the list.

Examples

```
# Assuming 'model_list' is a list of models with the appropriate structure
# likelihood_df <- .get_like_df(model_list)
```

`.get_mnage_df` *Extract mean age from pollock model run*

Description

Extract mean age from pollock model run

Usage

```
.get_mnage_df(M)
```

Arguments

`M`	list object created by read_admb function

Value

dataframe of spawning biomass

`.get_Nage_3_df` *Extract Numbers at age 3*

Description

Extract Numbers at age 3

Usage

```
.get_Nage_3_df(M)
```

Arguments

`M`	list object created by read_admb function

Value

dataframe of age3+ biomass

`.get_R_rel_df` *Extract Recruits (R) from pollock run*

Description

Extract Recruits (R) from pollock run

Usage

```
.get_R_rel_df(M, rel = TRUE)
```

Arguments

`M`	list object created by read_admb function
`rel`	logical flag for relative recruitment

Value

dataframe of spawning biomass

`.get_recruitment_df` *Get recruitment data*

Description

Extracts predicted recruitment and approximate asymptotic error-bars

Usage

```
.get_recruitment_df(M)
```

Arguments

`M`	list object(s) created by read_admb function

Value

dataframe of recruitment

Author(s)

SJD Martell, DN Webber

`.get_ser_df` *Extract spawning exploitation rate*

Description

Extract spawning exploitation rate

Usage

```
.get_ser_df(M)
```

Arguments

`M`	list object created by read_admb function

Value

dataframe of spawning biomass

`.get_srr_df` *Extract stock recruitment results*

Description

Extract stock recruitment results

Usage

```
.get_srr_df(M)
```

Arguments

`M`	list object created by read_admb function

Value

dataframe of expected and observed SRR

`.get_ssb_df` *Extract spawning stock biomass (ssb)*

Description

Spawning biomass may be defined as all males or some combination of males and females

Usage

```
.get_ssb_df(M)
```

Arguments

`M`	list object created by read_admb function

Value

dataframe of spawning biomass

`.get_ssb_rel_df` *Extract spawning stock biomass (ssb)*

Description

Spawning biomass may be defined as all males or some combination of males and females

Usage

```
.get_ssb_rel_df(M)
```

Arguments

`M`	list object created by read_admb function

Value

dataframe of spawning biomass

`.run_mod` *run pollock model*

Description

run pollock model

Usage

```
.run_mod(moddir = mod_dir)
```

Arguments

`moddir`	directory where pollock model should be run

`.run_mod_windows` *run pollock model on windows*

Description

run pollock model on windows

Usage

```
.run_mod_windows(moddir = mod_dir)
```

Arguments

`moddir`	directory where pollock model should be run

`.run_proj` *run projection model DOESN"T WORK...dunno why*

Description

run projection model DOESN"T WORK...dunno why

Usage

```
.run_proj(moddir = mod_dir)
```

Arguments

`moddir`	directory where pollock model should be run

`.run_proj_windows` *run projection model for windows*

Description

run projection model for windows

Usage

```
.run_proj_windows(moddir = mod_dir)
```

Arguments

`moddir`	directory where pollock model should be run

`get_input_data` *Input Data for Model Configuration*

Description

A function that returns a list containing various parameters, settings, and data inputs required for model configuration. This includes start years, recruitment ages, maturity proportions, weights, catch data, survey data, and more.

Usage

```
get_input_data()
```

Value

A list with the following components:

styr Start year for the model (e.g., 1964).

styr_bts Start year for the bottom trawl survey (e.g., 1982).

styr_ats Start year for the acoustic trawl survey (e.g., 1994).

endyr End year for the model (e.g., 2024).

recage Age at recruitment (e.g., 1).

nages Number of ages in the model (e.g., 15).

p_mature Proportion mature at age (vector of length `nages`).

ew_ind End-weight index (vector of length `nages`).

wt_fsh Fishery weight-at-age matrix (rows = years, columns = ages).

wt_ssb Spawning stock biomass weight-at-age matrix (rows = years, columns = ages).

obs_catch Observed catch data (vector of length `n_fsh`).

obs_effort	Observed effort data (vector of length `n_fsh`).
n_cpue	Number of CPUE data points.
yrs_cpue	Years for CPUE data (vector of length `n_cpue`).
obs_cpue	Observed CPUE data (vector of length `n_cpue`).
obs_cpue_std	Standard deviation for CPUE data (vector of length `n_cpue`).
n_avo	Number of acoustic survey data points.
yrs_avo	Years for acoustic survey data (vector of length `n_avo`).
ob_avo	Observed acoustic survey data (vector of length `n_avo`).
ob_avo_std	Standard deviation for acoustic survey data (vector of length `n_avo`).
wt_avo	Weight-at-age matrix for acoustic survey (rows = years, columns = ages).
ngears	Number of gears.
minind	Minimum index for each gear (vector of length `ngears`).
n_fsh	Number of fishery years.
n_bts	Number of bottom trawl survey years.
n_ats	Number of acoustic trawl survey years.
yrs_fsh_data	Years for fishery data (vector of length `n_fsh`).
yrs_bts_data	Years for bottom trawl survey data (vector of length `n_bts`).
yrs_ats_data	Years for acoustic trawl survey data (vector of length `n_ats`).
sam_fsh	Sample size for fishery data (vector of length `n_fsh`).
sam_bts	Sample size for bottom trawl survey data (vector of length `n_bts`).
sam_ats	Sample size for acoustic trawl survey data (vector of length `n_ats`).
err_fsh	Error for fishery data (vector of length `n_fsh`).
err_bts	Error for bottom trawl survey data (vector of length `n_bts`).
err_ats	Error for acoustic trawl survey data (vector of length `n_ats`).
oac_fsh_data	Observed age composition for fishery data (matrix of size `n_fsh` x `nages`).
ob_bts	Observed bottom trawl survey biomass (vector of length `n_bts`).
ob_bts_std	Standard deviation for bottom trawl survey biomass (vector of length `n_bts`).
wt_bts	Weight-at-age matrix for bottom trawl survey (rows = years, columns = ages).
ot_bts	Observed bottom trawl survey numbers-at-age (matrix of size `n_bts` x `nages`).
oac_bts	Observed age composition for bottom trawl survey (matrix of size `n_bts` x `nages`).
ot_ats_std	Standard deviation for acoustic trawl survey numbers-at-age (vector of length `n_ats`).
oac_ats	Observed age composition for acoustic trawl survey (matrix of size `n_ats` x `nages`).
ob_ats	Observed acoustic trawl survey biomass (vector of length `n_ats`).
ob_ats_std	Standard deviation for acoustic trawl survey biomass (vector of length `n_ats`).
wt_ats	Weight-at-age matrix for acoustic trawl survey (rows = years, columns = ages).
bottom_temp	Bottom temperature data (vector of length `n_fsh`).
n_age_err	Number of ageing error matrices.
age_err	Ageing error matrix (matrix of size `nages` x `nages`).
nlbins	Number of length bins.
olc_fsh	Observed length composition for fishery data (vector of length `nlbins`).
age_len	Age-length transition matrix (matrix of size `nages` x `nlbins`).
test	Test value (e.g., 1234567).

Examples

```
## Not run:
input_data <- get_input_data()

## End(Not run)
```

`get_results` *Fetch Model Results in Parallel*

Description

This function fetches model results based on specified model names and directories.

Usage

```
get_results(
  mod_names. = mod_names,
  rundir = "runs",
  moddir = mod_dir,
  run_on_mac = TRUE
)
```

Arguments

`mod_names.`	A character vector of model names. Default is 'mod_names'.
`rundir`	The main sub directory path for the models. Default is 'runs'
`moddir`	The main directory path for the models. Default is 'mod_dir'
`run_on_mac`	Logical. Whether to use macOS-specific run behavior.

Value

A list containing model results.

`get_tier3_res` *Extract Tier 3 Results*

Description

Extracts results relevant to Tier 3 analyses.

Usage

```
get_tier3_res(proj_file)
```

Arguments

`proj_file`	A file path to the projection file.

Value

A list or vector of results relevant to Tier 3 analyses.

Examples

```
## Not run:
proj_file_path <- "path/to/proj.csv"
tier3_results <- get_tier3_res(proj_file_path)

## End(Not run)
```

`get_vars` *Extract and Compute Several Metrics from Model Output*

Description

This function processes the output of a certain model (possibly related to fisheries) to extract, compute, and format various metrics.

Usage

```
get_vars(M, proj_file = NULL, ord = dec_tab_ord)
```

Arguments

`M`	A list or data structure that holds the model output/results. Expected to have several named elements, including `fit`, `R`, `SSB`, etc.
`proj_file`	(Optional) A file path to a projection file for Tier 3 results. Default is NULL.
`ord`	The order and extent of reporting for the decision table aspect Default is dec_tab_ord.

Value

A list `B` that contains various extracted and computed metrics.

Examples

```
## Not run:
model_result <- list(...) # Example model result here
metrics <- get_vars(model_result)

## End(Not run)
```

`model_settings` *Model Settings List*

Description

A named list containing parameters and settings that control the behavior of a model. This list includes options for stock recruitment, selectivity, natural mortality, and more.

Usage

```
model_settings
```

Format

A list with the following components:

DoCovBTS stan's covariance matrix option.

SrType Stock Recruitment Type (1 = Ricker, 2 = Beverton-Holt, 3 = Constant, 4 = Old Ricker).

Do_Combined Do combined surveys (0 = No, 1 = Yes).

use_age_err Use ageing error matrix (0 = No, 1 = Yes).

use_age1_ATS Use Age1 ATS Index (0 = No, 1 = Yes).

age1_sigma_ATS Age1 ATS Index Sigma (log-normal distribution).

use_endyr_len Use end-year length data (0 = No, 1 = Yes).

use_popwts_ssb Use population weights for spawning (otherwise fishery weights) (0 = No, 1 = Yes).

natmortprior Natural mortality prior.

cvnatmortprior CV of natural mortality prior.

natmort_in Natural mortality at age (vector for ages 1 to 15).

q_all_prior Prior for all q values.

q_all_sigma Sigma for all q values (ignored if greater than 1).

q_bts_prior Prior for BTS q values.

q_bts_sigma Sigma for BTS q values (ignored if greater than 1).

sigrprior Prior for sigma R (based on 1978 onwards).

cvsigrprior CV of sigma R prior.

phase_sigr Phase for estimating sigma R.

steepnessprior Prior for steepness.

cvsteepnessprior CV of steepness prior.

phase_steepness Phase for estimating steepness.

use_spr_msy_pen Use prior as SPRF35 = Fmsy (0 = No, 1 = Yes).

sigma_spr_msy Sigma for SPRF35 = Fmsy.

use_last_ATS_ac Use last ATS age composition (0 = No, >0 = Yes).

nyrs_sel_avg Number of years to average recent selectivity.

do_bts_bio Use BTS Biomass (0 = Numbers, 1 = Biomass).

do_ATS_bio Use ATS Biomass (0 = Numbers, 1 = Biomass).

srprior_a Stock-Recruitment Prior Beta distribution alpha.

srprior_b Stock-Recruitment Prior Beta distribution beta.

nyrs_future Number of future years.

next_yrs_catch Next year's catch.

nscen Number of scenarios.

fixed_catch_fut2 Fixed catch in future scenario 2.

fixed_catch_fut3 Fixed catch in future scenario 3.

phase_F40 Phase to calculate F40%.

robust_phase Phase to start robustness.

ATS_robust_phase Phase for ATS robustness.

ATS_like_type Likelihood for ATS (0 = standard, 1 = log-normal for each age).

phase_logist_fsh Phase for fishery logistic selectivity.

phase_logist_bts Phase for BTS logistic selectivity.

phase_seldevs_fsh Phase for fishery selectivity deviations.

phase_seldevs_bts Phase for BTS selectivity deviations.

phase_age1devs_bts Phase for BTS Age1 deviations.

phase_selcoffs_ATS Phase for ATS survey selectivity coefficients.

phase_selcoffs_ATS_dev Phase for ATS survey selectivity deviations.

phase_natmort Phase for natural mortality estimation.

phase_q_bts Phase for BTS q estimation.

phase_q_std_area Phase for standard BTS area q estimation.

phase_q_ATS Phase for ATS q estimation.

phase_bt Phase for bottom temperature effect on q.

phase_rec_devs Phase for regular recruitment deviations.

phase_larv Phase for larval recruitment deviations.

phase_sr Phase to estimate stock-recruitment parameters (negative means constant recruitment).

wt_fut_phase Phase of future weight uncertainty.

last_age_sel_group_fsh Fishery: Number of oldest age groups to have the same selectivity.

last_age_sel_group_bts BTS: Number of oldest age groups to have the same selectivity.

last_age_sel_group_ATS ATS: Number of oldest age groups to have the same selectivity.

ctrl_flag A numeric vector of control flags, where each element corresponds to a specific setting:

  1 Catch Biomass.

  2 Survey Emphasis.

3 Recruitment Deviations.
4 Fishing mortality deviations (F_devs).
5 Lambda on ATS survey.
6 AVO (Acoustic Visual Observation) data.
7 Age Composition.
8 Age Composition Fishery.
9 Age Composition Survey Fishery.
10 Selectivity Trend (Fishery).
11 Selectivity Curvature (Fishery).
12 CPUE Data.
13 Fishery Selectivity Dome-shapedness.
14 Bottom Trawl Survey Dome-shapedness.
15 Non-Increasing Selectivity Penalty for Hydraulic Survey.
16 Number of years selectivity fishery stays the same.
17 Number of years selectivity stays the same in the survey.
18 Reserved.
19 Selectivity Deviation Surveys Curvature (Surveys).
20 BTS Time Variability.
21 BTS Smoothness (if nonparametric).
22 ATS Second-Difference.
23 Lambda on larval recruitment deviations.
24 Recruits from 1978 onwards only.
25 Ignore 1978 year class in estimation.
26 Reserved.
27 Third differencing in space (negative means no smoothing, positive = lambda).
28 Retrospective year.
29 Omit recent years from stock-recruitment estimation.
30 SRR Prior only (0 = use prior and data, otherwise weight on prior).

sel_dev_shift Selectivity block shift.

phase_coheff Phase for cohort effect.

phase_yreff Phase for year effect.

switch_temp_recruitment Switch for temperature-dependent recruitment (0 = No, 1 = Yes).

phase_temp_recruitment Phase for temperature-dependent recruitment.

switch_pred_mort Switch for predator mortality (0 = No, 1 = Yes).

phase_pred_mort Phase for predator mortality.

switch_multispecies_functional_response Switch for multispecies functional response (0 = No, 1 = Yes).

Examples

```
model_settings <- list(
  DoCovBTS = 1,
  SrType = 1,
  Do_Combined = 0,
  use_age_err = 1,
  use_age1_ATS = 1,
  age1_sigma_ATS = 1,
  use_endyr_len = 0,
  use_popwts_ssb = 0,
  natmortprior = 0.3,
  cvnatmortprior = 0.1,
 natmort_in = c(0.9, 0.45, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3),
  q_all_prior = 0,
  q_all_sigma = 2,
  q_bts_prior = 0,
  q_bts_sigma = 2,
  sigrprior = 1,
  cvsigrprior = 0.2,
  phase_sigr = -6,
  steepnessprior = 0.6,
  cvsteepnessprior = 0.12,
  phase_steepness = 5,
  use_spr_msy_pen = 0,
  sigma_spr_msy = 0.20,
  use_last_ATS_ac = 1,
  nyrs_sel_avg = 2,
  do_bts_bio = 1,
  do_ATS_bio = 1,
  srprior_a = 14.93209877,
  srprior_b = 14.93209877,
  nyrs_future = 5,
  next_yrs_catch = 1350,
  nscen = 8,
  fixed_catch_fut2 = 1400,
  fixed_catch_fut3 = 1200,
  phase_F40 = 6,
  robust_phase = 1350,
  ATS_robust_phase = 1350,
  ATS_like_type = 0,
  phase_logist_fsh = -1,
  phase_logist_bts = 2,
  phase_seldevs_fsh = 4,
  phase_seldevs_bts = 5,
  phase_age1devs_bts = 3,
  phase_selcoffs_ATS = 3,
  phase_selcoffs_ATS_dev = -5,
  phase_natmort = -6,
  phase_q_bts = 3,
  phase_q_std_area = -4,
  phase_q_ATS = 4,
  phase_bt = -6,
```

```
    phase_rec_devs = 3,
    phase_larv = -3,
    phase_sr = 5,
    wt_fut_phase = 6,
    last_age_sel_group_fsh = 4,
    last_age_sel_group_bts = 8,
    last_age_sel_group_ATS = 8,
   ctrl_flag = c(200, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 12.5, 1, 1, 1, 1, 1, 1, 3.125, 5, 0.1, 5, 1, 1, 2, 1, 0, 2, 1),
    sel_dev_shift = 0,
    phase_coheff = 1,
    phase_yreff = 1,
    switch_temp_recruitment = 0,
    phase_temp_recruitment = 6,
    switch_pred_mort = 0,
    phase_pred_mort = 1,
    switch_multispecies_functional_response = 1
  )
```

`plot_agefit` *Age Composition Data Plotter t(t() This function generates a series of plots visualizing the age composition data from fisheries assessment. It compares observed and predicted data across different years.*

Description

Age Composition Data Plotter t(t() This function generates a series of plots visualizing the age composition data from fisheries assessment. It compares observed and predicted data across different years.

Usage

```
plot_agefit(
  x,
  case_label = "2021 assessment",
  gear = "bts",
  type = "survey",
  styr = NULL,
  ageplus = NULL
)
```

Arguments

`x`	Model object or data needed to build age composition plots.
`case_label`	Label for the specific case or dataset being visualized. Default is "2021 assessment".
`gear`	Gear identifier (e.g., "bts").
`type`	Data type to plot (e.g., "survey").

`styr`	Start year for plotting (optional).
`ageplus`	Plus group age (optional).

Details

The function plots observed fishery age composition data using a bar plot, overlaying predicted data with points. It organizes the plots in a multi-panel figure, with each panel representing a year. Age classes are represented on the x-axis.

Value

A multi-panel plot where each panel visualizes age composition data for a specific year.

Note

The function uses the `rainbow` color palette, and the colors loop around for each cohort. The function expects certain naming conventions in the input dataset `dat`.

Examples

```
# Example usage (ensure you have an appropriate dataset 'mod1'):
# AgeFits(dat=mod1, main="Fishery Age Composition", case_label="Sample Assessment")
```

`plot_ats` *Plot Acoustic Trawl Survey Biomass Data*

Description

This function visualizes the acoustic trawl survey data, either biomass or another metric, depending on the choice. It provides options for customization of the resulting plot.

Usage

```
plot_ats(
  M,
  xlab = "Year",
  ylab = "Acoustic trawl survey biomass",
  xlim = NULL,
  ylim = NULL,
  alpha = 0.1,
  biomass = TRUE,
  color = "red",
  error_bars = TRUE
)
```

Arguments

`M`	List of model outputs. A list object like the one accepted by `.get_ats_df`.
`xlab`	Label for the x-axis. Default is "Year".
`ylab`	Label for the y-axis. Default is "Acoustic trawl survey biomass".
`xlim`	Optional range for the x-axis.
`ylim`	Optional range for the y-axis.
`alpha`	Opacity for the ribbon, if used. Default is 0.1.
`biomass`	Logical flag. If `TRUE`, plots data related to biomass; if `FALSE`, plots data related to another metric. Default is `TRUE`.
`color`	Color for the data points. Default is "red".
`error_bars`	Logical flag to draw error bars. Default is `TRUE`.

Value

A ggplot2 plot object visualizing the acoustic trawl survey data.

`plot_avo` *Plot avo data*

Description

This function takes a list of models and plots the observed, predicted and confidence intervals for the bottom trawl survey data.

Usage

```
plot_avo(
  M,
  xlab = "Year",
  ylab = "Acoustic return (Sa from AVO) ",
  ylim = NULL,
  error_bars = TRUE
)
```

Arguments

`M`	A list of models where each model has the needed attributes.
`xlab`	Label for the x-axis.
`ylab`	Label for the y-axis.
`ylim`	Limits for the y-axis.
`xlim`	Limits for the x-axis.
`color`	Color for the points.
`biomass`	Logical indicating if biomass data should be used. If FALSE, other measurements are used.
`error_bars`	Logical flag to draw error bars. Default is `TRUE`.

Value

A ggplot object with the plotted data.

`plot_bts` *Plot bottom trawl survey data*

Description

This function takes a list of models and plots the observed, predicted and confidence intervals for the bottom trawl survey data.

Usage

```
plot_bts(
  M,
  xlab = "Year",
  ylab = "Bottom trawl survey biomass",
  xlim = NULL,
  ylim = NULL,
  color = "purple",
  biomass = TRUE,
  error_bars = TRUE
)
```

Arguments

`M`	A list of models where each model has the needed attributes.
`xlab`	Label for the x-axis.
`ylab`	Label for the y-axis.
`xlim`	Limits for the x-axis.
`ylim`	Limits for the y-axis.
`color`	Color for the points.
`biomass`	Logical indicating if biomass data should be used. If FALSE, other measurements are used.
`error_bars`	Logical flag to draw error bars. Default is `TRUE`.

Value

A ggplot object with the plotted data.

`plot_cope` *Plot copepod something*

Description

Plot copepod something

Usage

```
plot_cope(
  M,
  xlab = "Year",
  ylab = "Copepod index ",
  ylim = NULL,
  color = "red"
)
```

Arguments

`M`	List object(s) created by read_admb function
`xlab`	the x-label of the figure
`ylab`	the y-label of the figure
`ylim`	is the upper limit of the figure
`color`	Color for the points.

Value

Plot of model estimates of spawning stock biomass

`plot_cpue` *Plot predicted CPUE*

Description

Plot predicted CPUE

Usage

```
plot_cpue(
  M,
  xlab = "Year",
  ylab = "Early trawl fishery CPUE",
  ylim = NULL,
  color = "red"
)
```

Arguments

`M`	List object(s) created by read_admb function
`xlab`	the x-label of the figure
`ylab`	the y-label of the figure
`ylim`	is the upper limit of the figure
`color`	Color for the points.

Value

Plot of model estimates of observed and predicted CPUE

`plot_likes` *Plot Likelihood Values*

Description

This function plots the negative log likelihood values for different components against a specified predictor (e.g., natural mortality). The function uses `ggplot2` for visualization.

Usage

```
plot_likes(
  M,
  xlab = "Natural Mortality",
  ylab = "relative -ln Likelihood",
  ylim = NULL,
  xlim = NULL,
  alpha = 0.1,
  legend = TRUE
)
```

Arguments

`M`	A list of models from which likelihood values will be extracted.
`xlab`	Label for the x-axis. Default is "Natural Mortality".
`ylab`	Label for the y-axis. Default is "relative -ln Likelihood".
`ylim`	Limits for the y-axis. Default is NULL.
`xlim`	Limits for the x-axis. Default is NULL.
`alpha`	Alpha transparency level for the plotted lines. Default is 0.1.
`legend`	Logical indicating whether to display a legend. Default is TRUE.

Value

A ggplot object displaying the negative log likelihood values.

Examples

```
# Assuming 'model_list' is a list of models with the appropriate structure
# plot_likes(model_list)
```

`plot_mnage` *Plot predicted mean age by gear type*

Description

Plot predicted mean age by gear type

Usage

```
plot_mnage(M, xlab = "Year", ylab = "Mean age", xlim = NULL, ylim = NULL)
```

Arguments

`M`	List object(s) created by read_admb function
`xlab`	the x-label of the figure
`ylab`	the y-label of the figure
`xlim`	is the year range to plot of the figure
`ylim`	is the upper limit of the figure

Value

Plot of model estimates of mean age against observed (and implied confidence bounds)

`plot_Nage_3` *Plot predicted Numbers at age 3*

Description

Plot predicted Numbers at age 3

Usage

```
plot_Nage_3(
  M,
  xlab = "Year",
  ylab = "Numbers at age 3",
  ylim = NULL,
  xlim = c(1990, 2020),
  breaks = seq(1990, 2022, 2),
  alpha = 0.8,
  legend = TRUE,
  order = NULL
)
```

Arguments

`M`	List object(s) created by read_admb function
`xlab`	the x-label of the figure
`ylab`	the y-label of the figure
`ylim`	is the upper limit of the figure
`xlim`	limits for the x-axis
`breaks`	tick marks for the x-axis
`alpha`	the opacity of the ribbon
`legend`	logical flag to draw a legend
`order`	optional ordering of model series

Value

Plot of model estimates of age3+ stock biomass

`plot_R_rel` *Plot predicted recruitment*

Description

Plot predicted recruitment

Usage

```
plot_R_rel(
  M,
  xlab = "Year",
  ylab = "Relative age-1 recruits",
  ylim = NULL,
  xlim = NULL,
  alpha = 0.1,
  legend = TRUE,
  rel = TRUE
)
```

Arguments

`M`	List object(s) created by read_admb function
`xlab`	the x-label of the figure
`ylab`	the y-label of the figure
`ylim`	is the upper limit of the figure
`xlim`	limits for the x-axis
`alpha`	the opacity of the ribbon
`legend`	logical flag to draw a legend
`rel`	logical flag for relative recruitment

Value

Plot of model estimates of spawning stock biomass

`plot_recruitment` *Plot predicted recruitment and approximate asymptotic error-bars*

Description

Plot predicted recruitment and approximate asymptotic error-bars

Usage

```
plot_recruitment(
  M,
  xlab = "Year",
  ylab = "Recruitment (millions)",
  xlim = c(1990.5, 2023.5),
  fatten = 0.8,
  fill = "yellow",
  alpha = 0.9
)
```

Arguments

`M`	list object created by read_admb function
`xlab`	the x-axis label for the plot
`ylab`	the y-axis label for the plot
`xlim`	limits for the x-axis
`fatten`	line width multiplier
`fill`	fill color for the ribbon
`alpha`	opacity for the ribbon

Value

Plot of predicted recruitment

Author(s)

SJD Martell, DN Webber

`plot_sel` *Plot selectivity*

Description

Plot selectivity

Usage

```
plot_sel(
  Year = M$Yr,
  sel = M$sel_fsh,
  styr = 1977,
  fage = NULL,
  lage = NULL,
  alpha = 0.2,
  scale = 3.8,
  fill = "purple"
)
```

Arguments

`Year`	Vector of years.
`sel`	Selectivity matrix or vector.
`styr`	Start year.
`fage`	First age to plot (optional).
`lage`	Last age to plot (optional).
`alpha`	the opacity of the ribbon
`scale`	Scaling factor for plotting.
`fill`	Fill color.

Value

Plot of model estimates of spawning stock biomass

`plot_ser` *Plot predicted spawning stock biomass (SER)*

Description

Plot predicted spawning stock biomass (SER)

Usage

```
plot_ser(
  M,
  xlab = "Year",
  ylab = "Spawning Exploitation rate",
  ylim = NULL,
  xlim = NULL,
  breaks = seq(1960, 2017, 5),
  alpha = 0.1,
  legend = TRUE
)
```

Arguments

`M`	List object(s) created by read_admb function
`xlab`	the x-label of the figure
`ylab`	the y-label of the figure
`ylim`	is the upper limit of the figure
`xlim`	limits for the x-axis
`breaks`	tick marks for the x-axis
`alpha`	the opacity of the ribbon
`legend`	logical flag to draw a legend

Value

Plot of model estimates of spawning stock biomass

`plot_srr` *Plot Stock-Recruitment Relationship (SRR)*

Description

This function plots the stock-recruitment relationship (SRR) using ggplot2. It can handle multiple models and provides flexibility in display options.

Usage

```
plot_srr(
  M,
  ylab = "Recruits (age 1, millions)",
  xlab = "Female spawning biomass (kt)",
  ylim = NULL,
  xlim = NULL,
  alpha = 0.05,
  ebar = "FALSE",
  leglabs = NULL,
  coverlap = FALSE,
  sizein = 3,
  sizeout = 2,
  yrsin = 1977:2019
)
```

Arguments

`M`	A list or data structure containing model results.
`ylab`	Label for the y-axis. Default is "Recruits (age 1, millions)".
`xlab`	Label for the x-axis. Default is "Female spawning biomass (kt)".
`ylim`	Limits for the y-axis. Default is `NULL`.
`xlim`	Limits for the x-axis. Default is `NULL`.
`alpha`	Alpha for the ribbons indicating uncertainty. Default is 0.05.
`ebar`	Logical, if `TRUE` error bars will be plotted. Default is `FALSE`.
`leglabs`	Custom labels for the legend. Default is `NULL`.
`coverlap`	Logical, if `TRUE` overlapping text labels will be checked and prevented. Default is `FALSE`.
`sizein`	Font size for the in-sample text labels. Default is 3.
`sizeout`	Font size for the out-of-sample text labels. Default is 2.
`yrsin`	Years to be considered for in-sample. Default is 1977:2019.

Value

A ggplot object containing the SRR plot.

Examples

```
# Assuming 'model_list' contains the relevant model results:
# plot_srr(M = model_list, ylim = c(0, 1e6), xlim = c(0, 2000))
```

`plot_ssb` *Plot predicted spawning stock biomass (ssb)*

Description

Spawning biomass may be defined as all males or some combination of males and females

Usage

```
plot_ssb(
  M,
  xlab = "Year",
  ylab = "Female spawning biomass (kt)",
  ylim = NULL,
  xlim = NULL,
  breaks = seq(1990, 2022, 2),
  alpha = 0.1,
  legend = TRUE,
  order = NULL
)
```

Arguments

`M`	List object(s) created by read_admb function
`xlab`	the x-label of the figure
`ylab`	the y-label of the figure
`ylim`	is the upper limit of the figure
`xlim`	limits for the x-axis
`breaks`	tick marks for the x-axis
`alpha`	the opacity of the ribbon
`legend`	logical flag to draw a legend
`order`	optional ordering of model series

Value

Plot of model estimates of spawning stock biomass

`plot_ssb_rel` *Plot predicted spawning stock biomass (ssb)*

Description

Spawning biomass may be defined as all males or some combination of males and females

Usage

```
plot_ssb_rel(
  M,
  xlab = "Year",
  ylab = "Relative female spawning biomass",
  ylim = NULL,
  xlim = NULL,
  legend = TRUE,
  breaks = seq(1990, 2022, 5),
  alpha = 0.1
)
```

Arguments

`M`	List object(s) created by read_admb function
`xlab`	the x-label of the figure
`ylab`	the y-label of the figure
`ylim`	is the upper limit of the figure
`xlim`	limits for the x-axis
`legend`	logical flag to draw a legend
`breaks`	tick marks for the x-axis
`alpha`	the opacity of the ribbon

Value

Plot of model estimates of spawning stock biomass

`print_Tier3_tbl` *Generate and Print Tier 3 Summary Tables*

Description

This function generates and prints summary tables for a specified model configuration. It calculates mean values for various metrics across different scenarios and displays formatted tables for Catch, Spawning Stock Biomass (SSB), Fishing Mortality (F), and Allowable Biological Catch (ABC).

Usage

```
print_Tier3_tbl(mod_number, run = FALSE)
```

Arguments

`mod_number`	Integer. Specifies the model configuration number.
`run`	Logical. If `TRUE`, runs the projection model for the specified configuration. Default is `FALSE`.

Details

The function reads projection results from a CSV file, groups data by alternative and year, calculates mean values for specific metrics, and formats these into tables for each metric. The tables are printed with captions based on the model configuration name.

Value

Prints formatted tables for Catch, SSB, F, and ABC. Does not return a value.

Examples

```
# Example usage
print_Tier3_tbl(mod_number = 1, run = TRUE)
```

`read_admb` *Read ADMB output files*

Description

Read ADMB output files .rep, .par, and .cor and return an R object of type ’list’

Usage

```
read_admb(repfile)
```

Arguments

`repfile`	ADMB output files to be read (no extension needed)

Value

object of type 'list' with ADMB outputs as list elements

`read_ctl` *Read ADMB .ctl file*

Description

Read ADMB .ctl file and return an R object of type 'list'. DOES NOT WORK

Usage

```
read_ctl(fn)
```

Arguments

`fn`	name of ADMB .ctl file to be read

Value

object of type 'list' with ADMB outputs therein

Author(s)

D'Arcy N. Webber

`read_dat` *Read ADMB Data File*

Description

Reads an ADMB data file and returns a list of outputs in R.

Usage

```
read_dat(fn)
```

Arguments

`fn`	Character. The name of the ADMB data file.

Value

A list containing various ADMB outputs from the data file.

`read_fit` *Read ADMB .par, .std, and .cor file and return an R object of type 'list' of estimates and correlations*

Description

Read ADMB .par, .std, and .cor file and return an R object of type 'list' of estimates and correlations

Usage

```
read_fit(repfile)
```

Arguments

`repfile`	name of ADMB output file to be read (no extension needed)

Value

object of type 'list' with ADMB outputs therein

Author(s)

Steve Martell, Anders Nielsen, Athol Whitten, D'Arcy N. Webber

---

`read_psv` *Read ADMB .psv file*

Description

Read ADMB .psv file and return an R object of type 'list'

Usage

```
read_psv(fn, nsamples = 10000)
```

Arguments

`fn`	name of ADMB .psv file to be read (no extension needed)
`nsamples`	number of posterior samples to read

Value

object of type 'list' with ADMB outputs therein

Author(s)

Steve Martell

`read_rep` *Read ADMB .rep file*

Description

Read ADMB .rep file and return an R object of type 'list'

Usage

```
read_rep(fn)
```

Arguments

`fn`	name of ADMB output file to be read (no extension needed)

Value

object of type "list" with ADMB outputs therein

Author(s)

Steve Martell, D'Arcy N. Webber

`rescale_cov` *Rescale a Covariance Matrix*

Description

This function rescales a covariance matrix using either the Cholesky decomposition method or a simple diagonal adjustment.

Usage

```
rescale_cov(Sigma, Scale_factor, DoChol = FALSE)
```

Arguments

`Sigma`	A covariance matrix that needs to be rescaled.
`Scale_factor`	A numeric value or vector used for scaling.
`DoChol`	Logical indicating if the Cholesky decomposition method should be used. If `FALSE`, the simple diagonal adjustment method is used. Default is `FALSE`.

Value

A rescaled covariance matrix.

`run_model` *Run Multiple ADMB Models in Parallel*

Description

This function uses parallel processing to run multiple ADMB models and gather their outputs. After running the models, it also fetches certain variables from a 'proj/spm_detail.csv' file associated with each model.

Usage

```
run_model(
  moddir = mod_dir,
  rundir = "runs",
  modnames = mod_names,
  Output = TRUE,
  run_on_mac = TRUE
)
```

Arguments

`moddir`	Character vector. The directory paths for the models, default is `mod_dir`.
`rundir`	Character string. The base run directory path for the models, default is `runs`.
`modnames`	Character vector. The names of the models, default is `mod_names`.
`Output`	Logical. A flag indicating if any output should be displayed (e.g., messages, progress, etc.), default is `FALSE`.
`run_on_mac`	Logical. Whether to use macOS-specific run behavior.

Value

A list containing the outputs of the ADMB models and the fetched variables.

`run_proj` *Run Multiple projection models in Parallel*

Description

This function uses parallel processing to run multiple ADMB models and gather their outputs. After running the models, it also fetches certain variables from a 'proj/spm_detail.csv' file associated with each model.

Usage

```
run_proj(
  moddir = mod_dir,
  rundir = "runs",
  modnames = mod_names,
  run_on_mac = TRUE
)
```

Arguments

`moddir`	Character vector. The directory paths for the models, default is `mod_dir`.
`rundir`	Character string. The base run directory path for the models, default is `runs`.
`modnames`	Character vector. The names of the models, default is `mod_names`.
`run_on_mac`	Logical. Whether to use macOS-specific run behavior.

`tab_fit` *Tabulate fits from models using extended metrics.*

Description

This function takes a list of model outputs and compiles a dataframe with a broader set of fit metrics.

Usage

```
tab_fit(M, mod_scen = NULL)
```

Arguments

`M`	List containing model outputs.
`mod_scen`	Optional vector of integers indicating which models in M to use. Default is all models.

Value

Returns a dataframe with extended fit metrics.

`tab_ref` *Tabulate fits from models using extended metrics.*

Description

This function takes a list of model outputs and compiles a dataframe with a broader set of fit metrics.

Usage

```
tab_ref(M, mod_scen = NULL)
```

Arguments

`M`	List containing model outputs.
`mod_scen`	Optional vector of integers indicating which models in M to use. Default is all models.

Value

Returns a dataframe with extended fit metrics.

`upperCaseFirst` *Helper function to convert the first character of a string to uppercase.*

Description

Helper function to convert the first character of a string to uppercase.

Usage

```
upperCaseFirst(s)
```

Arguments

`s`	The input string.

Value

Returns the input string with the first character capitalized.

`write_dat` *Write Data to Text File*

Description

This function takes a list of data elements (matrices, vectors, or other values) and writes them to a specified text file. Each data element in the list is preceded by a comment line indicating its name.

Usage

```
write_dat(output_file = "runs/dat/newfile.dat", indata = in_data)
```

Arguments

`output_file`	A character string specifying the name of the output text file. Default is "output.txt".
`indata`	A list containing the data elements to be written to the file. Each element can be a matrix, a vector, or any other value. Default is `in_data`.

Value

This function doesn't return a value; it writes to the specified output file.

Examples

```
# Assuming 'data_list' is a list of data elements
# write_dat(output_file = "sample_output.txt", indata = data_list)
```

Index