

Package: Rceattle (via r-universe)

October 26, 2026

Title Fits the Multispecies Assessment Model (CEATTLE) Using TMB

Version 5.66.1

Description Implements the CEATTLE model using Template Model Builder ('TMB'; Kristensen et al. 2015), which can be installed following <https://github.com/kaskr/adcomp/wiki/Download>. Structured similar to the original manuscript in terms of modularization. Separate functions estimate retrospective temperature- and size-specific predator rations, prey preference, and weight-at-age. These are then used as inputs to the CEATTLE model to evaluate how predation mortality, recruitment, and survival of three target species change under historical climate conditions and harvest rates.

URL <https://afsc-assessments.github.io/Rceattle/>,
<https://github.com/afsc-assessments/Rceattle>

BugReports <https://github.com/afsc-assessments/Rceattle/issues>

License GPL (>= 2)

Encoding UTF-8

LazyData true

LazyDataCompression xz

Roxygen list(markdown = TRUE)

Imports dplyr, ggplot2 (>= 3.5.0), MASS, readxl, TMB (>= 1.7.16), writexl, parallel, tidyr, R.utils, stats, graphics, grDevices, cowplot, rlang, reformulas, plyr, tidyselect, magrittr, Matrix, yaml

LinkingTo TMB, RcppEigen

Suggests testthat (>= 3.2.0), covr, rmarkdown, knitr, mvtnorm, pkgdown, TMBhelper, RcppEigen, tibble, withr

Remotes TMBhelper=github::kaskr/TMB_contrib_R/TMBhelper

VignetteBuilder knitr

Depends R (>= 4.1.0)

Config/testthat/edition 3

Config/testthat/parallel true

Config/testthat/start-first report-tables, golden-regression,
convergence-variance-collapse,
selectivity-nonparametric-integrable, fit-config-overlay,
functions-sim-mod-diet-roundtrip-dm,
functions-sim-mod-comp-fold,
functions-sim-mod-diet-roundtrip-mn,
functions-sim-mod-diet-dispersion, functions-retrospective

Config/roxygen2/version 8.1.0

Config/pak/sysreqs libicu-dev zlib1g-dev

Repository <https://noaa-fisheries-integrated-toolbox.r-universe.dev>

Date/Publication 2026-10-26 23:01:40 UTC

RemoteUrl <https://github.com/grantdadams/Rceattle>

RemoteRef HEAD

RemoteSha 2b86271b70cffe0710938fb4e56b6f287881b23

Contents

adjust_map_shared_params	5
as.data.frame.Rceattle	6
Atka2022	7
BS2017MS	7
BS2017SS	8
build_bounds	14
build_catchability	14
build_composition	15
build_data	16
build_growth	17
build_hcr	20
build_hcr_map	22
build_M1	23
build_map	25
build_map_catchability	26
build_map_debug	26
build_map_f_and_data_weights	27
build_map_fixed_natage	27
build_map_growth	28
build_map_m1	28
build_map_predation	29
build_map_recruitment	29
build_map_selectivity	30
build_params	31
build_selectivity	31
build_srr	33

check_mse	36
clean_data	36
coef.Rceattle	37
combine_data	37
compare_sim	38
convergence_diagnostics	38
data_requirements	39
EBS_ms_run	41
EBS_ss_M_run	41
EBS_ss_run	42
fit_control	42
fit_mod	44
GeorgesBank3spp	48
get_growth_matrix_r	48
get_weight_at_age_r	49
GOA2018SS	50
GOAatf	51
GOAatf2023	51
GOAcod	52
GOApollack	52
GOAsafe2018	52
jitter	53
linkage_spec	55
load_config	58
load_mse	58
logLik.Rceattle	59
model_average	60
model_config	61
mse_summary	62
NorthernRockfish2022	64
osa_diagnostics	64
osa_residuals	65
parameter_dictionary	68
parameter_index	69
plot.Rceattle	70
plot.rceattle_osa	71
plot.Rceattle_profile	72
plot_b_eaten	73
plot_b_eaten_prop	74
plot_biomass	76
plot_catch	78
plot_catchability	79
plot_comp	81
plot_data	82
plot_depletion	84
plot_depletionSSB	85
plot_diet_comp	86
plot_diet_comp1	86

plot_diet_comp2	87
plot_exploitable_biomass	88
plot_f	90
plot_form	92
plot_index	92
plot_index_residual	93
plot_indexresidual	94
plot_m_at_age	95
plot_m2_at_age_prop	96
plot_maturity	98
plot_mortality	99
plot_profile	100
plot_ratio	102
plot_recruitment	104
plot_selectivity	105
plot_selectivity_vs_maturity	107
plot_ssb	108
plot_ssb_depletion	110
plot_stock_recruit	112
plot_timeseries	113
print.Rceattle	115
print.Rceattle_data	116
print.Rceattle_jitter	117
print.Rceattle_mse_summary	117
print.Rceattle_profile	118
print.rceattle_report	119
print.Rceattle_retro	119
print.Rceattle_selftest	120
print.summary.Rceattle	121
prior_beta	121
prior_gamma	122
prior_lognormal	122
prior_normal	123
process_residuals	123
profile.Rceattle	124
profile_components	127
quantity_dictionary	129
rceattle-plot-args	130
rceattle-refit-args	132
read_data	133
rearrange_data	133
remove_F	134
rename_output	134
report_tables	135
residuals.Rceattle	137
retrospective	139
reweight_comps	141
rich.colors.short	143

run_config	143
run_mse	144
sample_rec	147
save_config	148
self_test	149
set_phases	151
sim_mod	152
simulate.Rceattle	153
standard_output	155
summary.Rceattle	156
switch_check	156
TMBAIC	157
TMBphase	157
vcov.Rceattle	158
whamGrowthData	159
write_data	159
write_template	160
Index	161

adjust_map_shared_params	<i>Helper to adjust map for shared catchability/selectivity indices</i>
--------------------------	---

Description

Enforces parameter sharing by mapping parameters for fleets with a common Selectivity_index or Catchability_index to the same value as the initial index.

Usage

```
adjust_map_shared_params(map_list, data_list)
```

Arguments

- map_list The current TMB map list.
- data_list The data list containing model settings.

Value

Updated map_list.

as.data.frame.Rceattle

Tidy long-format derived quantities from an Rceattle fit

Description

Returns the model's derived population quantities in long form so that custom plots and post-processing don't have to walk the deeply nested quantities list or inherit the dimnames decisions in [rename_output\(\)](#). Two shapes are flattened into one tidy frame: species-by-year quantities (e.g. biomass, ssb, R, F_spp) and species-by-sex-by-age-by-year quantities (e.g. N_at_age, biomass_at_age, M_at_age). For the species-level shapes, sex and age are returned as NA. Cells of the 4D arrays that are padded out to $\max(\text{nsex}) / \max(\text{nages})$ for species with fewer sexes or ages are dropped.

Usage

```
## S3 method for class 'Rceattle'
as.data.frame(
  x,
  row.names = NULL,
  optional = FALSE,
  which = c("biomass", "ssb", "R", "biomass_depletion", "ssb_depletion", "F_spp"),
  ci_level = 0.95,
  ...
)
```

Arguments

x	An object of class "Rceattle" returned by fit_mod() .
row.names, optional	Ignored; present for the as.data.frame() generic.
which	Character vector of quantity names to extract, or "all" for every quantity with a known shape. Defaults to a common population-level summary. See <code>names(Rceattle:::RCEATTLE_QUANTITIES)</code> for the full list.
ci_level	Confidence level for lwr and upr. Default 0.95.
...	Currently unused.

Details

Standard errors (se) and confidence intervals (lwr, upr) are populated from the TMB sdreport for any quantity that was ADREPORT'd (biomass, ssb, R, exploitable_biomass, biomass_depletion, ssb_depletion); other quantities and fits produced with `getsd = FALSE` get NA for se / lwr / upr. Set `ci_level` to widen or narrow the band.

A strictly positive series takes its interval on the log scale, so it is right-skewed and cannot reach zero. This is the same construction [plot_timeseries\(\)](#) draws, so the table and the figure agree.

Value

A `data.frame` with columns `year`, `species`, `sex`, `age`, `quantity`, `value`, `se`, `lwr`, `upr`. `species` is the character species name from `data_list$spnames`. Rows are sorted in the order which was given.

Atka2022	<i>Data inputs for Atka mackerel CEATTLE model (2022)</i>
----------	---

Description

A data list containing inputs for a single-species CEATTLE model fit to Aleutian Islands Atka mackerel data through 2022. See [BS2017SS](#) for format details.

Usage

Atka2022

Format

A list with the same structure as [BS2017SS](#).

BS2017MS	<i>Data inputs for multispecies CEATTLE of the Bering Sea from 1979 to 2017</i>
----------	---

Description

A data list containing inputs for the three-species (walleye pollock, Pacific cod, arrowtooth flounder) multispecies CEATTLE model for the Eastern Bering Sea. See [BS2017SS](#) for format details.

Usage

BS2017MS

Format

A list with the same structure as [BS2017SS](#).

BS2017SS

*Data inputs for single species CEATTLE of the Bering Sea from 1979 to 2017***Description**

A data list of inputs for the single-species Eastern Bering Sea CEATTLE model (walleye pollock, Pacific cod, and arrowtooth flounder), covering 1979-2017.

Usage

BS2017SS

Format

Control

nspp Number of species included in CEATTLE**styr** Start year of the hindcast**endyr** End year of the hindcast**projyr** End year of the forecast**nsex** Number of sexes to model in the population (1 = sexes combined, 2 = females and males modeled separately)**spawn_month** Spawning month of the population to adjust the numbers spawning**R_sex** Percent of recruitment that is female (ignored if nsex = 1)**nages** Number of ages of each species included in the hindcast**minage** Minimum age for each population (i.e. does recruitment correspond to age 0, 1, 2?)**nlengths** Number of lengths of each species included in the hindcast**pop_wt_index** Weight-at-age (weight) index to use for calculation of each species population derived quantities (SSB, Consumption/Ration, Suitability, etc)**ssb_wt_index** Weight-at-age index used to compute each species' female spawning-stock biomass (SSB)**pop_age_transition_index** Age transition matrix (e.g. growth trajectory) index to use for derived quantities of the population to convert age to length (also used in length-based predation estimation)**sigma_rec** Standard deviation of the log-scale recruitment deviations (fixed, or an initial value if estimated)**other_food** Other food in the ecosystem for each species, in metric tons: it is summed into the available-food denominator beside numbers-at-age (thousands) times weight-at-age (kg)**estDynamics** Estimate or fix numbers-at-age. Accepts integer codes or the equivalent readable strings: 0 / "Estimated" = estimate dynamics; 1 / "Fixed" = use input numbers-at-age in NByageFixed; 2 / "FixedScaled" = multiply input numbers-at-age (NByageFixed) by a single scaling coefficient, estimated under predation (msmMode > 0) and fixed at 1 otherwise.

M1_model Estimate residual (multi-species mode) or total natural mortality (single-species mode).
 0 = use fixed natural mortality from M1_base, 1 = estimate sex- and age-invariant M1, 2 = sex-specific (two-sex model), age-invariant M1, 3 = estimate sex- and age-specific M1.

fleet_control Survey and fishery data specifications

index_data Survey index in weight (mt) or numbers (thousands of fish) data. The Observation column is on the NATURAL scale in the units fleet_control\$observation_units names, never a log: the likelihood takes the log itself, and a fitted observation must be positive. A negative Year predicts the row without fitting it.

index_cov Optional named list of survey-index variance-covariance matrices, keyed by Fleet_name (or Fleet_code), used only by fleets whose fleet_control Index_distribution == "MVN". Each matrix must be square and symmetric with dimension equal to the number of fitted survey observations for that fleet (Year in styr:endyr, Observation > 0), ordered as in index_data. Inverted once internally to the precision matrix used in $0.5 * r' \Sigma^{-1} r$. Leave unset for the default lognormal likelihood.

catch_data Total catch in weight (mt) or numbers (thousands of fish) data

comp_data Survey/fishery age or length composition data. Note if sex is 3, put female composition data then male composition data (similar to SS).

emp_sel Empirical/fixed selectivity for surveys and fisheries (leave empty if not used)

age_trans_matrix Age transition matrix (e.g. growth trajectory) used to convert age to length for length comp data. Can have multiple matrices for a species specified by Age_transition_index.

age_error Ageing error matrices, one row per true age. One per species by default; give several and select them per fleet with the 'Ageing_error_index' column of both this table and fleet_control.

weight Weight-at-age (weight) to use for calculation of derived quantities (SSB, Consumption/Ration, Suitability, Total Catch, Survey Biomass, etc). Can have multiple weight-at-age data-sets for each species, but must be full for all years of the hindcast.

maturity Maturity-at-age for each species

sex_ratio Percent female at age for each species

M1_base Residual natural mortality for each species

aLW Parameters for weight-at-length power function for each species. Used when estimating time-variant length-based gamma suitability (suitMode = 2) or time-variant length-based lognormal suitability (suitMode = 5)

Ceq Which bioenergetics equation to use for each species for ft to scale max consumption: 1 = Exponential (Stewart et al 1983), 2 = Temperature-dependence for warm-water species (Kitchell et al 1977; sensu Holsman et al 2015), 3 = temperature dependence for cool and cold-water species (Thornton and Lessem 1979)

Cindex Which environmental index in env_data to use to drive bioenergetics

Pvalue This scales the maximum consumption used for ration for each species; Pvalue is in $C_{max}/TP_{value} * \text{ration_data}$

fday Number of foraging days per year for each species

CA Intercept of allometric mass function for calculating maximum consumption: $CA * \text{Weight}^{CB}$

CB Slope of allometric mass function for calculating maximum consumption: $CA * \text{Weight}^{CB}$

- Qc** Parameter for temperature scaling function of maximum consumption specified by *Ceq*
- Tco** Parameter for temperature scaling function of maximum consumption specified by *Ceq*
- Tcm** Parameter for temperature scaling function of maximum consumption specified by *Ceq*
- Tcl** Parameter for temperature scaling function of maximum consumption specified by *Ceq*
- CK1** Parameter for temperature scaling function of maximum consumption specified by *Ceq*
- CK4** Parameter for temperature scaling function of maximum consumption specified by *Ceq*
- env_data** Environmental indices such as bottom temperature data to incorporate into ration equation specified by *Ceq* and *Cindex*. Also used to drive catchability if *Catchability* = 5. Will use the mean for missing years. Temperature should be in Celsius.
- ration_data** Annual relative foraging rate by age or input consumption at age. Multiplied by *pvalue* and *fday* to scale maximum consumption to the number of days in a year that foraging occurs.
- UobsWtAge** Stomach proportion by weight for each predator, prey, predator age, prey age combination. Can also be year specific by including the column, "Year"
- fleet_control**: controls for survey and fisheries data
- Fleet_name** Name of survey or fishery
- Fleet_code** Index of survey/fishery ACROSS species
- Fleet_type** 0 or 'Off' = Do not estimate; 1 = 'Fishery'; 2 = 'Survey'
- Species** Species number
- Selectivity_index** Index used to give fleets the same selectivity (otherwise, same as *Fleet_code*). Fleets sharing a value share one selectivity parameter block, with its penalties and priors accumulated once on the group's first non-Off fleet. Sharing the parameters is not enough on its own: the columns that shape the curve are read per fleet and must agree across the group, or the fleets end up with different selectivities. *data_check()* reports any that differ. To mirror a fleet, copy its *fleet_control* row and change only the identity and catchability columns. See *vignette("model-options-and-functionality")*.
- Selectivity** Selectivity form: 0 = "Fixed"; 1 = "Logistic"; 2 = "NonParametric" (Ianelli et al. 2018); 3 = "DoubleLogistic"; 4 = "DescendingLogistic"; 5 = "Hake" non-parametric (Taylor et al. 2014); 6 = "2DAR1"; 7 = "3DAR1" (Cheng et al. 2024); 8 = "DoubleNormal"; 9 = "NonParametricPM" (non-parametric random walk with penalized increments, AMAK "pm"; not integrable under *random_sel*); 11 = "LogisticPM" (logistic with a free age-1 selectivity, AMAK "pm"); 13 = "NonParametricIntegrable" (the Ianelli base curve whose deviations have a proper density, so *random_sel* integrates them; *Time_varying_sel* picks the structure: "Off", "IID" or "RandomWalk"); 15 = "DoubleNormalSS3" (Stock Synthesis size pattern 24, six parameters on SS3's scales, varied through selectivity linkages). Whether a form is age- or length-based is set by *Selectivity_dimension*.
- Selectivity_dimension** "Age" or "Length".
- N_sel_bins** Number of bins (ages or length bins) to estimate for non-parametric selectivity.
- Time_varying_sel** Whether a time-varying selectivity should be estimated. 0 = "Off"; 1 = "IID" penalized deviates given *Time_varying_sel_sd*, or random effects with that sd estimated under *random_sel*; 2 = "AR1" REMOVED in 5.16.0, it was scored as "IID", so *data_check()* errors on it; use "IID" for the same fit, or *linkage_spec(~ ar1(1 | Year))* for a real one; 3 = "Block" time blocks with no penalty; 4 = "RandomWalk" following Dorn; 5 = "RandomWalkAscending" on the ascending limb of double logistic only.

Time_varying_sel_sd The fixed or initial sd to use for time varying selectivity; always required by the 2DAR1 and 3DAR1 forms.

Sel_start_year First year the fleet's time-varying selectivity deviations are estimated and penalized. Deviations before this year are fixed at 0: they have no data and no penalty (the selectivity penalties are anchored at this year), so estimating them would leave unidentified flat directions. Defaults to the earliest year of data across all fleets sharing the fleet's Selectivity_index (fleets sharing an index share one selectivity curve), bounded below by styr. Only used for time-varying selectivity; set explicitly to override.

Bin_first_selected Age/length bin at which selectivity is non-zero

Sel_shape_sd, Sel_curvature_sd, Sel_devmag_sd Selectivity penalties expressed as standard deviations (an intuitive alternative to the raw penalty weights in Sel_curve_pen1/2/3). Each penalty is a Gaussian SSQ, so the weight is $1 / (2 * sd^2)$: Sel_shape_sd is the shape/smoothness penalty (Sel_curve_pen1), Sel_curvature_sd the 2nd-difference curvature penalty (Sel_curve_pen2), Sel_devmag_sd the deviation-magnitude penalty (Sel_curve_pen3). Supplying an SD sets the corresponding Sel_curve_pen column; a legacy Sel_curve_pen value is never overwritten. Sel_shape_sd applies to NonParametric (2/9/13) and LogisticPM (11), Sel_devmag_sd to "NonParametricPM" (9) and "LogisticPM" (11) only, the slot being unread by "NonParametric" (2) and "NonParametricIntegrable" (13); Sel_curvature_sd is NonParametric-only (2/9/13; LogisticPM does not use Sel_curve_pen2). The 2DAR1/3DAR1 forms reuse Sel_curve_pen for logit-scale correlations and keep those columns.

Sel_shape_dir Direction of the shape penalty when Sel_shape_sd is used: "Decreasing" (default, penalize decreasing selectivity) or "Increasing" (sets the sign of Sel_curve_pen1). Only "NonParametricPM" (9) with Sel_shape_mode = "Directional" branches on the sign; every other form, that one under "Smooth" included, applies Sel_curve_pen1 as written, so "Increasing" is refused there rather than negating it.

Sel_curve_pen1 Raw shape/smoothness penalty weight for non-parametric (type 2/9/13) and LogisticPM (11) selectivity. The intuitive alternative is Sel_shape_sd, which sets this to $1 / (2 * sd^2)$. The 2DAR1/3DAR1 forms reuse this column as a logit-scale correlation.

Sel_curve_pen2 Raw 2nd-difference curvature penalty weight for non-parametric selectivity (the intuitive alternative is Sel_curvature_sd). Reused as a logit-scale correlation by the 2DAR1/3DAR1 forms.

Sel_curve_pen3 Raw deviation-magnitude penalty weight, read only by "NonParametricPM" (9) and "LogisticPM" (11) (the intuitive alternative is Sel_devmag_sd).

Sel_pen_first_bin First bin (age or length, on the fleet's Selectivity_dimension) for the non-parametric shape penalty. NA defaults to Bin_first_selected.

Sel_pen_last_bin Last (left) bin of the shape-penalty pairs. NA defaults to nbins - 2.

Sel_shape_mode Shape-penalty mode: "Directional" (default) or "Smooth" (two-sided second-difference penalty, RTMB).

Sel_avgsel_pen Weight on the AMAK average-selectivity base-level penalty (type 9 only): 0 = off (default), 10 matches AMAK.

Sel_cap_bin NonParametricPM selectivity bin cap. NA (default) applies no cap.

Sel_norm_bin Where selectivity is normalized to 1. "Max" normalizes by the largest value, "Off" (or blank, the default) not at all, and "All" on a LogisticPM fleet covers its whole penalty range. A number is an absolute age for an age-based fleet (6 means age 6, not the 6th bin) or a 1-based length-bin ordinal for a length-based one, and must lie in the range the fleet

is selected over; below it the curve is zeroed, so a value there is read as "Max". In a two-sex model this says only where the reference is taken, whether it is pooled across the sexes is `Sel_norm_scope`. See the sex-structure section of `vignette("model-options-and-functionality")`.

Sel_norm_bin_upper Optional upper age/length bin. "Off" (or blank, the default) normalizes at a single bin; a number normalizes by mean selectivity between `Sel_norm_bin` and this bin, on the same scale and within the same range.

Sel_norm_scope Whether selectivity normalization pools its reference across sexes, orthogonal to `Sel_norm_bin` (which says where the reference is taken). "WithinSex" divides each sex by its own reference, so both reach 1 and only the shape differs by sex; "AcrossSexes" (default) pools one reference over both sexes, so the less-selected sex stays below 1 and relative sex-specific selectivity is retained. No effect on a one-sex species or where `Sel_norm_bin` is NA.

Observation_units Units of the observation: 1 = weight (mt), 2 = numbers caught (thousands of fish). Drives both catch and index prediction.

Weight_index Weight-at-age (weight) index to use for calculation of derived quantities

Age_transition_index Age transition matrix (e.g. growth trajectory) index to use for derived quantities to convert age to length

Ageing_error_index Ageing error matrix index, matching `Ageing_error_index` in `age_error`. Fleets sharing a value read the same matrix. Omit the column, or leave it NA, to use the fleet's own species, which is what a model with one matrix per species has always done.

Catchability_index Index used to give fleets the same catchability (otherwise, same as `Fleet_code`). Fleets sharing a value share one q parameter, so the group holds only one answer to whether q is estimated: the group's first non-Off fleet decides for all of them, regardless of fleet type. Give a fleet its own value when it should be estimated independently. "Analytical" and "AnalyticalArith" are the exception, they solve q from each fleet's own index observations, so a group containing one does *not* share a catchability. `data_check()` reports that case, a Fixed lead that leaves fleets on different inits, and a `Catchability` or `Time_varying_q` that differs within a group. See `vignette("model-options-and-functionality")`.

Catchability Estimate catchability? 0 or "Fixed" = fixed at prior; 1 or "Estimated" = estimate a single parameter; 2 or "Estimated-with-prior" = estimate a single parameter with a prior; 3 or "Analytical" = geometric-mean analytical q (Walters and Ludwig 1994); 4 = power equation (not yet implemented); 5 or "Environmental" = linear equation $\log(q_y) = q_{\mu} + \beta * \text{index}_y$; 6 or "AR1" = removed, and `data_check()` errors on it, the deviates were never estimated, so q came back constant; express the Rogers et al. (2024) form as a q linkage with `ar1(1 | Year)` and observe, passed to `fit_mod(qFun = )`; 7 or "AnalyticalArith" = arithmetic-mean analytical $q = \text{mean}(\text{obs})/\text{mean}(\text{pred})$, the AMAK/ebswp form used with the MVN survey likelihood.

Catchability_init Starting value (or fixed value) for catchability. When `Catchability` = "Estimated-with-prior" this also centers the q prior.

Catchability_prior_sd Log-scale standard deviation of the q prior: `dnorm(log_q, log(Catchability_init) - bias_adjust_proc * sd^2/2, sd)`. Used only when `Catchability` = "Estimated-with-prior".

Time_varying_q Whether a time-varying q should be estimated. 0 = "Off"; 1 = "IID" penalized deviate or random effect; 2 = "AR1" REMOVED in 5.16.0, it was scored as "IID", so `data_check()` errors on it; use "IID" for the same fit, or `linkage_spec(~ ar1(1 | Year))` for a real one; 3 = "Block" time blocks with no penalty; 4 = "RandomWalk" from the mean

following Dorn 2018. Under `Catchability = "Environmental"` this column instead holds the `env_data` column index, not a mode.

Time_varying_q_sd The sd to use for the random walk of time varying q if set to 1

Estimate_index_sd Estimate survey/index observation SD. Accepts integer codes or readable strings: 0 / "Fixed" = use the fixed SD from the data CV; 1 / "Estimated" = estimate; 2 / "Analytical" = analytically estimate following Walters and Ludwig 1994.

Index_sd Starting/fixed value for the index observation SD, used when `Estimate_index_sd = "Estimated"`.

Estimate_catch_sd Estimate fishery/catch observation SD. Accepts integer codes or readable strings: 0 / "Fixed" = use the fixed SD from the data CV; 1 / "Estimated" = estimate; 2 / "Analytical" = analytically estimate following Walters and Ludwig 1994.

Catch_sd Starting/fixed value for the catch observation SD, used when `Estimate_catch_sd = "Estimated"`.

Index_distribution Survey/index biomass observation likelihood family. 0 or "Lognormal" (default) = independent lognormal on the log observation. 1 or "MVN" and 2 or "MVNORM" = multivariate normal on the natural-scale residual ($\text{obs} - q \cdot \text{pred}$), with a user-supplied covariance in `index_cov` (e.g. a VAST-derived Sigma); "MVN" reports the bare quadratic form matching AMAK/ebswp and ADMB, "MVNORM" the properly normalized density, and the two give an identical fit. Pair either with `Catchability = "AnalyticalArith"`. 3 or "Normal" = independent normal on the same residual, with an ABSOLUTE sd from `Log_sd`, not a log-scale CV; reproduces AMAK `avo_like/cpue_like` term for term. 4 or "TruncatedNormal" = that normal left-truncated at zero. Prefer it to "Normal" unless an exact ADMB comparison is needed: an index cannot be negative, so it is the only natural-scale family whose likelihood and simulator are the same distribution. See vignette("model-options-and-functionality").

Comp_distribution Age/length composition data distribution. Accepts integer codes or readable strings: -1 or "MultinomialAFSC" (default) = AFSC multinomial; 0 or "Multinomial" = full multinomial; 1 or "DirichletMultinomial".

Comp_weights Composition weight. Its scale follows `Comp_distribution`: under a multinomial it is the natural-scale multiplier on the input sample size, while under a Dirichlet-multinomial the model uses $\exp(\text{Comp_weights})$, so the column holds the *log* of the starting weight, a value of 1 is a starting weight of e ; use 0 for a weight of 1. It is read when a model is built from scratch, so a refit keeps whatever weight it is given. A fit reports the weight its residuals imply, a harmonic mean of the ratio of effective to input sample size (McAllister & Ianelli 1997), in `Comp_weights_mcallister`; [reweight_comps](#) tunes towards it.

CAAL_distribution Conditional age-at-length composition distribution. Accepts integer codes or readable strings: -1 or "MultinomialAFSC" = AFSC multinomial; 0 or "Multinomial" (default) = full multinomial; 1 or "DirichletMultinomial".

CAAL_weights Conditional age-at-length (CAAL) composition weight. Its scale follows `CAAL_distribution` exactly as `Comp_weights` follows `Comp_distribution`: a natural-scale sample-size multiplier under a multinomial, the *log* of the starting weight under a Dirichlet-multinomial.

Comp_accum_young Young-tail composition accumulation bin (AFSC `ac_yng`): a 1-based ordinal on the fleet's composition dimension; age/length bins below it are folded into it before the likelihood. NA (default) or 1 = no young accumulation.

Comp_accum_old Old-tail composition accumulation bin (AFSC `ac_old`): a 1-based ordinal on the fleet's composition dimension; age/length bins above it are folded into it before the likelihood. NA (default), 0, or a value at/above the number of bins = no old accumulation.

Month Observation month for the fleet (0 = not specified).

Proj_F_proportion The proportion of future fishing mortality assigned to this fleet.

build_bounds	<i>Build parameter bounds</i>
--------------	-------------------------------

Description

Function to build parameter bounds based on Holsman et al 2015 and Kinzey and Punt 2010

Usage

```
build_bounds(param_list = NULL, data_list)
```

Arguments

param_list	Parameter list object built from build_params
data_list	a Rceattle data object

Value

List of upper and lower bounds

build_catchability	<i>Catchability specification</i>
--------------------	-----------------------------------

Description

Holds environmental linkages on survey/index catchability q. The effect of an env_data covariate is written as a formula and can hold priors, bounds, and an estimation phase like any other linkage.

Usage

```
build_catchability(linkages = NULL)
```

Arguments

linkages	Optional named list of linkage_spec() objects keyed by catchability parameter. The only parameter is q. Coefficients are per fleet by default (by = ~ fleet), except that fleets sharing a Catchability_index estimate one parameter block and so take ONE coefficient for the whole group. Use the fleet argument of linkage_spec() to restrict a spec to particular fleets.
----------	---

Details

Catchability is the one process that accepts `link = "exponential"`, Stock Synthesis's environmental link type 1: $q^{\exp(\beta * x)}$, which MULTIPLIES $\log q$ where "log" shifts it. It needs an estimated base q and a lognormal or t index likelihood, and cannot share a fleet with a random-effect q linkage; each of those is refused with the reason. See `vignette("environmental-linkages-and-priors")`.

Value

A list of catchability settings for `fit_mod()`.

Examples

```
# One temperature effect on q per fleet. `by = ~ fleet` is the default, and a
# fleet without an estimated survey catchability is not linkable, so at fit time
# restrict the linkage to the estimated-q fleets (here fleet 7 of BS2017SS).
build_catchability(linkages = list(
  q = linkage_spec(~ temp, fleet = 7)))

# Restrict it to fleets 1 and 3, with a prior on the slope
build_catchability(linkages = list(
  q = linkage_spec(~ temp, fleet = c(1, 3),
    priors = list(temp = prior_normal(0, 1))))))
```

build_composition	<i>Composition-weighting specification</i>
-------------------	--

Description

Holds **priors** on the Dirichlet-multinomial composition-weighting overdispersion. The DM weight (the "theta" that scales the effective sample size) is otherwise an unpenalized free parameter; a linkage lets you put a prior on it through the same grammar as every other parameter. The three parameters target the three DM likelihoods:

`theta_comp` age / length composition weight (`comp_weights`), per fleet.
`theta_caal` conditional-age-at-length weight (`caal_weights`), per fleet.
`theta_diet` diet composition weight (`diet_comp_weights`), per predator.

This is a **prior-only** process: the DM weight is a scalar, not a year-varying quantity, so only intercept formulas (~ 1) with a `priors` entry are meaningful. The prior is placed on the natural-scale DM weight $\theta = \exp(\text{weight})$, so a `gamma()` prior reads naturally. A linkage on a fleet whose `Comp_distribution` (or `CAAL_distribution`) is not "DirichletMultinomial" errors, since the weight has no effect there.

Usage

```
build_composition(linkages = NULL)
```

Arguments

linkages Optional named list of `linkage_spec()` objects keyed by `theta_comp` / `theta_caal` (per fleet by default, by = ~ fleet) or `theta_diet` (per predator by default, by = ~ species).

Value

A list of composition-weighting settings for `fit_mod()`.

Examples

```
# A weak gamma prior on the DM overdispersion of every fleet's age comps
build_composition(linkages = list(
  theta_comp = linkage_spec(~ 1, by = ~ fleet,
    priors = list(`(Intercept)` = gamma(2, 0.5))))))
```

 build_data

Build an Rceattle data list in R

Description

Assemble (or edit) an Rceattle data list in R code rather than from an xlsx workbook. Supply only the data blocks a model uses, dimensions, biology (weight, maturity, sex_ratio, M1_base), a fleet_control, and the observation tables (catch_data, index_data, comp_data, ...), and the optional blocks a single-species model does not need (caal_data, emp_sel, diet_data, ...) are default-filled by `clean_data()`. The result is the same bare list `read_data()` returns and round-trips through `write_data()` unchanged.

Usage

```
build_data(base = NULL, file = NULL, ..., .check = TRUE)
```

Arguments

base Optional existing Rceattle data list to start from and override.

file Optional path to an Rceattle xlsx workbook; read via `read_data()` to form the starting object. Supply at most one of base / file.

... Named top-level data-list elements to set or override (e.g. nspp, styr, endyr, fleet_control, weight, maturity, catch_data, index_data, comp_data). Every argument must be named.

.check Logical; run the presence pre-check (default TRUE). Set FALSE to assemble a deliberately partial object.

Details

Three entry points, which may be combined:

- **from blocks:** pass the elements as named arguments: `build_data(nspp = 1, styr = 1977, ..., fleet_control = fc, catch_data = catch)`.
- **from a file:** `build_data(file = "model.xlsx", projyr = 2060)` reads the workbook via `read_data()`, then applies the overrides.
- **from an existing object:** `build_data(base = BS2017SS, projyr = 2060)` starts from a data list (e.g. a bundled dataset or a `combine_data()` result) and overrides the named blocks. This is the common copy-and-edit / combine-and-restamp workflow.

Element names are checked against the recognized schema: an override whose name is a near-miss of a known element (e.g. `maturty`) errors with a suggestion, so a typo is caught here rather than surfacing much later in a fit. Legacy top-level names (`fsh_biom`, `srv_biom`, `wt`, `pmature`, `PyrS`) are mapped to their canonical equivalents.

Full validation runs at fit time, inside `fit_mod()`. `build_data()` runs only a light presence pre-check (`.check = TRUE`) so a missing *required* block is reported at construction with a clear message. The pre-check reads an attached `model_config()`, so a configuration held on the object is accounted for. Requirements that depend on fit-time settings passed directly to `fit_mod()` and stored nowhere on the data list are not knowable here and are left to that fit-time validation; see `data_requirements()` to preview them.

Value

An Rceattle data list (a bare list, as from `read_data()`).

See Also

`read_data()`, `clean_data()`, `data_requirements()`, `combine_data()`, `fit_mod()`.

Examples

```
# Copy-and-edit a bundled dataset.
dat <- build_data(base = BS2017SS, projyr = 2060)

# Preview what a configuration needs before building it.
data_requirements(msmMode = 0)
```

build_growth

Specify the growth model for Rceattle

Description

Specify the growth model for Rceattle

Usage

```
build_growth(
  fun = "empirical",
  growth_age_L1 = NA,
  sd_plus_group = NA,
  linkages = NULL,
  sd_form = NA,
  plus_group_length = NA,
  plus_group_decay = NA,
  pop_lengths = NULL
)
```

Arguments

fun	Growth function: a string (GROWTH_FUNS : "empirical" (default), "vonBertalanffy", "Richards") or the equivalent integer code (0, 1, 2). The canonical string is stored on the returned object.
growth_age_L1	Von Bertalanffy / Richards anchor age, where mean length equals L1; SS3's Growth_Age_for_L1. Falls back to $\max(0.5, \text{minage}[\text{sp}])$, which keeps minage ≥ 1 models backwards-compatible and gives minage = 0 models an SS3-consistent half-year anchor.
sd_plus_group	How the oldest age class's SD-at-age is treated, and only for estimated growth. "WHAM" (the fallback) pins the plus-group SD to the upper anchor $\exp(\text{sd_Linf})$, the WHAM SDAA convention; "SS3" interpolates it by length like any interior age. Integer codes 1/2 are accepted. Where SS3's Growth_Age_for_L2 is 999 it pins the plus group to the upper anchor, which is "WHAM" here.
linkages	Optional named list of linkage_spec() objects keyed by parameter name (one of GROWTH_LINKAGE_PARAMS). The mean-growth keys (K, L1, Linf, m) take arbitrary one-sided formulas and make that parameter year-varying, as a per-year offset around its mean. The SD-endpoint keys (sd_L1, sd_Linf) honour only intercept-bearing formulas, typically ~ 1 , threading init, bounds and priors onto the growth SD-at-age so the SDs get the same prior, fix and starting-value contract as the means. A slope row on an SD spec warns and has no effect; a slope-only formula ($\sim 0 + \text{temp}$) is an error.
sd_form	What the two variability endpoints (sd_L1, sd_Linf) are: "SD" (the fallback), standard deviations of length-at-age in cm, SS3's CV_Growth_Pattern 2; or "CV", coefficients of variation, so the SD is CV x mean length, SS3 pattern 0.
plus_group_length	How the plus group's mean length is set: "M1" (the fallback), "none", "SS3.24" or "decay". See Details.
plus_group_decay	Annual decay rate for plus_group_length = "decay", SS3's positive Linf_decay, roughly the plus group's total mortality. Required for, and used only by, "decay".
pop_lengths	Population length bins (lower edges, cm) on which the age-length key, weight-at-length and maturity-at-length are computed before being summed into the data length bins: SS3's population length bins. Every data-bin lower edge must

also be a population-bin edge. A vector applies to every species, a list gives one per species, and the fallback is the data bins.

Details

Every switch above takes a scalar, recycled across species, or a length-nspp vector for per-species values. Each defaults to NA, which inherits the matching field from `data_list` when it is set – so a refit, or a model built by the SS3 converter, keeps the original choice – and otherwise uses the fallback named with it.

Plus-group mean length. Fish older than the oldest age are pooled, so the plus group's mean length lies between the growth curve at the oldest age, L_A , and L -infinity:

- "M1": mean of $L_A + (a/n)(L_{\infty} - L_A)$ over $a = 0..n$ ($n = \text{nages}$), weighted by survival at the oldest age's base M1.
- "none": L_A (SS3 `Linf_decay = -998`).
- "SS3.24": as "M1" with weights $\exp(-0.2 a)$ and $a = 0..A$, A the oldest age (SS3 `Linf_decay = -999`).
- "decay": L_A and $2A$ further ages, each one more year along the von Bertalanffy curve, weighted by $\exp(-d a)$ with $d = \text{plus_group_decay}$ (SS3 `Linf_decay = d`).

Under "none", "SS3.24" and "decay" the plus group also grows within the year like every other age, as SS3 does; under "M1" it keeps its Jan-1 length through the year.

Maturity-at-length is set in the data, not here: the per-species control columns `L50_mat_len` and `slope_mat_len`.

Value

A list of switches defining the growth model.

Examples

```
# Sex-specific von Bertalanffy with temperature on K, by species + sex
build_growth(
  fun = "vonBertalanffy", # or fun = 1
  linkages = list(
    K = linkage_spec(
      formula = ~ temp,
      by      = ~ species + sex,
      priors  = list(temp = normal(0, 1))
    )
  )
)
```

build_hcr

*Specify the harvest control rule (HCR) used for Rceattle***Description**

Defines the harvest control rule and its associated reference points. The rules are tabulated, with which ones work under predation, in `vignette("projections-and-reference-points")`; the formulations of the three ramped rules are below.

Usage

```
build_hcr(
  HCR = 0,
  DynamicHCR = FALSE,
  Ftarget = 0.4,
  Flimit = 0.35,
  Ptarget = 0.4,
  Plimit = 0,
  Alpha = 0.05,
  Pstar = 0.45,
  Sigma = 0.5,
  Fmult = 1,
  HCRorder = 1
)
```

Arguments

HCR	Harvest control rule, as an integer or string alias: 0/"NoFishing" (no catch, hindcast only), 1/"CMSY" (maximize joint catch, optionally holding depletion at or above Plimit), 2/"ConstantF" (constant input F at Ftarget), 3/"ConstantFSSB" (the F reaching Ftarget percent of SSB0 by the end of the projection), 4/"ConstantFSpr" (constant Fspr, scalable by Fmult following NEFSC), 5/"NPFMC", 6/"PFMC" or 7/"SESSF". Default 0. Only 0, 1, 2, 3 and 6 work in multispecies mode.
DynamicHCR	TRUE/FALSE. Use dynamic rather than static reference points (default FALSE).
Ftarget	Target fishing mortality rate (yr^{-1}), SPR- or depletion-based, or the input F for projections. For SPR F40%, enter 0.40.
Flimit	Limit fishing mortality rate (yr^{-1}), SPR- or depletion-based. For SPR F35%, enter 0.35.
Ptarget	Target spawning-stock biomass as a proportion of static or dynamic spawning-stock biomass at F = 0. For B40%, enter 0.40.
Plimit	Limit spawning-stock biomass on the same basis. For B10%, enter 0.10.
Alpha	Parameter used in the NPFMC Tier 3 rule.
Pstar	Quantile for the uncertainty buffer, given Flimit and Sigma.
Sigma	Standard deviation of the normal uncertainty buffer, given Flimit and Pstar.

Fmult	Multiplier on the target F (default 1), scaling it following NEFSC convention.
HCRorder	For multispecies models, the order in which to project fishing (predators first, then prey, for instance).

Details

hcr = 5 or "NPFMC", the North Pacific Fishery Management Council Tier 3 spawner-per-recruit rule:
 Stock status: $SB > SB_{BatFtarget}$

$$F_{ofl} = F_{limit}$$

$$F_{use} = F_{target}$$

Stock status: $Alpha < SB/SB_{BatFtarget} \leq 1$

$$F_{ofl} = F_{limit} * (SB/SB_{Ftarget} - Alpha)/(1 - Alpha)$$

$$F_{use} = F_{target} * (SB/SB_{Ftarget} - Alpha)/(1 - Alpha)$$

Stock status: $SB/SB_{BatFtarget} \leq Alpha$ or $SB < P_{limit} * SB_0$

$$F_{ofl} = 0$$

$$F_{use} = 0$$

hcr = 6 or "PFMC", the Pacific Fishery Management Council category 1 40-10 annual catch limit rule (PFMC 2020), using Fspr in single species or the F reaching X% of SSB0 under predation. The uncertainty buffer is $qnorm(Pstar, Flimit, Sigma)$, which equals $Flimit + qnorm(Pstar, 0, Sigma)$ and so sits *below* $Flimit$ when $Pstar < 0.5$. The taper runs between $SB_0 * P_{limit}$ and $SB_0 * P_{target}$, so the 40-10 shape needs $P_{target} = 0.40$ and $P_{limit} = 0.10$; the default $P_{limit} = 0$ gives a 40-0 rule. Stock status: $SB > SB_0 * P_{target}$

$$F_{ofl} = F_{limit}$$

$$F_{use} = qnorm(Pstar, Flimit, Sigma)$$

Stock status: $SB_0 * P_{limit} < SB \leq SB_0 * P_{target}$

$$F_{ofl} = F_{limit}$$

$$F_{use} = qnorm(Pstar, Flimit, Sigma) * \frac{SB_0 * P_{target} * (SB - SB_0 * P_{limit})}{SB * SB_0 * (P_{target} - P_{limit})}$$

Stock status: $SB < SB_0 * P_{limit}$

$$F_{ofl} = 0$$

$$F_{use} = 0$$

hcr = 7 or "SESSF", the Southern and Eastern Scalefish and Shark Fishery Tier 1 spawner-per-recruit rule, with $F_{Limit} = F_{(20\%)}$ and $B_{Limit} = SB_{20}$ (AFMA 2017): Stock status: $SB > SB_0 * P_{target}$

$$F_{ofl} = F_{limit}$$

$$F_{use} = F_{target}$$

Stock status: $SB0 * P_{target} > SB > SB0 * P_{limit}$

$$F_{ofl} = F_{limit} * (SB / (SB0 * P_{limit}) - 1)$$

$$F_{use} = F_{target} * (SB / (SB0 * P_{limit}) - 1)$$

Stock status: $SB < SB0 * P_{limit}$

$$F_{ofl} = 0$$

$$F_{use} = 0$$

Value

A list containing the harvest control rule and associated biological reference points.

Examples

```
# Tier 3 NPFMC control rule: F40% as FABC, F35% as FOFL.
build_hcr(HCR = "NPFMC", DynamicHCR = FALSE, Ftarget = 0.4, Flimit = 0.35)

# No fishing: the projection a B0 reference point is read from.
build_hcr(HCR = "NoFishing")
```

build_hcr_map	<i>Function to construct the TMB map argument for CEATTLE for projecting under alternative harvest control rules</i>
---------------	--

Description

Reads a data list and map to update the map argument based on the HCR specified in [build_hcr](#)

Usage

```
build_hcr_map(
  data_list,
  map,
  debug = FALSE,
  all_params_on = FALSE,
  HCRiter = 1
)
```

Arguments

data_list	an Rceattle data_list
map	a map object created from build_map .
debug	logical. If TRUE, turns off all parameters for debugging (default = FALSE).
all_params_on	logical. If TRUE, leaves all hindcast parameters turned on (default = FALSE).
HCRiter	for multi-species models, the order in which to project fishing (e.g. predators first, then prey)

Value

a list of map arguments for each parameter

build_M1	<i>Specify the residual natural mortality (M1) model for Rceattle</i>
----------	---

Description

Specify the residual natural mortality (M1) model for Rceattle

Usage

```
build_M1(
  M1_model = 0,
  M1_re = 0,
  updateM1 = FALSE,
  M1_use_prior = FALSE,
  M2_use_prior = FALSE,
  M_prior = 0.4,
  M_prior_sd = 0.35,
  M1_indices = NA,
  linkages = NULL
)
```

Arguments

M1_model	Vector or scalar specifying the M1 structural fixed- effects model. Either an integer code or the equivalent string alias (both forms are accepted; the integer code is canonical): • 0 / "fixed", use the input M1_base (no estimation). • 1 / "sex_age_invariant", estimate one M1_{spp}. • 2 / "sex_specific", estimate M1_{spp, sex}. • 3 / "sex_age_specific", estimate M1_{spp, sex, age}. • 4, 5, soft-deprecated env-driven codes; use the linkages argument instead. See vignette("environmental-linkages-and-priors").
M1_re	Vector or scalar specifying the M1 random-effects model, as an integer code or the equivalent string alias: 0 / "none", 1 / "iid_age", 2 / "iid_year", 3 / "iid_age_year", 4 / "ar1_age", 5 / "ar1_year", 6 / "ar1_age_year". Default 0. Which combinations with M1_model are allowed is under Details.
updateM1	If using initial parameters, use M1 fixed effects from data (M1_base) instead. Default FALSE.
M1_use_prior	Vector or scalar; if TRUE and M2_use_prior is FALSE, apply the lognormal M_prior / M_prior_sd to M1 directly, which is refused alongside a prior on that species' M1 linkage intercept.

M2_use_prior	Vector or scalar; if TRUE, apply the lognormal prior to M1 + M2 in multi-species models.
M_prior	Natural-scale mean of the lognormal prior on M (its median when bias_adjust_proc = FALSE).
M_prior_sd	SD (log-scale) of the lognormal prior on M.
M1_indices	Soft-deprecated. Vector of column indices into env_data (excluding Year) for environmentally linked M1 when M1_model %in% c(4, 5). Use the linkages argument instead; see vignette("environmental-linkages-and-priors").
linkages	Optional named list of <code>linkage_spec()</code> objects keyed by M parameter name (currently the only valid key is "M1"). Each spec describes how M1 depends on environmental covariates and on stratifying factors (species, sex, age). The offset enters additively (on the log scale) inside the M1_at_age compute. A row's age_bin == NA broadcasts the offset across ages; specific values pin it to that age slice.

Details

Which M1_re families work with which M1_model. Anything but 0 needs an M1_model of 0, 1, 2, or 3 with a by-year family (2 / 5); the by-age families are refused under 3 because an age-varying deviation and an age-specific level are not separately identifiable, and 4 and 5 are refused entirely (use a catchability-style linkage instead). M1_model = 2 needs two sexes under every M1_re family, because a single-sex species would take its deviations in the padding sex slot, free and scored by no density. `fit_mod()` downgrades M1_model to 1 there, so the distinction shows only in a direct `build_map()` call.

Value

A list of switches for defining the M1 model.

M1_base completeness

M1_base needs a finite, positive residual M (per year) for every age of every sex the species has; it is taken on the log scale, so a blank or a zero is refused, naming the offending (species, sex, age) cells. Columns past a species' own nages are ignored, so a row may stop at its last age.

Examples

```
# Sex/age-invariant M with a temperature linkage on M1
build_M1(
  M1_model = "sex_age_invariant",
  linkages = list(
    M1 = linkage_spec(
      formula = ~ temp,
      by       = ~ species,
      priors   = list(temp = normal(0, 0.5))
    )
  )
)
```

build_map

*Main function to construct the TMB map argument for CEATTLE***Description**

Builds the TMB map, which tells TMB which parameters to estimate and which to hold fixed (a fixed parameter is mapped to NA), and which parameters share a single estimated value. One helper handles each process block (recruitment, M1, predation, selectivity, catchability, ...).

Usage

```
build_map(
  data_list,
  params,
  debug = FALSE,
  random_rec = FALSE,
  random_sel = FALSE,
  random_q = NULL
)
```

Arguments

data_list	an Rceattle data_list
params	A parameter list created from build_params .
debug	Logical. If TRUE, fixes every parameter except the dummy (maps all to NA), so the model runs with no parameters estimated.
random_rec	Logical. If TRUE, treats recruitment deviations as random effects, meaning the variance parameter (R_log_sd) is estimated.
random_sel	Logical. If TRUE, treats selectivity deviations as random effects, meaning the variance parameter (sel_dev_log_sd) is estimated.
random_q	Logical. If TRUE, treats catchability deviations as random effects, meaning the variance parameter (index_q_dev_log_sd) is estimated. Defaults to data_list\$random_q, which is what fit_mod() stores there.

Details

Fleets sharing a Selectivity_index or a Catchability_index have the lead fleet's map slice copied over the rest of the group, so a per-fleet setting that differs within a group is resolved here rather than reported. Those disagreements are checked in data_check(), which fit_mod() runs first; build_map() does not call it, so a caller invoking build_map() directly should run data_check() too.

Value

A list containing the factorized TMB map (mapFactor) and the original map matrix/array list (mapList).

 build_map_catchability

Helper to set map for Catchability parameters

Description

Maps catchability base parameters (index_log_q), time-varying deviations (index_q_dev), and environmental linkages (index_q_beta) for every fleet that holds fitted index_data, a fishery with a CPUE series as much as a survey. A fleet with no index rows gets none of them, whatever its Catchability says, since a q with no index to inform it is a flat direction. Sharing overrides this: adjust_map_shared_params() then copies the lead fleet's slice across each Catchability_index group.

Usage

```
build_map_catchability(map_list, data_list, nyrs_hind, random_q = FALSE)
```

Arguments

map_list	The current TMB map list.
data_list	The data list containing model settings.
nyrs_hind	Number of historical years.
random_q	Logical. If TRUE, the deviation sd (index_q_dev_log_sd) is estimated rather than held at Time_varying_q_sd.

Value

Updated map_list.

 build_map_debug

Helper to set map for debug mode

Description

Sets all parameters in the map list to NA, except the dummy parameter, for use in debug or testing modes.

Usage

```
build_map_debug(map_list, debug)
```

Arguments

map_list	The current TMB map list.
debug	Logical. If TRUE, debug mode is activated.

Value

Updated map_list.

build_map_f_and_data_weights

Helper to set map for Fishing Mortality and Data Weights

Description

Maps fishing mortality parameters (log_F) and related targets. Also maps data weighting parameters (catch_log_sd, comp_weights, caal_weights).

Usage

```
build_map_f_and_data_weights(map_list, data_list, nyrs_hind)
```

Arguments

map_list	The current TMB map list.
data_list	The data list containing model settings.
nyrs_hind	Number of historical years.

Value

Updated map_list.

build_map_fixed_natage

Helper to set map for Fixed N-at-Age models

Description

Turns off (sets to NA) most population and fleet parameters for species where the dynamics are fixed (estDynamics > 0).

Usage

```
build_map_fixed_natage(map_list, data_list)
```

Arguments

map_list	The current TMB map list.
data_list	The data list containing model settings.

Value

Updated map_list.

build_map_growth	<i>Helper to set map for growth parameters</i>
------------------	--

Description

Maps the growth parameters (log_growth_pars, growth_log_sd, weight_length_pars) according to growth_model. Everything starts mapped off and each growth function turns on the parameters it uses.

Time-varying growth comes from the linkage grammar (build_growth(linkages =)), whose random effects hold their own density and map.

Usage

```
build_map_growth(map_list, data_list, nyrs_hind)
```

Arguments

map_list	The current TMB map list.
data_list	The data list containing model settings.
nyrs_hind	Number of historical years.

Value

Updated map_list.

build_map_m1	<i>Helper to set map for Natural Mortality (M1) parameters</i>
--------------	--

Description

Maps the fixed parameters (log_M1) and the random effects parameters (log_M1_dev, M1_dev_log_sd, M1_rho) based on M1_model and M1_re settings.

Usage

```
build_map_m1(map_list, data_list, nyrs_hind)
```

Arguments

map_list	The current TMB map list.
data_list	The data list containing model settings.
nyrs_hind	Number of historical years.

Value

Updated map_list.

build_map_predation	<i>Helper to set map for Predation Mortality (M2) parameters</i>
---------------------	--

Description

Maps predation suitability parameters (`log_gam_a`, `log_gam_b`, `log_phi`) and diet weight parameters based on `msmMode` and `suitMode`.

Usage

```
build_map_predation(map_list, data_list)
```

Arguments

<code>map_list</code>	The current TMB map list.
<code>data_list</code>	The data list containing model settings.

Value

Updated `map_list`.

build_map_recruitment	<i>Helper to set map for Recruitment parameters</i>
-----------------------	---

Description

Maps the recruitment deviations (`rec_dev`, `init_dev`), stock-recruitment parameters (`rec_pars`) and Rec dev variances (`R_log_sd`)
see `build_srr()` for options,

Usage

```
build_map_recruitment(map_list, data_list, nyrs_hind, nyrs_proj, random_rec)
```

Arguments

<code>map_list</code>	The current TMB map list.
<code>data_list</code>	The data list containing model settings.
<code>nyrs_hind</code>	Number of historical years.
<code>nyrs_proj</code>	Total number of years (historical + projected).
<code>random_rec</code>	Logical indicating if recruitment deviations are random effects.

Value

Updated `map_list`.

build_map_selectivity *Helper to set map for Selectivity parameters*

Description

Maps base selectivity parameters (`log_sel_slp`, `sel_inf`, `sel_coff`) and time-varying deviations, based on `Selectivity` and `Time_varying_sel` settings in `fleet_control`.

Usage

```
build_map_selectivity(map_list, data_list, nyrs_hind, random_sel)
```

Arguments

<code>map_list</code>	The current TMB map list.
<code>data_list</code>	The data list containing model settings.
<code>nyrs_hind</code>	Number of historical years.
<code>random_sel</code>	Logical indicating if selectivity deviations are random effects.

Details

Selectivity sets the curve's shape and `Time_varying_sel` whether it varies over time; for the allowed values of both, see "3. Selectivity (per fleet)" in `vignette("model-options-and-functionality")`, or [rceattle_switch_maps](#) for the code-to-name maps themselves. `N_sel_bins` is how many age or length bins the non-parametric and AR1 forms estimate (Selectivity 2, 5, 6, 7, 9 or 13); it is unused otherwise. `random_sel` in `fit_mod()` treats the deviations as random effects and estimates their variance.

The double-normal forms blend Gaussian ascending and descending limbs at the peak. `DoubleNormal` reads its four parameters from the shared arrays: `sel_inf[1] = peak`, `sel_inf[2] = logit(right_floor)` (the right-tail floor, SS3 P6/end_logit), `log_sel_slp[1] = log(sigma_asc)` and `log_sel_slp[2] = log(sigma_desc)`. `right_floor -> 0` is dome-shaped; `right_floor -> 1` is logistic ascending only. `DoubleNormalSS3` instead carries its own six-parameter array (`sel_dn6`).

Value

Updated `map_list`.

build_params	<i>Build parameter list from cpp file</i>
--------------	---

Description

Construct the TMB parameter list, with every parameter set to its starting value, for an Rceattle model.

Usage

```
build_params(data_list)
```

Arguments

data_list an Rceattle data_list

Value

a named list of model parameters at their starting values

build_selectivity	<i>Selectivity specification</i>
-------------------	----------------------------------

Description

Holds environmental linkages on selectivity parameters. An effect is written as a formula and composes additively with any Time_varying_sel process error on the same fleet, the two being separate mechanisms: a covariate effect against a deviation.

The linkages keys are the shape parameters of the parametric forms, so which apply depends on the fleet's Selectivity:

slp_asc, slp_desc ascending / descending logistic slope (log scale); the double-normal widths, aliased sigma_asc / sigma_desc.

inf_asc, inf_desc ascending / descending inflection age or length (natural scale); the double-normal peak and logit right-floor, aliased peak / right_floor.

coff non-parametric selectivity-at-bin coefficients.

apical a log-scale multiplier on one sex's whole curve, applied after the form and before normalization, so every estimated form takes it. See Details.

DoubleNormalSS3 (code 15) carries its own six under Stock Synthesis's names – dn_peak, top_logit, ascend_se, descend_se, start_logit, end_logit – and a linkage is the **only** way to vary that form over time, since Time_varying_sel is refused on it. A time block is ~ cut(Year, ...); annual deviations are a random-effect term.

Every parameter but apical takes link = "log" (multiplicative on the natural parameter) or link = "identity" (additive); apical is already a multiplier and takes "log" only.

Usage

```
build_selectivity(linkages = NULL)
```

Arguments

linkages Optional named list of `linkage_spec()` objects keyed by selectivity parameter. Coefficients are per fleet by default (`by = ~ fleet`), except that fleets sharing a `Selectivity_index` estimate one parameter block and so take ONE coefficient for the whole group, which is what makes them mirror each other. Use `linkage_spec()`'s `fleet` argument to restrict a spec to particular fleets.

Details

The apical offset. Fishing mortality is one $\log_F$ per fleet and year shared by the sexes, and no form has a height parameter, so a sex difference in F can only come from selectivity. `apical` multiplies one sex's curve by $\exp(\log_sel_apical)$ bin by bin, as Stock Synthesis's male-offset option does. Name the fleet and the sex that holds it (`fleet = 3`, `sex = "male"`; by defaults to `~ fleet + sex` here) and the other sex is the reference. It equals the ratio of the sexes' peak heights only where their shapes peak equally, so for a sex-specific dome read it as the multiplier and take the peak ratio from `fit$quantities$sel_at_age`.

Only the contrast between sexes is identified, the common level being $\log_F$, so one sex holds it and the fit is refused if both do, if no fleet or sex is named, if the species has one sex, on a Fixed, ARI or `Fleet_type = "Off"` fleet, on a fleet sharing another's `Selectivity_index`, under `link = "identity"` (which could drive the multiplier negative), or where `Sel_norm_scope = "WithinSex"` would divide it straight back out. The contrast is informed only by joint composition (`comp_data$Sex = 3`); with single-sex compositions it rests on its prior, and `fit_mod()` warns when a fleet has neither. An intercept prior is on the multiplier's natural scale, so a `lognormal()` centred on 1 means no offset. See `vignette("environmental-linkages-and-priors")`.

Priors on a selectivity parameter. An intercept-only formula (`~ 1`) with a `priors` entry puts a prior on the parameter itself, adding no year-to-year offset. Read it on the parameter's own scale: the slopes are log scale (`lognormal()`), the inflections natural scale (`normal()`). A prior targets one parameter, so in a two-sex model an unstratified `~ 1` prior constrains sex 1 only – use `by = ~ sex`. An `init` on a selectivity intercept has no effect, since the starting value comes from the data, and a prior on the double-normal `right_floor` is unsupported. Fleets sharing a `Selectivity_index` estimate one block, so place the prior on the group's lead fleet (its first that is not `Off`): a prior on a follower would penalize the shared block once per sharing fleet. On an `Off` fleet a prior is dropped with a warning and the fleet's linkage coefficients are held at their starting values, since the fleet has no data; on an `Off` member of a shared block it moves to the block's lead instead.

Like every selectivity linkage, a covariate acts in the hindcast years; projection years hold the last hindcast year's curve.

Value

A list of selectivity settings for `fit_mod()`.

Examples

```
# A cold-pool effect on the ascending inflection of a logistic fleet
```

```

build_selectivity(linkages = list(
  inf_asc = linkage_spec(~ cold_pool, by = ~ fleet)))

# A normal prior on the ascending inflection (intercept-only formula)
build_selectivity(linkages = list(
  inf_asc = linkage_spec(~ 1, priors = list(`(Intercept)` = normal(0, 3)))))

```

build_srr

*Specify the stock-recruit relationship (SRR) for Rceattle***Description**

Sets the stock-recruit curve and how recruitment is estimated.

Relationships currently implemented:

- `srr_fun = 0` or "mean": no stock-recruit relationship. Recruitment is R_0 (log scale) plus annual deviates, and the reported steepness holds its placeholder 0.99 rather than being estimated.

$$R_y = \exp(R_0 + R_{dev,y})$$

- `srr_fun = 2` or "BevertonHolt":

$$R_y = \frac{\alpha_{srr} * SB_{y-minage}}{1 + \beta_{srr} * SB_{y-minage}}$$

- `srr_fun = 4` or "Ricker":

$$R_y = \alpha_{srr} * SB_{y-minage} * \exp(-\beta_{srr} * SB_{y-minage})$$

Both curves are the deterministic mean; realized recruitment applies the annual log deviation, $R_y \cdot \exp(R_{dev,y})$. For numerical stability the Ricker β_{srr} is estimated divided by 1,000,000, so the fitted beta is 1e6 times the coefficient above. `Bmsy_lim` is on the natural scale: the template divides it by 1e6 before comparing, so supply it in metric tons.

With `srr_pred_fun > 0` and `srr_fun = 0`, the hindcast estimates recruitment as in the mean form while the curve is fitted between `srr_hat_styr` and `srr_hat_endyr` as a penalty, then used in the projection.

Multispecies models. Spawning biomass per recruit is undefined once mortality includes predation, so under `msmMode > 0` a curve fitted in the hindcast estimates its initial recruitment level (R_0) rather than deriving it, and under the fished `initModes` (3, 4) that level trades off against the initial F. A steepness prior is refused and steepness is reported as 0; priors on alpha or beta go through linkages.

Two refusals worth knowing before you write a linkage: under a curve fitted in the hindcast a single-species R_0 is derived from alpha and beta, so an R_0 linkage is refused (a multispecies model accepts an intercept-only one), and a Ricker lognormal linkage prior on alpha duplicates the deprecated `srr_est_mode = "LognormalPrior"`, so using both is refused.

Usage

```

build_srr(
  srr_fun = 0,
  srr_pred_fun = srr_fun,
  proj_mean_rec = TRUE,
  srr_mse_switchyr = NULL,
  srr_hat_styr = NULL,
  srr_hat_endyr = NULL,
  srr_est_mode = 1,
  srr_prior = 4,
  srr_prior_sd = 1,
  srr_alpha_init = NULL,
  srr_beta_init = NULL,
  srr_indices = NA,
  Bmsy_lim = NA,
  linkages = NULL
)

```

Arguments

srr_fun	Stock-recruit function used in the hindcast estimation (see below). Default = 0
srr_pred_fun	Stock-recruit function used for projection, reference points and penalties. When <code>srr_fun == 0</code> the curve is added as a penalty on the annually estimated hindcast recruitment (the AMAK / Jim Ianelli pollock form); when <code>srr_fun > 0</code> it equals <code>srr_fun</code> and no penalty is added.
proj_mean_rec	Recruitment used in the projection: TRUE/1 (default) is mean recruitment over the hindcast, FALSE/0 follows <code>srr_pred_fun</code> . Equilibrium and dynamic reference points follow the curve whenever <code>srr_pred_fun</code> is a stock-recruit form, whatever this is set to.
srr_mse_switchyr	Year at which an MSE switches from the annual recruitment-penalty estimate to the stock-recruit function.
srr_hat_styr, srr_hat_endyr	First and last year used to estimate the recruitment-penalty function (active when <code>srr_pred_fun > 0</code> and <code>srr_fun = 0</code>), defaulting to <code>styr + 1</code> and <code>endyr</code> . Set them where the environmental data conditioning the curve do not span the full series but projections are still wanted.
srr_est_mode	The curve's built-in prior, as a code or string: 1 / "Estimated" (default, no prior), or 2 / "LognormalPrior" or 3 / "BetaPrior" on Beverton-Holt steepness, single-species only. 0 / "Fixed" and a Ricker "LognormalPrior" are deprecated in favour of an alpha linkage.
srr_prior	Natural-scale prior mean of Beverton-Holt steepness under modes 2 and 3 (the median of a lognormal prior when <code>bias_adjust_proc = FALSE</code>). Its other uses, as a Ricker alpha prior, a fixed alpha, or alpha's starting value, are deprecated.
srr_prior_sd	Prior standard deviation: log scale for the lognormal prior (mode 2), natural scale for the beta prior (mode 3).

srr_alpha_init, srr_beta_init	Optional natural-scale starting values for alpha and beta, one per species, used only where the curve estimates them.
srr_indices	Defunct: supplying it is an error; express an environmental effect through linkages.
Bmsy_lim	Upper limit on Ricker SSB-MSY (1/beta) in metric tons, adding a likelihood penalty once estimated SSB-MSY rises above it. Default NA is unused.
linkages	Named list of <code>linkage_spec()</code> objects keyed by "R0", "alpha", "beta" or "R_init" – note the key is a unitless log-scale MULTIPLIER on the initial age structure, not the reported quantity R_init, which is equilibrium recruitment at $F = F_{\text{init}}$ in thousands of fish. It is the recommended way to put a prior on, fix, or add an environmental effect to those parameters. See vignette("environmental-linkages-and-pri

Value

A list containing the stock recruitment relationship settings

Starting values

None of the starting values knows the stock's scale, and a bad one fails the fit rather than fitting badly. Mean recruitment (R_0) starts at $e^9 = 8103$ thousand fish; alpha starts at `srr_prior` (default 4) wherever that is an alpha and at e^3 otherwise; beta starts at 3. Set them with `srr_alpha_init` / `srr_beta_init` or a linkage init.

β sets the density dependence in $R = \alpha S / (1 + \beta S)$, so it must be on the order of $(\alpha - 1/\phi_0)/R_0$, typically 10^{-3} or smaller for a stock measured in tonnes. Starting three orders of magnitude away drives predicted recruitment to near zero and the optimizer returns NA/NaN gradient evaluation. From a steepness h and unfished spawning biomass per recruit ϕ_0 :

$$\alpha = \frac{4h}{\phi_0(1-h)}, \quad \beta = \frac{\alpha - 1/\phi_0}{R_0}.$$

Under predation, where ϕ_0 is undefined, seed them from a curve fitted to an earlier fit's SSB-recruitment pairs.

Examples

```
# Mean recruitment: no stock-recruit relationship fitted.
build_srr(srr_fun = "mean")

# Beverton-Holt with a lognormal prior on alpha (mean 5, log-scale SD 0.5),
# species 1 only.
build_srr(srr_fun = "BevertonHolt",
          linkages = list(alpha = linkage_spec(~ 1, species = 1,
          priors = list(`(Intercept)` = prior_lognormal(log(5), 0.5))))))

# Mean recruitment with the curve as a penalty (Ianelli form).
build_srr(srr_fun = "mean", srr_pred_fun = "BevertonHolt")

# A prior on Beverton-Holt steepness, the one prior linkages cannot express
# (single-species only).
build_srr(srr_fun = "BevertonHolt", srr_est_mode = "LognormalPrior",
          srr_prior = 0.8, srr_prior_sd = 0.2)
```

check_mse	<i>Check which saved MSE simulation files can be loaded</i>
-----------	---

Description

Check which saved MSE simulation files can be loaded

Usage

```
check_mse(dir = NULL, file = NULL)
```

Arguments

dir	Directory used to save files from run_mse
file	file name used to save files from run_mse

Value

list of MSE simulations/run

clean_data	<i>Default-fill and tidy a data list before fitting</i>
------------	---

Description

Fills optional blocks with correctly-shaped empties, filters observations to the model's year range, extends catch to the projection years, and re-keys any survey-index covariance matrices. Whether a missing block is actually a problem is left to the fit-time validation, which knows the model configuration.

Usage

```
clean_data(data_list)
```

Arguments

data_list	Rceattle data list
-----------	--------------------

coef.Rceattle	<i>Extract estimated parameters from an Rceattle fit</i>
---------------	--

Description

Extract estimated parameters from an Rceattle fit

Usage

```
## S3 method for class 'Rceattle'  
coef(object, ...)
```

Arguments

object	An object of class "Rceattle" returned by <code>fit_mod()</code> .
...	Currently unused.

Value

A named numeric vector of estimated fixed-effect parameters (the optimizer's par). Returns NULL if the model was not estimated.

combine_data	<i>Combine two Rceattle data lists</i>
--------------	--

Description

Concatenate two data lists into one. env_data is taken from data_list1, and diet data must be updated afterward.

Usage

```
combine_data(data_list1 = NULL, data_list2 = NULL)
```

Arguments

data_list1	Rceattle data_list 1
data_list2	Rceattle data_list 2

compare_sim	<i>Evaluate simulation performance</i>
-------------	--

Description

Function to evaluate the simulation performance with regard to bias using the median relative error (MRE) and precision using the coefficient of variation.

Usage

```
compare_sim(operating_mod, simulation_mods, object = "quantities")
```

Arguments

operating_mod	CEATTLE model object exported from Rceattle to be used as the operating model
simulation_mods	List of CEATTLE model objects exported from Rceattle fit to simulated data
object	character string specifying which part of the model to compare (default = "quantities")

Value

A data frame summarizing simulation performance metrics

Note

Compares against operating_mod only, so it is valid when the replicates redrew the observations alone. If they were produced with sim_mod(process =) / self_test(process =), the operating model's deviations are no longer what generated the data and the bias this reports is an artefact. Compare against attr(sims, "process_sim") in that case. Passing a [self_test](#) result that holds process draws warns, since the attribute makes that case detectable rather than only documented.

convergence_diagnostics	<i>Convergence diagnostics for a fitted Rceattle model</i>
-------------------------	--

Description

Runs the post-fit convergence battery and returns a single structured object. Each check yields a record with a common schema (id, tier, severity, message, data); severity is one of "OK", "NOTE", "WARN", "FAIL". The object's status is the worst severity present.

Usage

```
convergence_diagnostics(object, ...)
```

Arguments

object An object of class "Rceattle" returned by `fit_mod()`.
 ... Currently unused.

Details

`fit_mod()` runs this automatically and attaches the result as `fit$convergence`; call `convergence_diagnostics()` directly to re-run it on any fit. Checks cover the optimizer gradient, a requested `sdreport` that did not return, an `sdreport` that was never requested, Hessian positive-definiteness and conditioning, parameters on bounds, parameters past them, a deviation variance estimated to zero, phasing, parameter estimability, a numbers-at-age, Ricker-intercept or recruitment floor that was reached, and the stock-recruit curve.

Value

An object of class "Rceattle_convergence": a list with `status` (overall worst severity) and `checks` (named list of records).

data_requirements	<i>Report which data inputs a model configuration requires</i>
-------------------	--

Description

Introspects the Rceattle data-element catalogue and reports, for a given model configuration, which top-level `data_list` elements are **Required**, **Optional** (used if supplied, otherwise default-filled by `clean_data()`), or **Ignored** (not consulted because the feature that would use them is switched off). It answers "what do I actually need to supply for *this* model?" without having to read the validation code or the switch tables, the same conditions enforced at fit time (they share one declarative table).

Usage

```
data_requirements(
  data_list = NULL,
  msmMode = 0,
  growth_model = 0,
  estDynamics = 0,
  Selectivity = NULL,
  Index_distribution = NULL,
  Ceq = NULL
)
```

Arguments

data_list Optional. An existing Rceattle data list (e.g. from `read_data()` / `build_data()`, or a bundled dataset). When supplied, the configuration is read from it and the convenience arguments are ignored.

msmMode	Predation mode used to build the configuration when <code>data_list</code> is NULL. 0 single-species (default), > 0 multispecies.
growth_model	Growth mode (scalar or per-species). > 0 estimates growth and requires <code>caal_data</code> .
estDynamics	Numbers-at-age mode (scalar or per-species). > 0 fixes numbers-at-age and requires <code>NByageFixed</code> .
Selectivity	Optional character vector of per-fleet selectivity forms (used only when <code>data_list</code> is NULL) so the <code>emp_sel</code> requirement (<code>Selectivity = "Fixed"</code>) can be evaluated. Named for the <code>fleet_control</code> column it stands in for.
Index_distribution	Optional character vector of per-fleet index likelihood families (used only when <code>data_list</code> is NULL) so the <code>index_cov</code> requirement (<code>Index_distribution = "MVN"</code>) can be evaluated. Named for the <code>fleet_control</code> column it stands in for.
Ceq	Optional per-species consumption-equation codes (used only when <code>data_list</code> is NULL); > 1 requires an environmental temperature index.

Details

The configuration can be given either as an existing (possibly partial) `data_list`, its switches are normalized through `clean_data()` / `switch_check()` so the conditions evaluate against filled defaults, or, when no `data_list` is supplied, built from the convenience arguments.

Requirements are *conditional*: e.g. `diet_data`, `ration_data` and the bioenergetics scalars are Ignored under single-species (`msmMode = 0`) but Required under multispecies (`msmMode > 0`); `caal_data` is Optional unless `growth_model > 0`; `NByageFixed` is Ignored unless `estDynamics > 0`; `emp_sel` is Required only for fleets with `Selectivity = "Fixed"`; `index_cov` only for `Index_distribution = "MVN"`.

Value

A data.frame with one row per data element and columns:

`element` the `data_list` element name.

`category` grouping (dimensions / biology / fishery / composition / predation / environment).

`status` "Required", "Optional", or "Ignored".

`condition` the condition under which the element is required ("always" for the core backbone).

`default` for Optional elements, the `clean_data()` default used when the element is absent.

Rows are ordered Required, then Optional, then Ignored.

See Also

`clean_data()`, `build_data()`.

Examples

```
# Single-species: predation/diet inputs are Ignored, comp_data Optional.
data_requirements(msmMode = 0)

# Multispecies: diet_data and bioenergetics become Required.
data_requirements(msmMode = 1)

# From a bundled dataset.
data_requirements(BS2017SS)
```

EBS_ms_run

*Fitted multispecies CEATTLE model for the Eastern Bering Sea***Description**

A fitted Rceattle model object for the multispecies run of CEATTLE for the Eastern Bering Sea (walleye pollock, Pacific cod, arrowtooth flounder).

Usage

EBS_ms_run

Format

An object of class Rceattle.

EBS_ss_M_run

*Fitted single-species CEATTLE model with estimated M for the Eastern Bering Sea***Description**

A fitted Rceattle model object for the single-species run of CEATTLE for the Eastern Bering Sea with natural mortality estimated.

Usage

EBS_ss_M_run

Format

An object of class Rceattle.

EBS_ss_run

Fitted single-species CEATTLE model for the Eastern Bering Sea

Description

A fitted Rceattle model object for the single-species run of CEATTLE for the Eastern Bering Sea (walleye pollock, Pacific cod, arrowtooth flounder) without estimating natural mortality.

Usage

EBS_ss_run

Format

An object of class Rceattle.

fit_control

Bundle the optimizer / sdreport / phasing controls for fit_mod()

Description

Collects the optimizer and reporting settings into one object, so a call to `fit_mod()` stays about the model rather than how it is fit. Pass the result through `fit_mod()`'s `fit_control` argument, where it overrides the equivalent individual arguments, which are kept for back-compatibility.

Usage

```
fit_control(
  bias.correct = FALSE,
  getsd = TRUE,
  getJointPrecision = TRUE,
  getReportCovariance = FALSE,
  projection_uncertainty = FALSE,
  selectivity_se = FALSE,
  comp_offset = NULL,
  bias_adjust_obs = TRUE,
  bias_adjust_proc = TRUE,
  use_gradient = TRUE,
  rel_tol = 1,
  loopnum = 5,
  newtonsteps = 0,
  phase = FALSE,
  TMBfilename = NULL,
  verbose = 1,
  nlminb_control = list(eval.max = 1e+09, iter.max = 1e+09, trace = 0)
)
```

Arguments

bias.correct	logical. Apply bias correction via <code>TMB::sdreport()</code> . Default FALSE.
getsd	logical. Run <code>TMB::sdreport()</code> after optimization. Default TRUE.
getJointPrecision	logical. Return the full Hessian of fixed and random effects. Default TRUE.
getReportCovariance	logical. Return the variance-covariance of ADREPORT variables. Default FALSE.
projection_uncertainty	logical. Carry hindcast parameter uncertainty into an HCR projection, by re-fitting with the hindcast and biological-reference-point parameters turned on. Default FALSE for speed.
selectivity_se	logical. Also return a standard error for log selectivity-at-age. Default FALSE; see Details.
comp_offset	Numeric or NULL. Small proportion added to the observed and predicted composition and CAAL bins before the multinomial / Dirichlet-multinomial likelihood, so an empty bin is not $\log(0)$. Stored on <code>data_list</code> , so the fitted likelihood, the OSA observation vector and any internal refit share it. NULL (default) inherits <code>data_list\$comp_offset</code> if set, else $1e-5$. Diet compositions take the same offset.
bias_adjust_obs	logical, default TRUE. Apply a bias adjustment ($\text{mean} - \text{sd}^2/2$) to lognormal data likelihoods.
bias_adjust_proc	logical, default TRUE. Shift lognormal process likelihoods, lognormal priors and the Ianelli stock-recruit penalty by $-\text{sd}^2/2$ on the log scale, making each prior value or curve a mean rather than a median. A value between 0 and 1 scales the shift.
use_gradient	logical. Use the analytic gradient during phasing. Default TRUE.
rel_tol	Numeric tolerance for flagging a discontinuous likelihood, comparing <code>nlminb</code> 's objective with a fresh evaluation of the object it came from. Default 1.
loopnum	Integer. Times to restart optimization; 3 sometimes reaches a lower final gradient than 1. Default 5.
newtonsteps	Integer. Extra unconstrained Newton steps after optimization, an alternative to <code>loopnum</code> . Default 0.
phase	TRUE/FALSE, or a list of parameter names with their phases. Default FALSE.
TMBfilename	Optional path (without <code>.cpp</code>) to an alternate TMB template for development. Default NULL uses the bundled <code>ceattle</code> .
verbose	0 silent, 1 model-fit updates, 2 model-fit and TMB progress. Default 1.
nlminb_control	Control parameters passed to <code>stats::nlminb()</code> ; see <code>?nlminb</code> . Default <code>list(eval.max = 1e9, iter.max = 1e9, trace = 0)</code> .

Value

A list of class "Rceattle_fit_control".

Selectivity standard errors

selectivity_se needs getsd = TRUE, and reports the error belonging to whichever sdreport the fit ends on. Under estimateMode = "Estimate" with an HCR that re-optimizes, that is the *projection* fit, where every selectivity parameter is mapped off, so every error comes back exactly 0; "Projection" estimates no selectivity at all. Use "Hindcast", or "Estimate" with projection_uncertainty = TRUE, to get an error from a fit that estimated the curve. `fit_mod()` warns in each of these cases.

It is off by default because the delta method forms a Jacobian of every reported value against every parameter, so the cost is their product: on Atka2022, 1,012 values against 584 parameters. The error is on the log scale rather than the logit, because the non-parametric forms normalize to mean selectivity 1 instead of a maximum of 1 – 58% of Atka2022's sel_at_age exceeds 1, up to 3.06 – so a logit is undefined over most of the array.

Rows cover estimated, age-based lead fleets only, starting at each fleet's first selected bin. Four kinds of cell are a structural zero (a Fixed fleet's empirical curve, a length-based fleet's growth-matrix projection, a bin below Bin_first_selected, and array padding), and one $\log(0) = -\text{Inf}$ on the tape turns *every* quantity in the sdreport to NaN, biomass and SSB included. All four are identified from the data, so the reported set never depends on a parameter value, and a value that underflows to zero is floored so it cannot reintroduce the $-\text{Inf}$. See `plot_selectivity()`, which draws the interval.

Examples

```
# Quick-and-dirty fit: skip sdreport, single optimizer pass
ctl <- fit_control(getsd = FALSE, loopnum = 1)

# Production fit with bias correction and joint precision
ctl <- fit_control(bias.correct = TRUE, getJointPrecision = TRUE)
```

fit_mod

Fit the CEATTLE assessment model

Description

Estimate CEATTLE population parameters by maximum likelihood, and optionally project the stock and apply a harvest control rule.

Usage

```
fit_mod(
  data_list = NULL,
  inits = NULL,
  map = NULL,
  bounds = NULL,
  file = NULL,
  estimateMode = 0,
  random_rec = FALSE,
```

```

    random_q = FALSE,
    random_sel = FALSE,
    HCR = build_hcr(),
    niter = 3,
    recFun = build_srr(),
    M1Fun = build_M1(),
    growthFun = build_growth(),
    qFun = build_catchability(),
    selFun = build_selectivity(),
    compFun = build_composition(),
    msmMode = 0,
    avgnMode = 0,
    initMode = "NonEquilibrium",
    suitMode = 0,
    suit_styr = NULL,
    suit_endyr = NULL,
    fit_control = NULL,
    config = NULL,
    quiet_data_check = FALSE,
    ...
)

```

Arguments

data_list	A data list read in via read_data or built directly in R; see vignette("data-without-excel", package = "Rceattle").
inits	(Optional) A named list of initial parameter values, as returned by build_params or taken from a previous fit (model\$estimated_params). NULL initializes from scratch.
map	(Optional) A map object from build_map .
bounds	(Optional) A bounds object from build_bounds .
file	(Optional) Filename to save to. If NULL, no file is saved.
estimateMode	What to fit, as a string alias or integer code: "Estimate" (0) fits the hindcast and the HCR projection, "Hindcast" (1) the hindcast alone, "Projection" (2) the reference points and projection alone from inits, "DebugBuild" (3) builds through MakeADFun without optimizing, leaving a usable obj\$fn() / obj\$gr(), and "DebugOptimize" (4) maps every parameter out, so its objective is a placeholder rather than a likelihood. Default "Estimate".
random_rec	logical; treat recruitment deviations as random effects via the Laplace approximation. Default FALSE.
random_q	logical; integrate the Time_varying_q deviations as random effects with one estimated sd per Catchability_index group instead of fixing them at Time_varying_q_sd. Default FALSE. Linkage random effects are integrated either way.
random_sel	logical; the same for Time_varying_sel deviations, one estimated sd per Selectivity_index group. Default FALSE.
HCR	HCR list object from build_hcr

niter	Number of iterations for multispecies model
recFun	The stock recruit-relationship parameterization from build_srr .
M1Fun	M1 parameterizations and priors. Use build_M1 .
growthFun	The weight-at-age parameterization from build_growth .
qFun	Catchability specification from build_catchability , holding any environmental linkages on q.
selFun	Selectivity specification from build_selectivity , holding any environmental linkages on selectivity parameters.
compFun	Composition-weighting specification from build_composition , holding any priors on the Dirichlet-multinomial weights.
msmMode	The predation-mortality mode, as a string alias or integer code: "SingleSpecies" (0, the default, no predation), "MSVPA" (1, the Type-II MSVPA predation of Holsman et al. 2015) or "TypeIIIMSVPA" (2). Codes 3 to 9 (Kinzey & Punt 2009 functional responses, Holling forms, Ecosim) are declared but refused by <code>data_check()</code> until validated against the current parameter set; see <code>src/TMB/predation.hpp</code> .
avgnMode	the average abundance-at-age approximation in the predation equations. Only mode 0, $N/Z(1 - \exp(-Z))$ (the MSVPA form), is implemented; the alternatives are declared but have no effect.
initMode	how the population is initialized, as a string alias or integer code: "FreeParams" (0), "Equilibrium" (1), "NonEquilibrium" (2, the default), "FishedNonEquilibrium" (3), "FishedNonEquilibriumScaled" (4), "OffsetEquilibrium" (5), "FishedNonEquilibriumSelect" (6). What each one estimates, and how F_{init} reaches an age under 3, 4 and 6, is tabulated in <code>vignette("model-options-and-functionality")</code> . In multi-species mode the initial decay uses M1 only, so predation does not shape the initial age structure under any mode.
suitMode	how predator-prey suitability is derived, per predator (one value or a vector of length nspp): 0 = empirical from diet data (Holsman et al. 2015), 2 = weight-based gamma, 4 = weight-based lognormal, 6 = weight-based normal. The length-based forms (1, 3, 5) are declared but refused by the data check.
suit_styr	The first year used to calculate mean suitability, as one integer for every predator or a vector of length nspp. Defaults to <code>styr</code> . Set it where diet data were sampled from a subset of years.
suit_endyr	The last year used to calculate mean suitability, in the same form. Defaults to <code>endyr</code> .
fit_control	A list from fit_control() bundling the optimizer, sdreport and phasing settings. Defaults to <code>fit_control()</code> , which documents each field.
config	(Optional) An <code>Rceattle_run_config</code> from load_config() or run_config() whose stored settings overlay the ones you did not pass; NULL (default) applies none. See Details.
quiet_data_check	Drop the warnings fit-time validation raises; errors still stop the fit. FALSE (default) for an ordinary fit. The diagnostic refits (retrospective() , jitter() , self_test() , profile() , run_mse() , remove_F() , sample_rec() , reweight_comps()) set it, since they re-validate a <code>data_list</code> the caller has already fitted and would

otherwise repeat each warning per peel or iteration. It also drops a linkage filter's "has no effect" warning, though a filter that drops its whole spec still warns. Convergence and TMB warnings are unaffected.

... Deprecated optimizer / sdreport / phasing arguments, forwarded into `fit_control` with a deprecation warning. Pass them via `fit_control()` instead.

Details

CEATTLE is an age-structured population model, fit with or without predation mortality; `msmMode` selects which.

What config **overlays**, at two levels. The estimation controls (`estimateMode`, `random_rec` / `random_q` / `random_sel`, `suit_styr` / `suit_endyr`, `fit_control`) overlay only arguments you did not pass, so an explicit argument always wins. The stored `model_config` merges into the data object's **field by field**: only fields the config actually set are imposed, so a config built with `model_config()` keeps the linkages held on the data object. Where both set a field and disagree, the config wins and the difference is warned about by name. A config written before that field record existed counts as having set its non-default fields.

Value

A list of class "Rceattle" including:

- `data_list`: List of data inputs
- `initial_params`: List of starting parameters
- `bounds`: Parameter bounds used for estimation
- `map`: List of map used in TMB
- `obj`: TMB model object
- `opt`: Optimized model object from `nlminb`
- `sdrep`: Object of class `sdreport` from TMB, with standard errors
- `estimated_params`: List of estimated parameters
- `quantities`: Derived quantities from CEATTLE
- `run_time`: Model run time

Examples

```
data(BS2017SS)
ss_run <- fit_mod(
  data_list    = BS2017SS,
  estimateMode = 0,
  msmMode      = 0,
  fit_control  = fit_control(phase = FALSE, verbose = 0)
)
```

GeorgesBank3spp

Data inputs for a three-species Georges Bank CEATTLE model

Description

A data list containing inputs for a multispecies CEATTLE model fit to three groundfish species on Georges Bank. See [BS2017SS](#) for format details.

Usage

```
GeorgesBank3spp
```

Format

A list with the same structure as [BS2017SS](#).

get_growth_matrix_r

Generate Length-at-Age Transition Matrix

Description

This function calculates a probability transition matrix that defines the probability of a fish of a given age belonging to specific length bins. It supports Von Bertalanffy and Richards growth models and includes a Stock Synthesis (SS) style plus-group correction.

Usage

```
get_growth_matrix_r(
  fracyr,
  nsex_sp,
  nages_sp,
  nlengths_sp,
  nyrs,
  lengths_sp,
  minage_sp,
  maxage_sp,
  growth_params_sp,
  growth_log_sd_sp,
  growth_model_sp,
  M_plus_sp
)
```

Arguments

fracyr	Numeric. Fraction of the year (0 = Jan 1st).
nsex_sp	Integer. Number of sexes for the species.
nages_sp	Integer. Number of age classes.
nlengths_sp	Integer. Number of length bins.
nyrs	Integer. Number of years in the simulation.
lengths_sp	Vector. Boundaries of the length bins.
minage_sp	Numeric. The reference age (L1) for growth estimation.
maxage_sp	Unused by the body; helpers-make-msm-data.R passes it.
growth_params_sp	Array. Dimensions (sex, yr, 4). Params: K, L1, Linf, Richards m.
growth_log_sd_sp	Array. Dimensions (sex, 2). Log-SD of length: 1st param is SD at minage, 2nd param is SD at maxage.
growth_model_sp	Integer. 1 = Von Bertalanffy, 2 = Richards.
M_plus_sp	Numeric. Natural mortality at the oldest age (one value, or one per sex), which weights the ages pooled in the plus group by survival, as the template does.

Value

A 4D array of probabilities with dimensions (sex, age, length, year).

get_weight_at_age_r	<i>Calculate Predicted Weight-at-Age</i>
---------------------	--

Description

Converts a growth matrix (length-at-age probabilities) into mean weight-at-age using a length-weight relationship ($W = a * L^b$).

Usage

```
get_weight_at_age_r(
  nsex_sp,
  nages_sp,
  nlengths_sp,
  nyrs,
  lengths_sp,
  length_at_age,
  growth_matrix,
  lw_params
)
```

Arguments

nsex_sp	Integer. Number of sexes.
nages_sp	Integer. Number of age classes.
nlengths_sp	Integer. Number of length bins.
nyrs	Integer. Number of years.
lengths_sp	Vector. Boundaries of the length bins.
length_at_age	Array. Mean length at age from get_growth_matrix_r.
growth_matrix	Array. 4D array (sex, age, length, year) from get_growth_matrix_r.
lw_params	Array. Dimensions (sex, yr, 2). Params: 1st is alpha (a), 2nd is beta (b).

Details

The function calculates midpoints for length bins to avoid bias. For the first bin, it assumes the width is equal to the second bin’s width. The final weight-at-age is the expected value across all length bins for that age.

Value

A 3D array of mean weights with dimensions (sex, age, year).

GOA2018SS	<i>Data inputs for a single-species Gulf of Alaska CEATTLE model (2018)</i>
-----------	---

Description

A data list containing inputs for a single-species CEATTLE model fit to Gulf of Alaska groundfish data through 2018. See [BS2017SS](#) for format details.

Usage

GOA2018SS

Format

A list with the same structure as [BS2017SS](#).

GOAatf	<i>Data inputs for Gulf of Alaska arrowtooth flounder CEATTLE model</i>
--------	---

Description

A data list containing inputs for a single-species CEATTLE model fit to Gulf of Alaska arrowtooth flounder data. See [BS2017SS](#) for format details.

Usage

GOAatf

Format

A list with the same structure as [BS2017SS](#).

GOAatf2023	<i>Data inputs for Gulf of Alaska arrowtooth flounder CEATTLE model (2023)</i>
------------	--

Description

A data list containing inputs for a single-species CEATTLE model fit to Gulf of Alaska arrowtooth flounder data through 2023. See [BS2017SS](#) for format details.

Usage

GOAatf2023

Format

A list with the same structure as [BS2017SS](#).

GOAcod

Data inputs for Gulf of Alaska Pacific cod CEATTLE model

Description

A data list containing inputs for a single-species CEATTLE model fit to Gulf of Alaska Pacific cod data. See [BS2017SS](#) for format details.

Usage

GOAcod

Format

A list with the same structure as [BS2017SS](#).

GOApollock

Data inputs for Gulf of Alaska walleye pollock CEATTLE model

Description

A data list containing inputs for a single-species CEATTLE model fit to Gulf of Alaska walleye pollock data. See [BS2017SS](#) for format details.

Usage

GOApollock

Format

A list with the same structure as [BS2017SS](#).

GOAsafe2018

Gulf of Alaska 2018 SAFE report reference values

Description

A list containing biomass, spawning stock biomass, and recruitment reference values from the 2018 Gulf of Alaska Stock Assessment and Fishery Evaluation (SAFE) report.

Usage

GOAsafe2018

Format

A list with components:

biomass Total biomass time series

ssb Spawning stock biomass time series

recruitment Recruitment time series

jitter

Jitter analysis

Description

Refits the Rceattle model from starting values perturbed by $N(0, sd)$ around the model's initial (pre-fit) parameters, to check convergence robustness.

Usage

```
jitter(
  object = NULL,
  njitter = 50,
  sd = 0.2,
  phase = FALSE,
  seed = 123,
  cores = NULL,
  getsd = NULL,
  timeout = Inf,
  fit_control = NULL,
  ...,
  Rceattle = NULL
)
```

Arguments

object	an Rceattle model fit using <code>fit_mod()</code> .
njitter	the number of jitters to run
sd	standard deviation for jitter (default = 0.2)
phase	as in <code>fit_mod</code> default = FALSE. Jitters restart from perturbed <i>starting</i> values, so a model that needed phasing to fit its real data needs it here too; leave this at FALSE for such a model and the jitters end far from any optimum and are dropped as non-converged, which reads as multimodality rather than as an unphased fit.
seed	random number seed. Each jitter <i>i</i> uses <code>seed + i</code> so results are reproducible under both sequential and parallel execution.
cores	Number of cores for the parallel refits. Default NULL picks <code>parallel::detectCores()</code> - 6, capped at 2 under R CMD check (which sets <code>_R_CHECK_LIMIT_CORES_</code>). Set to 1 to force sequential execution.

<code>getsd</code>	whether each jitter runs <code>TMB::sdreport</code> . Jitter compares objectives and point estimates across starts, so <code>FALSE</code> is faster with no effect on that comparison. Default <code>NULL</code> inherits the input model's setting (<code>TRUE</code> only if it holds an <code>sdrep</code>).
<code>timeout</code>	elapsed-second limit per jitter, <code>Inf</code> (default) for none. A jitter is a deliberately perturbed start and the optimizer runs with no iteration cap, so this is the diagnostic most likely to send one somewhere pathological and stall the whole run, a hang no convergence check can catch, because the fit never returns. One that exceeds the limit is stopped, counted as non-converged and reported separately. Approximate: the limit is checked when control returns to R, so it fires between the optimizer's function evaluations rather than inside one.
<code>fit_control</code>	optional <code>fit_control()</code> bundle for the refits. Only <code>phase</code> and <code>getsd</code> are read; see What <code>fit_control()</code> reaches .
<code>...</code>	for a numeric object, passed to <code>base::jitter()</code> ; for a fitted model, refused, so that a misspelled argument is not silently ignored.
<code>Rceattle</code>	deprecated name for object, still accepted so existing scripts keep working. Supplying both is an error.

Value

a list of 1. `Rceattle_list`, the converged jitters, and 2. `nll`, their objective values. Non-converged (or timed-out) starts are dropped and reported in a message, so both can be shorter than `njitter`, and that count is itself the result, since the whole point is what fraction of random starts reach the same optimum.

Note

Attaching `Rceattle` masks `jitter`, because that function is not generic and this name is exported, so the mask cannot be removed by making this an S3 method. It is harmless for the spellings base callers use: a numeric first argument is handed straight to `base::jitter()`, so `jitter(1:5)` and `jitter(1:5, factor = 2)` behave as base does. `jitter(x = 1:5)` and a positional second argument do not. `x` is not a parameter here, so it is refused as an unused argument; a positional second argument binds `njitter`, so `jitter(1:5, 2)` jitters with base's defaults and silently drops the 2. Name base's arguments, or call `base::jitter()` directly.

Examples

```
data(BS2017SS)
ss_run <- fit_mod(data_list = BS2017SS,
  inits = NULL, file = NULL,
  estimateMode = 0, random_rec = FALSE,
  msmMode = 0, avgnMode = 0,
  phase = FALSE, verbose = 0)
jitters <- jitter(ss_run, njitter = 10)
```

linkage_spec	Capture a linkage specification
--------------	---------------------------------

Description

Capture a linkage specification

Usage

```
linkage_spec(
  formula,
  param = NULL,
  data = NULL,
  by = ~species,
  species = NULL,
  sex = NULL,
  fleet = NULL,
  link = "log",
  init = NULL,
  bounds = NULL,
  priors = NULL,
  est_phase = 1L,
  observe = NULL,
  obs_sd = NULL,
  obs_sd_est = FALSE,
  integrate = TRUE
)
```

Arguments

formula	one-sided R formula whose RHS describes the linear predictor for param (e.g. ~ 1 , $\sim \text{temp}$, $\sim \text{temp} + (1 \mid \text{Year})$), where a $(1 \mid g)$, $\text{rw}(1 \mid g)$ or $\text{ar1}(1 \mid g)$ term adds a random effect with one deviation per level of g .
param	target parameter name on the natural scale (e.g. "alpha", "M1", "K"). May be NULL when the spec is built inside a <code>build_*</code> () call that infers it from the enclosing list key; see build_growth() .
data	Optional data frame for formula validation; validation happens at fit time inside fit_mod() .
by	one-sided formula naming stratifying factors that each get their own coefficients: species, sex, age_bin, or fleet. Omitted, it defaults to the base stratum of the process the spec is attached to – $\sim \text{fleet}$ for catchability, selectivity and the fleet composition weights, $\sim \text{species}$ for recruitment, M, growth and the diet weight – so the base case needs no by. Pass $\sim \text{species} + \text{sex}$ for per-(species, sex) coefficients, or NULL to share one coefficient across every stratum; an explicit by, NULL included, is kept as given. In a selectivity or catchability linkage under $\sim \text{fleet}$, fleets sharing a <code>Selectivity_index</code> or <code>Catchability_index</code> still take one coefficient between them, because they share one parameter block.

species	optional species this spec applies to, as 1-based ids (<code>c(1L, 2L)</code>) or as names matching <code>data_list\$spnames</code> (<code>c("Pollock", "Cod")</code>). Names are matched exactly after trimming whitespace, and an unrecognized one is an error listing the model's species. Give ids or names, not a mix, since R coerces <code>c(1, "Cod")</code> to character. NULL (default) means every species. Register several specs against one parameter to give species different formulas.
sex	optional sex ids, as integers (1L female, 2L male) or strings (<code>"Females"/"Males"</code> , case-insensitive; <code>"f"/"m"</code> accepted). NULL (default) means every sex. by must include sex or it warns and has no effect.
fleet	optional fleets, as 1-based <code>Fleet_codes</code> or as names matching <code>fleet_control\$Fleet_name</code> . Prefer names: a <code>Fleet_code</code> that is wrong but in range attaches the linkage to a different fleet and still fits, where a misspelled name cannot. Names are matched exactly after trimming; one that is unrecognized or not unique is an error listing the fleets. Give ids or names, not a mix. NULL (default) means every fleet, and by must include fleet or it warns and has no effect.
link	how the linear predictor reaches the natural-scale parameter: <code>"log"</code> (default), so $\log(\text{param}) = X * \text{beta}$ and slopes are multiplicative; <code>"identity"</code> , so $\text{param} = X * \text{beta}$ and slopes are additive; or <code>"exponential"</code> , where the predictor multiplies the log rather than shifting it, $\text{param}^{\exp(X * \text{beta})}$. The exponential form reproduces Stock Synthesis's environmental link type 1 and is accepted on catchability only, for a lognormal index, with an estimated intercept.
init	optional named list of initial values keyed by design-matrix column (e.g. <code>list(`(Intercept)` = 4, temp = 0)</code>), an intercept on the parameter's natural scale and a slope on the link scale; missing entries default to 0. An intercept's value is logged onto the parameter's base (<code>rec_pars</code> , <code>log_M1</code> , ...), or onto the coefficient itself for recruitment <code>R_init</code> , which has no base.
bounds	optional named list of <code>c(lower, upper)</code> , keyed and scaled as <code>init</code> .
priors	optional named list of <code>Rceattle_priors</code> objects keyed by design-matrix column. Inside this argument <code>normal()</code> , <code>lognormal()</code> , <code>gamma()</code> and <code>beta()</code> may be written directly, so <code>priors = list(temp = normal(0, 1))</code> is the same as <code>prior_normal(0, 1)</code> .
est_phase	optional integer estimation phase, default 1L. 0 fixes the coefficient at its <code>init</code> , which then holds over any <code>inits</code> given to <code>fit_mod()</code> (an estimated intercept's <code>init</code> is only a starting value, and <code>inits</code> win). It applies to the fixed-effect rows in <code>beta_linkage</code> only: a random-effect term's deviations live in a separate vector it cannot reach, so <code>est_phase < 1</code> on a formula containing one is an error rather than a silent no-op. Drop the term, or fix a small SD with <code>init = list(sigma =)</code> , to remove the time variation. Values above 1 are inert for every linkage row.
observe	optional <code>env_data</code> column measuring the AR1 latent of an <code>ar1(1 group)</code> term, making it a state-space covariate (sensu Rogers et al. 2024): the latent reaches the linked parameter through an estimated effect size and is observed against this column. NULL (default) leaves the AR1 a plain random effect. The latent is zero-mean, so standardize the column to mean 0; a non-zero-mean column confounds its level with the intercept and warns.

obs_sd	optional positive measurement SD for the observe covariate, one per observed group. Required with observe and unused otherwise. Held fixed by default, or the starting value when obs_sd_est = TRUE.
obs_sd_est	estimate the observe measurement SD instead of fixing it, as the state-space survey-catchability (GOA pollock) model does. Default FALSE. Caveat: the effect size and obs_sd are jointly identified only when the observed covariate is informative; on a smooth series the AR1 latent tracks it exactly and a free obs_sd collapses toward 0.
integrate	whether the random effect's deviations are integrated out by the Laplace approximation, default TRUE. FALSE estimates them as a penalized fixed effect instead; see Details for when that is permitted.

Details

The reserved keys `sigma` and `rho` in `init / priors` route the random-effect deviation SD and, for `ar1`, the correlation. `sigma` means different things by structure – the innovation (per-step) SD for `rw()`, the marginal (stationary) SD for `ar1()` – so the two are not comparable across structures; see `vignette("environmental-linkages-and-priors")`.

`integrate = FALSE` leaves the deviations in the objective as a plain penalty and reports them with standard errors. That reproduces the ADMB/AMAK convention behind the legacy `Time_varying_sel / Time_varying_q` switches, which a Laplace-integrated `rw()` cannot match, because the marginal likelihood carries a log-determinant term the penalized form has no counterpart for. Permitted only with a fixed SD (`init = list(sigma = )`), no `sigma` prior, plus a fixed `rho` for `ar1`), since estimating deviations and their SD jointly as fixed effects is degenerate, and never with `observe`, which must stay integrated.

Value

An `Rceattle_linkage_spec` object.

Examples

```
# A covariate on a growth parameter, one coefficient per species. Attach to a
# parametric build_growth(); the default "empirical" form consumes no linkages.
linkage_spec(~ temp, param = "K")

# A covariate on catchability for one fleet, named rather than numbered.
# `by` defaults to ~ fleet for catchability, so it need not be given.
linkage_spec(~ temp, fleet = "Pollock_survey_1_shelikof_acoustic")

# IID annual deviations; rw() for a random walk, ar1() for AR1.
linkage_spec(~ (1 | Year))

# A random walk estimated as a penalized fixed effect, which requires a fixed
# SD: the ADMB/AMAK convention behind the legacy Time_varying_* switches.
linkage_spec(~ rw(1 | Year), init = list(sigma = 0.05), integrate = FALSE)

# Intercept-only: a prior on the base parameter itself.
linkage_spec(~ 1, priors = list(`(Intercept)` = normal(0, 1)))
```

load_config	<i>Load a model run configuration from a YAML file</i>
-------------	--

Description

The inverse of `save_config()`: reads a run-config YAML file and rebuilds the `model_config()` structure, estimation controls, and `fit_control()` bundle (reconstructing linkage formulas and priors). Attach the result to a fit with `fit_mod(data_list, config = load_config("run.yaml"))`, or read `$model_config` off it to attach to a data list.

Usage

```
load_config(file)
```

Arguments

file Path to a YAML file written by `save_config()`.

Value

An Rceattle_run_config.

See Also

`save_config()`, `fit_mod()`.

Examples

```
f <- file.path(tempdir(), "run.yaml")
save_config(model_config(msmMode = 1, initMode = "Equilibrium"), f)

cfg <- load_config(f)
cfg$model_config$msmMode
# Apply with fit_mod(data_list, config = cfg).
unlink(f)
```

load_mse	<i>Load saved MSE simulation runs</i>
----------	---------------------------------------

Description

Load saved MSE simulation runs

Usage

```
load_mse(dir = NULL, file = NULL, exclude = NULL, include_em = TRUE)
```

Arguments

dir	Directory used to save files from run_mse
file	file name used to save files from run_mse
exclude	index of MSE simulations not to load
include_em	whether the EMs should be loaded or not (default = TRUE)

Value

list of MSE simulations/run

logLik.Rceattle	<i>Log-likelihood of an Rceattle fit</i>
-----------------	--

Description

Returns the joint negative-objective from nlminb, sign-flipped, as a "logLik" object. df is the number of estimated fixed-effect parameters. The nobs attribute is intentionally omitted: counting "observations" in a stock assessment likelihood (with composition cells, indices, catches, and priors) is not well-defined, so [stats::AIC\(\)](#) works (uses df) while [stats::BIC\(\)](#) does not.

Usage

```
## S3 method for class 'Rceattle'
logLik(object, ...)
```

Arguments

object	An object of class "Rceattle" returned by fit_mod() .
...	Currently unused.

Value

An object of class "logLik", or NULL if the model was not estimated.

model_average	<i>Model average of derived quantities</i>
---------------	--

Description

Model average of derived quantities

Usage

```
model_average(
  object = NULL,
  weights = NULL,
  uncertainty = FALSE,
  nboot = 10000,
  Rceattle = NULL
)
```

Arguments

object	list of Rceattle model objects
weights	vector of weights to be used for weighting models
uncertainty	TRUE/FALSE Sample uncertainty across derived quantities using weighted bootstrap from the asymptotic distribution of MLEs
nboot	Number of bootstraps taken from asymptotic distribution of MLEs. Default = 10000
Rceattle	deprecated name for object, still accepted so existing scripts keep working. Supplying both is an error.

Value

an Rceattle object with derived quantities weighted by the specified weights. The length of the derived quantities spans the years which overlap across all models.

Examples

```
## Not run:
# Equal weights across two candidate models.
model_average(list(fit1, fit2), weights = c(1, 1))
# Weighted by AIC, with bootstrapped uncertainty.
model_average(list(fit1, fit2), weights = c(0.7, 0.3), uncertainty = TRUE)

## End(Not run)
```

model_config

Build a model-configuration slot for a data list

Description

Bundles the model-structure arguments of `fit_mod()`, predation mode, initialization, harvest control rule, and the `build_*`() process specifications, into a single validated object that can be stored on a data list (`data_list$model_config`, e.g. via `build_data(..., model_config = model_config(...))`). A configuration then travels with the data instead of living only in the `fit_mod()` call.

Usage

```
model_config(
  msmMode = 0,
  initMode = "NonEquilibrium",
  avgnMode = 0,
  suitMode = 0,
  niter = 3,
  HCR = build_hcr(),
  recFun = build_srr(),
  M1Fun = build_M1(),
  growthFun = build_growth(),
  qFun = build_catchability(),
  selFun = build_selectivity(),
  compFun = build_composition()
)
```

Arguments

msmMode	Predation-mortality mode (see <code>fit_mod()</code>). Default 0.
initMode	Population initialization (see <code>fit_mod()</code>). Default "NonEquilibrium".
avgnMode	Average-N mode. Default 0.
suitMode	Suitability mode. Default 0.
niter	Number of predation iterations. Default 3.
HCR	A harvest-control-rule specification from <code>build_hcr()</code> .
recFun	A stock-recruit specification from <code>build_srr()</code> .
M1Fun	A natural-mortality specification from <code>build_M1()</code> .
growthFun	A growth specification from <code>build_growth()</code> .
qFun	A catchability specification from <code>build_catchability()</code> .
selFun	A selectivity specification from <code>build_selectivity()</code> .
compFun	A composition specification from <code>build_composition()</code> .

Details

The defaults are exactly `fit_mod()`'s own argument defaults, so a data list holding `model_config()` fits identically to one with no slot at all. When a data list has a `model_config`, `fit_mod()` reads each field only for arguments the caller did **not** pass explicitly, an argument passed to `fit_mod()` always wins, even when passed at its default. Omit an argument to let the stored configuration take effect for that field.

Value

An object of class "Rceattle_model_config" (a named list) to be stored as `data_list$model_config`.

Persistence

The configuration is code-side model structure, not one of the workbook data sheets, so it is **not** written by `write_data()` and does not survive an xlsx round-trip, `build_data(base = x, model_config = cfg)` piped through `write_data()` then `read_data()` returns without the slot. Re-attach it in code, store it alongside the data, or persist it as a documented, git-diffable YAML with `save_config()` / `load_config()` and apply it with `fit_mod(data_list, config = load_config("run.yaml"))`.

See Also

`fit_mod()`, `build_data()`, `save_config()`, `load_config()`.

Examples

```
cfg <- model_config(msmMode = 1, initMode = "NonEquilibrium")
dat <- build_data(base = BS2017MS, model_config = cfg)
# fit_mod(dat) would then fit as multispecies without passing msmMode.
```

mse_summary

Management strategy evaluation performance metric summary

Description

Management strategy evaluation performance metric summary

Usage

```
mse_summary(mse, om_only = FALSE)
```

Arguments

mse	MSE runs from <code>run_mse</code> or <code>load_mse</code>
om_only	If TRUE, report only operating-model (true) status and skip the estimation-model (EM) perception metrics.

Value

A named list, one element per entity dimension so each metric lives only where it applies (no NA padding):

- `species`, a data.frame with one row per species (keyed by `Species`) of the conservation / status metrics: per-species average catch, catch inter-annual variability (IAV), and `P(Closed)` = the probability the fishery is closed (catch ~ 0); the relative mean-squared error of terminal and average SSB; the estimation-model- and operating-model- perceived overfishing / overfished probabilities (via `build_hcr`) and the probability each status is misclassified (estimation model disagrees with the operating-model truth); and terminal biomass, SSB, dynamic SB0, SSB depletion (equilibrium and dynamic), average SSB depletion, and SSB-collapse counts.
- `fleet`, a data.frame with one row per fishery fleet (keyed by `Fleet_code` / `Fleet_name`) of average catch, catch IAV, and `P(Closed)`.
- `total`, a named numeric of the across-fleet total average catch and catch IAV.
- `meta`, run provenance (`nsim`, `nspp`, `nflts`, `HCR`, projection-year range).

All metrics are averaged across projection years and simulations.

Metric columns hold syntactic names, so they can be typed without backticks: `avg_catch`, `catch_iav`, `p_closed`, `ssb_rmse_avg`, `ssb_rmse_terminal`, `em_p_overfishing`, `em_p_overfished`, `om_p_overfishing`, `om_p_overfished`, `p_overfishing_false_pos`, `p_overfishing_false_neg`, `p_overfished_false_pos`, `p_overfished_false_neg`, `om_terminal_biomass`, `om_terminal_ssb`, `om_terminal_dynamic_sb0`, `om_terminal_depletion`, `om_terminal_depletion_dynamic`, `om_avg_depletion`, `om_sims_collapsed`, `om_no_f_sims_collapsed`, `om_sims_collapsed_from_f`. An `om_` prefix is the operating model's truth and `em_` the estimation model's perception; the four `*_false_pos` / `*_false_neg` metrics are the probability the two disagree. The three `*_sims_collapsed` metrics are **counts of simulations**, not probabilities.

The three depletions divide by different references:

- `om_terminal_depletion_dynamic`: terminal SSB over the OM's own `DynamicSB0`, its history with no fishing on the stock-recruit curve with the realized recruitment deviations and, under predation, the suitability fitted in the hindcast. Defined for single- and multispecies OMs. A species with input numbers-at-age (`estDynamics > 0`) keeps those numbers in the dynamic run, so its dynamic depletion is near 1 and says nothing about fishing.
- `om_terminal_depletion`: terminal SSB over `SB0`, the equilibrium value for a single-species OM. Under `msmMode > 0` it is `MSSB0`, SSB at the end of a no-fishing projection, which `fit_mod()` derives only when a harvest control rule is present; without one (`HCR = "NoFishing"`) this is NA.
- `om_avg_depletion`: the OM's own `ssb_depletion` averaged over the projection. Under predation with no HCR that series divides by SSB in the last projection year, not by an unfished reference.

The collapse metrics compare the OM with `OM_no_F`, the OM refit with no fishing after the original operating model's endyr (`remove_F()`).

Each frame holds a `"labels"` attribute mapping those names to the long display strings (e.g. `om_terminal_depletion_dynamic -> "OM: Terminal SSB Depletion (Dynamic)"`) for plots and tables: `attr(summ$species, "labels")`.

Examples

```
## Not run:
mse <- load_mse(dir = "mse_runs")
s <- mse_summary(mse)
s$species # one row per species
s$fleet   # one row per fleet
s$total   # across the whole system
s$meta    # nsim, nspp, nflts, HCR

## End(Not run)
```

NorthernRockfish2022 *Data inputs for Northern Rockfish CEATTLE model (2022)*

Description

A data list containing inputs for a single-species CEATTLE model fit to Gulf of Alaska northern rockfish data through 2022. See [BS2017SS](#) for format details.

Usage

```
NorthernRockfish2022
```

Format

A list with the same structure as [BS2017SS](#).

osa_diagnostics *Statistical diagnostics for OSA residuals*

Description

Computes the Stewart and Monnahan (2025) statistical diagnostics for a set of OSA residuals: the standard deviation of the normalized residuals (SDNR) and the lower/upper tail statistics, each with the 95% interval expected under the standard normal null hypothesis (so departures can be judged objectively rather than by eye). Computed per data source (type x fleet) and overall.

Under a correctly specified model OSA residuals are already iid standard normal, so the SDNR is simply their sample standard deviation. Its null interval follows the chi-square result for the sample standard deviation of n standard normals (Francis 2014). Each tail statistic is the r -th order statistic of the residuals, at index $\text{round}(p * (n + 1))$, and its null interval is exact: the r -th order statistic of n uniforms is $\text{Beta}(r, n - r + 1)$. Nothing is simulated.

Usage

```
osa_diagnostics(osa, nsim = 10000, probs = c(0.025, 0.975), seed = 123)
```

Arguments

osa	An rceattle_osa object from <code>osa_residuals()</code> , or a data frame with residual and (optionally) type/fleet columns.
nsim	Ignored from 5.29.0, when the tail null intervals became exact; retained so existing calls keep working, and warns when supplied.
probs	Lower/upper tail probabilities. Default <code>c(0.025, 0.975)</code> .
seed	Ignored from 5.29.0, as <code>nsim</code> is; the residuals themselves are randomized-quantile, so seed <code>osa_residuals()</code> instead.

Value

A data frame (class "rceattle_osa_diagnostics", so it prints as a compact severity-tagged summary; every column is still there and `$` works as before) with one row per data source plus an "all" row, with columns: group (the "<source> fleet <n>" label), source, fleet, n, sdnr, sdnr_lo, sdnr_hi, lower, lower_lo, lower_hi, upper, upper_lo, upper_hi, the order statistic each tail was read at (lower_r, upper_r) and its exact nominal probability $r/(n + 1)$ (lower_p, upper_p, which differ from probs at small n), and the logical flags sdnr_ok, lower_ok, upper_ok (TRUE when the statistic is inside its null interval). On the "all" row source and fleet are NA.

References

Francis, R.I.C.C. 2014. Replacing the multinomial in stock assessment models: a first step. Fish. Res. 151:70-84.

Stewart, I.J., and Monnahan, C.C. 2025. Can. J. Fish. Aquat. Sci. 82:1-13.

See Also

`osa_residuals()`

osa_residuals	<i>One-step-ahead (OSA) residuals for an Rceattle model</i>
---------------	---

Description

One-step-ahead residuals, also called forecast or quantile residuals (Thygesen et al. 2017), computed via `TMB::oneStepPredict()`. Unlike Pearson residuals they are iid standard normal under a correctly specified model even when observations are correlated through composition bins or the model holds random effects, so they support objective goodness-of-fit testing (Trijoulet et al. 2023; Stewart and Monnahan 2025). The residualization runs inside the assessment, so it also accounts for correlation induced by the model's own random effects.

Usage

```
osa_residuals(
  object = NULL,
  source = c("ecov", "index", "catch", "comp", "caal"),
  method = "oneStepGaussianOffMode",
  discrete = NULL,
  parallel = TRUE,
  seed = 123,
  trace = FALSE,
  ...,
  fit = NULL
)
```

Arguments

object	A fitted Rceattle object, from <code>fit_mod()</code> .
source	Observation sources to residualize: any of "ecov", "index", "catch", "comp", "caal", "diet" or "all", defaulting to the five non-diet sources. Mirrors the source argument of <code>residuals.Rceattle()</code> ; a source with no observations is skipped silently, "diet" is opt-in because it needs a multispecies model and is expensive, and "ecov" is the state-space covariate, residualized first against its own series as in WHAM's <code>make_osa_residuals()</code> .
method	How the conditional distribution is read: one of "oneStepGaussianOffMode" (default, the WHAM/SAM choice), "oneStepGaussian", "fullGaussian", "oneStepGeneric" or "cdf", passed to <code>TMB::oneStepPredict()</code> .
discrete	Whether to treat composition observations as the discrete counts they are. NULL (default) picks TRUE under method = "cdf", where the residual is not standard normal without it, and FALSE otherwise. TRUE makes composition residuals randomized quantile residuals (Dunn and Smyth 1996) and so stochastic – set seed. A Gaussian method cannot score a discrete observation, so that group falls back to "oneStepGeneric"; "cdf" with discrete = FALSE is allowed and says in a message that those residuals are biased up.
parallel	Compute the per-observation loop with <code>mclapply</code> , default TRUE, which is the main speedup for a model with random effects; set <code>options(mc.cores =)</code> to choose how many. Where a forked worker aborts instead of returning, the loop recomputes serially and prints the worker's own "irrecoverable exception" message, which comes from C and cannot be suppressed – it does not mean the call failed.
seed	Seed passed to <code>TMB::oneStepPredict()</code> for reproducible randomized-quantile residuals. Default 123.
trace	Print <code>TMB::oneStepPredict()</code> progress. Default FALSE.
...	Further arguments passed to <code>TMB::oneStepPredict()</code> .
fit	Deprecated name for object, still accepted; supplying both is an error.

Details

Computed *post hoc* and expensive – TMB re-optimizes the random effects as each observation is added – so `fit_mod()` does not produce them, and the fit must have been optimized at `estimateMode < 3`. Composition data are decomposed into univariate conditional residuals (binomial / beta-binomial; Trijoulet et al. 2023), and the last bin of each composition is fixed by the sum-to-N constraint, so it has no residual (NA).

Choosing between the methods, what each costs, how they behave under random effects, the ceiling on `|residual|` under "cdf", and where compositions at scale defeat it are in `vignette("model-diagnostics")`, with the measurements behind each.

Value

A data frame of class `rceattle_osa`, one row per residualized observation, with columns `source`, `fleet`, `fleet_name`, `species`, `sex`, `year`, `age_length_bin`, `accumulated` (the bin folds neighbouring ages, so it stands for a range rather than the age named), `length` (the conditioning length bin for caal, NA otherwise), `index_label`, `observed`, `predicted`, `sd` and `residual`.

`observed` and `predicted` are on the residualization scale: `log` for lognormal catch and index, natural scale for "Normal" and "TruncatedNormal", whitened by the lower Cholesky of the survey covariance for "MVN"/"MVNORM", and bin counts for compositions. Three cases read differently, and none is an error:

- Under `method = "cdf"`, `predicted` is NA on every row because the method forms no conditional mode. `sd` is NA under the default method too; only "oneStepGaussian" returns one.
- On a "TruncatedNormal" fleet under a Gaussian method, `predicted` is the truncated conditional mean $E[x \mid x > 0]$, which sits above the fitted index (163.8 against a fitted 100.0 on the package fixture), and `sd` is NA. The residual is unaffected.
- A composition predicted can go slightly negative where a bin holds almost no fish, and the function warns, naming the count and the years. The bin's own count drives it, not the composition's sample size – median observed count 0.05 on those rows against 4.9 elsewhere – so the warning is not by itself evidence of thin data. Those residuals are biased positive.

Adding a "TruncatedNormal" fleet can move the *other* fleets' residuals on a random-effects model, because its rows interleave with the other index fleets by year instead of forming a contiguous block: a contiguous group reproduces a single call to `1.5e-14`, while an interleaved split moved a residual by `5.8e-2`. Both orderings are valid probability-integral-transform sequences, so neither is wrong. Fixed-effect models are unaffected.

Attributes carry `seed` and `method` – the string that was passed, or a named vector where a group of rows ran under its own, such as `TruncatedNormal = "oneStepGeneric"`. Where composition types are present a "pearson" attribute holds the matching Pearson residuals, on proportions with the sample size in `sample_size` rather than the bin counts above, so the two are not comparable column by column.

References

- Thygesen, U.H., et al. 2017. Validation of ecological state space models using the Laplace approximation. *Environ. Ecol. Stat.* 24:317-339.
- Trijoulet, V., et al. 2023. Model validation for compositional data in stock assessment models. *Fish. Res.* 257:106487.

Stewart, I.J., and Monnahan, C.C. 2025. Diagnosing common sources of lack of fit to composition data using one-step-ahead residuals. Can. J. Fish. Aquat. Sci. 82:1-13.

See Also

[osa_diagnostics\(\)](#), [plot.rceattle_osa\(\)](#), [process_residuals\(\)](#), [vignette\("model-diagnostics"\)](#)

Examples

```
## Not run:
data(BS2017SS)
fit <- fit_mod(BS2017SS, estimateMode = "Hindcast")
osa <- osa_residuals(fit, source = c("index", "comp"))
plot(osa)

## End(Not run)
```

parameter_dictionary *Look up what a CEATTLE parameter is*

Description

The TMB parameter vector uses transformed, abbreviated names, `log_M1`, `R_log_sd`, `index_log_q`. This returns the table mapping each one to the quantity it represents on its natural scale, which process it belongs to, what it means, and its dimensions.

Usage

```
parameter_dictionary(internal = NULL, process = NULL)
```

Arguments

<code>internal</code>	Character vector of internal parameter names, as they appear in <code>names(fitted\$estimated_params)</code> or in an <code>sreport()</code> . Default <code>NULL</code> returns every parameter.
<code>process</code>	Character vector restricting the table to one or more parts of the model: "recruitment", "mortality", "growth", "fishing", "catchability", "selectivity", "observation", "predation", "linkage" or "internal".

Details

A name beginning `log_`, or ending `_log_sd`, is estimated on the log scale, so `exp()` turns it into the quantity named in the natural column. Most fitted values are already back-transformed in `fitted$quantities`, so this is mainly needed when reading `estimated_params` or an `sreport()` directly.

Dimensions use the model's own notation: `nspp` species, `nsex` sexes, `nages` age bins, `nyrs` years including the projection, `nyrs_hind` hindcast years only, `n_flt` fleets and `n_sel` selectivity blocks. `n_sel` equals `n_flt`: fleets sharing a `Selectivity_index` share one block through the map, not through a narrower array.

Value

A data frame with one row per parameter and columns internal, natural, process, meaning and dims.

See Also

`fit_mod()` for the fitted object these names come from.

Examples

```
# The whole table
head(parameter_dictionary())

# What is R_log_sd?
parameter_dictionary("R_log_sd")

# Everything governing selectivity
parameter_dictionary(process = "selectivity")

# Search the meanings by keyword
dict <- parameter_dictionary()
dict[grep("catchability", dict$meaning, ignore.case = TRUE), ]
```

parameter_index	<i>Locate every estimated parameter in the model's own coordinates</i>
-----------------	--

Description

One row per estimated (fixed-effect) parameter, giving the fleet, species, sex, age, bin, year or slot it occupies.

Usage

```
parameter_index(object)
```

Arguments

object A fitted Rceattle model from `fit_mod()`.

Details

TMB names each element of `obj$par` after its parameter block, so a convergence diagnostic can report that `sel_coff_dev` is non-identifiable but not which of its elements.

The mapping is recovered by pushing a tagged parameter vector back through TMB's own `parList()`, so it reflects `build_map()` exactly: parameters mapped off are absent, and fleets sharing a `Selectivity_index` or `Catchability_index` appear once, as the single parameter they are, with `n_cells` recording how many array cells they drive.

Ages are absolute. `init_dev` holds one deviation per age from `minage + 1` upward, since age `minage` in the first year is recruitment.

Coordinate columns are NA where the axis does not apply to a block. The linkage and environmental-covariate blocks (`beta_linkage`, `M1_beta`, ...) are indexed by a linkage-table row rather than a model coordinate, so they hold no coordinates and are reported by element number.

A selectivity slot is named from the fleet's Selectivity: slot 2 of `sel_inf` is a descending inflection for the double-logistic family, the right-tail floor under `DoubleNormal`, and the age-1 selectivity under `LogisticPM`.

Under `estimateMode = "Estimate"` with any HCR but `"NoFishing"`, `object$obj` is the projection object, whose only free parameters are `log_Ftarget / log_Flimit`, so the index describes those, not the hindcast. `fit$convergence` reads the hindcast index that `fit_mod()` captured before the projection remapped the model, and checks it against the vector it is labelling either way.

Value

A data frame with one row per estimated parameter and columns `par_index` (position in `obj$par`), `block`, `species`, `fleet`, `sex`, `age`, `bin`, `year`, `slot`, `n_cells` and `label`.

See Also

`parameter_dictionary()` for what each block means.

plot.Rceattle	<i>Plot method for fitted Rceattle models</i>
---------------	---

Description

Thin S3 dispatcher around the package's existing `plot_*`() functions so that `plot(fit)` works the way users expect. Pick the panel with `what`; everything in ... is forwarded to the underlying function.

Usage

```
## S3 method for class 'Rceattle'
plot(x, what = "biomass", ...)
```

Arguments

<code>x</code>	An object of class "Rceattle" returned by <code>fit_mod()</code> .
<code>what</code>	Character. One of "biomass" (default), "ssb", "recruitment", "depletion" (total biomass / B0), "ssb_depletion" (female spawning biomass / SB0, the quantity a Tier 3 HCR compares against B40%), "index", "catch", "selectivity", "mortality", or "data".
...	Passed to the underlying plotting function.

Value

Invisibly returns NULL. Called for the side effect of producing a plot.

plot.rceattle_osa *Plot one-step-ahead (OSA) residual diagnostics*

Description

Diagnostic plots for an `rceattle_osa` object (from `osa_residuals()`), following the recommendations of Stewart and Monnahan (2025) and the styling of the NOAA-AFSC `afscOSA` package. Under a correctly specified model OSA residuals are iid standard normal, so the headline diagnostic is statistical (the Q-Q plot with its annotated SDNR / tail statistics).

Up to two separate figures are drawn, depending on which data are present:

1. **Aggregate** (index / catch): a Q-Q panel faceted by data source. These series have no age/length bin, so no bubble plots are drawn.
2. **Composition** (comp / caal): a Q-Q panel, a signed OSA-residual bubble panel, and a signed Pearson-residual bubble panel (the Pearson residuals held on the `rceattle_osa` object). By default age-based bins (age composition and conditional age-at-length) are shown in the left column and length-based bins in the right column, each with its own bin axis; set `combine = FALSE` to draw the age and length composition as two separate figures instead (useful when a model has many fleets/species).

Panel headers use the fleet name from `fleet_control`. Process residuals (from `process_residuals()`) are drawn as a Q-Q panel plus a residual-by-year panel.

Usage

```
## S3 method for class 'rceattle_osa'
plot(
  x,
  source = "all",
  species = NULL,
  combine = TRUE,
  add_sdnr_ci = TRUE,
  add_qq_quantiles = TRUE,
  ...
)
```

Arguments

<code>x</code>	An <code>rceattle_osa</code> object from <code>osa_residuals()</code> or <code>process_residuals()</code> .
<code>source</code>	Data source(s) to plot: any of "index", "catch", "comp", "caal", "diet", or "all" (default). Mirrors the source argument of <code>residuals.Rceattle()</code> ; filters which figures are produced.
<code>species</code>	Optional species code(s) to include (matched against the species column). Default NULL keeps all species.
<code>combine</code>	Logical. When TRUE (default), age and length composition share one figure (age in the left column, length in the right). When FALSE, they are drawn as separate <code>composition_age</code> / <code>composition_length</code> figures.

add_sdnr_ci	Logical. Show the chi-square null interval beside the SDNR annotation on each Q-Q panel. Default TRUE.
add_qq_quantiles	Logical. Annotate each Q-Q panel with the tail order statistics and their exact null intervals. Default TRUE.
...	Unused.

Value

Invisibly, a named list of the assembled ggplot / cowplot objects (some of aggregate, composition / composition_age / composition_length, and process). Called for its side effect of drawing the plot(s).

References

Stewart, I.J., and Monnahan, C.C. 2025. Can. J. Fish. Aquat. Sci. 82:1-13.

See Also

[osa_residuals\(\)](#), [osa_diagnostics\(\)](#)

plot.Rceattle_profile *Plot method for a likelihood profile*

Description

Shorthand for [plot_profile\(\)](#): draws the likelihood components against the profiled parameter, with the total overlaid.

Usage

```
## S3 method for class 'Rceattle_profile'
plot(x, ...)
```

Arguments

x An "Rceattle_profile" object from [profile.Rceattle\(\)](#).
 ... Passed to [plot_profile\(\)](#).

Value

A ggplot object.

plot_b_eaten

Plot biomass eaten by predation

Description

Total biomass of each species consumed by its predators, summed over sex and age. For an MSE object the projection is summarized across simulations with a mean and a 95% interval.

Usage

```
plot_b_eaten(
  Rceattle,
  file = NULL,
  model_names = NULL,
  line_col = NULL,
  species = NULL,
  spnames = NULL,
  add_ci = FALSE,
  lwd = 3,
  save = FALSE,
  right_adj = 0,
  width = 7,
  height = 6.5,
  minyr = NULL,
  incl_proj = FALSE,
  mod_cex = 1,
  alpha = 0.4,
  mod_avg = rep(FALSE, length(Rceattle)),
  mse = FALSE,
  OM = TRUE,
  maxyr = NULL,
  lty = 1,
  incl_mean = FALSE,
  top_adj = 0.15
)
```

Arguments

Rceattle	A single <code>fit_mod()</code> object or a list of them (overlaid).
file	Optional file stem; the figure is written to <code><file>_<suffix>.png</code> if given.
model_names	Legend labels for the models.
line_col	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.

species	Species to include, as indices (c(1, 3)), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in spnames; a character species that matches no species name is read as labels for back-compatibility.
spnames	Species labels, length nspp. Default: the model's own.
add_ci	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.
lwd	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
save	Ignored. Only the time-series plotters write their series to CSV.
right_adj	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
width, height	Saved figure size in inches.
minyr, maxyr	First / last year to plot.
incl_proj	Include the projection years, with a dashed divider at the last hindcast year.
mod_cex	Ignored. Base-graphics leftover: legend text size. Use p + ggplot2::theme(legend.text = ggplot2::element_text(size = ...)).
alpha	Transparency of confidence ribbons and shaded areas, between 0 and 1.
mod_avg	TRUE/FALSE
mse	Is an MSE object from load_mse or run_mse
OM	if mse == TRUE, use the OM (TRUE) or EM (FALSE) for plotting?
lty	Line type. A vector varies it across the levels of whatever the figure separates by line type.
incl_mean	Add a horizontal line at the hindcast mean of each series.
top_adj	Ignored. Base-graphics leftover; see right_adj.

plot_b_eaten_prop	<i>Plot biomass consumed of each prey species by predator</i>
-------------------	---

Description

Function that plots the biomass consumed trends as estimated from Rceattle. Returns and saves a figure with the biomass eaten trajectory.

Usage

```
plot_b_eaten_prop(
  Rceattle,
  file = NULL,
  model_names = NULL,
  line_col = NULL,
  spnames = NULL,
  species = NULL,
```

```

    lwd = 3,
    right_adj = 0,
    top_adj = 0.15,
    minyr = NULL,
    mohns = NULL,
    width = 7,
    height = 6.5,
    incl_proj = FALSE,
    incl_mean = FALSE,
    add_ci = FALSE,
    mod_cex = 1,
    maxyr = NULL,
    lty = 1
  )

```

Arguments

<code>Rceattle</code>	A single <code>fit_mod()</code> object or a list of them (overlaid).
<code>file</code>	Optional file stem; the figure is written to <code><file>_<suffix>.png</code> if given.
<code>model_names</code>	Legend labels for the models.
<code>line_col</code>	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
<code>spnames</code>	Species labels, length <code>nspp</code> . Default: the model's own.
<code>species</code>	Species to include, as indices (<code>c(1, 3)</code>), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in <code>spnames</code> ; a character species that matches no species name is read as labels for back-compatibility.
<code>lwd</code>	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
<code>right_adj</code>	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
<code>top_adj</code>	Ignored. Base-graphics leftover; see <code>right_adj</code> .
<code>minyr, maxyr</code>	First / last year to plot.
<code>mohns</code>	Ignored. Formerly annotated the figure with Mohn's rho from <code>retrospective()</code> ; add it with <code>ggplot2::labs(subtitle = ...)</code> instead.
<code>width, height</code>	Saved figure size in inches.
<code>incl_proj</code>	Include the projection years, with a dashed divider at the last hindcast year.
<code>incl_mean</code>	Add a horizontal line at the hindcast mean of each series.
<code>add_ci</code>	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.
<code>mod_cex</code>	Ignored. Base-graphics leftover: legend text size. Use <code>p + ggplot2::theme(legend.text = ggplot2::element_text(size = ...))</code> .
<code>lty</code>	Line type. A vector varies it across the levels of whatever the figure separates by line type.

Details

Colour separates predators; line type separates models. Panels are prey species.

plot_biomass	<i>Plot biomass</i>
--------------	---------------------

Description

Plots the mean minage+ biomass (million mt) and 95\

Usage

```
plot_biomass(  
  Rceattle,  
  file = NULL,  
  model_names = NULL,  
  line_col = NULL,  
  species = NULL,  
  spnames = NULL,  
  add_ci = FALSE,  
  lwd = 3,  
  save = FALSE,  
  right_adj = 0,  
  legend.pos = "topright",  
  width = 7,  
  height = 6.5,  
  minyr = NULL,  
  maxyr = NULL,  
  incl_proj = FALSE,  
  mod_cex = 1,  
  lty = rep(1, length(Rceattle)),  
  alpha = 0.4,  
  mod_avg = rep(FALSE, length(Rceattle)),  
  mse = FALSE,  
  OM = TRUE,  
  reference = NULL,  
  ylab = NULL  
)
```

Arguments

- Rceattle A single `fit_mod()` object or a list of them (overlaid).
- file Optional file stem; the figure is written to `<file>_<suffix>.png` if given.
- model_names Legend labels for the models.

line_col	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
species	Species to include, as indices (c(1, 3)), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in snames; a character species that matches no species name is read as labels for back-compatibility.
snames	Species labels, length nspp. Default: the model's own.
add_ci	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.
lwd	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
save	Write the plotted series to CSV alongside the figure.
right_adj	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
legend.pos	Ignored. Base-graphics leftover: legend placement. Use p + ggplot2::theme(legend.position = "right").
width, height	Saved figure size in inches.
minyr, maxyr	First / last year to plot.
incl_proj	Include the projection years, with a dashed divider at the last hindcast year.
mod_cex	Ignored. Base-graphics leftover: legend text size. Use p + ggplot2::theme(legend.text = ggplot2::element_text(size = ...)).
lty	Line type. A vector varies it across the levels of whatever the figure separates by line type.
alpha	Transparency of confidence ribbons and shaded areas, between 0 and 1.
mod_avg	TRUE/FALSE
mse	Is an MSE object from load_mse or run_mse
OM	if mse == TRUE, use the OM (TRUE) or EM (FALSE) for plotting?
reference	A model to overlay for comparison, drawn at 1.5x lwd and labelled "Reference". It takes the next colour from the palette, or black if line_col is supplied.
ylab	Y-axis label. NULL (default) derives one from output and the model's minage.

Value

Returns and saves a figure with the biomass trajectory.

plot_catch	<i>Fishery catch fits</i>
------------	---------------------------

Description

Plots fitted fishery catch: observed points with lognormal 95% intervals and the model-predicted catch, faceted by fishery fleet. For MSE objects the projection period is summarized with 50% / 95% ribbons across simulations.

Usage

```
plot_catch(
  Rceattle,
  file = NULL,
  model_names = NULL,
  species = NULL,
  incl_proj = FALSE,
  width = 7,
  height = 6.5,
  error = TRUE,
  maxyr = NULL,
  mse = FALSE,
  fleets = NULL,
  line_col = NULL,
  right_adj = 0,
  top_adj = 1.2,
  single.plots = FALSE,
  alpha = 0.4,
  lwd = 2,
  ymax = NULL
)
```

Arguments

Rceattle	A single <code>fit_mod()</code> object or a list of them (overlaid).
file	Optional file stem; if given the figure is written to <code><file>_survey_indices.png</code> .
model_names	Legend labels for the models.
species	Species (indices) to include. Default all.
incl_proj	Include projection years.
width, height	Saved figure size (inches).
error	Draw observed points and error bars.
maxyr	Last year to plot.
mse	Is Rceattle an MSE object (operating models)?
fleets	Fishery fleets to include (indices into the sorted fleet codes). Default all.

line_col, right_adj, top_adj, single.plots

Deprecated base-graphics arguments, retained for back-compatibility and ignored.

alpha, lwd, ymax Deprecated base-graphics arguments, ignored.

Details

The predicted line is the MEDIAN, as in `plot_index()`: $\text{catch_hat} * \exp(-\text{bias_adjust_obs} * \text{sd}^2 / 2)$. The catch likelihood is lognormal for every fishery, so this applies throughout.

Value

A ggplot object.

plot_catchability	<i>Survey catchability over time</i>
-------------------	--------------------------------------

Description

Plots the fitted survey catchability q by year, faceted by fleet. Every fleet holding `index_data` is drawn, a fishery with a CPUE series as much as a survey.

Usage

```
plot_catchability(
  Rceattle,
  file = NULL,
  model_names = NULL,
  species = NULL,
  spnames = NULL,
  mse = FALSE,
  minyr = NULL,
  maxyr = NULL,
  incl_mean = FALSE,
  log = FALSE,
  line_col = NULL,
  lwd = 3,
  lty = 1,
  width = 7,
  height = 6.5,
  right_adj = 0,
  top_adj = 0.05,
  single.plots = FALSE
)
```

Arguments

Rceattle	A single <code>fit_mod()</code> object or a list of them (overlaid).
file	Optional file stem; the figure is written to <code><file>_<suffix>.png</code> if given.
model_names	Legend labels for the models.
species	Species to include, as indices (<code>c(1, 3)</code>), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in <code>spnames</code> ; a character species that matches no species name is read as labels for back-compatibility.
spnames	Species labels, length <code>nspp</code> . Default: the model's own.
mse	Is Rceattle an MSE object? Its operating models are drawn.
minyr, maxyr	First / last year to plot.
incl_mean	Add a horizontal line at the hindcast mean of each series.
log	Plot catchability on the log scale.
line_col	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
lwd	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
lty	Line type. A vector varies it across the levels of whatever the figure separates by line type.
width, height	Saved figure size in inches.
right_adj	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
top_adj	Ignored. Base-graphics leftover; see <code>right_adj</code> .
single.plots	Ignored. Base-graphics leftover: one device per panel. The ggplot facets instead.

Details

The value plotted is the realized catchability the model scaled the survey by, whichever route produced it: an estimated or fixed mean, annual deviations under `Time_varying_q`, an environmental regression (`Catchability = "Environmental"`), a `q` linkage such as `rw(1 | Year)` or `ar1(1 | Year)`, or the closed-form value under `"Analytical"` / `"AnalyticalArith"`. A fleet with a time-invariant catchability draws a flat line, which is the honest picture of a constant `q` rather than an empty panel.

Hindcast years only. The model applies catchability over the hindcast and does not project it, so the series stops at `endyr` regardless of the projection horizon. There is no `incl_proj`.

`line_col` and `lty` separate the **models**, so overlaying several fits compares their catchability series panel by panel.

Value

A ggplot object.

Examples

```
data(BS2017SS)
fit <- Rceattle::fit_mod(data_list = BS2017SS, msmMode = 0)
plot_catchability(fit)
plot_catchability(fit, log = TRUE)
```

plot_comp

Plot composition fits and residuals

Description

Diagnostic plots for age / length composition data from a fitted [Rceattle](#) model, drawn with `ggplot2` for a consistent look with [plot.rceattle_osa\(\)](#). With the default `residual_type = "pearson"` three figures are produced per fleet x composition type:

- **Pearson residual bubbles** by year and bin, faceted by fleet (and, for joint-sex data, by sex); red = positive, blue = negative, sized by magnitude. The Pearson residual is the same form used by [residuals.Rceattle\(\)](#), standardized by the variance the fleet's own likelihood assumes, so the weight column and any Dirichlet-multinomial overdispersion are already in the denominator.
- **Annual composition**: observed (shaded area) vs fitted (line) proportion at age / length, one panel per year. Joint-sex data are mirrored (females up, males down).
- **Aggregated composition**: observed vs fitted, counts pooled over the fitted hindcast years. The interval holds 95% of the data the model predicts (not a confidence interval on the mean), so observations outside it indicate misfit. Poor where the expected count is below about 10, and slightly narrow throughout, because it treats the fitted proportions as known rather than estimated from these same data.

The shaded area and fitted line span only the observed bins (they do not extend past the first/last bin), and bins with zero observed proportion are retained (only NA bins are dropped), so the curves are not interpolated across gaps.

Usage

```
plot_comp(
  Rceattle,
  file = NULL,
  model_names = NULL,
  species = NULL,
  cex = 3,
  lwd = 3,
  right_adj = 0,
  residual_type = c("pearson", "osa"),
  add_agg_ci = TRUE,
  add_agg_n = TRUE
)
```

Arguments

Rceattle	A single fitted Rceattle model.
file	Optional filename stem; when supplied, each figure is also saved as a PNG.
model_names	Unused (kept for back-compatibility).
species	Optional species code(s) to plot (matched against the Species column); NULL (default) plots all. Mirrors the species argument of <code>residuals.Rceattle()</code> and <code>plot.rceattle_osa()</code> .
cex	Unused (kept for back-compatibility).
lwd	Width of the fitted-composition line. Default 3.
right_adj	Unused (kept for back-compatibility).
residual_type	"pearson" (default) for the ggplot2 Pearson-residual and composition-fit figures drawn here, or "osa" to instead draw the one-step-ahead residual diagnostics via <code>osa_residuals()</code> and <code>plot.rceattle_osa()</code> , a Q-Q plot (with SDNR / tail annotation) alongside signed OSA- and Pearson-residual bubbles. The "osa" path builds its observation data on demand, so it works with any fit.
add_agg_ci	Logical. Draw the 95% prediction interval on the aggregated composition figure. Default TRUE.
add_agg_n	Logical. Annotate the aggregated composition figure with the input sample size, the effective sample size the likelihood assumed, and the McAllister-Ianelli effective sample size this fit implies. Default TRUE.

Value

Invisibly, a named list of the ggplot objects. Called for its side effect of drawing (and optionally saving) the figures.

plot_data	<i>Timeline of data used in the model likelihoods</i>
-----------	---

Description

Plots a timeline of presence/absence (and optionally relative quantity) of the data sources contributing to an Rceattle model’s likelihood, by year and fleet. Modeled after Stock Synthesis’s `r4ss::SSplotData()`.

Usage

```
plot_data(  
  Rceattle,  
  file = NULL,  
  subplots = 1:2,  
  datatypes = "all",  
  fleets = "all",  
  species = "all",  
)
```

```

ghost = FALSE,
fleetcol = "default",
width = 7,
height = 5,
res = 200,
ptsize = 10,
margins = c(5.1, 2.1, 2.1, 10.1),
cex = 2,
lwd = 12,
maxsize = 1,
alphasize = 1,
mainTitle = FALSE,
cex.main = 1
)

```

Arguments

Rceattle	Either a single Rceattle model object exported from <code>Rceattle::fit_mod()</code> , or a raw <code>data_list</code> (e.g. one of the bundled datasets such as BS2017MS).
file	Path/prefix used to save the figure. If NULL (default) the plot is drawn to the active device only. When supplied, the figure is written to <code>paste0(file, "_data_plot.png")</code> .
subplots, res, ptsize, margins, cex, lwd, maxsize, mainTitle, cex.main	Deprecated base-graphics arguments, retained for back-compatibility and ignored. One faceted figure is drawn; size, resolution and titles come from the ggplot theme and from width/height.
datatypes	Either "all" or a subset of <code>c("catch", "index", "agecomp", "lencomp", "caal", "diet")</code> .
fleets	Either "all" or an integer vector of <code>Fleet_codes</code> to include. Diet "fleets" are encoded after the regular fleets (one per predator species).
species	Either "all" or an integer vector of species indices to include.
ghost	Logical. If TRUE, also include rows excluded from the likelihood ($\text{Year} < 0$). Defaults to FALSE.
fleetcol	Either "default" or a vector of colors (one per fleet shown after subsetting). Diet entries get an additional color appended.
width, height	Figure dimensions in inches.
alphasize	Bubble fill transparency (0-1).

Details

Years with $\text{Year} > 0$ contribute to the likelihood; rows with $\text{Year} < 0$ (kept for fitting comparison only) are shown when `ghost = TRUE`. Diet data uses $\text{Year} == 0$ for the fixed reference stomach composition; those entries are plotted at the model endyr and labeled per predator species.

Value

Invisibly, a list with `typetable`, the long data frame underlying the plot (year, fleet, data type, relative size). Point area is scaled within each data type: catch in metric tons, 1/SE for indices, sample size for compositions.

plot_depletion	<i>Plot biomass depletion</i>
----------------	-------------------------------

Description

Plots the mean biomass depletion and 95\

Usage

```
plot_depletion(
  Rceattle,
  file = NULL,
  model_names = NULL,
  line_col = NULL,
  species = NULL,
  spnames = NULL,
  add_ci = FALSE,
  lwd = 3,
  save = FALSE,
  right_adj = 0,
  legend.pos = "topright",
  width = 7,
  height = 6.5,
  minyr = NULL,
  maxyr = NULL,
  incl_proj = FALSE,
  mod_cex = 1,
  lty = rep(1, length(Rceattle)),
  alpha = 0.4,
  mod_avg = rep(FALSE, length(Rceattle)),
  mse = FALSE,
  OM = TRUE,
  reference = NULL,
  ylab = NULL
)
```

Arguments

Rceattle	A single <code>fit_mod()</code> object or a list of them (overlaid).
file	Optional file stem; the figure is written to <code><file>_<suffix>.png</code> if given.
model_names	Legend labels for the models.
line_col	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.

species	Species to include, as indices (<code>c(1, 3)</code>), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in <code>spnames</code> ; a character species that matches no species name is read as labels for back-compatibility.
spnames	Species labels, length <code>nspp</code> . Default: the model's own.
add_ci	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.
lwd	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
save	Write the plotted series to CSV alongside the figure.
right_adj	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
legend.pos	Ignored. Base-graphics leftover: legend placement. Use <code>p + ggplot2::theme(legend.position = "right")</code> .
width, height	Saved figure size in inches.
minyr, maxyr	First / last year to plot.
incl_proj	Include the projection years, with a dashed divider at the last hindcast year.
mod_cex	Ignored. Base-graphics leftover: legend text size. Use <code>p + ggplot2::theme(legend.text = ggplot2::element_text(size = ...))</code> .
lty	Line type. A vector varies it across the levels of whatever the figure separates by line type.
alpha	Transparency of confidence ribbons and shaded areas, between 0 and 1.
mod_avg	TRUE/FALSE
mse	Is an MSE object from <code>load_mse</code> or <code>run_mse</code>
OM	if <code>mse == TRUE</code> , use the OM (TRUE) or EM (FALSE) for plotting?
reference	A model to overlay for comparison, drawn at 1.5x <code>lwd</code> and labelled "Reference". It takes the next colour from the palette, or black if <code>line_col</code> is supplied.
ylab	Y-axis label. NULL (default) derives one from output and the model's minage.

Value

Returns and saves a figure with the biomass depletion trajectory.

<code>plot_depletionSSB</code>	<i>Plot SSB depletion (deprecated name)</i>
--------------------------------	---

Description

Deprecated alias for `plot_ssb_depletion`, which is the snake_case spelling the rest of the package uses. Please use `plot_ssb_depletion()` instead.

Usage

```
plot_depletionSSB(...)
```

Arguments

... passed unchanged to [plot_ssb_depletion](#).

plot_diet_comp	<i>Plot diet composition fits</i>
----------------	-----------------------------------

Description

If year == 0, diet data are averaged from suit_styr to suit_endyr If prey_age >= 0 diet data are diet proportion of prey-at-age in predator-at-age If prey_age < 0 diet data are diet proportion of prey-spp in predator-at-age (sum across prey ages) If prey_age < 0 and pred_age < 0, diet data are mean diet proportion of prey-spp in predator-spp (sum across prey ages and take mean across predator ages) If prey_age < 0 and pred_age < -500, diet data are weighted mean diet proportion of prey-spp in predator-spp (sum across prey ages and take weighted mean across predator ages)

Usage

```
plot_diet_comp(Rceattle, file = NULL, species = NULL)
```

Arguments

Rceattle	Single or list of Rceattle model objects exported from Rceattle
file	Path/prefix for the saved figure. NULL (default) draws without saving.
species	Species names for legend

Value

Returns and saves a figure

plot_diet_comp1	<i>Plot diet composition fits (bubble/grid diagnostic)</i>
-----------------	--

Description

plot_diet_comp1() is an alias of [plot_diet_comp\(\)](#): it produces the predator-age x prey-age bubble grids (observed / estimated / Pearson residual) for each predator-prey-year interaction. It is provided as a named entry point alongside [plot_diet_comp2\(\)](#) (the aggregation-aware line / bar / bubble plots) for scripts that call both by name.

Usage

```
plot_diet_comp1(Rceattle, file = NULL, species = NULL)
```

Arguments

Rceattle	Single Rceattle model object exported from Rceattle.
file	Optional file-name prefix for saved figures.
species	Optional species names for the legend.

Value

Invisibly returns a list of the printed plot grids (see [plot_diet_comp\(\)](#)).

See Also

[plot_diet_comp\(\)](#), [plot_diet_comp2\(\)](#)

Examples

```
## Not run:
plot_diet_comp1(my_msm_fit)

## End(Not run)
```

plot_diet_comp2	<i>Plot diet composition fits (aggregation-aware)</i>
-----------------	---

Description

Diagnostic plots for diet-composition fits that adapt to how each predator-prey interaction is aggregated (see [plot_diet_comp\(\)](#) for the aggregation conventions):

- prey-age aggregated (predator age resolved): line plot of observed vs estimated diet proportion against predator age, with 95% CI ribbons;
- predator aggregated (prey age resolved): line plot against prey age;
- both aggregated: dodged bar plot of observed vs estimated proportion;
- fully disaggregated: predator-age x prey-age bubble grids (observed / estimated / Pearson residual).

The observed proportions, fitted proportions and Pearson residuals come from the diet data path (`quantities$diet_hat / residuals(source = "diet") / data_list$diet_data`) via the internal `.diet_plot_data()` helper. Observed 95% CI ribbons are a normal approximation to the binomial proportion from `Sample_size`. Estimated 95% CI ribbons are drawn only when the `sdreport` exposes a `diet_hat` standard error; the C++ template `REPORT()`s but does not `ADREPORT()` `diet_hat`, so the estimated ribbon is unavailable unless the template is changed to `ADREPORT(diet_hat)` (the code path is retained so it activates automatically if that ever happens, or when `sdrep` is `NULL`).

Usage

```
plot_diet_comp2(Rceattle, file = NULL, species = NULL)
```

Arguments

Rceattle	A single Rceattle model object.
file	Optional file-name prefix for saved figures.
species	Optional species names for the legend.

Value

Invisibly returns a list of the printed plot objects.

See Also

[plot_diet_comp\(\)](#), [plot_diet_comp1\(\)](#)

Examples

```
## Not run:
plot_diet_comp2(my_msm_fit)

## End(Not run)
```

plot_exploitable_biomass

Plot exploitable biomass

Description

Plots the mean exploitable biomass (million mt) and 95\

Usage

```
plot_exploitable_biomass(
  Rceattle,
  file = NULL,
  model_names = NULL,
  line_col = NULL,
  species = NULL,
  spnames = NULL,
  add_ci = FALSE,
  lwd = 3,
  save = FALSE,
  right_adj = 0,
  legend.pos = "topright",
  width = 7,
  height = 6.5,
  minyr = NULL,
  maxyr = NULL,
  incl_proj = FALSE,
```

```

    mod_cex = 1,
    lty = rep(1, length(Rceattle)),
    alpha = 0.4,
    mod_avg = rep(FALSE, length(Rceattle)),
    mse = FALSE,
    OM = TRUE,
    reference = NULL,
    ylab = NULL
  )

```

Arguments

Rceattle	A single <code>fit_mod()</code> object or a list of them (overlaid).
file	Optional file stem; the figure is written to <code><file>_<suffix>.png</code> if given.
model_names	Legend labels for the models.
line_col	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
species	Species to include, as indices (<code>c(1, 3)</code>), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in <code>spnames</code> ; a character species that matches no species name is read as labels for back-compatibility.
spnames	Species labels, length <code>nspp</code> . Default: the model's own.
add_ci	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.
lwd	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
save	Write the plotted series to CSV alongside the figure.
right_adj	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
legend.pos	Ignored. Base-graphics leftover: legend placement. Use <code>p + ggplot2::theme(legend.position = "right")</code> .
width, height	Saved figure size in inches.
minyr, maxyr	First / last year to plot.
incl_proj	Include the projection years, with a dashed divider at the last hindcast year.
mod_cex	Ignored. Base-graphics leftover: legend text size. Use <code>p + ggplot2::theme(legend.text = ggplot2::element_text(size = ...))</code> .
lty	Line type. A vector varies it across the levels of whatever the figure separates by line type.
alpha	Transparency of confidence ribbons and shaded areas, between 0 and 1.
mod_avg	TRUE/FALSE
mse	Is an MSE object from <code>load_mse</code> or <code>run_mse</code>
OM	if <code>mse == TRUE</code> , use the OM (TRUE) or EM (FALSE) for plotting?
reference	A model to overlay for comparison, drawn at 1.5x <code>lwd</code> and labelled "Reference". It takes the next colour from the palette, or black if <code>line_col</code> is supplied.
ylab	Y-axis label. NULL (default) derives one from output and the model's minage.

Value

Returns and saves a figure with the exploitable biomass trajectory.

<code>plot_f</code>	<i>plot F</i>
---------------------	---------------

Description

Fishing mortality over time, one panel per species, with the Ftarget (blue) and Flimit (red) reference points drawn per panel.

Usage

```
plot_f(  
  Rceattle,  
  file = NULL,  
  model_names = NULL,  
  line_col = NULL,  
  species = NULL,  
  spnames = NULL,  
  add_ci = FALSE,  
  lwd = 3,  
  save = FALSE,  
  right_adj = 0,  
  legend.pos = "topright",  
  width = 7,  
  height = 6.5,  
  minyr = NULL,  
  maxyr = NULL,  
  incl_proj = FALSE,  
  mod_cex = 1,  
  lty = rep(1, length(Rceattle)),  
  alpha = 0.4,  
  mod_avg = rep(FALSE, length(Rceattle)),  
  mse = FALSE,  
  OM = TRUE,  
  reference = NULL,  
  ylab = NULL  
)
```

Arguments

- Rceattle A single `fit_mod()` object or a list of them (overlaid).
- file Optional file stem; the figure is written to `<file>_<suffix>.png` if given.
- model_names Legend labels for the models.

line_col	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
species	Species to include, as indices (c(1, 3)), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in snames; a character species that matches no species name is read as labels for back-compatibility.
snames	Species labels, length nspp. Default: the model's own.
add_ci	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.
lwd	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
save	Write the plotted series to CSV alongside the figure.
right_adj	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
legend.pos	Ignored. Base-graphics leftover: legend placement. Use p + ggplot2::theme(legend.position = "right").
width, height	Saved figure size in inches.
minyr, maxyr	First / last year to plot.
incl_proj	Include the projection years, with a dashed divider at the last hindcast year.
mod_cex	Ignored. Base-graphics leftover: legend text size. Use p + ggplot2::theme(legend.text = ggplot2::element_text(size = ...)).
lty	Line type. A vector varies it across the levels of whatever the figure separates by line type.
alpha	Transparency of confidence ribbons and shaded areas, between 0 and 1.
mod_avg	TRUE/FALSE
mse	Is an MSE object from load_mse or run_mse
OM	if mse == TRUE, use the OM (TRUE) or EM (FALSE) for plotting?
reference	A model to overlay for comparison, drawn at 1.5x lwd and labelled "Reference". It takes the next colour from the palette, or black if line_col is supplied.
ylab	Y-axis label. NULL (default) derives one from output and the model's minage.

Value

Returns and saves a figure with the fishing mortality trajectory.

plot_form	<i>Plot functional form</i>
-----------	-----------------------------

Description

Function to plot the functional form estimated or specified by Rceattle

Usage

```
plot_form(params = NULL, pred = 1, pred_age = 1, prey = 1, msmMode = 3)
```

Arguments

params	Parameter list object from build_params or Rceattle
pred	Predator index
pred_age	Predator age
prey	Prey index
msmMode	Multispecies mode integer specifying functional form

plot_index	<i>Survey index fits</i>
------------	--------------------------

Description

Plots fitted index series: observed points with 95% observation intervals and the model-predicted index, faceted by fleet. Every fleet holding index_data is drawn, a fishery with a CPUE series as much as a survey.

Usage

```
plot_index(
  Rceattle,
  file = NULL,
  model_names = NULL,
  species = NULL,
  incl_proj = FALSE,
  width = 7,
  height = 6.5,
  error = TRUE,
  log = FALSE,
  line_col = NULL,
  right_adj = 0,
  top_adj = 0.05,
  single.plots = FALSE
)
```

Arguments

Rceattle	A single <code>fit_mod()</code> object or a list of them (overlaid).
file	Optional file stem; if given the figure is written to <code><file>_survey_indices.png</code> .
model_names	Legend labels for the models.
species	Species (indices) to include. Default all.
incl_proj	Include projection years.
width, height	Saved figure size (inches).
error	Draw observed points and error bars.
log	Plot the index on the log scale.
line_col, right_adj, top_adj, single.plots	Deprecated base-graphics arguments, retained for back-compatibility and ignored.

Details

The interval follows the fleet's `Index_distribution`: lognormal quantiles on a "Lognormal" fleet, and observed ± 1.96 sd truncated at zero on a natural-scale one ("MVN", "MVNORM", "Normal", "TruncatedNormal"), whose sd is absolute and in the units of the index.

The predicted line is the MEDIAN of the fitted observation distribution, $\text{index_hat} * \exp(-\text{bias_adjust_obs} * \text{sd}^2 / 2)$, which is where the lognormal likelihood is centred, so about half the observations should fall below it. It equals the reported `index_hat` where `bias_adjust_obs = FALSE`. A natural-scale fleet has no such offset and is drawn unchanged.

Value

A ggplot object.

plot_index_residual	<i>Survey index residuals</i>
---------------------	-------------------------------

Description

Plots survey index residuals by year, faceted by survey fleet (`residual_type = "pearson"`, the default), or one-step-ahead (OSA) residual diagnostics for a single fit (`residual_type = "osa"`, via `osa_residuals()` / `plot.rceattle_osa()`).

Usage

```
plot_index_residual(
  Rceattle,
  file = NULL,
  model_names = NULL,
  species = NULL,
  width = 7,
```

```

height = 6.5,
residual_type = c("pearson", "osa"),
line_col = NULL,
right_adj = 0,
top_adj = 0.05,
incl_proj = FALSE,
single.plots = FALSE
)

```

Arguments

<code>Rceattle</code>	A single <code>fit_mod()</code> object or a list of them (overlaid).
<code>file</code>	Optional file stem; if given the figure is written to <code><file>_survey_indices.png</code> .
<code>model_names</code>	Legend labels for the models.
<code>species</code>	Species (indices) to include. Default all.
<code>width, height</code>	Saved figure size (inches).
<code>residual_type</code>	"pearson" (standardized index residuals) or "osa".
<code>line_col, right_adj, top_adj, single.plots</code>	Deprecated base-graphics arguments, retained for back-compatibility and ignored.
<code>incl_proj</code>	Include projection years.

Details

The residual is observed minus predicted, standardized by the sd the row was fitted with – the same quantity `residuals.Rceattle()` returns for type = "pearson". On a log-scale fleet it is measured against the likelihood's centre, $\log(\text{hat}) - \text{bias_adjust_obs} * \text{sd}^2 / 2$; on a natural-scale fleet ("MVN", "MVNORM", "Normal", "TruncatedNormal" in `Index_distribution`), whose sd is absolute, it is $(\text{obs} - \text{hat}) / \text{sd}$. A covariance fleet ("MVN", "MVNORM") takes that sd from $\text{sqrt}(\text{diag}(\text{index_cov}))$, which its likelihood actually scores, so the residual is marginal rather than whitened. A positive residual means the survey saw more than the model predicted. The y scale stays free per fleet, since residual spread differs by fleet.

Value

A ggplot object.

<code>plot_indexresidual</code>	<i>Plot survey index residuals (deprecated name)</i>
---------------------------------	--

Description

Deprecated alias for `plot_index_residual`, which is the snake_case spelling the rest of the package uses. Please use `plot_index_residual()` instead.

Usage

```
plot_indexresidual(...)
```

Arguments

... passed unchanged to [plot_index_residual](#).

plot_m_at_age	<i>Plot natural mortality by age</i>
---------------	--------------------------------------

Description

Function that plots the natural mortality at age ($M1 + M2$) as estimated from Rceattle. Returns and saves a figure with the M-at-age trajectory.

Usage

```
plot_m_at_age(
  Rceattle,
  file = NULL,
  age = 1,
  model_names = NULL,
  line_col = NULL,
  spnames = NULL,
  species = NULL,
  lwd = 3,
  lty = 1,
  right_adj = 0,
  minyr = NULL,
  width = 7,
  height = 6.5,
  incl_proj = FALSE,
  incl_mean = FALSE,
  add_ci = FALSE,
  maxyr = NULL,
  top_adj = 0.15
)
```

Arguments

Rceattle	A single fit_mod() object or a list of them (overlaid).
file	Optional file stem; the figure is written to <file>_<suffix>.png if given.
age	Age to plot M at, on the species' own age scale (so age = 1 is age 1, not the first age bin, and means nothing to a species whose minage is 2). A species that has no such age is dropped, with a warning.
model_names	Legend labels for the models.

line_col	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
spnames	Species labels, length nspp. Default: the model's own.
species	Species to include, as indices (c(1, 3)), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in spnames; a character species that matches no species name is read as labels for back-compatibility.
lwd	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
lty	Line type. A vector varies it across the levels of whatever the figure separates by line type.
right_adj	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
minyr, maxyr	First / last year to plot.
width, height	Saved figure size in inches.
incl_proj	Include the projection years, with a dashed divider at the last hindcast year.
incl_mean	Add a horizontal line at the hindcast mean of each series.
add_ci	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.
top_adj	Ignored. Base-graphics leftover; see right_adj.

Details

Colour separates the models; line type separates the sexes. A sex-combined model has one sex, so a varying lty has nothing to key on and warns.

plot_m2_at_age_prop	<i>Plot predation mortality by age and predator</i>
---------------------	---

Description

Share of the predation mortality (M2) on each prey age that is attributable to each predator species. The shares sum to 1 across predators for every prey age and year; a year with no predation on that age leaves them undefined and draws nothing.

Usage

```
plot_m2_at_age_prop(
  Rceattle,
  file = NULL,
  age = 1,
  model_names = NULL,
  line_col = NULL,
  spnames = NULL,
```

```

    species = NULL,
    lwd = 3,
    right_adj = 0,
    top_adj = 0.15,
    minyr = NULL,
    width = 7,
    height = 6.5,
    incl_proj = FALSE,
    incl_mean = FALSE,
    add_ci = FALSE,
    maxyr = NULL,
    lty = 1
  )

```

Arguments

<code>Rceattle</code>	A single <code>fit_mod()</code> object or a list of them (overlaid).
<code>file</code>	Optional file stem; the figure is written to <code><file>_<suffix>.png</code> if given.
<code>age</code>	Prey age to plot the M2 proportions at, on the prey species' own age scale rather than as an age-bin index. A prey species that has no such age is dropped, with a warning.
<code>model_names</code>	Legend labels for the models.
<code>line_col</code>	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
<code>spnames</code>	Species labels, length <code>nspp</code> . Default: the model's own.
<code>species</code>	Species to include, as indices (<code>c(1, 3)</code>), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in <code>spnames</code> ; a character species that matches no species name is read as labels for back-compatibility.
<code>lwd</code>	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
<code>right_adj</code>	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
<code>top_adj</code>	Ignored. Base-graphics leftover; see <code>right_adj</code> .
<code>minyr, maxyr</code>	First / last year to plot.
<code>width, height</code>	Saved figure size in inches.
<code>incl_proj</code>	Include the projection years, with a dashed divider at the last hindcast year.
<code>incl_mean</code>	Add a horizontal line at the hindcast mean of each series.
<code>add_ci</code>	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.
<code>lty</code>	Line type. A vector varies it across the levels of whatever the figure separates by line type.

Details

Colour separates predators; line type separates models. Panels are prey species (and sex, where the prey is sexed).

plot_maturity

Plot maturity

Description

Function that plots the maturity of each species

Usage

```
plot_maturity(
  Rceattle,
  file = NULL,
  model_names = NULL,
  line_col = NULL,
  species = NULL,
  width = 4,
  height = 5.5,
  lwd = 3
)
```

Arguments

Rceattle	Single or list of Rceattle model objects exported from Rceattle
file	Path/prefix for the saved figure. NULL (default) draws without saving.
model_names	Names of models to be used in legend
line_col, lwd	Deprecated base-graphics arguments, retained for back-compatibility and ignored; colour comes from the shared model palette and the line width is fixed.
species	Species labels for the facet strips, length nspp. Default is the model's own names. It labels panels and does not subset.
width	Figure width in inches
height	Figure height in inches

plot_mortality	<i>Plot M1 or M2 at age</i>
----------------	-----------------------------

Description

Mortality-at-age over the hindcast for one model: predation mortality (M2, the default) or residual natural mortality (M1). One component at a time, never their sum, `plot_m_at_age()` draws total M ($M1 + M2$) as a time series.

Usage

```
plot_mortality(
  Rceattle,
  file = NULL,
  incl_proj = FALSE,
  zlim = NULL,
  type = "lines",
  width = 8,
  height = 5.5,
  title = NULL,
  log = FALSE,
  minyr = NULL,
  theta = 155,
  species = NULL,
  maxage = NULL,
  title_cex = 10,
  M2 = TRUE
)
```

Arguments

Rceattle	A single <code>fit_mod()</code> object. A list of more than one is an error; call it per model.
file	Path/prefix for the saved figure. NULL (default) draws without saving.
incl_proj	Include the projection years (TRUE/FALSE)
zlim	Ignored. Base-graphics leftover: the fill range of the tile plot. Use <code>p + ggplot2::scale_fill_viridis_c = ...</code> .
type	"lines" (default) draws M-at-age with one line per year; "heatmap" (or 0) draws the same series as an age-by-year tile plot. Any other value gives the lines.
width	Plot width when saved "inches"
height	Plot height when saved "inches"
title	Additional title to add. Will also add species names if not NULL
log	TRUE/FALSE plot the series on a log scale
minyr	First year to plot

theta	Ignored. Base-graphics leftover: viewing angle of a persp surface, which is no longer drawn.
species	Species to plot. Plots all if null.
maxage	Oldest age to draw, on the species' own age scale rather than as a count of age bins. NULL draws every age.
title_cex	Ignored. Base-graphics leftover: title font size. Use <code>p + ggplot2::theme(plot.title = ggplot2::element_text(size = ...))</code> .
M2	Draw predation mortality M2 (TRUE, the default) or residual natural mortality M1 (FALSE). Neither is the sum of the two.

Value

A ggplot.

plot_profile	<i>Plot the likelihood components across a profile</i>
--------------	--

Description

Draws each negative log-likelihood component against the profiled parameter, with the total overlaid, in the style of `r4ss::SSplotProfile()`. Where the curves disagree about which value they prefer, the data sources are in conflict, which is what the total on its own cannot show.

Usage

```
plot_profile(  
  Rceattle_profile,  
  model_names = NULL,  
  weighted = TRUE,  
  relative = c("own", "scaled", "minimum", "none"),  
  minfraction = 0.01,  
  add_cutoff = FALSE,  
  cutoff = 1.92,  
  xlab = NULL,  
  ylab = NULL,  
  line_col = NULL,  
  lwd = 3,  
  lty = 1,  
  file = NULL,  
  width = 7,  
  height = 5  
)
```

Arguments

<code>Rceattle_profile</code>	A single "Rceattle_profile" from <code>profile.Rceattle()</code> , or a list of them to compare models in facets.
<code>model_names</code>	Legend labels for the models.
<code>weighted, relative, minfraction</code>	Passed to <code>profile_components()</code> , which documents them. <code>minfraction</code> defaults to 0.01 here, as in <code>r4ss::SSplotProfile()</code> .
<code>add_cutoff</code>	Draw a horizontal line at cutoff. Off by default: the cutoff is a statement about the total, and drawing it across the components invites reading it as one about them.
<code>cutoff</code>	Height of that line. Default 1.92, the 95% profile-likelihood cutoff for one parameter, $\chi^2_1(0.95)/2$.
<code>xlab</code>	X-axis label. Default names the profiled parameter and cell in the units the grid is in.
<code>ylab</code>	Y-axis label. Default follows <code>relative</code> .
<code>line_col</code>	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
<code>lwd</code>	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
<code>lty</code>	Line type. A vector varies it across the levels of whatever the figure separates by line type.
<code>file</code>	Optional file stem; the figure is written to <code><file>_<suffix>.png</code> if given.
<code>width, height</code>	Saved figure size in inches.

Details

Read the figure by where each curve bottoms out, not by how deep it is. Every series is re-zeroed at its own minimum and a point marks that minimum, so the spread of the points along the x axis *is* the disagreement: components whose points sit together support the same value, and one whose point sits far from the total's is pulling against the rest. Where one component dwarfs the others it sets the y axis and the rest flatten onto the bottom; `relative = "scaled"` puts every curve on 0 to 1 so the minima stay comparable, and `relative = "minimum"` re-zeroes every series at the total's minimum, showing what each gives up by moving away from the fitted value. `profile_components()` documents all of them, and the magnitude "scaled" discards.

The total is drawn solid and black at 1.6 times `lwd`, and kept out of the colour legend, so it stays the reference line whatever the components are styled as; `line_col`, `lwd` and `lty` therefore apply to the **components**. Series are ordered, in the legend and the palette, by how much each moves over the grid. Under `joint = "multiply"` or `"add"` a dotted vertical line marks the fitted model (a multiplier of 1, an offset of 0). Non-converged grid points leave a gap.

Value

A ggplot object. Its `$data` is the plotted frame, so the numbers behind the figure can be read back off it.

References

Taylor, I.G., et al. (2021) Beyond visualizing catch-at-age models: Lessons learned from the r4ss package about software to support stock assessments. Fisheries Research 239: 105924.

See Also

`profile_components()` for the same numbers as a data frame, `profile.Rceattle()` to run the profile, `print.Rceattle_profile()` for whether the grid brackets the minimum.

Examples

```
data(BS2017SS)
ss_run <- fit_mod(data_list = BS2017SS, inits = NULL, file = NULL,
  estimateMode = 0, random_rec = FALSE, msmMode = 0, avgnMode = 0,
  phase = FALSE, verbose = 0)

# Profile M for species 1 and see which data sources disagree about it
prof <- profile(ss_run, param = "M1", slots = list(c(1, 1, 1)),
  values = list(seq(0.2, 0.5, by = 0.05)))

plot_profile(prof, xlab = "M at age 1")

# Two model configurations side by side
plot_profile(list(prof, prof), model_names = c("Base", "Alternative"))
```

plot_ration

Plot ration

Description

Population-level consumption for ages minage+: the individual annual ration (kg/yr) multiplied by average numbers-at-age and summed over age, in million mt. This is how the model forms total consumption (`avgN_at_age * ration`, `predation.hpp`), so it is the consumption that generates the predation mortality in `plot_m2_at_age_prop()` and the biomass in `plot_b_eaten()`, plus the other-food term.

Usage

```
plot_ration(
  Rceattle,
  file = NULL,
  minage = 1,
  model_names = NULL,
  line_col = NULL,
  spnames = NULL,
  species = NULL,
  lwd = 3,
```

```

    lty = 1,
    right_adj = 0,
    minyr = NULL,
    width = 7,
    height = 6.5,
    incl_proj = FALSE,
    incl_mean = FALSE,
    add_ci = FALSE,
    maxyr = NULL,
    top_adj = 0.15
  )

```

Arguments

<code>Rceattle</code>	A single <code>fit_mod()</code> object or a list of them (overlaid).
<code>file</code>	Optional file stem; the figure is written to <code><file>_<suffix>.png</code> if given.
<code>minage</code>	Youngest age to sum consumption over, so the figure is "age minage+". An age on the species' own scale, not an age-bin index; a species with no age that old is dropped, with a warning.
<code>model_names</code>	Legend labels for the models.
<code>line_col</code>	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
<code>spnames</code>	Species labels, length <code>nspp</code> . Default: the model's own.
<code>species</code>	Species to include, as indices (<code>c(1, 3)</code>), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in <code>spnames</code> ; a character species that matches no species name is read as labels for back-compatibility.
<code>lwd</code>	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
<code>lty</code>	Line type. A vector varies it across the levels of whatever the figure separates by line type.
<code>right_adj</code>	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
<code>minyr, maxyr</code>	First / last year to plot.
<code>width, height</code>	Saved figure size in inches.
<code>incl_proj</code>	Include the projection years, with a dashed divider at the last hindcast year.
<code>incl_mean</code>	Add a horizontal line at the hindcast mean of each series.
<code>add_ci</code>	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.
<code>top_adj</code>	Ignored. Base-graphics leftover; see <code>right_adj</code> .

Details

Colour separates the models; line type separates the sexes.

plot_recruitment	<i>Plot recruitment</i>
------------------	-------------------------

Description

Plots the mean minage recruitment (millions) and 95\

Usage

```
plot_recruitment(
  Rceattle,
  file = NULL,
  model_names = NULL,
  line_col = NULL,
  species = NULL,
  spnames = NULL,
  add_ci = FALSE,
  lwd = 3,
  save = FALSE,
  right_adj = 0,
  legend.pos = "topright",
  width = 7,
  height = 6.5,
  minyr = NULL,
  maxyr = NULL,
  incl_proj = FALSE,
  mod_cex = 1,
  lty = rep(1, length(Rceattle)),
  alpha = 0.4,
  mod_avg = rep(FALSE, length(Rceattle)),
  mse = FALSE,
  OM = TRUE,
  reference = NULL,
  ylab = NULL
)
```

Arguments

Rceattle	A single <code>fit_mod()</code> object or a list of them (overlaid).
file	Optional file stem; the figure is written to <code><file>_<suffix>.png</code> if given.
model_names	Legend labels for the models.
line_col	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
species	Species to include, as indices (<code>c(1, 3)</code>), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in <code>spnames</code> ; a character species that matches no species name is read as labels for back-compatibility.

sppnames	Species labels, length nspp. Default: the model's own.
add_ci	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.
lwd	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
save	Write the plotted series to CSV alongside the figure.
right_adj	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
legend.pos	Ignored. Base-graphics leftover: legend placement. Use <code>p + ggplot2::theme(legend.position = "right")</code> .
width, height	Saved figure size in inches.
minyr, maxyr	First / last year to plot.
incl_proj	Include the projection years, with a dashed divider at the last hindcast year.
mod_cex	Ignored. Base-graphics leftover: legend text size. Use <code>p + ggplot2::theme(legend.text = ggplot2::element_text(size = ...))</code> .
lty	Line type. A vector varies it across the levels of whatever the figure separates by line type.
alpha	Transparency of confidence ribbons and shaded areas, between 0 and 1.
mod_avg	TRUE/FALSE
mse	Is an MSE object from load_mse or run_mse
OM	if mse == TRUE, use the OM (TRUE) or EM (FALSE) for plotting?
reference	A model to overlay for comparison, drawn at 1.5x lwd and labelled "Reference". It takes the next colour from the palette, or black if line_col is supplied.
ylab	Y-axis label. NULL (default) derives one from output and the model's minage.

Value

Returns and saves a figure with the recruitment trajectory.

plot_selectivity	<i>Plot selectivity</i>
------------------	-------------------------

Description

Selectivity over the hindcast, one panel per fleet. Each fleet is drawn on the dimension it was estimated on: age for an age-based fleet, length bin for a length-based one (`Selectivity_dimension`). A model mixing the two gets one figure per dimension.

Usage

```
plot_selectivity(
  Rceattle,
  file = NULL,
  model_names = NULL,
  line_col = NULL,
  width = 7,
  height = 6.5,
  species = NULL,
  lwd = 3,
  spnames = NULL,
  lty = 1,
  minyr = NULL,
  maxyr = NULL,
  alpha = 0.25,
  colour_by = c("auto", "year", "model"),
  add_ci = FALSE
)
```

Arguments

Rceattle	A single <code>fit_mod()</code> object or a list of them (overlaid).
file	Optional file stem; the figure is written to <code><file>_<suffix>.png</code> if given.
model_names	Legend labels for the models.
line_col	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
width, height	Saved figure size in inches.
species	Species to include, as indices (<code>c(1, 3)</code>), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in <code>spnames</code> ; a character species that matches no species name is read as labels for back-compatibility.
lwd	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
spnames	Species labels, length <code>nspp</code> . Default: the model's own.
lty	Line type. A vector varies it across the levels of whatever the figure separates by line type.
minyr, maxyr	First / last year to plot.
alpha	Transparency of the faintest year when the fan is drawn by transparency, i.e. under <code>colour_by = "model"</code> .
colour_by	What colour separates: "year" (a fan), "model", or "auto" (the default) for year with a single model and model with several.
add_ci	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.

Value

A ggplot, or for a model mixing age- and length-based fleets a named list of them, one per dimension.

What the colours mean

With one model, colour is the year, so time-varying selectivity reads as a fan and a time-invariant fleet collapses to a single line. With several models, colour separates the models and the year fan moves to transparency, faintest at the earliest year drawn. The transparency scale spans the years shown across all models, so the same year is the same shade everywhere and a short retrospective peel stops short of solid. `colour_by` overrides the choice either way, and `alpha` sets the faintest end.

Line type separates the sexes, and `lty` supplies its values. Panels are fleets, so `sfnames` does not label anything here, it only lets species select by name.

Confidence intervals

`add_ci = TRUE` draws $\exp(\log(\text{sel}) \pm 1.96 * \text{sd})$, so the band is positive and right-skewed. It needs a fit run with `fit_control(selectivity_se = TRUE)`.

Only estimated, age-based fleets hold one. A length-based fleet is drawn on `sel_at_length`, which is what was fitted and holds no error; a `Selectivity = "Fixed"` fleet estimates nothing; and no fleet gets a band below its first selected bin (`Bin_first_selected`, a 1-based bin ordinal, not an absolute age), where selectivity is 0 by construction. A fleet mirroring another's `Selectivity_index` has no errors of its own and borrows its lead's, but only where the two curves agree, `data_check()` only warns when a shared group differs in a shaping column such as `Sel_norm_bin`.

Every year drawn gets its own band, so a time-varying fleet is dense. Pair it with `minyr / maxyr` on a single year, or use it on a time-invariant fleet.

The interval is **relative to the normalization anchor**, so quote the anchor with it. Selectivity is identified only up to a scalar, jointly with `F` for a fishery and `q` for a survey, so `Sel_norm_bin` fixes the level rather than measuring it: the bin it pins has a standard error of exactly 0 and the band widens away from it. On a penalized fit (`random_sel = FALSE`) the band is also conditional on the smoothing, and it is marginal rather than joint. See `vignette("model-options-and-functionality")` for both, and for the `estimateMode` that determines whether the fit reports an error at all.

plot_selectivity_vs_maturity

Plot fishery selectivity and maturity

Description

Overlays terminal-year fishery selectivity-at-age against input maturity-at-age, faceted by species. Useful for debugging SPR-based reference points (the two curves drive spawning-potential ratio).

Usage

```
plot_selectivity_vs_maturity(  
  Rceattle,  
  file = NULL,  
  model_names = NULL,  
  line_col = NULL,  
  width = 7,  
  height = 6.5,  
  species = NULL,  
  lwd = 3  
)
```

Arguments

- Rceattle A single `fit_mod()` object or a list of them (the first is used).
- file Optional file stem; the figure is written to `<file>_selectivity_vs_maturity.png` if given.
- model_names, line_col, lwd Deprecated base-graphics arguments, retained for back-compatibility and ignored.
- width, height Saved figure size (inches).
- species Species names for the facet labels. Default: model species names.

Value

A ggplot object.

plot_ssb	<i>Plot spawning stock biomass (SSB)</i>
----------	--

Description

Plots the mean SSB (million mt) and 95\

Usage

```
plot_ssb(  
  Rceattle,  
  file = NULL,  
  model_names = NULL,  
  line_col = NULL,  
  species = NULL,  
  spnames = NULL,  
  add_ci = FALSE,  
  lwd = 3,  
  save = FALSE,
```

```

    right_adj = 0,
    legend.pos = "topright",
    width = 7,
    height = 6.5,
    minyr = NULL,
    maxyr = NULL,
    incl_proj = FALSE,
    mod_cex = 1,
    lty = rep(1, length(Rceattle)),
    alpha = 0.4,
    mod_avg = rep(FALSE, length(Rceattle)),
    mse = FALSE,
    OM = TRUE,
    reference = NULL,
    ylab = NULL
  )

```

Arguments

Rceattle	A single <code>fit_mod()</code> object or a list of them (overlaid).
file	Optional file stem; the figure is written to <code><file>_<suffix>.png</code> if given.
model_names	Legend labels for the models.
line_col	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
species	Species to include, as indices (<code>c(1, 3)</code>), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in <code>spnames</code> ; a character species that matches no species name is read as labels for back-compatibility.
spnames	Species labels, length <code>nspp</code> . Default: the model's own.
add_ci	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.
lwd	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
save	Write the plotted series to CSV alongside the figure.
right_adj	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
legend.pos	Ignored. Base-graphics leftover: legend placement. Use <code>p + ggplot2::theme(legend.position = "right")</code> .
width, height	Saved figure size in inches.
minyr, maxyr	First / last year to plot.
incl_proj	Include the projection years, with a dashed divider at the last hindcast year.
mod_cex	Ignored. Base-graphics leftover: legend text size. Use <code>p + ggplot2::theme(legend.text = ggplot2::element_text(size = ...))</code> .
lty	Line type. A vector varies it across the levels of whatever the figure separates by line type.

alpha	Transparency of confidence ribbons and shaded areas, between 0 and 1.
mod_avg	TRUE/FALSE
mse	Is an MSE object from load_mse or run_mse
OM	if mse == TRUE, use the OM (TRUE) or EM (FALSE) for plotting?
reference	A model to overlay for comparison, drawn at 1.5x lwd and labelled "Reference". It takes the next colour from the palette, or black if line_col is supplied.
ylab	Y-axis label. NULL (default) derives one from output and the model's minage.

Value

Returns and saves a figure with the SSB trajectory.

plot_ssb_depletion	<i>Plot SSB depletion</i>
--------------------	---------------------------

Description

Plots the mean SSB depletion and 95\ Depletion reference lines for Ptarget and Plimit are drawn in blue and red respectively.

Usage

```
plot_ssb_depletion(
  Rceattle,
  file = NULL,
  model_names = NULL,
  line_col = NULL,
  species = NULL,
  spnames = NULL,
  add_ci = FALSE,
  lwd = 3,
  save = FALSE,
  right_adj = 0,
  legend.pos = "topright",
  width = 7,
  height = 6.5,
  minyr = NULL,
  maxyr = NULL,
  incl_proj = FALSE,
  mod_cex = 1,
  lty = rep(1, length(Rceattle)),
  alpha = 0.4,
  mod_avg = rep(FALSE, length(Rceattle)),
  mse = FALSE,
  OM = TRUE,
  reference = NULL,
  ylab = NULL
)
```

Arguments

<code>Rceattle</code>	A single <code>fit_mod()</code> object or a list of them (overlaid).
<code>file</code>	Optional file stem; the figure is written to <code><file>_<suffix>.png</code> if given.
<code>model_names</code>	Legend labels for the models.
<code>line_col</code>	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
<code>species</code>	Species to include, as indices (<code>c(1, 3)</code>), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in <code>spnames</code> ; a character species that matches no species name is read as labels for back-compatibility.
<code>spnames</code>	Species labels, length <code>nspp</code> . Default: the model's own.
<code>add_ci</code>	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.
<code>lwd</code>	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
<code>save</code>	Write the plotted series to CSV alongside the figure.
<code>right_adj</code>	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
<code>legend.pos</code>	Ignored. Base-graphics leftover: legend placement. Use <code>p + ggplot2::theme(legend.position = "right")</code> .
<code>width, height</code>	Saved figure size in inches.
<code>minyr, maxyr</code>	First / last year to plot.
<code>incl_proj</code>	Include the projection years, with a dashed divider at the last hindcast year.
<code>mod_cex</code>	Ignored. Base-graphics leftover: legend text size. Use <code>p + ggplot2::theme(legend.text = ggplot2::element_text(size = ...))</code> .
<code>lty</code>	Line type. A vector varies it across the levels of whatever the figure separates by line type.
<code>alpha</code>	Transparency of confidence ribbons and shaded areas, between 0 and 1.
<code>mod_avg</code>	TRUE/FALSE
<code>mse</code>	Is an MSE object from <code>load_mse</code> or <code>run_mse</code>
<code>OM</code>	if <code>mse == TRUE</code> , use the OM (TRUE) or EM (FALSE) for plotting?
<code>reference</code>	A model to overlay for comparison, drawn at 1.5x <code>lwd</code> and labelled "Reference". It takes the next colour from the palette, or black if <code>line_col</code> is supplied.
<code>ylab</code>	Y-axis label. NULL (default) derives one from output and the model's minage.

Details

The denominator differs by configuration, so say which one a figure uses. `DynamicHCR = TRUE` gives that year's `DynamicSB0`. Otherwise it is static `SB0`, except under `HCR = 0` with `msmMode > 0`, where it is female SSB in the **last projection year** – which approximates multispecies `SB0` only once `projyr` is far enough out to equilibrate. An `msmMode = 0` fit and an `msmMode = 1`, `HCR = 0` fit drawn on one axis are therefore two different ratios.

Value

Returns and saves a figure with the SSB depletion trajectory.

See Also

[report_tables\(\)](#), which states the same multispecies denominator.

plot_stock_recruit	<i>Plot stock recruit function</i>
--------------------	------------------------------------

Description

Plots the stock-recruit relationship estimated by Rceattle: spawning stock biomass (x) against recruitment (y) as points, with the fitted stock-recruit curve (mean recruitment, Beverton-Holt, or Ricker) overlaid, faceted by species. A 95% normal data ellipse of the SSB-R cloud is added when `add_ci = TRUE`.

Usage

```
plot_stock_recruit(
  Rceattle,
  file = NULL,
  model_names = NULL,
  line_col = NULL,
  width = 7,
  height = 6.5,
  species = NULL,
  spnames = NULL,
  lwd = 3,
  lty = 1,
  incl_proj = FALSE,
  plot_env = FALSE,
  mod_cex = 1,
  add_ci = TRUE
)
```

Arguments

Rceattle	A single fit_mod() object or a list of them (overlaid).
file	Optional file stem; the figure is written to <file>_stock_recruit.png if given.
model_names	Legend labels for the models.
line_col, lwd, lty, plot_env, mod_cex	Deprecated base-graphics arguments, retained for back-compatibility and ignored.
width, height	Saved figure size (inches).

species	Species (indices) to include. Default all.
spnames	Species names for the facet labels.
incl_proj	Currently unused (kept for back-compatibility).
add_ci	Add a 95% normal data ellipse of the SSB-R points.

Value

A ggplot object.

plot_timeseries	<i>Plot time-series</i>
-----------------	-------------------------

Description

Function that plots the time-series (SSB/B/R/Depletion) 95% CI trends as estimated from Rceattle

Usage

```
plot_timeseries(
  Rceattle,
  output = "biomass",
  ylab = NULL,
  file = NULL,
  model_names = NULL,
  line_col = NULL,
  species = NULL,
  spnames = NULL,
  add_ci = FALSE,
  lwd = 3,
  save = FALSE,
  legend.pos = "topright",
  right_adj = 0,
  width = 7,
  height = 6.5,
  minyr = NULL,
  maxyr = NULL,
  incl_proj = FALSE,
  mod_cex = 1,
  lty = rep(1, length(Rceattle)),
  alpha = 0.4,
  mse = FALSE,
  OM = TRUE,
  reference = NULL,
  zero_y = FALSE,
  ref_lines = NULL,
  suffix = NULL,
  mod_avg = rep(FALSE, length(Rceattle))
)
```

Arguments

<code>Rceattle</code>	A single <code>fit_mod()</code> object or a list of them (overlaid).
<code>output</code>	derived quantity of interest: recruitment, biomass, ssb, depletion, or ssb_depletion. Uses same name as ".cpp" file.
<code>ylab</code>	Y-axis label. NULL (default) derives one from output and the model's minage.
<code>file</code>	Optional file stem; the figure is written to <file>_<suffix>.png if given.
<code>model_names</code>	Legend labels for the models.
<code>line_col</code>	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
<code>species</code>	Species to include, as indices (<code>c(1, 3)</code>), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in <code>spnames</code> ; a character species that matches no species name is read as labels for back-compatibility.
<code>spnames</code>	Species labels, length <code>nspp</code> . Default: the model's own.
<code>add_ci</code>	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.
<code>lwd</code>	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
<code>save</code>	Write the plotted series to CSV alongside the figure.
<code>legend.pos</code>	Ignored. Base-graphics leftover: legend placement. Use <code>p + ggplot2::theme(legend.position = "right")</code> .
<code>right_adj</code>	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
<code>width, height</code>	Saved figure size in inches.
<code>minyr, maxyr</code>	First / last year to plot.
<code>incl_proj</code>	Include the projection years, with a dashed divider at the last hindcast year.
<code>mod_cex</code>	Ignored. Base-graphics leftover: legend text size. Use <code>p + ggplot2::theme(legend.text = ggplot2::element_text(size = ...))</code> .
<code>lty</code>	Line type. A vector varies it across the levels of whatever the figure separates by line type.
<code>alpha</code>	Transparency of confidence ribbons and shaded areas, between 0 and 1.
<code>mse</code>	Is an MSE object from <code>load_mse</code> or <code>run_mse</code>
<code>OM</code>	if <code>mse == TRUE</code> , use the OM (TRUE) or EM (FALSE) for plotting?
<code>reference</code>	A model to overlay for comparison, drawn at 1.5x <code>lwd</code> and labelled "Reference". It takes the next colour from the palette, or black if <code>line_col</code> is supplied.
<code>zero_y</code>	Anchor the y-axis at zero. TRUE for the absolute series (biomass, SSB, recruitment), where a truncated axis exaggerates change; FALSE for the depletions, which are already on a relative scale.
<code>ref_lines</code>	Internal. <code>function(models, sp_sel)</code> returning per-species reference-point layers, added before the figure is saved. Supplied by <code>plot_f()</code> and <code>plot_ssb_depletion()</code> through <code>.ts_wrapper()</code> ; <code>sp_sel</code> is the species resolution the panels were built from.

suffix	Internal. Overrides the <file>_<suffix>.png stem, which otherwise names the series.
mod_avg	TRUE/FALSE

Value

Returns and saves a figure with the population trajectory.

Units

The model holds numbers-at-age in **thousands** and weight-at-age in **kg**, so every biomass series (biomass, ssb, exploitable_biomass) comes out of the model in **mt** and recruitment comes out in **thousands of fish**. For display these are divided by 1e6 (million mt) and 1e3 (millions of recruits) respectively; depletion is a ratio and is not rescaled. Supply the model's inputs on that convention, catch and index in mt, weight-at-age in kg, or the axis labels will not describe what is plotted.

Confidence intervals

add_ci = TRUE draws a 95% interval. Every strictly positive series takes it on the log scale as $\exp(\log(x) \pm qnorm(0.975) * sd_log)$. These quantities are built multiplicatively, so $\log(x)$ is close to linear in the estimated parameters where x is exponential in them: the interval is both a better linearization and right-skewed, and it cannot cross zero the way a symmetric natural-scale interval does for weak year classes and depleted stocks.

sd_log comes from the model's own $\log_biomass / \log_ssb / \log_R$ where those are reported, and is otherwise recovered as $sd(x) / x$, the delta method's own identity, which matches the reported values to machine precision. That covers exploitable_biomass and the two depletions, which cannot be reported on the log scale (exploitable_biomass is identically 0 without projection F), and models fit before the log_* series existed. A non-positive or unreported value keeps the symmetric interval.

print.Rceattle	<i>Print method for fitted Rceattle models</i>
----------------	--

Description

Provides a compact summary so that auto-printing inside R Markdown / knitr / RStudio does not recurse into the (very deep) data and TMB objects stored on the fit. Only structural metadata, convergence status, headline derived quantities, and the package / TMB-DLL versions used to produce the fit are printed.

Usage

```
## S3 method for class 'Rceattle'
print(x, ...)
```

Arguments

x An object of class "Rceattle" returned by `fit_mod()`.
 ... Currently unused.

Details

For operational use, the package version line is meant to make it obvious which version of Rceattle produced an archived fit so that results can be reproduced even if main has moved on. Tag a release (`devtools::install_github("afsc-assessments/Rceattle@X.Y.Z")`), bare version, no leading v) and the same version string will reappear here on a fresh run.

Value

The input x, invisibly.

`print.Rceattle_data` *Print method for an Rceattle data list*

Description

Shows the model specification as an indented tree, dimensions, fleets and their selectivity / catchability forms, configured processes, active linkages, and any attached `model_config()`, rather than dumping the full data list.

Usage

```
## S3 method for class 'Rceattle_data'
print(x, config = TRUE, ...)

## S3 method for class 'Rceattle_data'
summary(object, config = FALSE, ...)
```

Arguments

x An "Rceattle_data" object from `build_data()`.
 config Show the attached `model_config()` block. Defaults to TRUE for `print()` and FALSE for `summary()`. Has no effect on an object that holds no configuration.
 ... Currently unused.
 object An "Rceattle_data" object.

Details

`print()` includes the configuration block; `summary()` omits it and reports the data structure alone. Either can be overridden with `config`.

Value

x, invisibly.

```
print.Rceattle_jitter
```

Print method for a jitter analysis

Description

Reports what the run was for: how many random starts reached the best optimum found. The objective values alone cannot say that, non-converged starts are dropped before the result is returned, so the count of returned fits is not the count attempted.

Usage

```
## S3 method for class 'Rceattle_jitter'
print(x, tol = 0.01, ...)
```

Arguments

x	A "Rceattle_jitter" object from jitter() .
tol	Objective units within which a start counts as reaching the same optimum. Default 0.01, which is far below any difference that would change management advice and far above optimizer noise.
...	Currently unused.

Value

x, invisibly.

```
print.Rceattle_mse_summary
```

Print method for an MSE summary

Description

Says what the summary holds. The object is deliberately ragged, per-species, per-fleet and whole-system metrics have different shapes and cannot share one frame, so it reports the blocks and their dimensions rather than printing them end to end.

Usage

```
## S3 method for class 'Rceattle_mse_summary'
print(x, ...)
```

Arguments

x	An "Rceattle_mse_summary" object from mse_summary() .
...	Currently unused.

Value

x, invisibly.

```
print.Rceattle_profile
```

Print method for a likelihood profile

Description

Reports whether the grid actually brackets the minimum. A profile whose lowest point is its first or last grid value has not found the optimum, it has run out of grid, and the curve drawn from it understates how far the parameter can move. That is the failure the numbers alone hide, since a partial profile plots as a perfectly ordinary line.

Usage

```
## S3 method for class 'Rceattle_profile'
print(x, cutoff = 1.92, ...)
```

Arguments

x	A "Rceattle_profile" object from <code>profile.Rceattle()</code> .
cutoff	Objective units above the minimum that bound the reported interval. Default 1.92, the 95% profile-likelihood cutoff for one parameter.
...	Currently unused.

Details

For a one-dimensional profile the interval reported is the usual profile-likelihood one: the grid values within cutoff of the minimum, where the default 1.92 is $\chi_1^2(0.95)/2$. It is read off the grid, so it is no finer than the spacing of values, and it is reported as open on either side the grid does not close. It is also referenced to the best GRID point rather than to the unconstrained MLE, which the object does not hold: the grid minimum sits at or above the MLE, so the interval errs wide. And it is reported as a range, so a profile with a second basin, or a failed point inside the range, is called out as not contiguous rather than left to read as one interval. No interval is given for a cross-profile over two or more cells, the cutoff would be $\chi_k^2(0.95)/2$ and the region is not an interval.

Under `random_rec = TRUE` the objective is the Laplace-approximated marginal likelihood, so this is a profile of that, with the usual caveat that the approximation is what is being profiled.

Value

x, invisibly.

```
print.rceattle_report
```

Print method for a set of assessment reporting tables

Description

Print method for a set of assessment reporting tables

Usage

```
## S3 method for class 'rceattle_report'
print(x, ...)
```

Arguments

x	An "rceattle_report" from report_tables() .
...	Currently unused.

Value

x, invisibly.

```
print.Rceattle_retro
```

Print method for a retrospective analysis

Description

Reports Mohn's rho against a reference band rather than as a bare number. The default band is +/- 0.2 on SSB, the rule of thumb vignette("model-diagnostics") states; Hurtado-Ferro et al. (2015) give the asymmetric, life-history-dependent alternatives (-0.15 to 0.20 for long-lived, -0.22 to 0.30 for short-lived), which is why the band is an argument rather than a constant.

Only the terminal-year peel (Forecast year 0) is judged. The forecast-skill rows are reported for information: a rho computed over a forecast horizon is not the quantity the +/- 0.2 rule was calibrated on.

Usage

```
## S3 method for class 'Rceattle_retro'
print(x, band = 0.2, ...)
```

Arguments

x	A "Rceattle_retro" object from retrospective() .
band	Symmetric reference band for Mohn's rho. Default 0.2.
...	Currently unused.

Value

x, invisibly.

References

Hurtado-Ferro, F., et al. 2015. Looking in the rear-view mirror: bias and retrospective patterns in integrated, age-structured stock assessment models. ICES J. Mar. Sci. 72:99-110.

```
print.Rceattle_selftest
```

Print method for a self-test

Description

Reports what a self-test is run to find out: how many simulations the estimator brought back to an optimum, and under what the replicates were generated. The returned list cannot say the first on its own: non-converged runs are dropped before it is returned, so the number of fits returned is not the number attempted.

Usage

```
## S3 method for class 'Rceattle_selftest'
print(x, rate = 0.9, ...)
```

Arguments

x	A "Rceattle_selftest" object from self_test() .
rate	Fraction of simulations that must converge before the run counts as clean. Default 0.9, matching jitter() 's.
...	Currently unused.

Details

The status is the convergence RATE, since that is the quantity the run produces: FAIL if nothing converged, WARN below rate, NOTE if any replicate was dropped, OK otherwise. The table beneath tallies each returned fit's own \$convergence\$status, which is a separate question, a replicate can reach a zero gradient and still hold a NOTE or WARN, and those are not folded into the header for that reason.

The last line says what generated the replicates, because it decides what the spread means. With processes held fixed (the default) `sim_mod()` redraws the observations alone, so the spread reflects observation error only and is a lower bound on estimation uncertainty, do not read it against the model's own uncertainty bands, which include process error. See **Interpreting the spread** in [self_test\(\)](#).

Value

x, invisibly.

```
print.summary.Rceattle
```

Print method for an Rceattle model summary

Description

Print method for an Rceattle model summary

Usage

```
## S3 method for class 'summary.Rceattle'
print(x, n = 10, ...)
```

Arguments

x	A "summary.Rceattle" object from summary.Rceattle() .
n	Number of parameters to show, in <code>coef()</code> order. Default 10; use <code>Inf</code> for all, or take <code>x\$coefficients</code> .
...	Currently unused.

Value

x, invisibly.

```
prior_beta
```

Beta prior on a linkage coefficient

Description

Standard `Beta(shape1, shape2)` on (0, 1), matching [stats::dbeta\(\)](#). For priors on stock-recruit steepness on the standard (0.2, 1) interval, transform inside the model and use [prior_beta\(\)](#) on the rescaled quantity.

Usage

```
prior_beta(shape1, shape2)
```

Arguments

shape1, shape2 positive shape parameters.

Value

An `Rceattle_prior` of family "beta".

prior_gamma	<i>Gamma prior on a linkage coefficient</i>
-------------	---

Description

Shape-rate parameterization, matching `stats::dgamma()`.

Usage

```
prior_gamma(shape, rate)
```

Arguments

shape	positive shape parameter.
rate	positive rate parameter.

Value

An Rceattle_prior of family "gamma".

prior_lognormal	<i>Lognormal prior on a linkage coefficient</i>
-----------------	---

Description

A normal density on the log of the coefficient. Under `fit_control(bias_adjust_proc = TRUE)`, the default, it is centred at $\text{meanlog} - \text{sdlog}^2/2$, so $\exp(\text{meanlog})$ is the prior mean; with `FALSE` it matches `stats::dlnorm()` and $\exp(\text{meanlog})$ is the prior median.

Usage

```
prior_lognormal(meanlog, sdlog)
```

Arguments

meanlog	log of the prior mean (of the median when <code>bias_adjust_proc = FALSE</code>).
sdlog	prior standard deviation of the log (must be positive).

Value

An Rceattle_prior of family "lognormal".

prior_normal	<i>Normal prior on a linkage coefficient</i>
--------------	--

Description

Normal prior on a linkage coefficient

Usage

```
prior_normal(mean, sd)
```

Arguments

mean	prior mean.
sd	prior standard deviation (must be positive).

Value

An Rceattle_prior of family "normal".

process_residuals	<i>Process residuals for an Rceattle model's random-effect processes</i>
-------------------	--

Description

One-sample process residuals for the model's random-effect (or penalized) process deviations, in the style of SAM's `procres()` (Nielsen and Berg 2014). They validate the *process* model, whether the deviations behave like their assumed iid normal process, a complement to the observation-based [osa_residuals\(\)](#).

Each set of deviations holds a Gaussian process prior in the model. The posterior *mode* of the deviations is shrunk toward that prior, so, following SAM, a single draw is taken from the joint posterior of the deviations (from the joint precision when they are random effects, or the fixed-effect covariance when they are penalized fixed effects) and standardized by the process standard deviation. Under a correctly specified process these are approximately iid $N(0, 1)$. Because a random draw is used the residuals are stochastic; set seed for reproducibility.

Supported processes and their deviations:

"recruitment" `rec_dev`, prior $N(-\text{bias_adjust_proc} * R_sd^2/2, R_sd^2)$ per species.

"initial" `init_dev`, prior $N(-\text{bias_adjust_proc} * R_sd^2/2, R_sd^2)$ per species.

"catchability" `index_q_dev`, prior $N(0, q_dev_sd^2)$ per index.

"all" returns every supported process present in the fit. Selectivity and natural-mortality deviations (which use random-walk / 2D-AR1 priors) are not yet supported. Recruitment and initial-abundance residuals are exact (their priors are iid). Catchability residuals are exact only for the iid deviate prior (`Time_varying_q = 1` or `2`); for a random-walk or AR1 catchability prior the marginal-SD standardization ignores the prior correlation, so those residuals are approximate and a warning is emitted.

Usage

```
process_residuals(  
  object = NULL,  
  process = c("recruitment", "initial", "catchability", "all"),  
  seed = 123,  
  fit = NULL  
)
```

Arguments

object	A fitted Rceattle model. The targeted deviations must be estimated, as random effects (e.g. random_rec = TRUE) or as penalized fixed effects, with a usable covariance.
process	One of "recruitment", "initial", "catchability", or "all".
seed	Seed for the posterior draw. Default 123.
fit	deprecated name for object, still accepted so existing scripts keep working. Supplying both is an error.

Value

A data frame of class rceattle_osa (so it can be passed to [osa_diagnostics\(\)](#) and [plot.rceattle_osa\(\)](#)) with one row per deviation: columns source (the process), fleet, species, year, age_length_bin, and residual.

References

Nielsen, A., and Berg, C.W. 2014. Estimation of time-varying selectivity in stock assessments using state-space models. Fish. Res. 158:96-101.

See Also

[osa_residuals\(\)](#), [osa_diagnostics\(\)](#)

profile.Rceattle	<i>Likelihood profile across one or more parameter cells</i>
------------------	--

Description

Re-fits an Rceattle model while holding selected cells of a parameter fixed. Profiles a single cell (R_log_sd[species = 1]) or crosses several, such as log_M1[1, 1, 1] against log_M1[1, 2, 1] for residual M in males versus females. For each grid point the targeted cells are fixed in the TMB map and everything else is re-estimated.

Usage

```
## S3 method for class 'Rceattle'
profile(
  fitted = NULL,
  param = NULL,
  slots = NULL,
  values = NULL,
  transform = "log",
  cores = NULL,
  getsd = NULL,
  joint = c("none", "multiply", "add", "value"),
  ...
)
```

Arguments

fitted	an Rceattle model fit using fit_mod
param	Name of the parameter to profile: a raw slot in Rceattle\$estimated_params, or one of the natural-scale aliases "sigmaR", "R_sd", "M1", "R0", "alpha", "beta", "q". See Details.
slots	A list of integer index vectors, one per cell to fix, each as long as the resolved parameter has dimensions: 1 for R_log_sd, 2 for rec_pars, 3 for log_M1. The "R0"/"alpha"/"beta" aliases take the species index alone. So list(c(1, 1), c(1, 2, 1)) crosses the two sex cells of species 1. If omitted it defaults to a single species-1 cell shaped to the parameter and warns, which requires length(values) == 1L.
values	A list of numeric vectors, one per entry of slots. The grid is expand.grid(values), so k slots give a k -D cross-profile.
transform	How user values map onto the internal scale before substitution: "log" (default), "identity", or a unary function such as qlogis, applied element-wise.
cores	Cores for the parallel fits. NULL (default) picks parallel::detectCores() - 6, capped at 2 under R CMD check; 1 forces sequential.
getsd	Whether each grid fit runs TMB::sdreport. The profile reads only nll, so FALSE is faster and changes nothing. NULL (default) inherits the input model's setting (TRUE only if it holds an sdrep).
joint	How the grid moves the cells in slots: "none" (default), "multiply", "add" or "value". See Details.
...	Unused; present for consistency with the stats::profile generic.

Details

Natural-scale aliases. An alias implies transform = "log" and, for rec_pars, fills in the column so slots needs only the species index. transform is ignored with a warning when an alias is used. A raw slot instead has slots index the full array and transform set the scale.

"sigmaR", "R_sd" R_log_sd

"M1" log_M1
 "R0", "alpha", "beta" rec_pars columns 1, 2 and 3
 "q" index_log_q, base catchability, one entry per fleet; slots takes a fleet index or name. Fleets sharing a Catchability_index share one q, so naming any of them profiles the group: the rest are added with a message and joint becomes "value".

Joint modes. A joint mode profiles a vector-valued parameter such as an empirical age-based M, where "none" over ten ages and 13 values would be 13^{10} fits.

"none" (default) each slot gets its own grid and they cross.

"multiply" one grid multiplying every cell, so the schedule keeps its shape; 1 is the fitted model.

"add" one grid added to every cell, so the schedule keeps the differences between cells; 0 is the fitted model.

"value" one grid; every cell takes that value.

A joint mode needs values to be one vector, and "multiply"/"add" work in natural units, so transform must be "log" or "identity". A value that would take a log-scale parameter to zero or below is an error naming the cell, not a failed fit.

Value

A list of class "Rceattle_profile" with elements:

Rceattle_list the fitted models, one per grid row, NULL where a fit did not converge so positions stay aligned with grid.

grid grid values on the user scale, one column per profiled cell, named slot_1, slot_2, ...

nll joint negative log-likelihoods, NA where the fit did not converge.

param the profiled parameter name.

slots the slots list, echoed for plotting.

alias the natural-scale name asked for, or NA. Profiling "M1" returns param = "log_M1" because the model estimates M on the log scale, while grid holds the M values supplied, so a figure can label the axis in the units profiled.

joint the mode used, so a figure can say whether grid holds a multiplier, an offset or a value.

Printing it reports whether the grid brackets the minimum; see `print.Rceattle_profile`.

Examples

```
data(BS2017SS)
ss_run <- fit_mod(data_list = BS2017SS, inits = NULL, file = NULL,
  estimateMode = 0, random_rec = FALSE, msmMode = 0, avgnMode = 0,
  phase = FALSE, verbose = 0)

# 1-D profile of sigmaR for species 1, natural scale via the alias
p1 <- profile(ss_run, param = "sigmaR", slots = list(1),
  values = list(seq(0.1, 1.5, by = 0.1)))

# 2-D cross-profile of M1 across species 1 and 2 (sex 1, age 1)
```

```

p2 <- profile(ss_run, param = "M1",
  slots = list(c(1, 1, 1), c(2, 1, 1)),
  values = list(seq(0.1, 0.4, length.out = 3),
    seq(0.1, 0.4, length.out = 3)))

# An empirical age-based M moved as a whole: one slot per age bin on ONE grid
# of multipliers, so the schedule keeps its shape and 1 is the fitted model.
sp <- 1
p3 <- profile(ss_run, param = "M1",
  slots = lapply(seq_len(ss_run$data_list$nages[sp]),
    function(a) c(sp, 1, a)),
  values = list(seq(0.6, 1.4, by = 0.05)),
  joint = "multiply")

# Profiling catchability by fleet name, and reading a profile for where the
# data disagree, are in vignette("model-diagnostics").

```

profile_components	<i>Likelihood components along a profile</i>
--------------------	--

Description

Pulls the per-fleet and per-species negative log-likelihood components out of every fit in a `profile()` and returns them as one long data frame, one row per grid point per component.

The total is the least informative curve on a profile: a well-behaved quadratic total can hide two data sources pulling the parameter in opposite directions, and their conflict shows only once the components are drawn separately. This is the extractor behind `plot_profile()`; use it to tabulate, or to draw the figure yourself.

Usage

```

profile_components(
  object,
  weighted = TRUE,
  relative = c("own", "scaled", "minimum", "none"),
  minfraction = 0,
  include_total = TRUE
)

```

Arguments

object	An "Rceattle_profile" object from <code>profile.Rceattle()</code> .
weighted	Report the weighted components the optimizer minimized (TRUE, the default) or the unweighted ones (FALSE). See Details.
relative	How to place each series on the y axis: "own" (default), "scaled", "minimum" or "none". See Details.

minfraction	Drop components whose change over the grid is less than this fraction of the total's change, as in <code>r4ss::SSplotProfile()</code> . Default 0 keeps everything; <code>plot_profile()</code> uses 0.01, so a total moving 60 units cuts at 0.6. "Total" is never dropped.
include_total	Include the "Total" series. Default TRUE; FALSE also disables minfraction, which is defined against the total.

Details

Which cells are reported. `jnll_comp`'s columns mean different things on different rows – fleets on the data, selectivity and catchability rows, species on the priors, penalties and predation rows – so each cell is labelled from the axis its row uses, giving e.g. "Shelikof acoustic: Index data". The unit is dropped from the label when the model has only one fleet (or species) to distinguish. Cells that are zero at every grid point are dropped as components the model does not fit.

What Total is. The "Total" series is `object$jnll`, the objective each grid fit minimized. Under `random_rec = TRUE` that is the Laplace-approximated marginal likelihood while the components are the inner joint negative log-likelihood, so they will not sum to it, and the function says so when they differ. Compare the shapes, not the sums.

Non-converged grid points keep their row with a value of NA, so a failed fit leaves a gap in the curve rather than a segment drawn across it.

Weighted or unweighted components (weighted). Conflict is normally read off the weighted components, since those are what moved the fit. `unweighted_jnll_comp` scores the composition, CAAL and stomach-content rows without their data weights (`Comp_weights`, `CAAL_weights`, `Diet_comp_weights`); every other row equals `jnll_comp`, so `weighted = FALSE` changes only those three components. A Dirichlet-multinomial row is scored at $\theta = 1$, an effective sample size of about $(1 + N)/2$ rather than N , so it is not comparable with a multinomial row.

Placing each series on the y axis (relative).

"own" (default) subtract each series' own minimum, so every curve starts at zero and its minimum marks the value that component prefers. Objective units, so depth still says how strongly.

"scaled" as "own", then divide by each series' own change over the grid, so every curve runs 0 to 1. Use it where one component dwarfs the rest and flattens them onto the axis. **It discards magnitude** – a component moving 0.02 draws like one moving 40 – so raise minfraction with it. A series that does not move stays at zero.

"minimum" subtract each series' value at the grid point where the total is lowest, showing each component's change away from the fitted optimum.

"none" the raw negative log-likelihoods.

minfraction and the series ordering are always computed on the raw change over the grid, before any of this, so "scaled" cannot quietly disable the filter by making every span 1.

Value

A data frame with one row per grid point per retained component: the profile's grid columns (`slot_1`, ...), then `fit` (grid row index), `component` (the `jnll_comp` row), `unit` (fleet or species name, NA for model-wide rows), `axis` ("fleet", "species" or "model"), `series` (the plotting label) and `value` (the re-zeroed negative log-likelihood). Series are ordered by decreasing change over the grid, "Total" first. The profile's `param`, `alias` and the `relative` used are attributes.

See Also

[plot_profile\(\)](#) to draw it, [profile.Rceattle\(\)](#) to produce the profile.

Examples

```
data(BS2017SS)
ss_run <- fit_mod(data_list = BS2017SS, inits = NULL, file = NULL,
  estimateMode = 0, random_rec = FALSE, msmMode = 0, avgnMode = 0,
  phase = FALSE, verbose = 0)

prof <- profile(ss_run, param = "M1", slots = list(c(1, 1, 1)),
  values = list(seq(0.2, 0.5, by = 0.05)))

comps <- profile_components(prof)
head(comps)
```

quantity_dictionary *Look up what a CEATTLE reported quantity is*

Description

`fit$quantities` uses the model's own abbreviated names. This returns the table mapping each one to what it means, the units it is in, how it is shaped, whether it holds a standard error, and what the same quantity is called in the NOAA standardized assessment output.

Usage

```
quantity_dictionary(quantity = NULL, process = NULL)
```

Arguments

quantity	Report names as they appear in <code>names(fit\$quantities)</code> , default NULL for every documented quantity.
process	Character vector restricting the table to one or more parts of the model: "population", "recruitment", "mortality", "fishing", "reference_points", "catchability", "selectivity", "growth", "composition", "predation", "likelihood", "linkage" or "internal".

Details

Units follow the model throughout: numbers-at-age are **thousands of fish** and weight-at-age is **kg**, so their product is **mt**. A quantity whose units read "mt or thousands of fish" follows its fleet's `Observation_units` column.

`se = TRUE` means the TMB template ADREPORTs the quantity, so `fit$sdrep` holds a standard error for it and [as.data.frame.Rceattle\(\)](#) can fill `se`, `lwr` and `upr`. `se = FALSE` means no standard error exists anywhere on the fit for that quantity. Nothing has a standard error when the fit was produced with `fit_control(getsd = FALSE)`, which leaves `sdrep` NULL.

standard_label gives the name the quantity takes in the NOAA standardized assessment output that stockplotr and asar consume, so a table can be relabelled for those tools; NA marks a quantity that standard has no equivalent for. Rceattle's own names are kept as the primary vocabulary because several quantities (predation, multispecies) have no standard name.

Every per-recruit reference point (SPR0, SPRlimit, SPRtarget, SPRFinit, NbyageSPR) is computed only under `msmMode = 0` and is exactly **zero on a multispecies fit**: `M` there holds predation mortality, which scales with predator abundance, so spawning output per recruit is not a property of the prey stock alone.

Value

A data frame with one row per quantity and columns quantity, process, meaning, units, dims, se and standard_label.

See Also

`parameter_dictionary()` for the estimated parameters, `as.data.frame.Rceattle()` for these quantities in tidy form, and `fit_mod()` for the fitted object they come from.

Examples

```
# The whole table
head(quantity_dictionary())

# What is ssb_depletion, and what units is it in?
quantity_dictionary("ssb_depletion")

# Everything that holds a standard error
dict <- quantity_dictionary()
dict[dict$se, c("quantity", "meaning")]

# Every reference point
quantity_dictionary(process = "reference_points")

# Search the meanings by keyword
dict[grep("depletion", dict$meaning, ignore.case = TRUE), ]
```

rceattle-plot-args *Shared plotting arguments*

Description

A common vocabulary, documented here once and inherited by the plotters that take it: `plot_timeseries()` and its wrappers (`plot_biomass()`, `plot_ssb()`, `plot_recruitment()`), the depletions, `plot_exploitable_biomass()`, `plot_f()`, the predation plotters (`plot_b_eaten()`, `plot_b_eaten_prop()`, `plot_m_at_age()`, `plot_m2_at_age_prop()`, `plot_ration()`), and `plot_selectivity()`. Each argument means the same thing wherever it appears, but not every plotter takes every one, `incl_mean` is on the predation plotters, `add_ci` only where the quantity holds standard errors, and `alpha` only where the figure has a ribbon or a fan. The remaining `plot_*`() functions still take their own arguments; see each one's help.

Arguments

<code>rceattle</code>	A single <code>fit_mod()</code> object or a list of them (overlaid).
<code>file</code>	Optional file stem; the figure is written to <code><file>_<suffix>.png</code> if given.
<code>model_names</code>	Legend labels for the models.
<code>species</code>	Species to include, as indices (<code>c(1, 3)</code>), names, a logical mask, or "all". Default NULL plots every species. Species labels belong in <code>spnames</code> ; a character species that matches no species name is read as labels for back-compatibility.
<code>spnames</code>	Species labels, length <code>nspp</code> . Default: the model's own.
<code>line_col</code>	Line colours; names, hex, or base-graphics integers. NULL uses the colorblind-safe Okabe-Ito palette. Applied to whichever variable the figure separates by colour, in legend order. Too few colours are recycled, with a warning.
<code>lwd</code>	Line width on the base-graphics scale: the default 3 renders as a standard-weight ggplot line. A vector varies it across series.
<code>lty</code>	Line type. A vector varies it across the levels of whatever the figure separates by line type.
<code>alpha</code>	Transparency of confidence ribbons and shaded areas, between 0 and 1.
<code>add_ci</code>	Add a 95% confidence interval. Only available where the plotted quantity holds standard errors; warns and draws none otherwise.
<code>minyr, maxyr</code>	First / last year to plot.
<code>incl_proj</code>	Include the projection years, with a dashed divider at the last hindcast year.
<code>incl_mean</code>	Add a horizontal line at the hindcast mean of each series.
<code>width, height</code>	Saved figure size in inches.
<code>right_adj</code>	Ignored. Base-graphics leftover: the figure widened its right margin to fit the legend. Set margins on the returned ggplot instead.
<code>top_adj</code>	Ignored. Base-graphics leftover; see <code>right_adj</code> .
<code>mod_cex</code>	Ignored. Base-graphics leftover: legend text size. Use <code>p + ggplot2::theme(legend.text = ggplot2::element_text(size = ...))</code> .
<code>legend.pos</code>	Ignored. Base-graphics leftover: legend placement. Use <code>p + ggplot2::theme(legend.position = "right")</code> .
<code>single.plots</code>	Ignored. Base-graphics leftover: one device per panel. The ggplot facets instead.
<code>theta</code>	Ignored. Base-graphics leftover: viewing angle of a persp surface, which is no longer drawn.
<code>ymax</code>	Ignored. Base-graphics leftover: y-axis maximum. Use <code>p + ggplot2::coord_cartesian(ylim = c(0, ymax))</code> .
<code>cex</code>	Ignored. Base-graphics leftover: point expansion.

Value

A ggplot object.

How line_col and lty are applied

They supply values for whichever **discrete variable the plot already encodes with that aesthetic**, matched in level order, which is not always the model. Each function's help says what its own figure separates.

Where colour encodes a continuous variable, the year fan in `plot_selectivity()`, `line_col` supplies the ramp anchors instead: one colour draws the fan in that colour, several interpolate between them.

rceattle-refit-args	<i>Shared refit-diagnostic arguments</i>
---------------------	--

Description

A common vocabulary, documented here once and inherited by the diagnostics that refit the model: `retrospective()`, `jitter()` and `self_test()`. Each argument means the same thing wherever it appears. `phase`, `getsd` and `timeout` are deliberately **not** here, their defaults and their consequences for the diagnostic differ, so each function documents its own. `profile()` refits too but takes only cores and a bare `getsd`.

Arguments

object	an Rceattle model fit using <code>fit_mod()</code> .
Rceattle	deprecated name for object, still accepted so existing scripts keep working. Supplying both is an error.
cores	Number of cores for the parallel refits. Default NULL picks <code>parallel::detectCores()</code> – 6, capped at 2 under R CMD check (which sets <code>_R_CHECK_LIMIT_CORES_</code>). Set to 1 to force sequential execution.
fit_control	optional <code>fit_control()</code> bundle for the refits. Only <code>phase</code> and <code>getsd</code> are read; see What <code>fit_control()</code> reaches .

What `fit_control()` reaches

These diagnostics refit through an internal helper that forwards `phase` and `getsd` and nothing else. Every other field either comes off the fitted model's own `data_list` (`loopnum`, `comp_offset`, the bias-adjustment flags) or takes `fit_control()`'s default. Setting one of those is an error rather than a silent no-op, so a field you set is a field the refit used.

Only the fields you set are applied, named in the `fit_control()` call, or assigned to afterwards (`ctl$getsd <- TRUE`). One you never touch keeps the diagnostic's own default, which is not always `fit_control()`'s, `retrospective()` phases its peels where `fit_control()` does not, so `fit_control(getsd = FALSE)` asks about standard errors and changes nothing else. Setting a field to the value that is already `fit_control()`'s default still counts as setting it.

read_data	<i>Read a CEATTLE excel data file</i>
-----------	---------------------------------------

Description

Read a CEATTLE excel data file

Usage

```
read_data(file = "Rceattle_data.xlsx")
```

Arguments

file Filename to read. Must end in '.xlsx'

See Also

[build_data\(\)](#) to assemble or edit a data list in R (and to read a workbook then override blocks in one call, [build_data\(file = ...\)](#)), [data_requirements\(\)](#) to see which inputs a configuration needs.

Examples

```
library(Rceattle)
data(BS2017SS)
out_file <- file.path(tempdir(), "BS2017SS.xlsx")
write_data(data_list = BS2017SS, file = out_file)
data_list <- read_data(file = out_file)
# Or read + override blocks in one call:
data_list <- build_data(file = out_file, projyr = 2060)
file.remove(out_file)
```

rearrange_data	<i>Rearrange a data_list for TMB</i>
----------------	--------------------------------------

Description

Function to rearrange a data_list object to be read into TMB

`rearrange_dat()` is a deprecated alias for `rearrange_data()`, kept so older scripts keep running. It is scheduled for removal in 6.0.0; use `rearrange_data()`.

Usage

```
rearrange_data(data_list, build_osa = FALSE)

rearrange_dat(data_list)
```

Arguments

data_list	an Rceattle data_list
build_osa	Logical. Passed to <code>build_osa_data()</code> ; when TRUE the full one-step-ahead (OSA) observation data is assembled so <code>osa_residuals()</code> can be computed. Default FALSE (the fast path used by simulation testing). The composition proportion offset is read from <code>data_list\$comp_offset</code> (default 1e-5).
remove_F	<i>Rerun with $F = 0$.</i>

Description

Refits the model with fishing mortality set to 0 from `styr` on, keeping every other parameter. The projection after `endyr` is always unfished. `run_mse()` uses it for the no-fishing run (OM_no_F) behind the collapse metrics.

Usage

```
remove_F(object = NULL, styr = NULL, Rceattle = NULL)
```

Arguments

object	A fitted Rceattle model object
styr	First year with $F = 0$; default <code>endyr + 1</code> , which leaves the hindcast unchanged.
Rceattle	deprecated name for <code>object</code> , still accepted so existing scripts keep working. Supplying both is an error.

Details

`styr` may be any year from the model's own `styr` to `endyr + 1`; the projection is unfished whatever harvest control rule the model was fit under. Under predation, empirical suitability (`suitMode = 0`) is derived from the fitted abundance over each predator's `suit_styr:suit_endyr`, so `styr` must fall after that window for every predator with `suitMode = 0` and diet data in it.

rename_output	<i>Label the derived quantities reported by a CEATTLE fit</i>
---------------	---

Description

Attaches dimension names (species, sex, age, length bin, year, fleet) to the report objects returned by the TMB model, and names the rows/columns of the likelihood-component matrices, so the fitted quantities read as labeled arrays rather than bare numbers.

Usage

```
rename_output(data_list = NULL, quantities = NULL)
```

Arguments

data_list	an Rceattle data_list
quantities	list of "report" objects from Rceattle.

report_tables	<i>Assessment reporting tables from one or more Rceattle fits</i>
---------------	---

Description

Collects the quantities a stock assessment reports into one set of tidy tables, so a SAFE chapter or a model comparison comes from a single call rather than a dozen ad-hoc extractions. Every table holds a model column, so passing several fits gives a like-for-like comparison.

Usage

```
report_tables(
  object,
  model_names = NULL,
  retro = NULL,
  jitter = NULL,
  osa = NULL,
  ci_level = 0.95,
  quantities = c("biomass", "ssb", "R", "biomass_depletion", "ssb_depletion", "F_spp")
)
```

Arguments

object	An Rceattle fit from fit_mod() , or a list of them.
model_names	Names for the models, defaulting to the names of object or to Model_1, Model_2, ...
retro	A retrospective() result per model, or NULL (default) to omit that section.
jitter	A jitter() result per model, or NULL (default) to omit that section.
osa	An osa_residuals() result per model, summarized here with osa_diagnostics() , or NULL (default) to omit that section.
ci_level	Confidence level for the timeseries interval, default 0.95.
quantities	Time-series quantities to include, from the set as.data.frame.Rceattle() accepts; quantity_dictionary() says what each one means.

Details

The sections follow the AFSC Alaska Groundfish Stock Assessment Guidelines for what a chapter reports:

model One row per fit: dimensions, switches, the marginal and joint negative log-likelihoods, estimated parameter count, AIC, maximum gradient, whether the Hessian was positive definite, and run time.

parameters Every estimated parameter with its standard error and its natural-scale name and process from `parameter_dictionary()`, which is where `sigma_R` and an estimated `M` are found. Estimates are on the parameter's own scale, so a `log_name` needs `exp()`. A **fixed** `M` is absent, never having been estimated: read it off `M_at_age`.

likelihood Negative log-likelihood by component and fleet or species, weighted and unweighted.

timeseries Biomass, female spawning-stock biomass, recruitment, depletion and fishing mortality by species and year, with standard errors and intervals, split into hindcast (`era = "time"`) and projection (`era = "fore"`).

reference_points The executive-summary quantities: SPR-based `F` proxies, unfished and target female spawning-stock biomass, the biomass proxies implied by `Ptarget / Plimit`, and terminal status. A basis column says whether each was estimated, and if not why.

fits Observed against predicted index and catch, with the standard deviation of normalized residuals (SDNR) per fleet.

retrospective, jitter, osa Present only when the corresponding object is supplied.

Nothing here refits: the three diagnostics are tens to hundreds of optimizations each, so they are passed in already computed, and a section whose object is `NULL` is absent from the result.

The standard harvest scenarios of guideline section 4.11.3 are **not** produced; they need a standard projection module, which Rceattle does not have. Projected biomass under the model's own control rule is in `timeseries` at `era = "fore"`.

Reference points a fit never estimated come back as NA, with the reason in basis. The reported array leaves a plausible *number* in each, which is how a wrong figure reaches an executive summary:

- `Ftarget / Flimit` are estimated only under a control rule that defines them, and are off for a species with no projected fishery or with fixed numbers-at-age. Unestimated, they sit at $\exp(0) = 1$, reading as an `F` of 1.0/yr.
- `SB0 / B0` under `msmMode > 0` come from the `MSSB0 / MSB0` inputs, which hold a placeholder until `fit_mod()` derives them from a no-fishing projection. `B_target` and `B_limit` are fractions of `SB0` and go with it.
- The per-recruit quantities (`SPR0`, `SPRtarget`, `SPRlimit`) are computed only under `msmMode = 0`.

The depletions are **not** blanked alongside `SB0`: under a no-fishing control rule in multispecies mode the model divides by last-projection-year biomass, the equilibrated unfished reference, so the series is meaningful.

Value

A list of data frames with class `"rceattle_report"`, one per section above, each holding a `model` column.

Two negative log-likelihoods

model reports both, and **both are minimized**, so smaller is the better fit. `marginal_nll` is what the optimizer minimized, random effects integrated out by Laplace, and is what AIC is built from. `joint_nll` is what the template evaluated at the conditional modes, so it is what likelihood sums to, on the same scale as `jnll_comp`. They are equal when `n_random` is 0 and differ by the Laplace correction otherwise.

Supplying diagnostics for several models

A diagnostics list is matched to models **by name**, so `list(alt = ..., base = ...)` pairs correctly whatever the order. An unnamed list is paired positionally, with a message. A name that is not a model name is an error, which catches passing one model's `osa_residuals()` result stored as a list of parts.

See Also

`quantity_dictionary()` for what each quantity means and its units, `as.data.frame.Rceattle()` for the time series alone, and `standard_output()` to relabel the result into the NOAA standardized assessment output that `stockplotr` and `asar` consume.

Examples

```
## Not run:
fit <- fit_mod(data_list = BS2017SS, estimateMode = 1)
tabs <- report_tables(fit)
tabs$reference_points

# Compare two models, with diagnostics attached
tabs <- report_tables(list(base = fit0, alt = fit1),
                      retro = list(retro0, retro1))

## End(Not run)
```

residuals.Rceattle	<i>Residuals from an Rceattle fit</i>
--------------------	---------------------------------------

Description

Returns a long-format data frame of residuals, following the convention of `stats::residuals.glm()` where type selects the *kind* of residual. By default residuals are returned for every applicable data source, survey indices, fishery catches, age/length composition (comp), and conditional age-at-length (caal); use `source` to restrict to particular ones.

Usage

```
## S3 method for class 'Rceattle'
residuals(
  object,
  type = "response",
  source = "all",
  scale = "log",
  species = NULL,
  ...
)
```

Arguments

object	An object of class "Rceattle" returned by <code>fit_mod()</code> .
type	Residual kind: one of "response" (default), "pearson", "osa", or "process".
source	Data source(s) to include: any of "index", "catch", "comp", "caal", or "all" (default). "diet" (predator stomach-content composition) is also accepted but, because it uses a predator/prey schema rather than the fleet/bin layout, must be requested on its own. Ignored when type = "process".
scale	"log" (default) or "natural". Only affects "response" residuals for index / catch.
species	Optional species code(s) to include (matched against the Species column). Default NULL keeps all species. Mirrors the species argument of <code>plot.rceattle_osa()</code> .
...	Passed to <code>osa_residuals()</code> (e.g. method, seed) when type = "osa", or to <code>process_residuals()</code> when type = "process".

Details

Residual kinds (type):

"response" Observed minus fitted. For index / catch this is on the log scale by default (matching the lognormal likelihood; set scale = "natural" for the arithmetic difference); for comp / caal it is the difference in proportions, observed minus fitted.

"pearson" Standardized residuals. For catch and a log-scale survey index, $(\log o - (\log \hat{o} - b\sigma^2/2))/\sigma$ using the model's realized observation log-SD σ and the observation bias-adjustment flag b (bias_adjust_obs, default 1). A natural-scale index fleet (Index_distribution "MVN", "MVNORM", "Normal" or "TruncatedNormal") holds an ABSOLUTE σ and is standardized as $(o - \hat{o})/\sigma$ instead. Two caveats there: a covariance fleet ("MVN", "MVNORM") takes σ from `sqrt(diag(index_cov))`, since its likelihood never reads the reported scalar sd, and so gets its marginal residual rather than the whitened one (use type = "osa" for that); and "TruncatedNormal" is standardized on the untruncated moments, so it is approximate where $\hat{o}/\sigma$ is small enough for truncation to shift the mean. For comp / caal, $(p - \hat{p})/\sqrt{\hat{p}(1 - \hat{p})/N}$ with input sample size N .

"osa" One-step-ahead residuals via `osa_residuals()`, which builds the composition observation data on demand from any fit. Pass its arguments through ...; note that method = "cdf" forms no conditional mean, so Fitted is NA for every row under it.

"process" Process residuals via `process_residuals()` for the model's random-effect deviations; source does not apply.

Composition rows are returned in long form (one row per observation x age/length bin) and hold the Age0_Length1 flag from `comp_data` (0 age, 1 length); CAAL rows hold both the conditioning Length and the age Bin.

Where a fleet uses tail accumulation (`Comp_accum_young` / `Comp_accum_old`), composition residuals describe the bins the likelihood actually fit: the tails are folded into their boundary bin and the bins outside the window are not reported, because the model never fit them separately. Bin names the age or length each residual belongs to and Accumulated marks the boundary bins, which stand for a range rather than the single bin they are named for. Fleets without accumulation are unchanged.

Value

A data.frame with columns Source, Fleet_code, Fleet_name, Species, Sex, Year, Length, Bin, Age0_Length1, Sample_size, Accumulated, Observed, Fitted, Residual. Columns are NA where they do not apply; Accumulated is FALSE except on a composition bin that absorbed a folded tail.

retrospective	<i>Retrospective peels</i>
---------------	----------------------------

Description

Calculate Mohn's rho and run retrospective peels for an Rceattle model, and evaluate retrospective forecast skill. Each peel uses TMB's map to:

1. filter the data and fixed inputs and map out time-varying parameters for the peeled years, holding them at that peel's terminal year;
2. fit the peeled model;
3. turn the hindcast parameters off and F for the peeled years on, and fit to the peeled catch series, updating the "forecast" dynamics given the projection assumptions and the observed catch.

Usage

```
retrospective(
  object = NULL,
  peels = 5,
  rescale = FALSE,
  nyrs_forecast = 3,
  cores = NULL,
  getsd = NULL,
  phase = TRUE,
  fit_control = NULL,
  Rceattle = NULL
)
```

Arguments

<code>object</code>	an Rceattle model fit using <code>fit_mod()</code> .
<code>peels</code>	number of retrospective peels used for rho and estimation
<code>rescale</code>	TRUE/FALSE whether to subset and rescale environmental predictors over the peel years.
<code>nyrs_forecast</code>	forecast years over which to calculate Mohn's rho, in addition to the terminal year
<code>cores</code>	Number of cores for the parallel refits. Default NULL picks <code>parallel::detectCores()</code> - 6, capped at 2 under R CMD check (which sets <code>_R_CHECK_LIMIT_CORES_</code>). Set to 1 to force sequential execution.
<code>getsd</code>	whether each peel runs <code>TMB::sdreport</code> , which costs an extra model build per peel (see Details). Mohn's rho uses point estimates only, so FALSE is faster with no effect on it. NULL (default) inherits the input model's setting (TRUE only if it holds an <code>sdrep</code>), and the returned peels hold standard errors only when it is TRUE.
<code>phase</code>	whether each peel is refitted in phases, default TRUE. A peel restarts from the unpeeled fit's starting values with a year removed, so without phasing the parameters barely move, the peels sit on top of the full model, and Mohn's rho is biased toward zero. Change it only deliberately.
<code>fit_control</code>	optional <code>fit_control()</code> bundle for the refits. Only phase and <code>getsd</code> are read; see What <code>fit_control()</code> reaches.
<code>Rceattle</code>	deprecated name for <code>object</code> , still accepted so existing scripts keep working. Supplying both is an error.

Details

Each peel is fitted twice: a peeled hindcast, then a forecast refit estimating only the peeled years' F. The second holds every hindcast parameter fixed, so on its own it reports a standard error of zero across the hindcast. Under `getsd = TRUE` the peel is therefore rebuilt at those same parameters with the hindcast free in the map and reported from there; nothing is re-estimated, so no point estimate moves. Where that rebuild fails – it throws, or comes back with a Hessian it could not factor, or with no finite standard error – the peel keeps its point estimates and carries no `sdrep` at all, rather than the forecast refit's zero standard errors. The reason is named in one warning per call and carried on the peel; `print()` reports how many peels could not report standard errors.

Value

a list of 1. a list of Rceattle models and 2. Mohn's rho per species.

A peel that did not converge is dropped, so `Rceattle_list` can be shorter than `peels + 1`, with a message. Entries are named for their own terminal year (`Year_2017, ...`), so index by name: `Rceattle_list[[3]]` is not necessarily the 3-year peel. With no peel left, rho is NaN and the function warns. Each peel reports its own terminal year as `data_list$endyr`, so plots draw it only as far as it was fit.

A peel still estimates the years it dropped: that is its retrospective forecast, fit to observed catch with the survey and composition data withheld. Its recruitment deviation is the one `sample_rec()`

sets with `sample_rec = FALSE`, computed from the peel's own fit. Three years matter, and every peel carries all three:

`endyr`, `endyr_peel` the peel's terminal year, what it was fit through. Equal to each other.

`endyr_full` the unpeeled model's terminal year, where the retrospective forecast ends.

`projyr` the end of the harvest-control-rule projection.

A peel whose standard errors could not be rebuilt carries `attr(peel, "sd_report_error")`, a one-line reason naming the peel and its terminal year; it is absent on every other peel.

Take the forecast years as `endyr_peel + seq_len(endyr_full - endyr_peel)`, which is empty for the unpeeled model, rather than `(endyr_peel + 1):endyr_full`, which counts *down* there. Mohn's rho is computed from `endyr_peel` and is unaffected by any of this.

Catchability is estimated only for a fleet holding fitted index rows (see [build_map](#)), and a peel moves `endyr`, so a survey whose index observations all fall in the peeled-off years has no `q` estimated in that peel: the parameter count is not constant across peels. That is deliberate, since a `q` with no index to inform it is a flat direction in the likelihood. Rho is computed from SSB and is unaffected, but `npar` and the reported catchability do differ between a shallow and a deep peel for such a fleet.

Examples

```
data(BS2017SS)
ss_run <- fit_mod(data_list = BS2017SS,
  inits = NULL, file = NULL,
  estimateMode = 0, random_rec = FALSE,
  msmMode = 0, avgnMode = 0,
  phase = FALSE, verbose = 0)
retro <- retrospective(ss_run, peels = 10)
```

reweight_comps

Iteratively reweight composition data (McAllister-Ianelli)

Description

Tunes multinomial composition weights by refitting: each pass sets a fleet's weight to the McAllister & Ianelli (1997) weight implied by the previous fit, and stops once the weights settle. The weight multiplies the input sample size, so changing it changes the fit and hence the next implied weight, which is why the weights are tuned iteratively rather than in one step.

Usage

```
reweight_comps(
  object = NULL,
  n_iter = 10,
  tol = 0.01,
  fleets = NULL,
  verbose = TRUE,
  fit = NULL
)
```

Arguments

<code>object</code>	An Rceattle model fitted at estimateMode "Estimate" (0) or "Hindcast" (1), the modes that optimize the hindcast.
<code>n_iter</code>	Maximum number of iterations (default 10).
<code>tol</code>	Relative change in the weights below which to stop (default 0.01).
<code>fleets</code>	Fleets to tune, as Fleet_code values or as names matching fleet_control\$Fleet_name (as in <code>linkage_spec()</code>). Give ids or names, not a mix, R coerces <code>c(1, "Survey")</code> to <code>c("1", "Survey")</code> . Default NULL tunes every eligible fleet.
<code>verbose</code>	Print the weights each iteration (default TRUE).
<code>fit</code>	deprecated name for object, still accepted so existing scripts keep working. Supplying both is an error.

Details

Only multinomial fleets are tuned. A Dirichlet-multinomial fleet estimates its own weight within the likelihood, so tuning it externally would compete with that estimate; those fleets are named and skipped, as are any requested through fleets that have no fitted composition data.

Only the age/length composition weight (Comp_weights) is tuned. The conditional age-at-length and diet weights have McAllister-Ianelli analogues of their own (CAAL_weights, Diet_comp_weights), but they are not tuned here; set them by hand if needed.

Tuning stops when the largest relative change in any weight falls below tol. Hitting n_iter first gives a warning, not an error, the partly tuned fit is returned either way, and \$reweight\$history shows whether the weights were still settling.

The weight is a parameter, not a data input, so each pass holds it to the next through inits; fleet_control\$Comp_weights supplies the value only when a model is built from scratch, and editing that column then refitting from an existing fit has no effect (fit_mod() warns when it detects this). Each pass writes the new weight to both, so the returned model's data list rebuilds the tuned model from scratch.

Value

The model refitted with the final weights, holding a reweight element: history (one row per fleet per iteration), iterations, and converged. The last rows of history are the weights the returned model was fitted with.

References

McAllister, M.K. and Ianelli, J.N. (1997) Bayesian stock assessment using catch-age data and the sampling-importance resampling algorithm. *Canadian Journal of Fisheries and Aquatic Sciences* 54:284-300.

Francis, R.I.C.C. (2011) Data weighting in statistical fisheries stock assessment models. *Canadian Journal of Fisheries and Aquatic Sciences* 68:1124-1138. (The main alternative; applied by hand rather than by this function.)

Examples

```
## Not run:
fit <- fit_mod(BS2017SS, estimateMode = "Hindcast")
tuned <- reweight_comps(fit)
tuned$reweight$history      # weight per fleet, per iteration

## End(Not run)
```

rich.colors.short	<i>Make a vector of colors.</i>
-------------------	---------------------------------

Description

A subset of rich.colors by Arni Magnusson from the gplots package, with the addition of alpha transparency (which is now available in the gplots version as well)

Usage

```
rich.colors.short(n, alpha = 1)
```

Arguments

n	Number of colors to generate.
alpha	Alpha transparency value for all colors in vector. Value is passed to rgb function.

Author(s)

Arni Magnusson, Ian Taylor

run_config	<i>Extract the run configuration from a fit, data list, or config object</i>
------------	--

Description

Returns the `model_config()` structure plus the estimation controls and `fit_control()` bundle as a single `Rceattle_run_config`. Accepts a fitted `Rceattle` object, a data list holding `$model_config`, an `Rceattle_run_config`, or an `Rceattle_model_config`. Estimation controls and `fit_control` supplied via `...` override any found on the object.

Usage

```
run_config(x, ...)
```

Arguments

`x` A fitted Rceattle object, a data list, an Rceattle_run_config, or an Rceattle_model_config.
`...` Estimation controls / `fit_control` to override (`estimateMode`, `random_rec`, `random_q`, `random_sel`, `suit_styr`, `suit_endyr`, `fit_control`).

Value

An Rceattle_run_config.

See Also

[save_config\(\)](#), [load_config\(\)](#), [model_config\(\)](#).

Examples

```
# Coerce a model_config() into a full run configuration, overriding the
# estimation controls. A fitted model or a data list works the same way.
rc <- run_config(model_config(msmMode = 1), estimateMode = "Hindcast")
rc$estimateMode

# Optimizer and uncertainty settings belong inside fit_control(), not here;
# an unrecognized field is an error rather than a silent no-op.
rc <- run_config(rc, fit_control = fit_control(getsd = FALSE))
```

run_mse

Run a management strategy evaluation

Description

Runs a forward-projecting management strategy evaluation (MSE). Projected selectivity, catchability, foraging days and weight-at-age are held at the operating model's terminal hindcast year, survey SD is the average over the historical series, and composition sample size is held at the last year. There is no implementation error and no observation error on catch. Fixed numbers-at-age (`estDynamics > 0`) are an input rather than a forecast, so they are NOT held forward: supply `NByageFixed` rows for every projection year, or the uncovered years read 0 and that species contributes nothing, with a warning naming it.

Usage

```
run_mse(
  om,
  em,
  nsim = 10,
  start_sim = 1,
  assessment_period = 1,
  sampling_period = 1,
  simulate_data = TRUE,
```

```

    regenerate_past = FALSE,
    sample_rec = TRUE,
    rec_trend = 0,
    fut_sample = 1,
    cap = NULL,
    catch_mult = NULL,
    seed = 666,
    regenerate_seed = seed,
    loopnum = 1,
    file = NULL,
    dir = NULL,
    timeout = 999,
    endyr = NA,
    cores = NULL
  )

```

Arguments

om	CEATTLE model object exported from Rceattle
em	CEATTLE model object exported from Rceattle
nsim	Number of simulations to run (default 10)
start_sim	First simulation number to start at, useful when a run stops at a particular seed (default 1).
assessment_period	Assessment schedule. A single number is the period in years between assessments, counted from the operating model's terminal year; a vector is the explicit set of assessment years, which must name two or more years inside the projection horizon. Default 1.
sampling_period	Period of years data sampling is conducted. Single value or vector the same length as the number of fleets.
simulate_data	Include simulated random error proportional to that estimated or provided for the data from the OM.
regenerate_past	Refit the EM to historical conditioning data before the MSE, generated from the OM with or without sampling error according to <code>simulate_data</code> .
sample_rec	Include resampled hindcast recruitment deviations in the OM projection. Resampled rather than drawn from $N(0, \sigma_R)$ because the initial deviations bias R_0 low. FALSE uses the single deviation described in <code>sample_rec()</code> .
rec_trend	Linear change in mean recruitment from endyr to projyr, as the terminal multiplier $\text{mean rec} * (1 + (\text{rec_trend}/\text{projection years}) * 1:\text{projection years})$. Length 1 or nspp.
fut_sample	Future sampling effort relative to the last year: $\text{Log_sd} * 1 / \text{fut_sample}$ for the index and $\text{Sample_size} * \text{fut_sample}$ for comps.
cap	A ceiling on projected catch. A single number is an ANNUAL ceiling on total removals summed across species, shared between them in proportion to the

	catch the control rule recommended; a vector of length nspp is a separate ceiling per species. Default NULL.
catch_mult	A multiplier on the catch the control rule recommends, applied before cap and before the exploitable-biomass limit. A single number or a vector of length nspp applies in every projection year; a data.frame with columns Year, Species and mult applies only to the pairs it lists, and any pair it omits is multiplied by 1. Year must be a year the schedule covers. Default NULL.
seed	seed for the simulation
regenerate_seed	seed for regenerating data
loopnum	Number of times to restart optimization; loopnum = 3 sometimes reaches a lower final gradient than 1.
file	(Optional) Filename where each OM simulation with EMs is saved. If NULL, no files are saved.
dir	(Optional) Directory where each OM simulation is saved
timeout	Minutes estimation runs before a sim is stopped (default 999)
endyr	Terminal year of the MSE projection. Default NA uses projyr from the operating model.
cores	Cores for the parallel simulations. NULL (default) picks parallel::detectCores() - 6, capped at 2 under R CMD check; 1 forces sequential.

Details

Designing a schedule, comparing monitoring scenarios, and what the common random numbers do and do not share across schedules are in vignette("hcrs-and-mses").

One piece of arithmetic to get right when buffering a missed assessment: the assessment in year Y sets catch for $Y + 1$ onward. So dropping 2031 from a biennial cycle leaves **2032 and 2033** on 2029's advice, and 2031 itself is set by the 2029 assessment either way – buffering 2031 changes a year the missed assessment never touched.

Catch recorded past a model's terminal year is blanked to NA at setup, with a warning naming the years: the MSE sets those years from the control rule, and the likelihood never scored them, since it fits only $\text{Year} \leq \text{endyr}$. That is what a workbook looks like when endyr has fallen behind the catch series, and it is worth resolving first, because conditioning on those years is a different question from projecting over them.

Value

A list of operating models (differing by simulated recruitment, per nsim) and estimation models fit to each (differing by terminal year).

Examples

```
## Not run:
data(BS2017SS)
om <- fit_mod(BS2017SS, estimateMode = "Estimate", HCR = build_hcr("NPFMC"))
# Closed loop: assess every 2 years, survey every year, 10 simulations.
```

```
mse <- run_mse(om = om, em = om, nsim = 10,
               assessment_period = 2, sampling_period = 1)
mse_summary(mse)$species

## End(Not run)
```

sample_rec

*Sample historical recruitment deviations into the projection***Description**

Sample historical recruitment deviations into the projection

Usage

```
sample_rec(
  object = NULL,
  sample_rec = TRUE,
  update_model = TRUE,
  rec_trend = 0,
  Rceattle = NULL
)
```

Arguments

object	CEATTLE model object exported from Rceattle
sample_rec	Include resampled recruitment deviations from the hindcast in the OM projection. Resampled deviations are used rather than drawing from $N(0, \sigma_R)$ because the initial deviations bias R_0 low. If FALSE, uses one deviation (plus the log of the rec_trend multiplier) for every projection year: $\log(\text{mean}(R)) - \log(R_0)$ under mean recruitment, and $\log(\text{mean}(R / \hat{R}))$ under a curve, over every hindcast year when the curve is fitted in the hindcast and over the penalty years for the penalty form.
update_model	Update model dynamics. Default = TRUE
rec_trend	Linear increase or decrease in mean recruitment from endyr to projyr. This is the terminal multiplier $\text{mean rec} * (1 + (\text{rec_trend}/\text{projection years}) * 1:\text{projection years})$. Can be of length 1 or of length nspp. If length 1, all species get the same trend.
Rceattle	deprecated name for object, still accepted so existing scripts keep working. Supplying both is an error.

Value

Rceattle model

save_config

Save a model run configuration to a documented YAML file

Description

Round-trips a full run configuration, the `model_config()` structure plus the estimation controls and `fit_control()` bundle, to a git-diffable YAML file, with each field's documentation emitted as a comment and a spec-tree + provenance header. Only fields that differ from their defaults are written, so two configurations diff to just their real differences. The parameter values (inits/map/bounds) are NOT stored; pair the config with a saved fit for those.

Usage

```
save_config(x, file = "Rceattle_config.yaml", ...)
```

Arguments

x	A fitted Rceattle object, a data list holding <code>\$model_config</code> , an <code>Rceattle_run_config</code> , or an <code>Rceattle_model_config</code> .
file	Output path for the .yaml file.
...	Estimation controls / <code>fit_control</code> to record (passed to <code>run_config()</code>).

Value

Invisibly, the `Rceattle_run_config` that was written.

See Also

`load_config()`, `run_config()`, `model_config()`, `fit_mod()`.

Examples

```
cfg <- model_config(msmMode = 1, initMode = "OffsetEquilibrium")
f <- file.path(tempdir(), "run.yaml")
save_config(cfg, f)
identical(load_config(f)$model_config$msmMode, 1)
```

self_test

*Self-test simulation analysis***Description**

Simulates data from an Rceattle model and refits the model to the simulated data, to check that the fitting procedure recovers the operating-model parameters. Pass process to redraw a process as well as the observations.

Usage

```
self_test(
  object = NULL,
  nsim = 50,
  simulate = TRUE,
  seed = 123,
  cores = NULL,
  getsd = NULL,
  phase = NULL,
  start = c("initial", "estimated"),
  debug = FALSE,
  timeout = Inf,
  process = FALSE,
  fit_control = NULL,
  Rceattle = NULL
)
```

Arguments

object	an Rceattle model fit using <code>fit_mod()</code> .
nsim	number of simulations
simulate	passed to <code>sim_mod</code> . TRUE (default) simulates data with observation error; FALSE uses the model's expected values.
seed	random number seed. Simulation <i>i</i> uses <code>seed + i</code> , so results are reproducible under sequential and parallel execution alike.
cores	Number of cores for the parallel refits. Default NULL picks <code>parallel::detectCores()</code> - 6, capped at 2 under R CMD check (which sets <code>_R_CHECK_LIMIT_CORES_</code>). Set to 1 to force sequential execution.
getsd	whether each refit runs <code>TMB::sdreport</code> . Self-test compares refit point estimates to the operating model, so FALSE is faster with no effect on that comparison. NULL (default) inherits the input model's setting (TRUE only if it holds an <code>sdrep</code>).
phase	as in <code>fit_mod</code> . Under the default <code>start = "initial"</code> each refit covers the ground the original fit did, so a model that needed phasing for its real data needs it for every simulated one; without it such a model's refits can end many orders

	of magnitude from a zero gradient and be dropped as non-converged. NULL (default) reads what <code>fit_mod()</code> recorded on the source fit, so a model fitted under the package default <code>phase = FALSE</code> is refitted unphased.
<code>start</code>	Which parameter set each refit starts from: "initial" (default) or "estimated". See Details.
<code>debug</code>	return every simulation rather than only the converged ones. The dropped runs are the interesting ones when a self-test comes back short, and each holds its own <code>\$convergence</code> diagnostics. See Value .
<code>timeout</code>	elapsed-second limit per simulation, Inf (default) for none. The optimizer runs with no iteration cap, so a replicate that wanders somewhere pathological can stall the run – a hang no convergence check can catch, because the fit never returns. One that exceeds the limit is stopped, counted as non-converged and reported separately. It is approximate: the limit is checked when control returns to R, so it fires between the optimizer's function evaluations rather than inside one.
<code>process</code>	passed to <code>sim_mod</code> . FALSE (default) keeps the fitted process deviations, so the test measures whether the estimator recovers its own parameters from new observations. Naming a process ("recruitment", "M", "growth", "dynamics", TRUE) redraws it, so the test instead measures recovery of a process the estimator has not been shown.
<code>fit_control</code>	optional <code>fit_control()</code> bundle for the refits. Only <code>phase</code> and <code>getsd</code> are read; see What <code>fit_control()</code> reaches .
<code>Rceattle</code>	deprecated name for object, still accepted so existing scripts keep working. Supplying both is an error.

Details

What start costs. "initial" uses `initial_params`, so the estimator travels the same distance to the optimum on simulated data as it did on the real data. "estimated" starts from the generating values instead – the fixed effects, and under `random_rec = TRUE` the inner Laplace problem too – so it is much faster and far likelier to converge, but on a multimodal or weakly identified surface the optimizer never leaves the basin containing them, so recovery is close to guaranteed by construction. Read it as optimistic about recovery, not merely less powerful. It is not a complete warm start either – `fit_mod()` resets `log_Ftarget`, `proj_F_prop` and the stock-recruit α/β under both settings – and non-identifiability stays visible either way through `$convergence`'s Hessian conditioning and estimability checks. What a warm start stops testing is *reachability*.

Value

A list of Rceattle models named `Sim_1`, `Sim_2`, ... By default only the converged simulations, renumbered contiguously, with a message reporting how many were dropped. The list has class "Rceattle_selftest" and carries the number attempted in `attr(, "nsim")`, so printing it reports the convergence rate; it is otherwise the plain list it always was.

With `debug = TRUE`, every simulation, `Sim_i` being simulation `i` so it pairs with seed `seed + i`, plus the convergence verdicts in `attr(, "converged")`. A refit that errored outright comes back as the condition object rather than a model, so it cannot abort the run.

Where process redraw something, `attr(, "process_sim")` holds the deviations that generated each replicate, keyed by the same `Sim_i` names and subset alongside the models. Each entry names whichever of `rec_dev`, `init_dev`, `log_M1_dev` and `beta_linkage_re` were drawn, each with a same-shaped `_drawn` logical marking the cells the draw touched. Restrict any recovery statistic to those cells, and compare estimates against these rather than against the operating model, whose fitted deviations are no longer what generated the data.

Interpreting the spread

`sim_mod` redraws the observations only, and warns about each row it leaves alone: a predator with empirical rather than estimated suitability has no predicted diet to draw from, and a covariance (MVN/MVNNORM) survey fleet has no covariance outside its fitted years. Those data are identical in every replicate, so recovery of whatever they inform is optimistic.

By default recruitment is not redrawn, so under `random_rec = TRUE` every replicate shares the operating model's single realization – its shrunk empirical-Bayes modes, not a draw from $N(0, \text{sigmaR})$. The spread across replicates is then observation error alone and a lower bound on estimation uncertainty in SSB and recruitment, so do not read it against the model's own uncertainty bands, which include process error; and `sigmaR` is re-estimated from deviations shrunk toward zero the same way every time, a downward bias averaging does not remove. `process = "recruitment"` removes both, at the cost of asking a different question. See `vignette("model-diagnostics")`.

Examples

```
data(BS2017SS)
ss_run <- fit_mod(data_list = BS2017SS,
  inits = NULL, file = NULL,
  estimateMode = 0, random_rec = FALSE,
  msmMode = 0, avgnMode = 0,
  phase = FALSE, verbose = 0)
sims <- self_test(ss_run, nsim = 10)
```

set_phases

Default phasing order for CEATTLE parameters

Description

Default phasing order for CEATTLE parameters

Usage

```
set_phases()
```

Value

Tagged list mapping each parameter name to its estimation phase.

sim_mod	<i>Simulate Rceattle data</i>
---------	-------------------------------

Description

Simulates the data an Rceattle model would have produced, as expected values or as a random draw. Every observation type is covered: survey biomass (under the fleet's own `Index_distribution` – lognormal, natural-scale normal, or the correlated MVN/MVNORM draw from its covariance), total catch (lognormal), age/length composition and conditional age-at-length, and stomach contents (multinomial or Dirichlet-multinomial).

Usage

```
sim_mod(object = NULL, simulate = FALSE, process = FALSE, Rceattle = NULL)
```

Arguments

object	A CEATTLE model object exported from Rceattle.
simulate	Logical. TRUE draws from the distributions, FALSE returns the expected values (hats).
process	Which process error to redraw alongside the observations. FALSE (default) or "none" keeps the fitted deviations; TRUE or "all" redraws every process; "dynamics" covers recruitment, natural mortality and growth, "observation" covers catchability and selectivity. Any subset of "recruitment", "M", "growth", "catchability" and "selectivity" may be given instead. Ignored when <code>simulate = FALSE</code> .
Rceattle	deprecated name for object, still accepted; supplying both is an error.

Details

Every draw is taken by the TMB model itself, in a `SIMULATE` block beside the likelihood that defines it, so `simulate = TRUE` needs a model to evaluate. A model loaded from a file has one, and a fit whose `$obj` was dropped to save space is rebuilt from its `data_list` and estimates, provided the rebuild reproduces the fit's own expected values. `model_average` output cannot be simulated from at all: its quantities are an average over models, so no parameters produced them. `simulate = FALSE` reads only `$quantities` and draws no random numbers, so it works on any model.

Rows the model predicts nothing for are left alone, and all but the last of these is reported by a warning: a composition for a fleet with no catch that year comes back empty, while a stomach whose predator has empirical suitability, and a covariance fleet's observations outside its fitted window, keep their observed values. A composition, CAAL or diet row whose sample size times its weight rounds below one observation also comes back empty, with `Sample_size` dropped to zero so a refit does not score a row with no data behind it. A hindcast survey year with no observation (NA) stays NA and is NOT warned about, since the likelihood never scored it, while the independent survey families still draw their projection rows (negative Year), which `run_mse` reveals as each assessment's new data.

One deliberate exception to draw-equals-density: compositions are drawn from the predicted proportions *before* comp_offset (the constant added to observed and predicted alike to keep $\log(0)$ out of the density, default $1e-5$), so a bin predicted at zero is drawn at zero while the density scores it at comp_offset. That is about $4e-4$ relative on a 20-bin composition and does not bias the round trip, since the offset is applied to both sides on the next fit.

Drawn values are left in the object's report environment under names ending _sim; the estimates, the data and the objective are untouched, so a later osa_residuals() or vcov() on the same model is unaffected.

Value

A data_list of the simulated or expected data, formatted for Rceattle. Where process redrew something, the deviations that generated the data are attached as attr(x, "process_sim"), a named list holding whichever of rec_dev, init_dev, log_M1_dev and beta_linkage_re were drawn. Those are the truth a refit has to recover; without them the only comparison available is against the original fitted deviations, which are no longer what generated the data.

Each comes with a same-shaped logical named *_drawn, TRUE where the draw touched that cell. Draws cover the hindcast only, and beta_linkage_re is one vector over every random-linkage slot whether or not its process was asked for, so the arrays hold fitted values beside simulated ones.

Restrict any recovery statistic to the _drawn cells; over the full array it reports perfect recovery on the cells that were never redrawn.

Examples

```
## Not run:
data(BS2017SS)
fit <- fit_mod(BS2017SS, estimateMode = "Hindcast")
# Expected values only, no observation error drawn.
sim_mod(fit)
# One stochastic replicate, drawn from each fleet's own likelihood.
sim_mod(fit, simulate = TRUE)

## End(Not run)
```

simulate.Rceattle	<i>Simulate data from a fitted Rceattle model</i>
-------------------	---

Description

The `stats::simulate()` method for CEATTLE fits: draws nsim replicate data sets from the fitted observation model, and optionally from the process model as well.

Usage

```
## S3 method for class 'Rceattle'
simulate(object, nsim = 1, seed = NULL, process = FALSE, ...)
```

Arguments

object	An object of class "Rceattle" returned by <code>fit_mod()</code> .
nsim	Number of replicate data sets to draw. Default 1.
seed	Optional seed. The caller's random state is restored afterwards, per <code>stats::simulate()</code> . Recorded as <code>attr(, "seed")</code> .
process	Which process error to redraw alongside the observations. FALSE (default) keeps the fitted deviations, so replicates differ only in observation error. See <code>sim_mod()</code> for the alternatives.
...	Currently unused.

Details

A wrapper on `sim_mod()`, which documents the observation model in full. For expected values rather than draws, call `sim_mod(simulate = FALSE)`.

Draws are taken by the TMB model, so `simulate()` needs a live `$obj`: a model loaded from disk has one, a `model_average()` result does not, see `sim_mod()`.

Value

A list of `nsim` `data_list` objects, always a list, including at `nsim = 1`, so callers do not have to special-case the length. When process redraw something, each element holds the deviations that generated it as `attr(, "process_sim")`: a named list of whichever of `rec_dev`, `init_dev`, `log_M1_dev` and `beta_linkage_re` were drawn, each with a `_drawn` logical of the same shape marking the cells the draw touched. Compare estimates against those rather than against object, see `sim_mod()`.

See Also

`sim_mod()` for the observation model and the process options, `self_test()` for simulating and refitting in one step.

Examples

```
## Not run:
fit <- fit_mod(data_list = BS2017SS, estimateMode = 1)
reps <- simulate(fit, nsim = 10, seed = 1)
# ...and with recruitment redrawn as well:
reps <- simulate(fit, nsim = 10, seed = 1, process = "recruitment")
truth <- attr(reps[[1]], "process_sim")$rec_dev

## End(Not run)
```

standard_output	<i>Rceattle output in the NOAA standardized assessment format</i>
-----------------	---

Description

Relabels `report_tables()` output into the standardized assessment output that the `stockplotr` and `asar` packages consume, so `Rceattle` results can be plotted and written into a report by the same tools used for `SS3`, `BAM`, `WHAM` and `FIMS`.

Usage

```
standard_output(x, species = NULL, model = NULL)
```

Arguments

<code>x</code>	An "rceattle_report" from <code>report_tables()</code> , or an <code>Rceattle</code> fit, which is passed through <code>report_tables()</code> first.
<code>species</code>	Which stock to emit: a species name, or an index into the species this report holds, required when it holds more than one.
<code>model</code>	Which model to emit when <code>x</code> holds several, defaulting to the first.

Details

Quantity names are translated through the `standard_label` column of `quantity_dictionary()`, so `ssb` becomes `spawning_biomass`, `R` becomes `recruitment`, `F_spp` becomes `fishing_mortality`, and so on. A quantity the standard has no name for keeps its `Rceattle` name, so nothing is silently dropped.

The standard has no species dimension. It describes one stock, so a multispecies CEATTLE fit cannot be represented in it as a whole. A `species` column is held alongside the standard columns and `species` selects one stock; with several species in the fit and no selection, this errors rather than returning a frame in which two stocks' biomass share a year.

`era` follows the standard's own vocabulary: "time" for the hindcast the model was fit to, "fore" for the projection.

Value

A data frame with the standardized columns, in the standard's own order, plus `species` and `model`.

See Also

`report_tables()` for the native tables and `quantity_dictionary()` for the name crosswalk.

Examples

```
## Not run:
fit <- fit_mod(data_list = BS2017SS, estimateMode = 1)
std <- standard_output(fit, species = 1)
# ...then, with the stockplotr package installed:
# stockplotr::plot_spawning_biomass(std)

## End(Not run)
```

summary.Rceattle	<i>Compact summary method for Rceattle fits</i>
------------------	---

Description

Compact summary method for Rceattle fits

Usage

```
## S3 method for class 'Rceattle'
summary(object, ...)
```

Arguments

- object An object of class "Rceattle" returned by `fit_mod()`.
- ... Currently unused.

switch_check	<i>Function to check for missing switches for map and parameter functions</i>
--------------	---

Description

Function to check for missing switches for map and parameter functions

Usage

```
switch_check(data_list)
```

Arguments

- data_list Rceattle data list

TMBAIC	<i>Calculate marginal AIC for a fitted model</i>
--------	--

Description

TMBAIC calculates AIC for a given model fit

Usage

```
TMBAIC(opt, p = 2, n = Inf)
```

Arguments

opt	the output from nlminb or optim
p	the penalty on additional fixed effects (default=2, for AIC)
n	the sample size, for use in AICc calculation (default=Inf, for which AICc=AIC)

Value

AIC, where a parsimonious model has a AIC relative to other candidate models

TMBphase	<i>Run TMB with ADMB-style phased estimation</i>
----------	--

Description

Fits a TMB model in phases, like ADMB: parameters are switched on in stages rather than all at once, which stabilizes a difficult optimization. The optimizer (nlminb by default) runs once per phase; at each phase the map holds fixed any parameter whose phase is greater than the current phase, so parameters turn on progressively as the phase counter advances. phases is a tagged list giving each named parameter its integer phase; parameters not listed are estimated from the first phase onward.

Usage

```
TMBphase(
  data,
  parameters,
  map,
  random,
  phases,
  model_name,
  silent,
  use_gradient = TRUE,
  control = list(eval.max = 1e+09, iter.max = 1e+09, trace = 0)
)
```

Arguments

data	A list to be passed to TMB
parameters	A list of parameters of the model
map	a list of map object from the model
random	A character vector of names of parameters that are random effects
phases	Tagged list assigning each named parameter its integer phase (as returned by <code>set_phases()</code>).
model_name	A string describing the model name. Must be the name of your .cpp file
silent	logical. Suppress TMB's output. No default; phased fits set it.
use_gradient	logical. If TRUE, uses gradient in optimization (default = TRUE).
control	A list of control parameters. For details see ?nlminb

Value

The parameter list estimated in the final phase, with a per-phase convergence log attached as the `phase_log` attribute.

Author(s)

Gavin Fay https://github.com/kaskr/TMB_contrib_R/blob/master/TMBphase/R/TMBphase.R

vcov.Rceattle

Variance-covariance matrix for an Rceattle fit

Description

Returns the fixed-effect covariance matrix produced by `TMB::sdreport()`. Random-effect covariance is not returned here, use `object$sdrep` for the full report.

Usage

```
## S3 method for class 'Rceattle'
vcov(object, ...)
```

Arguments

object	An object of class "Rceattle" returned by <code>fit_mod()</code> .
...	Currently unused.

Value

A numeric matrix, or NULL if `sdreport` was not run (i.e. the fit was produced with `get_sd = FALSE`).

whamGrowthData	<i>Data inputs for CEATTLE model with WHAM-estimated growth</i>
----------------	---

Description

A data list containing inputs for a CEATTLE model that uses growth estimated from the Woods Hole Assessment Model (WHAM). See [BS2017SS](#) for format details. Its two 2DAR1 fleets ship with `Time_varying_sel_sd = 0` and no `N_sel_bins`, so `data_check()` refuses it as shipped; the vignettes set a different selectivity before fitting.

Usage

```
whamGrowthData
```

Format

A list with the same structure as [BS2017SS](#).

write_data	<i>Write a data list to a CEATTLE xlsx workbook</i>
------------	---

Description

Write a data list to a CEATTLE xlsx workbook

Usage

```
write_data(data_list, file = "Rceattle_data.xlsx")
```

Arguments

<code>data_list</code>	Rceattle data_list object
<code>file</code>	Filename to write. Must end in '.xlsx'

See Also

[build_data\(\)](#) to assemble or edit a data list in R, [read_data\(\)](#) to read one back. A [model_config\(\)](#) slot is not written to the workbook and does not survive the round-trip (a warning is issued); persist it separately with [save_config\(\)](#) / [load_config\(\)](#).

Examples

```
library(Rceattle)
data(BS2017SS)
out_file <- file.path(tempdir(), "BS2017SS.xlsx")
write_data(data_list = BS2017SS, file = out_file)
file.remove(out_file)
```

write_template

Write a minimal starter CEATTLE data workbook

Description

Emits a small, structurally complete single-species workbook, one survey and one fishery, flat placeholder data, that a user can open and edit as a starting point. `fleet_control` is built on the canonical column names with schema defaults filled by [switch_check](#), so the template is always in sync with the current schema. The template round-trips through [read_data](#) and [data_check](#) and builds under `fit_mod(estimateMode = 3)`; replace the placeholder observations with real data before fitting.

Usage

```
write_template(
  file = "Rceattle_data_template.xlsx",
  nages = 10,
  nyrs = 30,
  nprojyrs = 12,
  minage = 1
)
```

Arguments

<code>file</code>	Output path for the .xlsx workbook.
<code>nages</code>	Number of ages to template (default 10).
<code>nyrs</code>	Number of hindcast years (default 30).
<code>nprojyrs</code>	Number of projection years beyond the hindcast (default 12).
<code>minage</code>	Minimum age (recruitment age; default 1).

Value

Invisibly, the minimal `data_list` that was written.

Examples

```
f <- file.path(tempdir(), "template.xlsx")
dat <- write_template(f, nages = 8, nyrs = 20)

# The workbook is correctly shaped but holds placeholder observations;
# data_requirements() reports what a given configuration still needs.
head(data_requirements(dat), 4)
unlink(f)
```

Index

* datasets

- Atka2022, [7](#)
 - BS2017MS, [7](#)
 - BS2017SS, [8](#)
 - EBS_ms_run, [41](#)
 - EBS_ss_M_run, [41](#)
 - EBS_ss_run, [42](#)
 - GeorgesBank3spp, [48](#)
 - GOA2018SS, [50](#)
 - GOAatf, [51](#)
 - GOAatf2023, [51](#)
 - GOAcod, [52](#)
 - GOApollock, [52](#)
 - GOAsafe2018, [52](#)
 - NorthernRockfish2022, [64](#)
 - whamGrowthData, [159](#)
-
- adjust_map_shared_params, [5](#)
 - as.data.frame(), [6](#)
 - as.data.frame.Rceattle, [6](#)
 - as.data.frame.Rceattle(), [129](#), [130](#), [135](#), [137](#)
 - Atka2022, [7](#)
 - BS2017MS, [7](#)
 - BS2017SS, [7](#), [8](#), [48](#), [50–52](#), [64](#), [159](#)
 - build_bounds, [14](#), [45](#)
 - build_catchability, [14](#), [46](#)
 - build_catchability(), [61](#)
 - build_composition, [15](#), [46](#)
 - build_composition(), [61](#)
 - build_data, [16](#)
 - build_data(), [39](#), [40](#), [62](#), [116](#), [133](#), [159](#)
 - build_growth, [17](#), [46](#)
 - build_growth(), [55](#), [61](#)
 - build_hcr, [20](#), [22](#), [45](#), [63](#)
 - build_hcr(), [61](#)
 - build_hcr_map, [22](#)
 - build_M1, [23](#)
 - build_M1(), [61](#)
 - build_map, [22](#), [25](#), [45](#), [141](#)
 - build_map_catchability, [26](#)
 - build_map_debug, [26](#)
 - build_map_f_and_data_weights, [27](#)
 - build_map_fixed_natage, [27](#)
 - build_map_growth, [28](#)
 - build_map_m1, [28](#)
 - build_map_predation, [29](#)
 - build_map_recruitment, [29](#)
 - build_map_selectivity, [30](#)
 - build_osa_data(), [134](#)
 - build_params, [14](#), [25](#), [31](#), [45](#), [92](#)
 - build_selectivity, [31](#), [46](#)
 - build_selectivity(), [61](#)
 - build_srr, [33](#), [46](#)
 - build_srr(), [61](#)
 - check_mse, [36](#)
 - clean_data, [36](#)
 - clean_data(), [16](#), [17](#), [39](#), [40](#)
 - coef.Rceattle, [37](#)
 - combine_data, [37](#)
 - combine_data(), [17](#)
 - compare_sim, [38](#)
 - convergence_diagnostics, [38](#)
 - data_check, [160](#)
 - data_requirements, [39](#)
 - data_requirements(), [17](#), [133](#)
 - EBS_ms_run, [41](#)
 - EBS_ss_M_run, [41](#)
 - EBS_ss_run, [42](#)
 - fit_control, [42](#)
 - fit_control(), [46](#), [47](#), [54](#), [58](#), [132](#), [140](#), [143](#), [148](#), [150](#)
 - fit_mod, [44](#), [53](#), [125](#), [149](#)
 - fit_mod(), [6](#), [15–17](#), [30](#), [32](#), [37](#), [39](#), [42](#), [44](#), [53](#), [55](#), [56](#), [58](#), [59](#), [61](#), [62](#), [66](#), [67](#), [69](#)

- 70, 73, 75, 76, 78, 80, 84, 89, 90,
93–95, 97, 99, 103, 104, 106, 108,
109, 111, 112, 114, 116, 130–132,
135, 136, 138, 140, 148, 149, 154,
156, 158
- GeorgesBank3spp, 48
get_growth_matrix_r, 48
get_weight_at_age_r, 49
GOA2018SS, 50
GOAatf, 51
GOAatf2023, 51
GOAcod, 52
GOApollack, 52
GOAsafe2018, 52
GROWTH_FUNS, 18
GROWTH_LINKAGE_PARAMS, 18
- jitter, 53, 54
jitter(), 46, 117, 120, 132, 135
- linkage_spec, 55
linkage_spec(), 14, 16, 18, 24, 32, 35, 142
load_config, 58
load_config(), 46, 62, 144, 148, 159
load_mse, 58, 62, 74, 77, 85, 89, 91, 105, 110,
111, 114
logLik.Rceattle, 59
- mclapply, 66
model_average, 60, 152
model_average(), 154
model_config, 61
model_config(), 17, 47, 58, 116, 143, 144,
148, 159
mse_summary, 62
mse_summary(), 117
- NorthernRockfish2022, 64
- osa_diagnostics, 64
osa_diagnostics(), 68, 72, 124, 135
osa_residuals, 65
osa_residuals(), 65, 71, 72, 82, 93, 123,
124, 134, 135, 137, 138
- parameter_dictionary, 68
parameter_dictionary(), 70, 130, 136
parameter_index, 69
plot.Rceattle, 70
plot.rceattle_osa, 71
plot.rceattle_osa(), 68, 81, 82, 93, 124,
138
plot.Rceattle_profile, 72
plot_b_eaten, 73
plot_b_eaten(), 102, 130
plot_b_eaten_prop, 74
plot_b_eaten_prop(), 130
plot_biomass, 76
plot_biomass(), 130
plot_catch, 78
plot_catchability, 79
plot_comp, 81
plot_data, 82
plot_depletion, 84
plot_depletionSSB, 85
plot_diet_comp, 86
plot_diet_comp(), 86–88
plot_diet_comp1, 86
plot_diet_comp1(), 88
plot_diet_comp2, 87
plot_diet_comp2(), 86, 87
plot_exploitable_biomass, 88
plot_exploitable_biomass(), 130
plot_f, 90
plot_f(), 114, 130
plot_form, 92
plot_index, 92
plot_index(), 79
plot_index_residual, 93, 94, 95
plot_indexresidual, 94
plot_m2_at_age_prop, 96
plot_m2_at_age_prop(), 102, 130
plot_m_at_age, 95
plot_m_at_age(), 99, 130
plot_maturity, 98
plot_mortality, 99
plot_profile, 100
plot_profile(), 72, 127–129
plot_ration, 102
plot_ration(), 130
plot_recruitment, 104
plot_recruitment(), 130
plot_selectivity, 105
plot_selectivity(), 44, 130, 132
plot_selectivity_vs_maturity, 107
plot_ssb, 108
plot_ssb(), 130

plot_ssb_depletion, 85, 86, 110
 plot_ssb_depletion(), 114
 plot_stock_recruit, 112
 plot_timeseries, 113
 plot_timeseries(), 6, 130
 print.Rceattle, 115
 print.Rceattle_data, 116
 print.Rceattle_jitter, 117
 print.Rceattle_mse_summary, 117
 print.Rceattle_profile, 118, 126
 print.Rceattle_profile(), 102
 print.rceattle_report, 119
 print.Rceattle_retro, 119
 print.Rceattle_selftest, 120
 print.summary.Rceattle, 121
 prior_beta, 121
 prior_beta(), 121
 prior_gamma, 122
 prior_lognormal, 122
 prior_normal, 123
 process_residuals, 123
 process_residuals(), 68, 71, 138, 139
 profile(), 46, 127, 132
 profile.Rceattle, 124
 profile.Rceattle(), 72, 101, 102, 118, 127, 129
 profile_components, 127
 profile_components(), 101, 102

 quantity_dictionary, 129
 quantity_dictionary(), 135, 137, 155

 Rceattle, 81
 rceattle-plot-args, 130
 rceattle-refit-args, 132
 Rceattle_priors, 56
 rceattle_switch_maps, 30
 read_data, 45, 133, 160
 read_data(), 16, 17, 39, 62, 159
 rearrange_dat (rearrange_data), 133
 rearrange_data, 133
 remove_F, 134
 remove_F(), 46, 63
 rename_output, 134
 rename_output(), 6
 report_tables, 135
 report_tables(), 112, 119, 155
 residuals.Rceattle, 137
 residuals.Rceattle(), 66, 71, 81, 82, 94

 retrospective, 139
 retrospective(), 46, 75, 119, 132, 135
 reweight_comps, 13, 141
 reweight_comps(), 46
 rich.colors.short, 143
 run_config, 143
 run_config(), 46, 148
 run_mse, 36, 59, 62, 74, 77, 85, 89, 91, 105, 110, 111, 114, 144, 152
 run_mse(), 46

 sample_rec, 147
 sample_rec(), 46, 140, 145
 save_config, 148
 save_config(), 58, 62, 144, 159
 self_test, 38, 149
 self_test(), 46, 120, 132, 154
 set_phases, 151
 set_phases(), 158
 sim_mod, 149–151, 152
 sim_mod(), 154
 simulate.Rceattle, 153
 standard_output, 155
 standard_output(), 137
 stats::AIC(), 59
 stats::BIC(), 59
 stats::dbeta(), 121
 stats::dgamma(), 122
 stats::dlnorm(), 122
 stats::nlminb(), 43
 stats::residuals.glm(), 137
 stats::simulate(), 153, 154
 summary.Rceattle, 156
 summary.Rceattle(), 121
 summary.Rceattle_data
 (print.Rceattle_data), 116
 switch_check, 156, 160
 switch_check(), 40

 TMB::oneStepPredict(), 65, 66
 TMB::sdreport(), 43, 158
 TMBAIC, 157
 TMBphase, 157

 vcov.Rceattle, 158

 whamGrowthData, 159
 write_data, 159
 write_data(), 16, 62
 write_template, 160