r4Casal2

C.Marsh

2023-06-18

Contents

Chapter 1

Welcolme to r4Casal2

This book demonstrates functionality of the `r4Casal2` R package. This is an R package that works with the `Casal2` base R-library found here, although it is advised to use the R-library that is included with the Casal2 binary and usermanual you acquired. The `Casal2` base R-library is responsible for reading in output and interacting with Casal2 configuration files. The `r4Casal2` R package has been built for summarising and visualising objects read in from the base `Casal2` R-library.

All functions in this package should be documented using the `roxygen` syntax with input parameters available using the `?` query. For example `?get_fisheries`. To get a list of functions and general info on the package you can use `library(help="r4Casal2")` or see Section `??` for another list

```
library(r4Casal2)
library(Casal2)
library(knitr)
library(ggplot2)
library(dplyr)
library(reshape2)
library(tidyr)
```

The core functionality of `r4Casal2` are its accessor functions. These are functions that will return a specific object from a range of Casal2 objects in long format that are `ggplot`, `dplyr` friendly. Most accessors start with `get_` and should be self explanatory. There are some plotting functions, but I have found that I often want to custom ggplots and so mainly have custom plots. The accessors are coded to deal with three types of outpus. These are;

- `extract.mpd()` where Casal2 has been run with default report style. These objects are of class `casal2MPD` which are set by the Casal2 base function

- `extract.mpd()` where Casal2 has been run with tabular reports `casal2 --tabular` or `casal2 -t`. These objects are of class `casal2TAB` which are set by the Casal2 base function
- `list` this is a list of `casal2MPD` which is a useful format for comparing MPD runs see Section **??**

Chapter 2

A list of key functions in the `r4Casal2` package

2.1 Accessor functions

- `get_derived_quantities()` or for lazy people (like myself) `get_dqs()`. These will return all the derived quantities for a model output.
- `get_selectivities` will return a data frame with all the selectivity reports for a model output.
- `get_selectivities_by_year` will return a data frame with all the reports of type `selectivity_by_year` from a model output.
- `get_catchabilities` will return a data frame with all the catchability reports for a model output.
- `get_fisheries` will return a data frame with information from an `instantaneous_mortality` process for a model output.
- `get_BH_recruitment` will return a data frame with information from a `recruitment_beverton_holt` process for a model output.
- `get_abundance_observations` will return a data frame with information from an `abundance` or `biomass` observation for a model output.
- `get_composition_observations` will return a data frame with information from an `proportion_at_length`, `proportion_at_age`, `process_removals_by_age` and `process_removals_by_length` observation for a model output.
- `get_composition_mean_bin` will return a data frame with information from an `proportion_at_length`, `proportion_at_age`,

process_removals_by_age and process_removals_by_length summarised as the mean length or mean age.

- get_tag_recapture_observations will return a data frame with information from an tag_recapture_by_length_for_growth, tag_recapture_by_length and tag_recapture_by_age observation for a model output.
- get_partition will return a data frame with partition data from partition report.
- get_initial_partition will return a data frame with initial partition initialisation_partition report.
- get_profile Will return a data frame for a profile report.
- get_estimated_values Will return a data frame for a estimate_value report.
- get_transformed_parameters Will return a data frame for a parameter_transformations report.
- get_timevarying_parameters Will return a data frame for a time_varying report.
- get_simulated_age_resids Will reformat simulated data read in by the read.simulated.data function.
- get_projections will return a data frame of all projection reports from a model output.
- get_growth will return a data frame of all age_length report from a model output.
- get_covariance will return a data frame of all covariance_matrix report from a model output.

2.2 Other useful functions

- aggregate_objective_report This reformats an objective function report to be "similar" to CASALs output.
- create_simulation_reports This will create a range of @report.type=simulated_observation Casal2 reports that can help set up simulations. See Section **??** on why you want to do this.
- build_assessment_bookdown This will create a bookdown template for an assessment model MPD run.
- summarise_config Will summarize input files see Section **??**
- calculate_composition_stage_two_weights Calculates the stage-two weights using **?** TA1.8 method.

- `get_high_correlations` Returns index of parameters that have high correlations from MPD. This requires the Casal2 model to have reported the `correlation_matrix`
- `run_automatic_reweighting` Automatically apply iterative reweighting methods for a Casal2 model
- `extract_reweighted_mpds` extract all the reweighted mpds that are created by `run_automatic_reweighting`. Useful to then plot the effect of reweighting
- `error_value_table` Create a data.frame of all observations from a casal2 mpd run outlining likelihood type, observation type and error value by year and observation.
- `summarise_estimated_parameters` If a model reports `estimate_summary` this function will extract two data frames that can be used to assess starting values and estimated values along with prior assumptions.
- `plot_profile` Will plot profiles for reports that have been run with `casal2 -p` format.

Chapter 3

Summarise configuration inputs

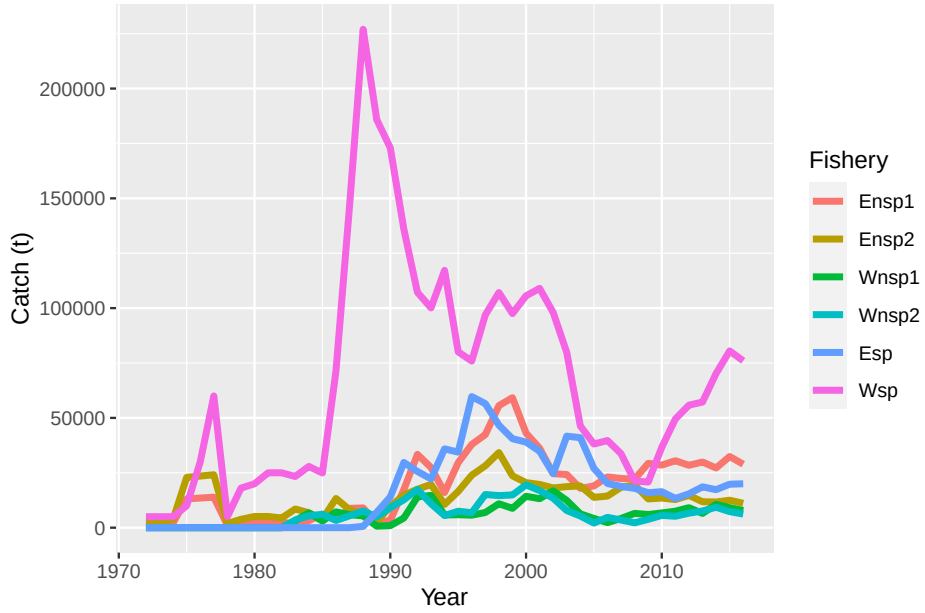
The `r4Casal2` has some functions that summarise a set of input files and returns a summary of the key model attributes. It can be difficult to know all the working parts in a Casal2 model. This is compouned when users often make tweaks during an assessment and so the initial assumptions will not correspond to the final assumptions. The key function is `summarise_config`

3.1 Example files

```
config_dir = system.file("extdata", "TestModelComplex", package = "r4Casal2", mustWork = TRUE)
## This function is the key function will read a Casal config file and report useful information
## should be used when describing model structures and assumptions
## as well as validation.
summary = summarise_config(config_dir, config_file = "config.csl2", quiet = T)
names(summary)
```

```
##  [1] "category_df"           "estimate_df"           "full_category_df"
##  [4] "method_df"             "catch_df"              "time_step_df"
##  [7] "time_step_df_just_lab" "obs_year_df"           "model_years"
## [10] "model_ages"            "model_length_bins"     "M_by_category"
## [13] "model_block"
```

```
ggplot(summary$catch_df, aes(x = year, y = catch, col = fishery)) +
  geom_line(size = 1.5) +
  labs(x = "Year", y = "Catch (t)", col = "Fishery")
```




```
ggplot(summary$obs_year_df, aes(x = year, y = observation, col = observation, size = a
  geom_point() +
  guides(colour = "none", size = "none")
```

```
## Warning: Removed 509 rows containing missing values (`geom_point()`).
```