Operating Model Specification and Software Upgrade Project

Details of progress since the Tokyo workshop

Darcy Webber

24 June 2024
Seattle

QUANTIFISH
Quantitative Fisheries Science

sbt

Prior to June 2024 OMMP

Ideally the following changes would be implemented before the June 2024 meeting so that they can be evaluated by the OMMP working group:

- ✓ Check age likelihood again (small difference in likelihood)
- ✓ Lump the LL3 and LL4 fisheries and cohort slice and treat as removals
- ❑ Specify the LL1, LL2, Australian and, Indonesian selectivity using GMRF (in progress)
- ❑ Review this years sensitivities and robustness tests and make sure all the code to do these is available
- ❑ Can filter out some of the POPs in get_data that result in likelihood values that are not used in the estimation
- ✓ Name the grid runs in run_grid
- ❑ Implement grid sampling in the R code (in progress)
- ✓ Re-code tag likelihoods to remove the H* parameters (harvest rate for mixing periods) and add the output for the PSIS-LOO diagnostic
- ✓ Implement the Dirichlet-multinomial likelihood for composition data
- ❑ Code prior distributions in short-hand (following R format; e.g., dnorm()) (in progress)
- ❑ Incorporate the age-uncertainty for the adult part of the POP calculations (the possible ages given length) (in progress)
- ❑ Update website to improve documentation (e.g., add vignette on "how to run the grid") (in progress)
- ❑ Evaluate if other "Stan" R packages (e.g., adnuts) can be used to help evaluate model runs.

Contents

Different model versions

Different model versions

- V1:
  - Matches the ADMB model
  - Not pdH
- V2:
  - Updates the tag likelihood (get_tag_like)
  - Adds the Dirichlet-multinomial distribution function and uses this in get_tag_like
  - Adds the student-t distribution function and uses it as prior for sigma_r
  - Adds option to fit to LFs/AFs using old method (which is wrong), multinomial, Dirichlet, or Dirichlet-multinomial
  - Adds cohort slicing of LFs
  - Adds option to specify direct removal of the catch
  - Not pdH
- V3:
  - working on GMRF selectivity - want to retain old selectivity at the same time

Cohort slicing LFs

Cohort slicing LFs

In the ADMB OM, time varying selectivity at age was estimated for the LL3 fishery, time invariant selectivity at age was estimated for the LL4 fishery (figure to right), and the LFs for these two fisheries were fitted to separately (see the next two slides).

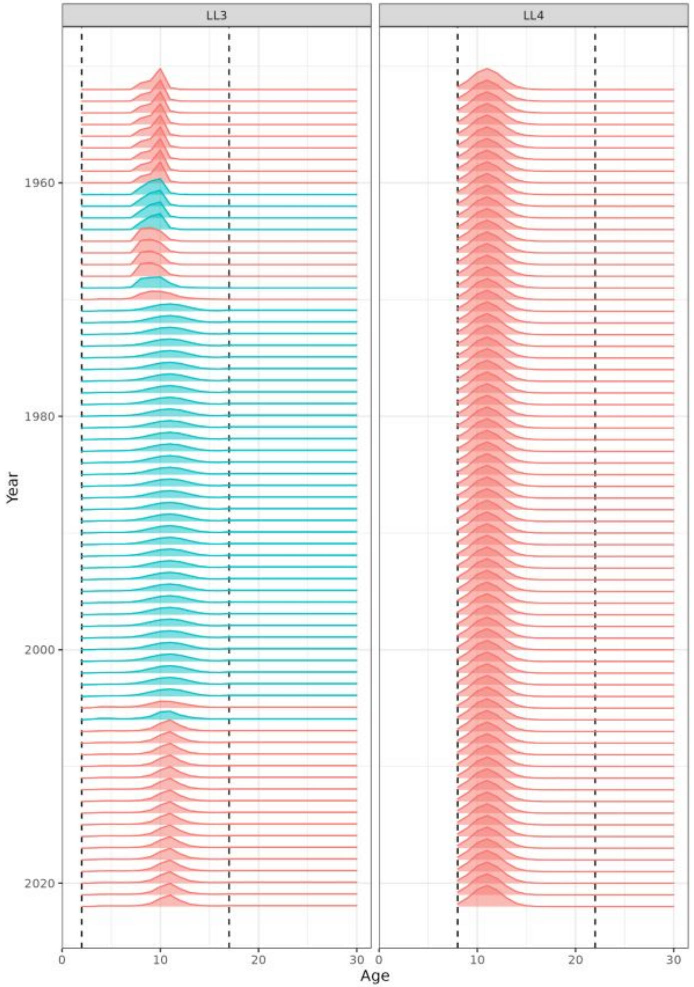


Figure 3: selectivity estimated by the previous OM for LL3 and LL4. The colours represent selectivity periods of that are assumed to be the same (i.e., there are nine periods of different selectivity for LL3 and LL4 selectivity is time invariant).

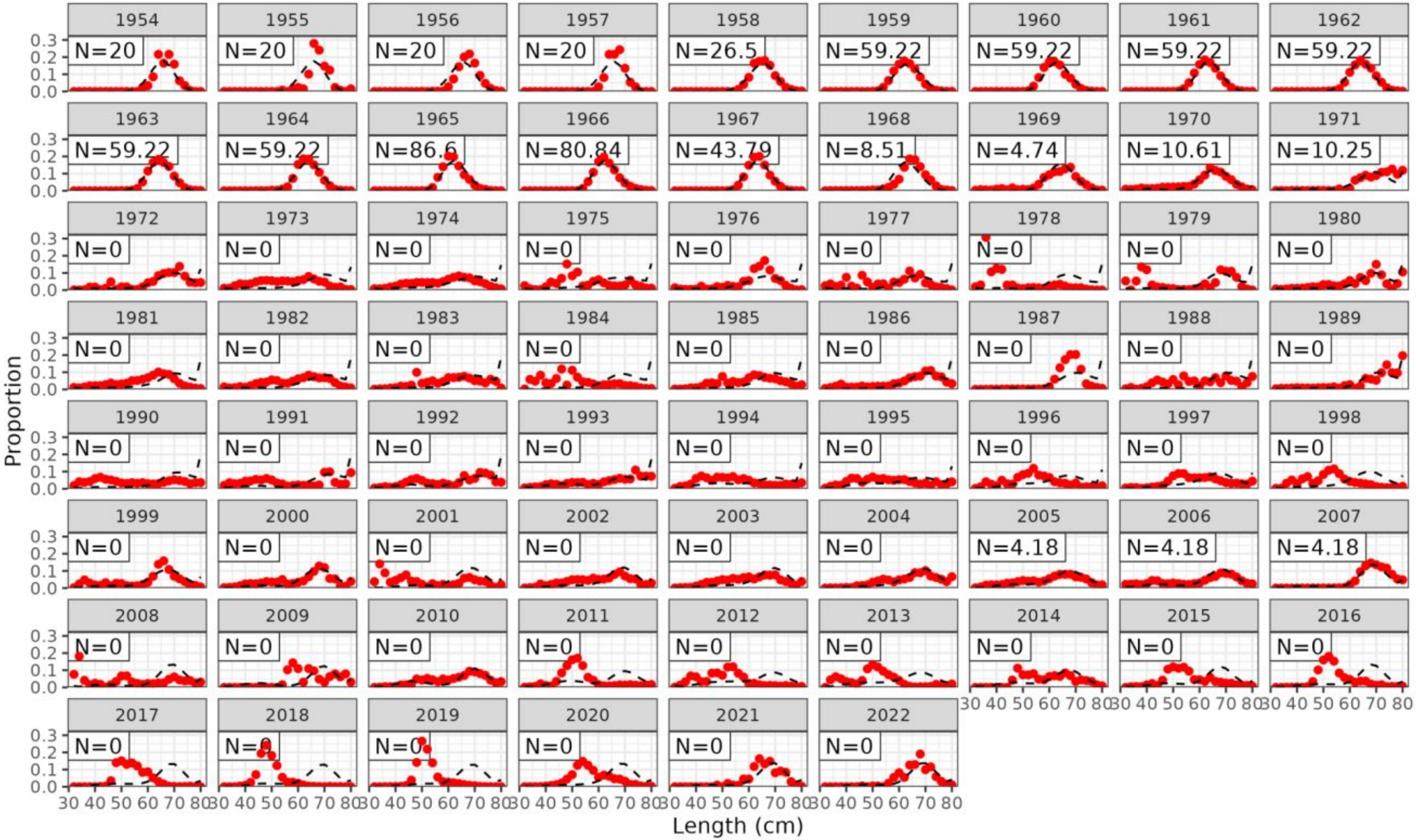


Figure 4: Observed LFs (red) and model fit (black) to the LL3 LFs. The effective sample size (N) is also shown for each year.

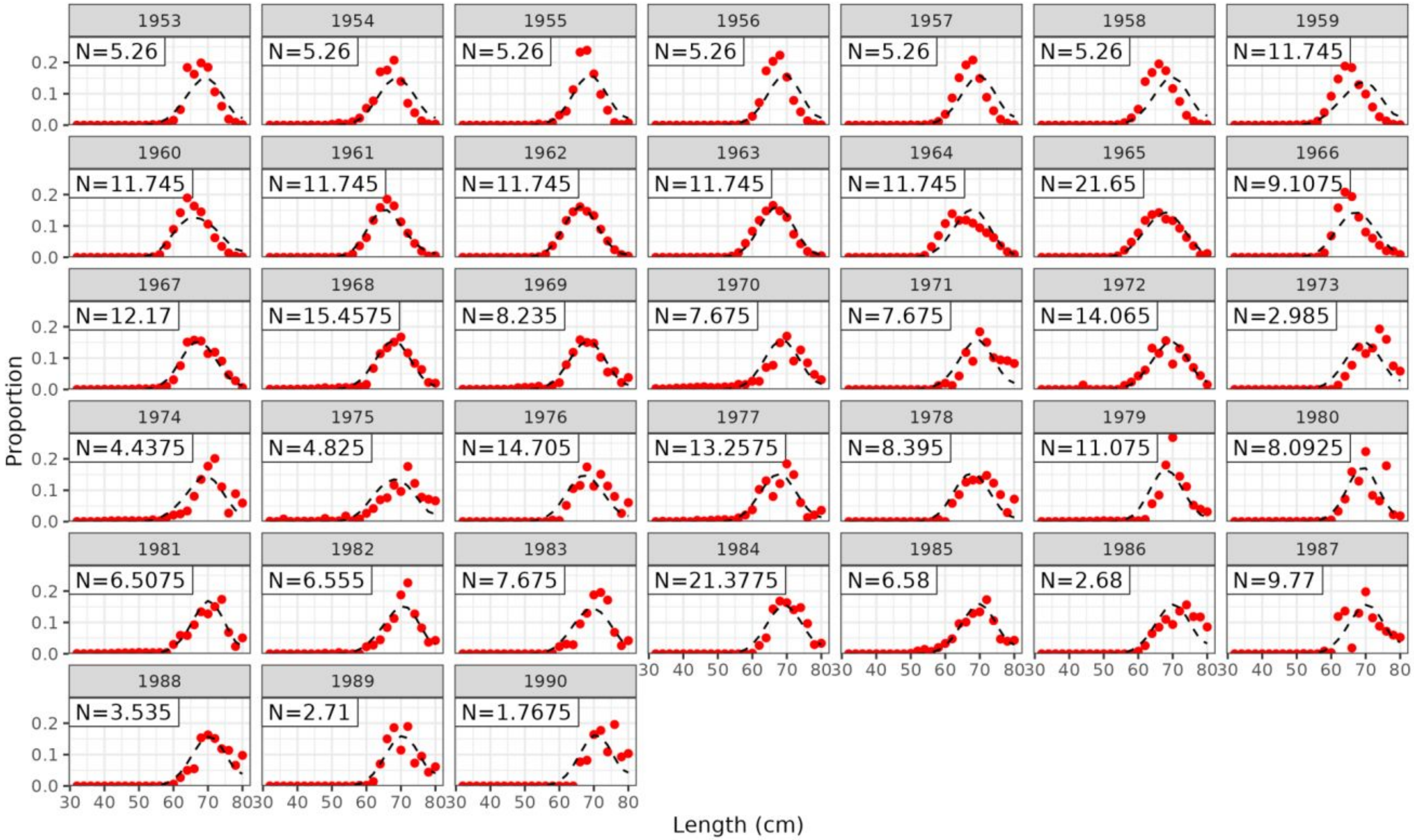


Figure 5: Observed LFs (red) and model fit (black) to the LL4 LFs. The effective sample size (N) is also shown for each year.

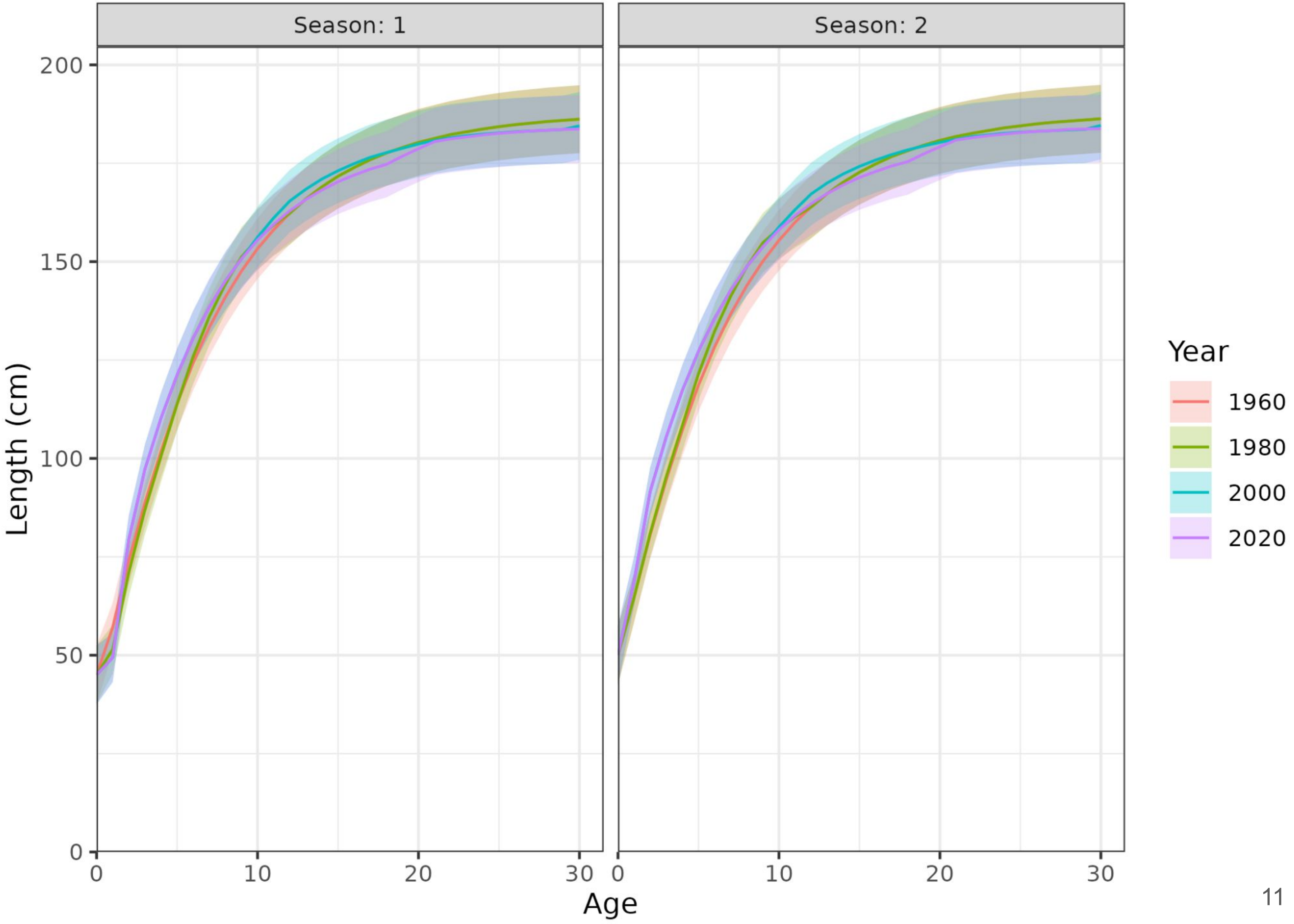
Cohort slicing LFs

Code was written to cohort slice LFs into AFs for the LL3 and LL4 fisheries. This is done outside of the TMB model code (i.e., in R code using the function `get_sliced_afs` which is embedded in the function `get_data`). This code actually slices all four longline fisheries (LL1, LL2, LL3, LL4), but the user can choose to fit to the original LFs or the sliced AFs for each fishery.

In short, the process of cohort slicing for each LF involves taking the mean length at age for each year and season for each LF ($l_{y,s,a}$), finding the midpoints between each length at age (and appending zero and infinity at start and end), then cutting the LFs at these midpoints (Figure 6, Figure 7).

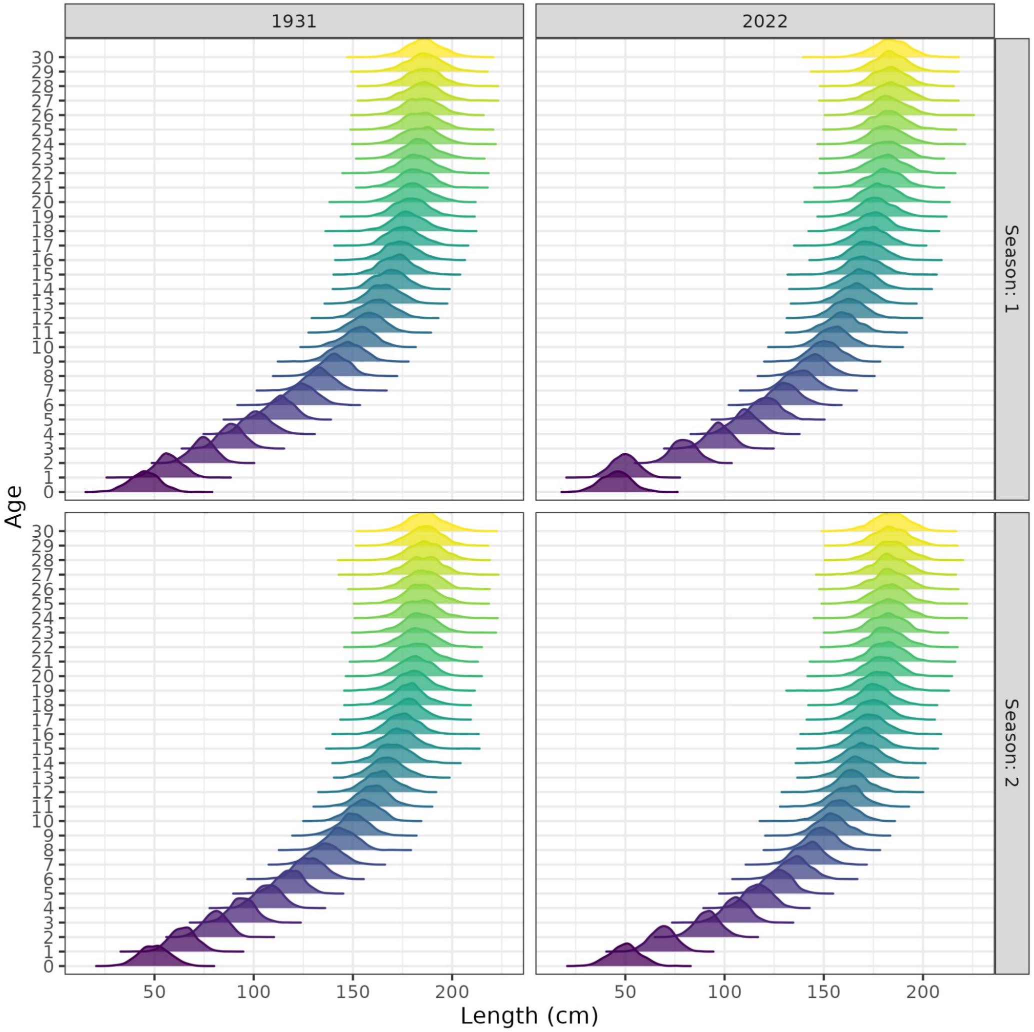
Cohort slicing LFs

Note that this is just a subset of years for each fishery.



Cohort slicing LFs

Note that this is just a subset of years for each fishery.



Cohort slicing LFs

Note that this is just a subset of years for each fishery.

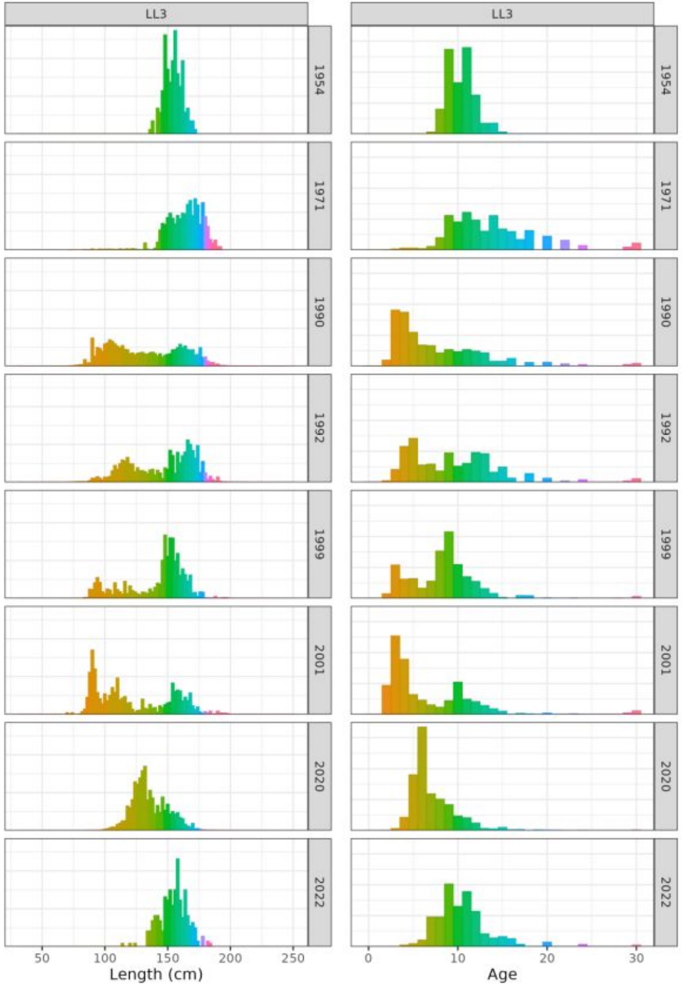


Figure 6: original LFs for a subset of years (left) and sliced LFs for those same years (right) for the LL3 fishery. Each colour represents an age.

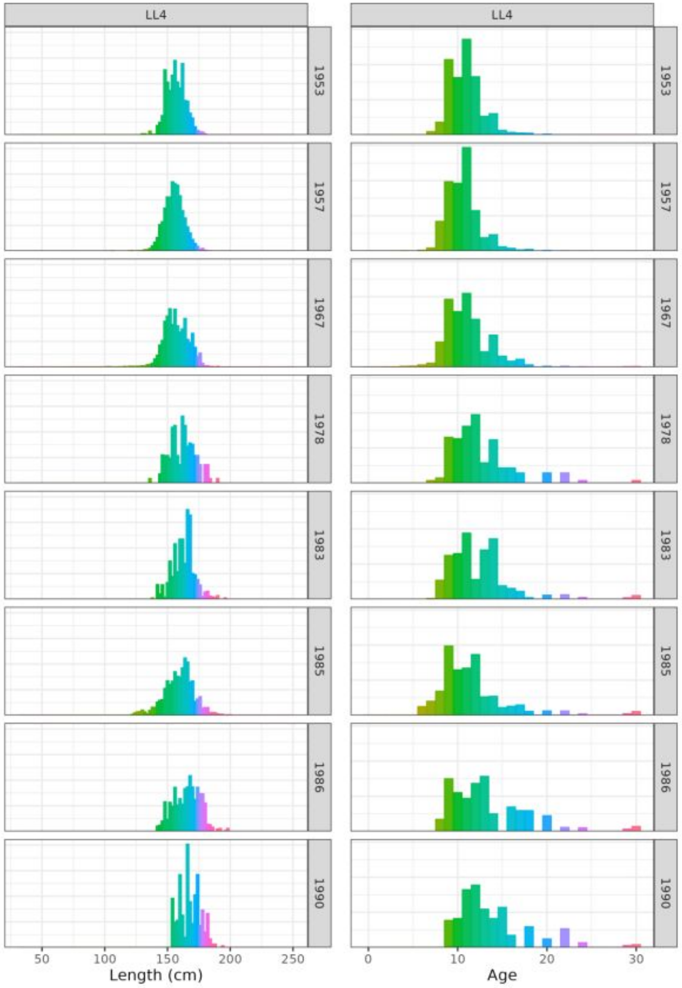


Figure 7: original LFs for a subset of years (left) and sliced LFs for those same years (right) for the LL4 fishery. Each colour represents an age.

Direct removal of catches

Direct removal of catches

- Typically, the catch for each year (y), season (s), and fishery (f, see next slide for other variable definitions) is removed by estimating selectivity ogives and using the process outlined in Table 2 on slide 18.
- The *sbt* code has been re-structured so that direct removals can optionally be specified for any fishery (all seasons and all years for the selected fishery).
- When specifying direct removals, the catch is removed from the model using the observed catch (tonnes) for each fishery ($C_{y,s,f}$) which is split proportionally by the AF or the sliced AF ($A_{y,s,f,a}$) for each fishery.

Direct removal of catches

Variable	Class	Description
a	Dimension	Age
y		Year
s		Season
f		Fishery
$N_{y,s,a}$	Derived quantity	Numbers at age in the population
$w_{y,f,a}$		Weight (tonnes) at age for each fishery
$S_{y,f,a}$		Selectivity at age for each fishery
M_a		Natural mortality at age
$H_{y,s,f,a}$		Harvest rate at age for each fishery
$C^N_{y,s,f,a}$		Catch (numbers) at age for each fishery
$C_{y,s,f}$	Covariate	Catch (tonnes) for each fishery
$A_{y,s,f,a}$		Proportion at age derived from an LF (i.e., cohort sliced LF) for each fishery

Direct removal of catches

Standard removals	Direct removals
$$U_{y,s,f,a} = \frac{C_{y,s,f}}{\sum_a N_{y,s,a} S_{y,f,a} w_{y,f,a}}$$	Define the catch biomass (tonnes) as $$C_{y,s,f} = X_{y,s,f} \sum_a A_{y,s,f,a} w_{y,f,a}$$ where $X_{y,s,f}$ is the catch in numbers obtained by dividing the catch in biomass by the average weight: $$X_{y,s,f} = \frac{C_{y,s,f}}{\sum_a A_{y,s,f,a} w_{y,f,a}}$$ The catch at age in numbers can be calculated as $$C^N_{y,s,f,a} = X_{y,s,f} A_{y,s,f,a}$$
$$H_{y,s,f,a} = U_{y,s,f,a} S_{y,f,a} \frac{C_{y,s,f} S_{y,f,a}}{\sum_a N_{y,s,a} S_{y,f,a} w_{y,f,a}}$$	$$H_{y,s,f,a} = \frac{C^N_{y,s,f,a}}{N_{y,s,a}} = \frac{X_{y,s,f} A_{y,s,f,a}}{N_{y,s,a}} = \frac{C_{y,s,f} A_{y,s,f,a}}{N_{y,s,a} \sum_a A_{y,s,f,a} w_{y,f,a}}$$

The catch from all fisheries is removed from the population using

$$N_{y,s+1,a} = N_{y,s,a} \left(1 - \sum_f H_{y,s,f,a} \right) e^{-0.5 M_a}$$

The catch for deriving LFs and AFs is calculated as

$$C^N_{y,s,f,a} = H_{y,s,f,a} N_{y,s,a}$$

And the catch biomass is

$$C'_{y,s,f} = H_{y,s,f,a} N_{y,s,a} w_{y,f,a}$$

Model input

```
Data <- list(last_yr = 2022, age_increase_M = 25,
             length_m50 = 150, length_m95 = 180,
             catch_UR_on = 0, catch_surf_case = 1, catch_LL1_case = 1,
             scenarios_surf = scenarios_surface, scenarios_LL1 = scenarios_LL1,
             sel_min_age_f = c(2, 2, 2, 8, 6, 0),
             sel_max_age_f = c(17, 9, 17, 22, 25, 7),
             sel_end_f = c(1, 0, 1, 1, 1, 0),
             sel_change_sd_fy = t(as.matrix(sel_change_sd[,-1])),
             sel_smooth_sd_f = lr$sel.smooth.sd,
             removal_switch = c(0, 0, 0, 0, 0, 0), # 0=standard removals, 1=direct removals
             pop_switch = 1,
             hsp_switch = 1, hsp_false_negative = 0.7467647,
             gt_switch = 1,
             cpue_switch = 1, cpue_a1 = 5, cpue_a2 = 17,
             aerial_switch = 4, aerial_tau = 0.3,
             troll_switch = 1,
             af_switch = 3, # 0=multinomial, 1=Dirichlet, 2=Dirichlet-multinomial, 3=old
             lf_switch = 3, lf_minbin = c(1, 1, 1, 11),
             tag_switch = 1, tag_var_factor = 1.82)
```

Direct removal of catches

An example of direct removal of the catch is provided below. In this example:

- the LL3 catch is all removed because there is an LF associated with the catch in every year, but
- the LL4 catch is not all removed because in some years there is catch but no LF (see figure to right).

Thus, as it is currently coded, when no AF or LF is available for a fishery in a year then the catch collapses to zero even if there is catch in that year/season/fishery. This can be amended by combining the LL3 and LL4 fisheries (this code change will be done in future updates to the TMB code).

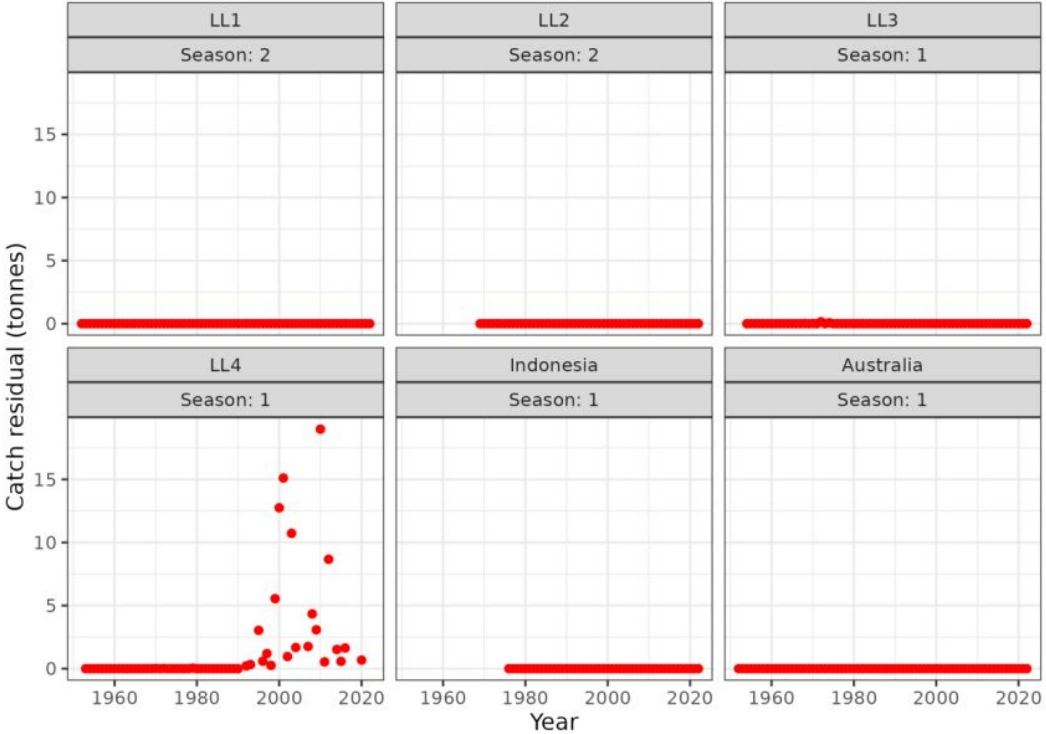


Figure 8: Catch residuals (input catch minus output catch, tonnes).

Direct removal of catches

Further to the example above, the predicted LFs for the LL3 and LL4 fisheries can be derived.

Note that the model predicted LFs do not match the observed LFs exactly because the catches are modified by the age-length-key (to convert from catch at age to catch at length) which distorts the predictions a little (Figure 9, Figure 10).

```
obs = lf_obs.row(i);
catch_a = catch_pred_fya(f, y);
pred = (alk_la * catch_a) / sum(catch_a);
```

However, when a fishery with AFs is specified to use as direct removals then the predicted AFs match the observed AFs exactly (Figure 11).

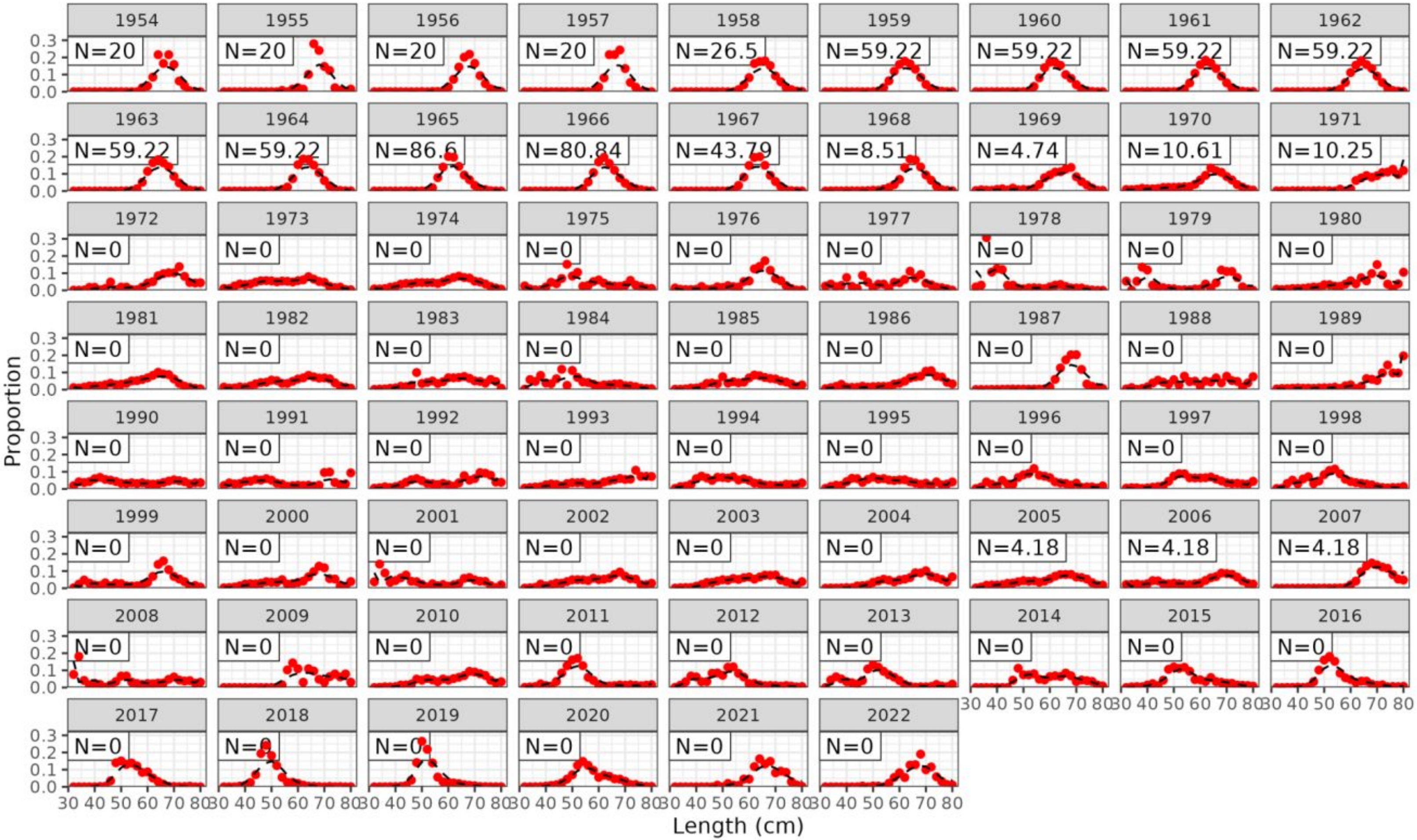


Figure 9: Observed LFs (red) and model fit (black) to the LL3 when treating as direct removals.

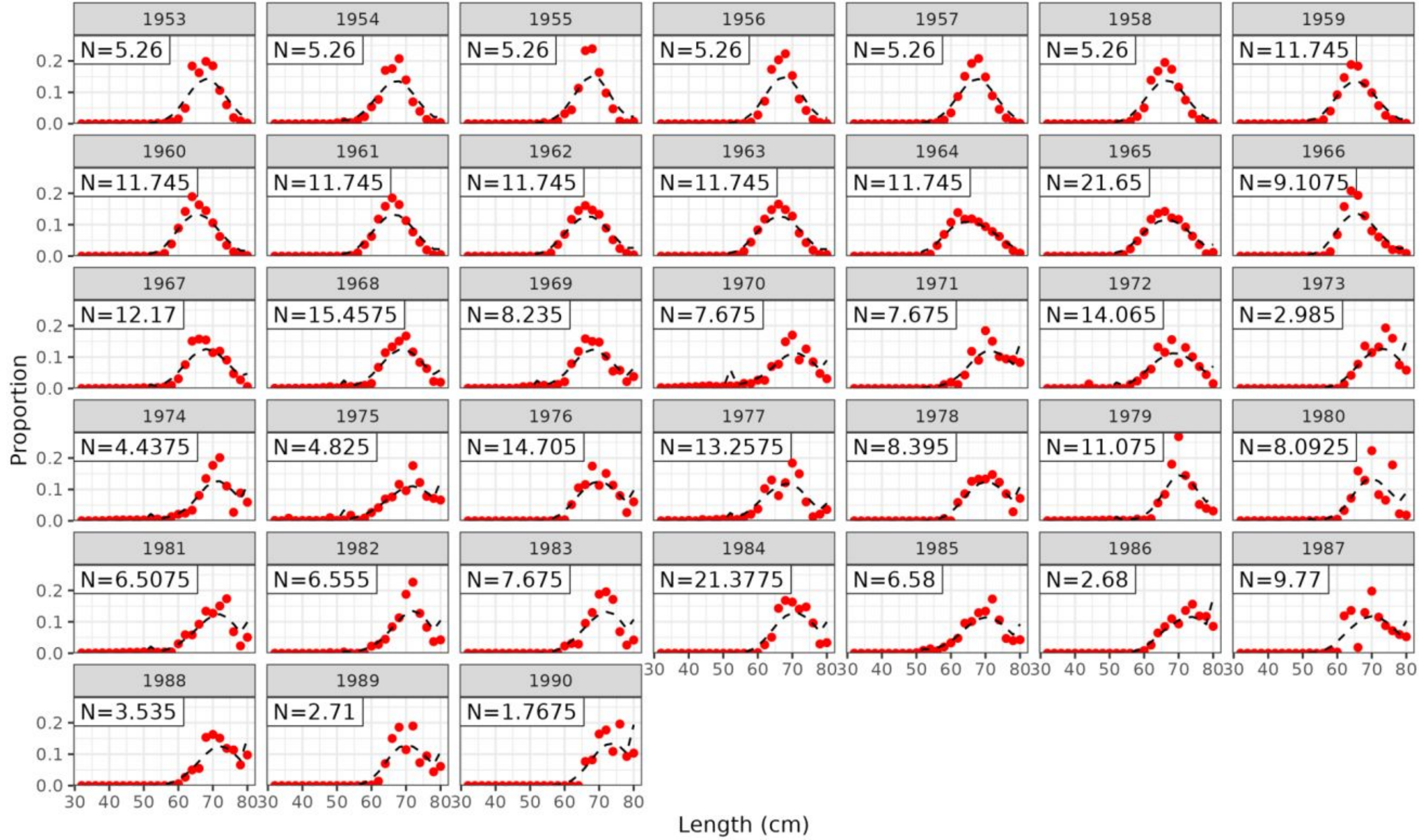


Figure *10*: Observed LFs (red) and model fit (black) to the LL4 when treated as direct removals.

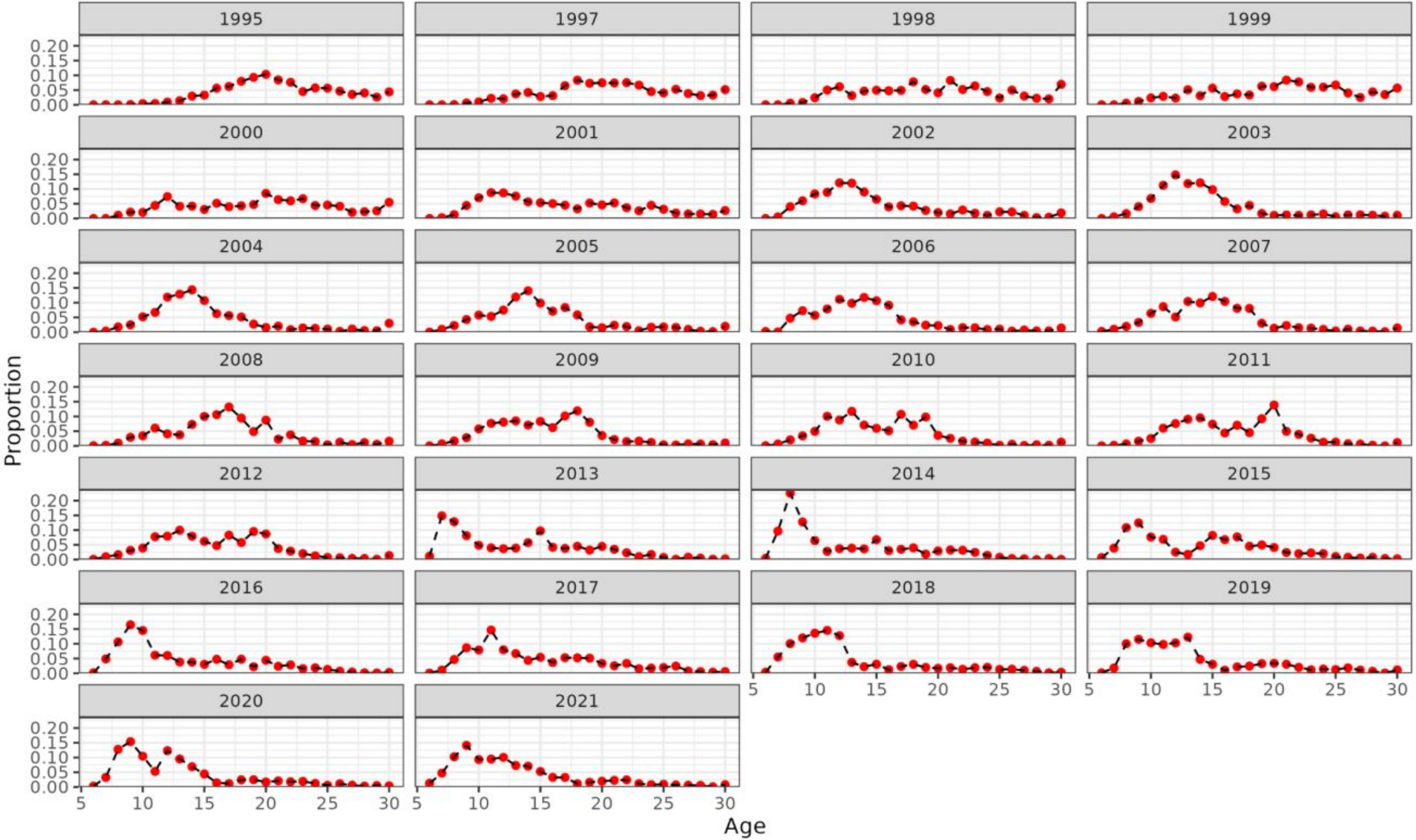


Figure 11: Observed AFs (red) and model fit (black) to the Indonesian fishery when treated as direct removals.

sbt

Changes to likelihoods

Changes to likelihoods

Both the tag and POP likelihoods have been updated - see paper for further details.

R format likelihoods/priors

Code prior distributions in short-hand (following R format; e.g., dnorm()) to improve readability of the code:

- ddm - Dirichlet-multinomial density, I coded this in TMB, used in get_tag_like and an option for fitting to AFs/LFs
- ddirichlet - Dirichlet density, I coded this in TMB, also a version in OSA_multivariate_dists-main/distr.hpp (not yet used), an option for fitting to AFs/LFs
- dmultinom - this is available in TMB, an option for fitting to AFs/LFs
- dlnorm - this is not available in TMB and had to be coded, used in get_cpue_like
- dstudent = I coded a different version of this in TMB, used as prior for sigma_r

Model input

```
Data <- list(last_yr = 2022, age_increase_M = 25,
             length_m50 = 150, length_m95 = 180,
             catch_UR_on = 0, catch_surf_case = 1, catch_LL1_case = 1,
             scenarios_surf = scenarios_surface, scenarios_LL1 = scenarios_LL1,
             sel_min_age_f = c(2, 2, 2, 8, 6, 0),
             sel_max_age_f = c(17, 9, 17, 22, 25, 7),
             sel_end_f = c(1, 0, 1, 1, 1, 0),
             sel_change_sd_fy = t(as.matrix(sel_change_sd[,-1])),
             sel_smooth_sd_f = lr$sel.smooth.sd,
             removal_switch = c(0, 0, 0, 0, 0, 0), # 0=standard removals, 1=direct removals
             pop_switch = 1,
             hsp_switch = 1, hsp_false_negative = 0.7467647,
             gt_switch = 1,
             cpue_switch = 1, cpue_a1 = 5, cpue_a2 = 17,
             aerial_switch = 4, aerial_tau = 0.3,
             troll_switch = 1,
             af_switch = 3, # 0=multinomial, 1=Dirichlet, 2=Dirichlet-multinomial, 3=old
             lf_switch = 3, lf_minbin = c(1, 1, 1, 11),
             tag_switch = 1, tag_var_factor = 1.82)
```

Age/length composition likelihoods

Will change the get_age_like TMB code:

```
// multinomial
obs += Type(1e-6);
pred += Type(1e-6);
lp(i) -= af_n(i) * (obs * log(pred)).sum();
lp(i) += af_n(i) * (obs * log(obs)).sum();
```

to:

Age/length composition likelihoods

```
if (af_switch == 0) {
   obs *= af_n(i);
   lp(i) -= dmultinom(obs, pred, true);
 }
 if (af_switch == 1) {
   obs += Type(1e-6);
   obs /= sum(obs);
   pred *= af_n(i) * exp(par_log_af_alpha(f - 4));
   lp(i) -= ddirichlet(obs, pred, true);
 }
 if (af_switch == 2) {
   obs *= af_n(i);
   pred *= exp(par_log_af_alpha(f - 4));
   lp(i) -= ddm(obs, pred, true);
 }
 if (af_switch == 3) {
   obs += Type(1e-6);
   pred += Type(1e-6);
   lp(i) -= af_n(i) * (obs * log(pred)).sum();
   lp(i) += af_n(i) * (obs * log(obs)).sum();
 }
```

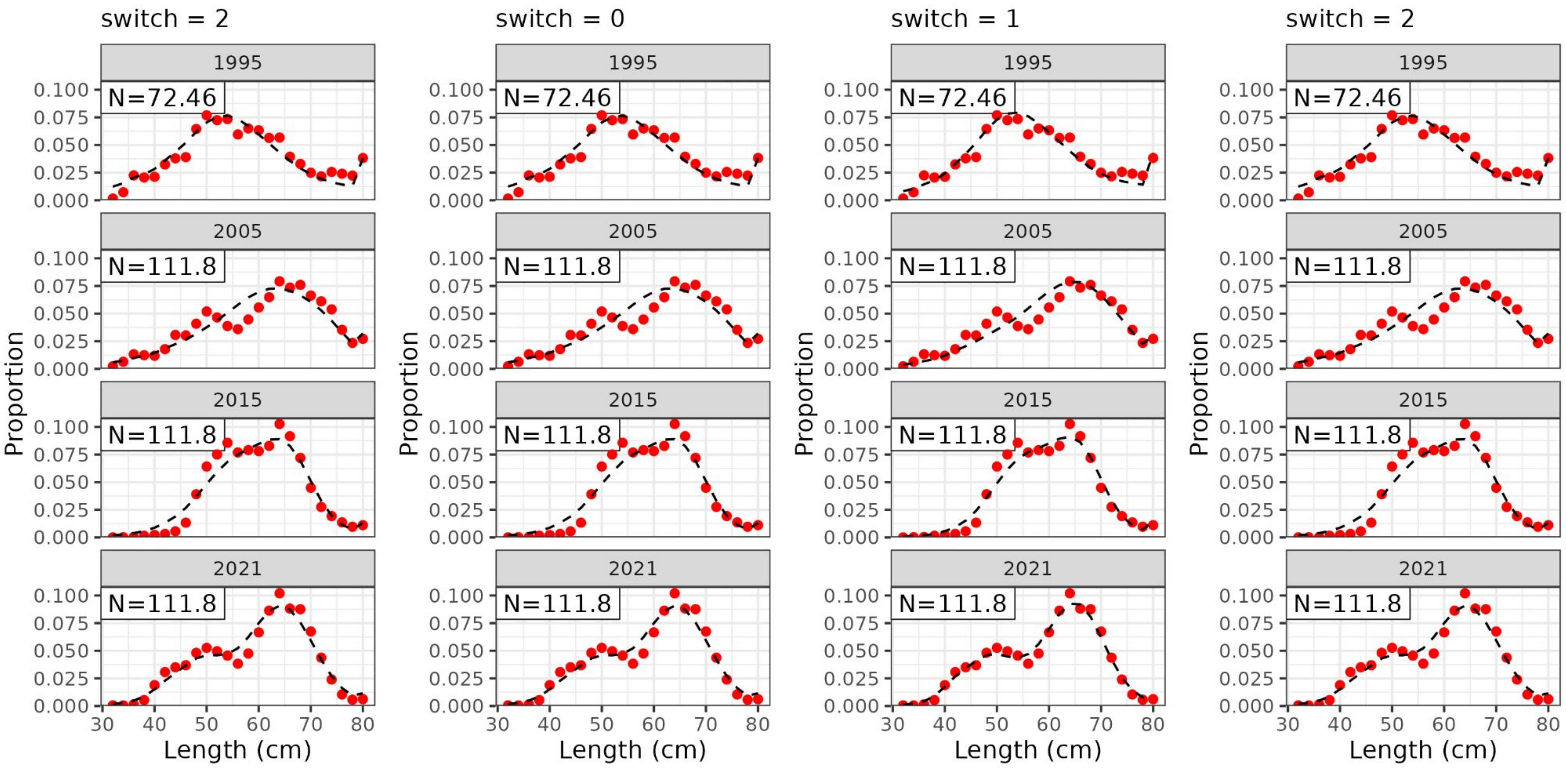
Age/length composition likelihoods

alpha0 is an estimated (or fixed) scaling parameter and:

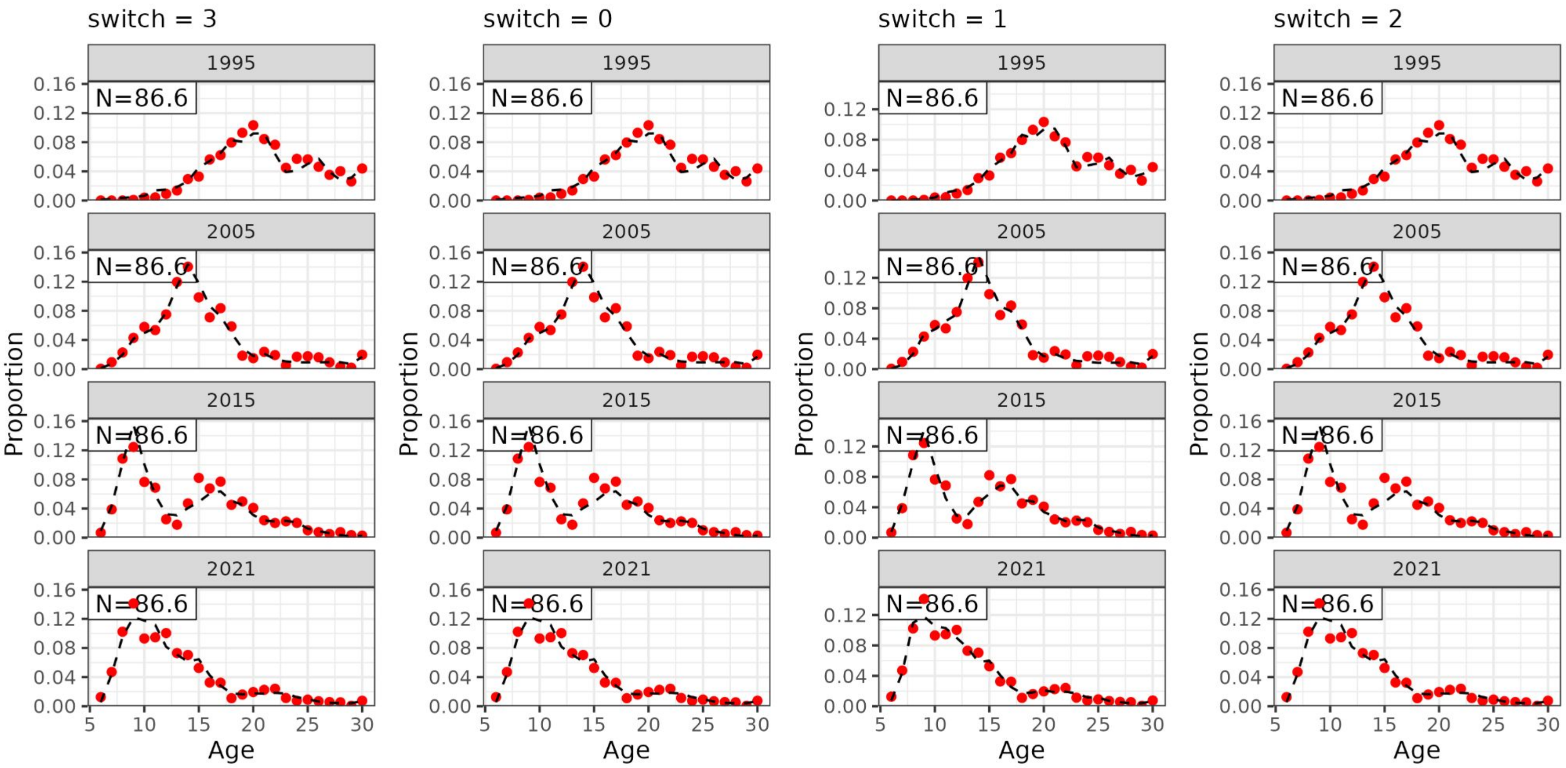
```
template <class Type>
Type ddm(vector<Type> x, vector<Type> alpha, int give_log) {
  Type sum_alpha = alpha.sum();
  Type logres = lgamma(sum_alpha) - lgamma(x.sum() + sum_alpha) +
    lgamma(x.sum() + Type(1.0)) - lgamma(vector<Type>(x + Type(1.0))).sum() + // constant, may
omit
    lgamma(vector<Type>(x + alpha)).sum() - lgamma(alpha).sum();
  if (give_log) return logres;
  else return exp(logres);
}
```

The ddm function is used in the new tag recapture likelihood also.

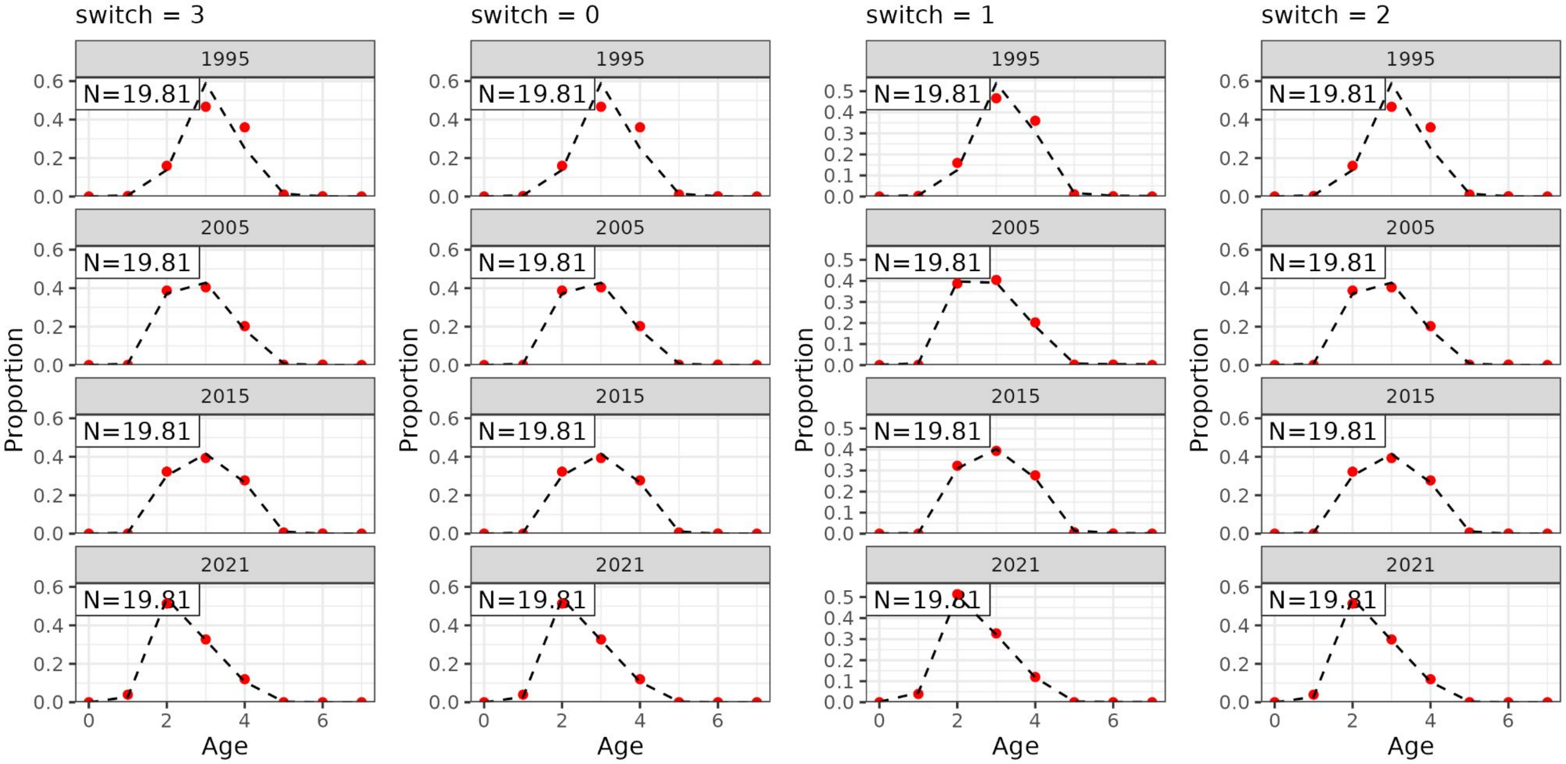
Length composition likelihoods - LL1



Age composition likelihoods - Indo



Age composition likelihoods - Aussie



Selectivity overhaul

Fisheries Research 266 (2023) 106755

Contents lists available at ScienceDirect

Fisheries Research

journal homepage: www.elsevier.com/locate/fishres

Short Communication

Unlocking the triad of age, year, and cohort effects for stock assessment: Demonstration of a computationally efficient and reproducible framework using weight-at-age

Matthew LH. Cheng [a,*], James T. Thorson [b], James N. Ianelli [c], Curry J. Cunningham [a]

[a] *Department of Fisheries at Lena Point, College of Fisheries and Ocean Sciences, University of Alaska Fairbanks, 17101 Point Lena Loop Rd, Juneau, AK 99801, USA*

[b] *Habitat and Ecological Processes Research Program, Alaska Fisheries Science Center, National Marine Fisheries Service, NOAA, 7600 Sand Point Way N.E., Seattle, WA 98115, USA*

[c] *Resource Ecology and Fisheries Management Division, Alaska Fisheries Science Center, NOAA, 7600 Sand Point Way N.E., Building 4, Seattle, WA 98115, USA*

Assuming time invariant selectivity (for catches at least) is wrong! I have implemented this in the SBT model.

The fisheries

Fishery	Type	Min age	Max age	N ages	First year	Last year	N years	Notes
LL1	LF	2	17	16	1952	2022	71	GMRF
LL2	LF	2	9	8	1997	2022	26	GMRF
LL3	LF	2	17	16	1954	2022	69	To be combined, cohort sliced, and treated as direct removals
LL4	LF	8	22	15	1953	1990	38	
Indo	AF	6	25	20	1995	2021	27	GMRF with cohort effect
Aussie	AF	0	7	8	1964	2022	59	GMRF with cohort effect

Selectivity inputs

Retain

- DATA_IVECTOR(sel_min_age_f);
- DATA_IVECTOR(sel_max_age_f);
- DATA_IVECTOR(sel_end_f);
- DATA_IMATRIX(sel_change_year_fy)

Drop

- DATA_MATRIX(sel_change_sd_fy);
- DATA_VECTOR(sel_smooth_sd_f);
- PARAMETER_VECTOR(par_sels_init_i);
- PARAMETER_VECTOR(par_sels_change_i);

Add

- DATA_MATRIX(Index_ay);
- DATA_INTEGER(sel_switch); // Variance parameterization of Precision Matrix 0=Conditional, 1=Marginal
- PARAMETER_VECTOR(par_rho_a); // Correlation by age
- PARAMETER_VECTOR(par_rho_y); // Correlation by year
- PARAMETER_VECTOR(par_rho_c); // Correlation by cohort
- PARAMETER_VECTOR(par_log_sigma2); // Variance of the GMRF process
- PARAMETER_MATRIX(par_log_sel_ay); // Random effects selectivity array

Positive definite Hessian (pdH) issues

pdH issues

I have not managed to get pdH for any of the model versions.

It seems like B0 and many of the recruitment deviates are the culprits.

There are also divergent transitions when running the MCMC.

```
ce <- TMBhelper::check_estimability(obj = obj)
> ce[[4]] %>% filter(Param_check != "OK")
         Param         MLE Param_chec
1  par_log_B0 16.188096890         Ba
2  par_rdev_y  0.241066963         Ba
3  par_rdev_y  0.010828190         Ba
4  par_rdev_y -0.099010364         Ba
5  par_rdev_y -0.721471259         Ba
6  par_rdev_y -0.894094926         Ba
7  par_rdev_y -1.247567051         Ba
8  par_rdev_y -0.906158525         Ba
9  par_rdev_y  0.047462311         Ba
10 par_rdev_y  0.509574893         Ba
11 par_rdev_y  0.367630732         Ba
12 par_rdev_y  0.017139926         Ba
13 par_rdev_y  0.014245081         Ba
14 par_rdev_y  0.125911155         Ba
15 par_rdev_y  0.492768309         Ba
16 par_rdev_y  0.515632636         Ba
17 par_rdev_y  0.186693565         Ba
18 par_rdev_y  0.655020127         Ba
19 par_rdev_y -0.005032565         Ba
20 par_rdev_y -0.055016461         Ba
21 par_rdev_y  0.149471082         Ba
22 par_rdev_y -0.379260918         Ba
23 par_rdev_y -0.642502473         Ba
24 par_rdev_y  0.290348950         Ba
25 par_rdev_y -0.020684511         Ba
26 par_rdev_y  0.105792142         Ba
27 par_rdev_y  0.390174332         Ba
28 par_rdev_y  0.402433542         Ba
29 par_rdev_y -0.105798458         Ba
```