LIME: An R package for simulation and estimation using length data to account for variable fishing mortality and recruitment

Merrill Rudd

2017-09-13

Introduction

This package contains functions to run the Length-based Integrated Mixed Effects (LIME) fisheries stock assessment method. The LIME package can be used in two ways: 1) simulating the expected age-structured population dynamics, length composition, catch data, and an abundance index using LIME and 2) fitting to empirical length data at a minimum and any available catch and/or abundance index data to provide an estimate of the spawning potential ratio (SPR).

LIME has been developed for data-limited fisheries, where few data are available other than a representative sample of the size structure of the vulnerable portion of the population (i.e., the catch) and an understanding of the life history of the species. LIME relaxes the equilibrium assumptions of other length-based methods by estimating annual fishing mortality and recruitment variation (among other parameters), deriving annual recruitment as a random effect.

See Rudd and Thorson (2017) in the reference list for full details of the model, including simulation testing to evaluate performance across life history types, population variability scenarios, and data availability scenarios, as well as violations of the model assumptions.

Bug Reports

Please alert me to any bugs or issues by using GitHub.

Comments and suggestions for additional features are welcome and we can discuss via email at mbrudd@uw.edu. GitHub pull requests are also welcome.

Finally, please make sure you understand the data and the biological parameters (and how the model treats these) and critically evaluate any output of LIME.

First Steps

Installing the Package

You can install the development version of the package from GitHub using the `devtools` package:

```
install.packages("devtools", repos='http://cran.us.r-project.org')
```

```
devtools::install_github("merrillrudd/LIME")
```

LIME also requires the package `TMBhelper`, which is not currently available on CRAN. You'll need to make sure this package is installed as well:

```
devtools::install_github("kaskr/TMB_contrib_R/TMBhelper")
```

Load the Package

```
library(LIME)
```

Simulation

The LIME package can be used to generate the expected size composition, other age-structured outputs (e.g. abundance-at-age, catch-at-age, annual recruitment), SPR, and MSY for a given set of biological parameters and fishing mortality and recruitment patterns.

Specify biological inputs and starting values

The LIME package simulation and estimation features require the biological inputs and starting values for selectivity parameters in a list. Using `create_lh_list` will make sure all required elements are included within the list and in the required format. The `create_lh_list` function requires inputs for some parameters and includes default values for others.

Minimum inputs for `create_lh_list`

The user is required to input the following parameters to `create_lh_list` for simulation and estimation:

Minium inputs: Biology

- von Bertalanffy asymptotic length (`linf`)
- von Bertalanffy growth coefficient (`vbk`)
- Annual natural mortality (`M`)
- Length-weight scaling parameter (`lwa`)
- Length-weight allometric parameter (`lwb`)
- Length or age at 50% maturity (`M50`)
- Whether `M50` input is in "length" or "age" (`maturity_input`)

Minimum inputs: Exploitation

- Length at 50% selectivity (`S50`)
- Whether `S50` input is in "length" or "age" (`selex_input`)

Assumptions using the default settings

- Width of the length classes (`binwidth`) equal to 1
- von Bertalanffy length at age=0 (`t0`) equal to -0.01
- Selectivity is assumed to follow a logistic function.
- If `M95` and `S95` are unspecified, their defaults are `NULL` and we assume one-parameter logistic maturity and/or selectivity. This results in close to knife-edge maturity or selectivity.

Other inputs for `create_lh_list`

Other inputs: Biology

- Length or age at 95% maturity (`M95`): Default=`NULL` for one-parameter logistic maturity; specifying `M95` will use two-parameter logistic maturity curve. It is assumed `M95` should be input as length if `M50` is length, same for age.
- Length at age=0 (`t0`): Default=-0.01, should be adjusted based on local studies or meta-analysis.

- Equilibrium recruitment (`R0`): Default=1.0. Changing `R0` will change the scale of the population size. It is recommended to set an initial value of `R0` more appropriate to the expected population size if using catch data to estimate population size with LIME.
- Steepness (`h`): Default=1.0 such that the expected recruitment calculated in the Beverton-Holt stock-recruit curve is not affected by the level of spawning biomass. Setting `h` below 1.0 will lead to the expected annual recruitment as a function of the spawning biomass. LIME will still calculate recruitment deviates around this expected value of annual recruitment.
- Maximum age (`AgeMax`): Default= the age at which 1% of the population is still alive based on the specified natural mortality rate `M`. This age can be adjusted based on local evidence of maximum age.
- Recruitment autocorrelation (`rho`): Used in simulation only. Default=0 (no recruitment autocorrelation). Changing the value of `rho` will add autocorrelation to the generated recruitment time series for a simulation study.

Other inputs: Exploitation

- Length or age at 95% selectivity (`S95`): Default=`NULL` for one-parameter logistic selectivity; specifying `S95` will use two-parameter logistic selectivity curve. It is assumed `S95` should be input as length if `S50` is length, same for age.
- Selectivity function (`selex_type`): Default=`logistic` using a one-parameter logistic selectivity curve if `S95` is not specified, or a two-parameter logistic curve if `S95` is specified. Alternate option=`dome` to generate a population exploited based on a dome-shaped selectivity curve. Dome-shaped selectivity cannot currently be estimated in LIME, but specifying dome-shaped selectivity in the `create_lh_list` function can generate an assumed dome-shaped curve to be fixed when using LIME for estimation.
- Dome-shaped selectivity right-hand standard deviation (`dome_sd`): Default=`NULL` for logistic selectivity, must be specified if `selex_type=dome`. The standard deviation of the normal distribution for the right side of the selectivity curve (after the logistic curve specified reaches 100% selectivity).
- Catchability coefficient (`qcoef`): Estimated when using an abundance index, default=1e-5, starting value should be adjusted to a reasonable number based on the abundance index and starting value for the scaling parameter `R0`.
- Dirichlet-multinomial parameter (`theta`): Default=10, a relatively large number coinciding with no variance inflation where the effective sample size will approach the input sample size (Thorson et al. 2017)
- Number of seasons in a year (`nseasons`): Default=1 for annual length composition data and annual growth and mortality parameters. For short-lived species, it is helpful to use a shorter-than-annual time step to account for the growth of short-lived fish during one year, if length data is collected on time steps less than one year (e.g. month, `nseasons`=12).

Other inputs: Variation

- Coefficient of variation around the growth curve (`CVlen`): Default=0.1; Should be adjusted based on the expected variability in the age-length curve.
- Recruitment standard deviation (`SigmaR`): Default=0.737, the median across all fish species (Thorson et al. 2014); Used for generating recruitment deviates in the simulation or as a starting value for its estimation using LIME.
- Fishing mortality standard deviation (`SigmaF`): Default=0.2; Used for generating fishing mortality deviates in the simulation or as the standard deviation of the fishing mortality penalty when estimating annual fishing mortality using LIME.
- Catch standard deviation (`SigmaC`): Default=0.2; Used as the fixed standard deviation in the lognormal likelihood when fitting LIME to catch data.
- Index standard deviation (`SigmaI`): Default=0.2; Used as the fixed standard deviation in the lognormal likelihood when fitting LIME to index data.

Now let's populate the `create_lh_list` function with example values.

```
lh <- create_lh_list(vbk=0.21,
                     linf=65,
                     t0=-0.01,
```

```
                    lwa=0.0245,
                    lwb=2.79,
                    M=0.27,
                    M50=34,
                    M95=NULL,
                    maturity_input="length",
                    S50=20,
                    S95=26,
                    selex_input="length",
                    binwidth=1,
                    CVlen=0.1,
                    SigmaR=0.737,
                    SigmaF=0.2,
                    R0=1,
                    rho=0.43,
                    nseasons=1)
```

Plot the biological and selectivity inputs

And then check out the biological parameters and selectivity we've created. Note that even though we input maturity and selectivity by length, `create_lh_list` converts to age and outputs both age (`lh$S_a`) and length (`lh$S_l`).

Generating data from the simulation model

Using the life history list output from `create_lh_list`, we can simulate a population and generate data.

The `generate_data` function has several settings for which the user is required to input a value:

- `modpath`: model path for saving simulated populations; can set as `NULL` to run within the R environment only without saving locally.
- `itervec`: vector of iterations of simulated data; can be 1 to run only one iteration, or 1:100 to run 100 iterations of simulated populations.
- `lh`: life history list, output from `create_lh_list`.
- `Fdynamics`: Pattern for fishing mortality dynamics; options include `Constant`, `Ramp`, `Increasing`, `None`, or `Endogenous`.
- `Rdynamics`: Pattern for recruitment dynamics; options include `Constant`, `AR` (for autocorrelated recruitment), `Pulsed`, `Pulsed_up`, or `BH` (for Beverton-Holt stock-recruit relationship).
- `Nyears`: number of years for your simulated population.
- `Nyears_comp`: number of years to generate length data; e.g. if 10 years and `Nyears` is 20 years, will be the last 10 years in the 20 year time series.
- `comp_sample`: nominal sample size of length data annually; e.g. 200 length measurements collected annually.
- `init_depl`: initial depletion, or the proportion of the unfished population biomass in the first year of the population to be modeled. Specifying a single value indicates the initial depletion (e.g. 0.5) or two values indicates the lower and upper bounds of a uniform distribution for which the population simulation will randomly draw a depletion value (e.g. c(0.1,0.9)).

There are also several other settings not required but may be useful:

- `rewrite`: default=TRUE to always re-simulated data. `rewrite`=FALSE may be useful to only simulate a new population if the `True.rds` output file already exists in the folder.
- `derive_quants`: default=FALSE to save time, changing this argument to TRUE will calculate MSY-based reference points.

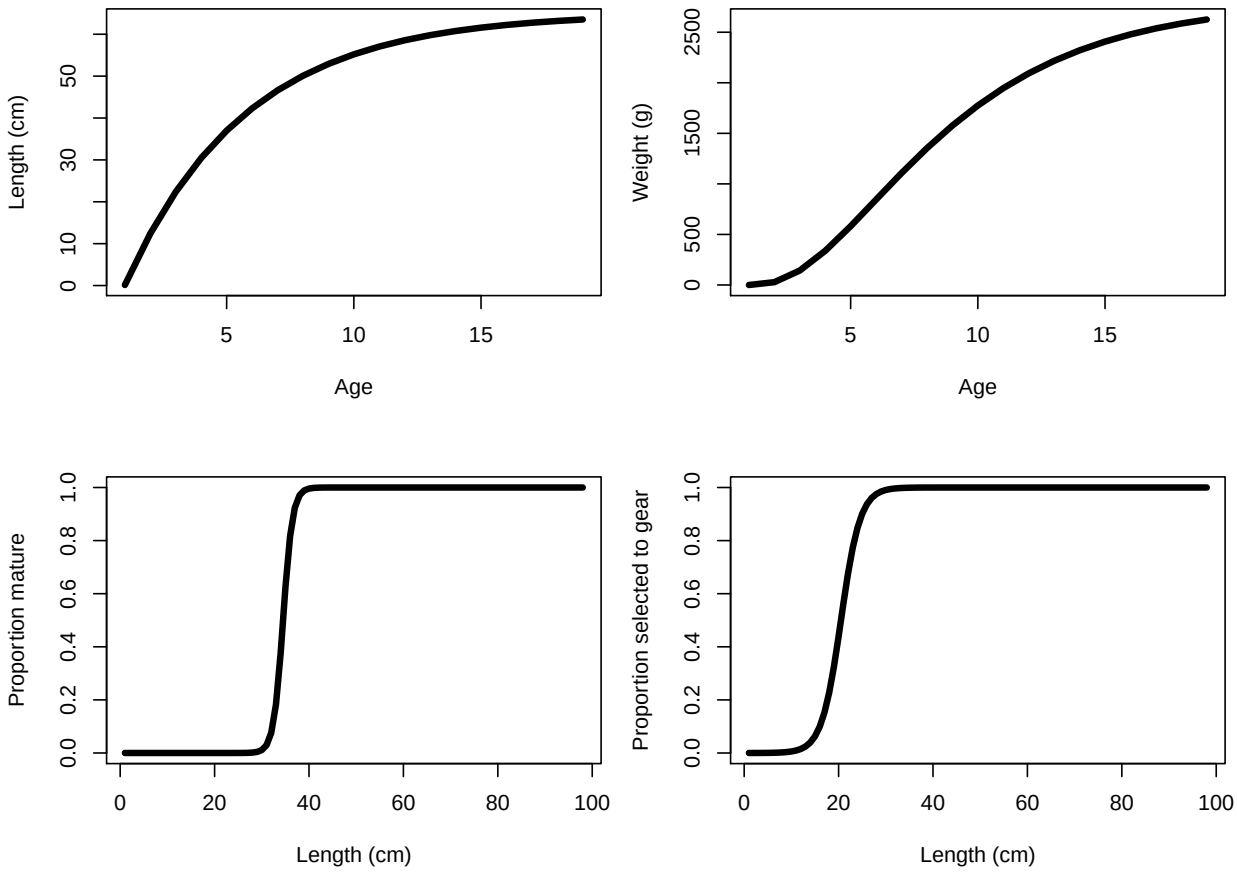


Figure 1: Example length-at-age, weight-at-age, maturity-at-length, and selectivity-at-length.

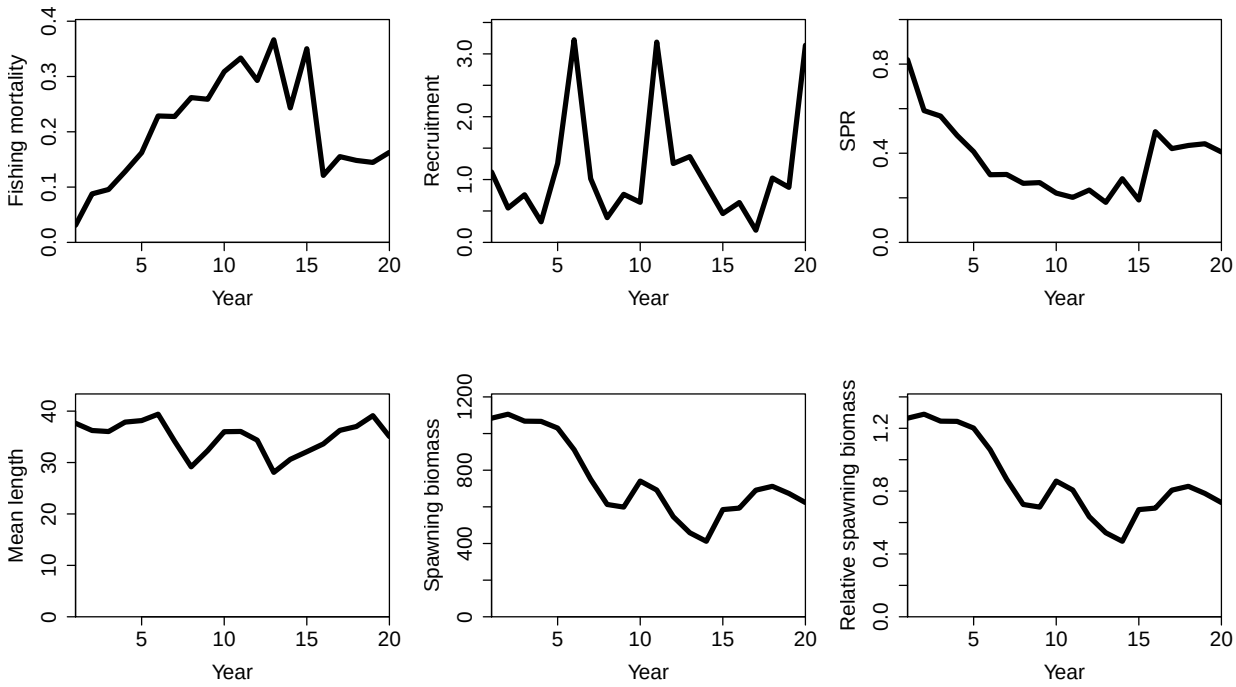


Figure 2: True population: fishing mortality ramped (increasing then decreasing), autocorrelated recruitment, spawning potential ratio (SPR), mean length, spawning biomass, and relative spawning biomass.

- `pool`: default=TRUE, meaning that if the number of seasons per year as specified in the `create_lh_list` function is more than 1, the length composition data collected in each time set is pooled together annually. `pool`=FALSE would mean the generated length composition data is on a time step shorter than 1 year (e.g. monthly if `nseasons`=12). The total sample size for the year will still be equal to `comp_sample`.
- `mismatch`: default=FALSE, indicating that the length data, catch, and abundance index are all generated from the same years. `mismatch`=FALSE forces the length data to be collected in separate years from the catch and abundance index, overlapping only 1 year.

Now let's use `generate_data` to simulate 1 example population (itervec=1) to generate length, catch, and an abundance index. We won't specify a `modpath` so the simulated population will just be used locally in this vignette.

```
true <- generate_data(modpath=NULL,
                      itervec=1,
                      lh=lh,
                      Fdynamics="Ramp",
                      Rdynamics="AR",
                      Nyears=20,
                      Nyears_comp=20,
                      comp_sample=200,
                      init_depl=0.8)
```

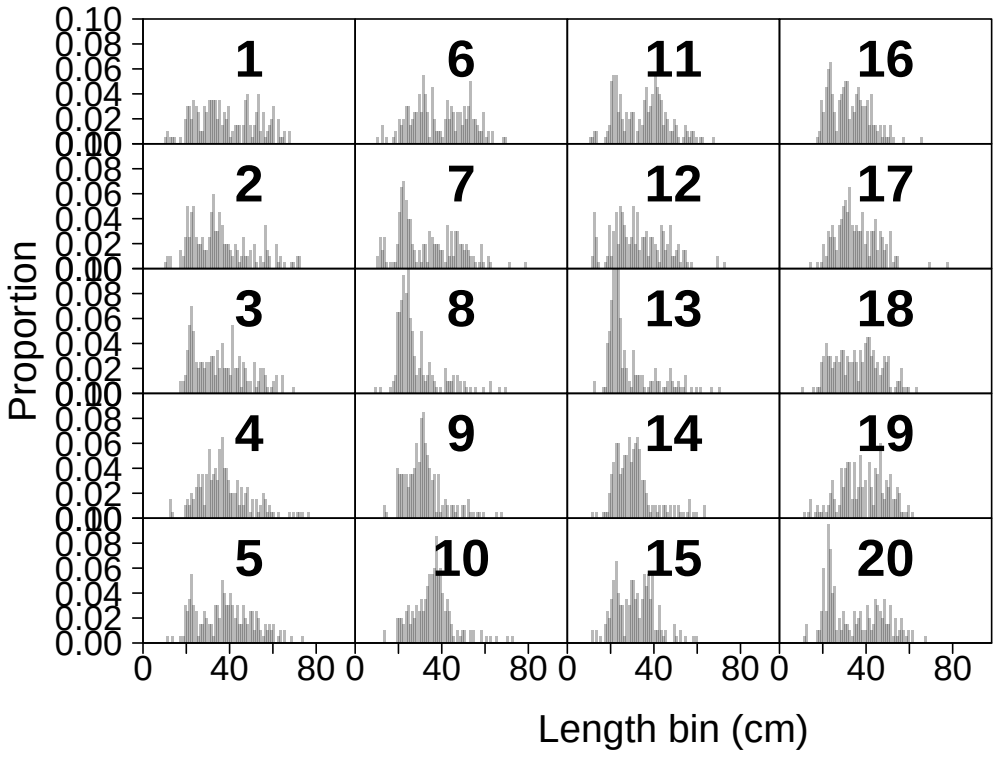


Figure 3: Generated length composition data, labeled by years 1-20.

Plot the simulated population

Plot the generated data

Each panel of length data is labeled with the year (1-20).

Estimation

Using LIME for estimation requires three vital steps:

1. Setup data correctly
2. Run the LIME model using TMB
3. Check convergence and plot output
4. Run sensitivities

Data setup

It is very important that the data are set up correctly for the LIME estimation to run.

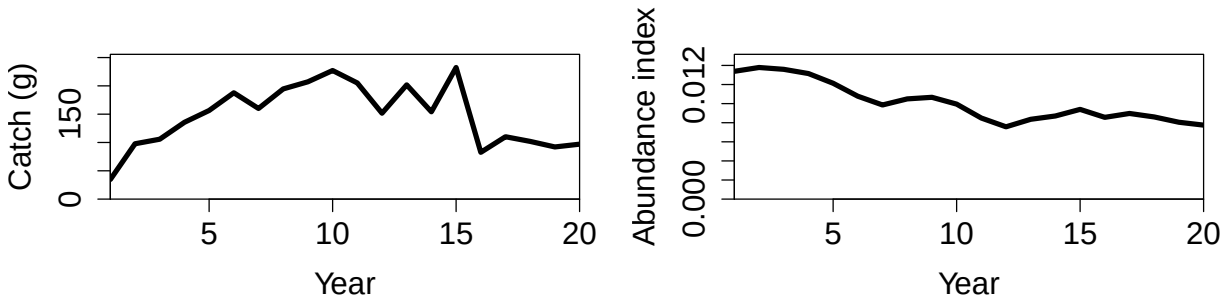


Figure 4: Generated catch and abundance index over time.

Length data

A matrix with the time step (e.g. year) along the rows and length bins along the columns.

- Rows should be named with the year names. Only years with length data should be included. Years can be in true years (1998-2017) or indexed years (1-20) as long as there is consistency between items in the data list (see "Input List" below).
- Columns should be named with the upper length bins. Columns should range from the first possible length bin (e.g. 1 for 1-cm length bins) to the last observed length bin. If there are no observations until 30cm, add a matrix of zeros for lengths 1,2,3,... 29cm. If the last observed length bin is less than 1.5 times the asymptotic length (linf), LIME will internally add a matrix of zeros to be able to account for potentially larger fish than were observed.

```
## years with length data -- rename with your own years with length data
length_years <- rownames(true$LF)

## length bins -- rename with your upper length bins
length_bins <- colnames(true$LF)

## matrix with length frequency data
LF <- true$LF

## set rownames as years
rownames(LF) <- length_years

## set colnames as length bins
colnames(LF) <- length_bins

## first six years of length data
head(LF)
```

```
##   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
## 1 0 0 0 0 0 0 0 0 0  0  1  2  1  1  1  0  0  1  0  4  6  6  4  7  6  5  2
## 2 0 0 0 0 0 0 0 0 0  0  1  2  2  0  0  0  0  3  2  5 10  5  9 10  5  4  4
## 3 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  2  2  3  7 11 14 10  5  4  5
## 4 0 0 0 0 0 0 0 0 0  0  0  0  3  1  0  0  0  0  0  2  3  2  4  3  5  7  5
## 5 0 0 0 0 0 0 0 0 0  0  0  1  0  1  0  0  0  1  1  6  5  7 11  6  5  1  3
## 6 0 0 0 0 0 0 0 0 0  0  1  0  3  0  1  0  0  1  2  0  4  3  4  6  6  3  4
##   28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
## 1  2  6  5  7  7  7  7  4  7  3  5  4  6  1  2  3  3  3  0  3  7  8  3  1
```

```
## 2  5  3  3  5  9 12  6  6  9  7  4  4  4  3  2  4  3  1  2  5  2  2  3  0
## 3  5  4  5  5  6  6  3  7  4  8  4  4  4  3 11  4  4  6  1  5  4  2  2  0
## 4  7  2  7 11  6  7  8  6 11 13  8  8  6  4  4  4  6  2  5  2  4  5  1  3
## 5  3  5  4  3  3  1  6  6  4 10  8  6  6  8  6  4  6  3  3  6  4  3  5  5
## 6  5  5  8  5 11  8  5  1  9  4  2  2  2  1  4  7  4  6  5  2  5  5  3  6
##    52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
## 1  3  5  8  2  5  1  2  3  4  6  0  4  1  1  2  0  2  0  0  0  0  0  0  0
## 2  4  1  0  2  1  7  3  2  0  0  4  1  2  0  1  0  0  1  1  0  2  2  0  0
## 3  5  1  2  4  4  3  1  0  1  2  3  0  0  3  0  0  0  0  1  0  0  0  0  0
## 4  0  3  1  2  4  3  1  2  1  1  0  1  0  0  0  0  1  0  1  0  1  1  1  0
## 5  2  5  4  2  1  3  2  3  2  3  0  1  2  0  2  0  0  1  0  0  0  0  1  0
## 6  4  6 10  4  4  3  2  2  5  1  2  0  2  0  0  0  0  1  1  0  0  0  0  0
##    76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98
## 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 4  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 6  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

It is possible length data exist for 10 years only, but either 1) catch or abundance index data are available for years prior to length data or 2) the first few years of length data hold information on recruitment from years prior.

In the scenario that you have ten years of length data but want to model over 20 years, just make sure the years of length data are labeled with the correct year in terms of the overall model:

```
## years with length data, shortened
years_short <- 11:20

LF_short <- LF[years_short,]
rownames(LF_short) <- years_short

LF_short
```

```
##    1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
## 11 0 0 0 0 0 0 0 0 0  0  1  1  2  2  0  0  0  1  2  3 10 11 11  5  8  4  2
## 12 0 0 0 0 0 0 0 0 0  0  0  2  9  5  1  0  0  1  2  7  2  6  9  4 10  9  7
## 13 0 0 0 0 0 0 0 0 0  0  0  0  2  0  0  0  1  1  8 10 15 26 26 24 12  4  7
## 14 0 0 0 0 0 0 0 0 0  0  0  1  0  1  0  0  1  1  1  4  7  9 12 12  7  8 10
## 15 0 0 0 0 0 0 0 0 0  0  0  2  0  2  0  1  0  3  5  4  7 10 13  8  6  6  1
## 16 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  1  2  7  5  9 12 13  8  5  2
## 17 0 0 0 0 0 0 0 0 0  0  0  0  0  0  1  0  0  1  0  1  4  2  6  5  7  5  3
## 18 0 0 0 0 0 0 0 0 0  0  1  0  0  0  0  1  0  1  1  5  6  8  6  6  4  5  6
## 19 0 0 0 0 0 0 0 0 0  0  0  1  0  1  3  0  1  2  1  1  2  1  2  4  6  3  1
## 20 0 0 0 0 0 0 0 0 0  0  0  1  3  0  0  0  0  2  2  5 12  5 19 15  8  9  2
##    28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
## 11  5  4  5  5  0  2  4  3  7  9  5  6  8 11  9  8  7  3  5  4  2  2  4  3
## 12  5  4  5 10  4  9  3  5  5  6  2  5  5  4  3  1  7  6  3  4  7  2  2  3
## 13  4  2  1  7  3  3  3  3  1  0  1  2  2  4  2  2  0  1  1  2  4  2  1  1
## 14 10 13  9 11 12 13 11  6  6  4  2  1  2  2  1  2  0  2  2  1  2  0  2  0
## 15  7  6 10 10  7  8  4  7 11  9 11  7 11  2  2  6  2  1  2  0  0  0  3  1
## 16  7  8  9 10 10  4  6  5  8  9  7  6  6  7  3  8  3  3  2  3  1  3  1  1
## 17  7  8 10 11  9 13  7  6  7  6  6  9  4  6  2  6  6  8  3  6  5  3  4  0
## 18  5  7  7  3  6  5  5  7  5  2  7  5  8  9  9  6  7  4  5  3  4  6  5  6
## 19  2  8  5  8  9  9  3  9  3  5 10  5  6  0  9  5  2  8  7 12  4  3  6  5
```

```
## 20  4  2  5  3  3  2  1  5  3  4  5  2  2  6  4  2  3  7  6  5  4  7  3  3
##    52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
## 11  1  0  1  3  2  1  2  2  1  0  1  1  0  0  0  0  1  0  0  0  0  0  0  0
## 12  4  1  3  3  1  0  1  0  0  0  0  0  0  0  0  0  0  0  2  0  0  1  0  0
## 13  2  1  1  3  0  0  1  0  1  0  1  0  0  0  0  1  0  0  0  1  0  0  0  0
## 14  2  1  1  0  1  3  0  1  1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0
## 15  0  2  0  1  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 16  3  1  0  0  0  0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
## 17  6  1  2  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
## 18  1  0  1  2  2  4  1  1  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
## 19  7  3  3  5  4  1  1  1  2  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
## 20  6  1  4  1  2  4  2  1  2  0  2  0  0  0  0  0  1  0  0  0  0  0  0  0
##    76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98
## 11  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 12  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 13  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 14  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 15  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 17  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 18  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 19  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 20  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Catch and index data

Vectors with each observation named with the time step (e.g. year) it was observed. Years can be in true years (1998-2017) or indexed years (1-20) as long as there is consistency between items in the data list (see "Input List" below).

Setup for the catch data:

```
## catch years
catch_years <- names(true$Cw_t)

## catch in weight
C_t <- true$Cw_t
names(C_t) <- catch_years

C_t
```

```
##         1         2         3         4         5         6         7
##  34.57798  97.74775 105.75883 135.62760 156.50493 187.77404 160.05154
##         8         9        10        11        12        13        14
## 194.60334 207.19326 227.12236 205.13366 151.69859 201.73241 154.20927
##        15        16        17        18        19        20
## 232.43909  82.61458 110.14214 101.91499  92.10815  97.01780
```

Same setup for the abundance index:

```
## index years
index_years <- names(true$I_t)

## catch in weight
I_t <- true$I_t
names(I_t) <- index_years
```

```
I_t
```

```
##           1           2           3           4           5           6
## 0.013383592 0.013783633 0.013592323 0.013149325 0.012132628 0.010773542
##           7           8           9          10          11          12
## 0.009859794 0.010495202 0.010661281 0.009949099 0.008489889 0.007572050
##          13          14          15          16          17          18
## 0.008358553 0.008707140 0.009396856 0.008562653 0.008969735 0.008605467
##          19          20
## 0.008048952 0.007748832
```

Input list

LIME requires data in an list form, with a minimum:

- `years`: the total, inclusive years to model.
  - If length data is available for years 1-5 and 10-20, `years` should be a vector 1-20.
  - Years can be in true years (1998-2017) or indexed years (1-20) as long as there is consistency between the length, catch, and/or index data. For example, if length data years are labeled 1998-2017, `years` should be 1998-2017 (or may start before 1998 if length data are informative for a few years prior to the first year of data).
- `LF`: the length frequency matrix. See "Length data" section above for formatting details.
  - Rows labeled with years.
  - Columns labeled with upper length bins.

With the minium inputs, the data list should look like:

```
## years with length data, make sure they are numbers and not characters
length_years <- as.numeric(rownames(LF))

## total years
total_years <- min(length_years):max(length_years)

data_list <- list("years"=total_years, "LF"=LF)
```

If catch and/or an abundance index are available, these should also be included in the data input list. See "Catch and index data" section above for formatting details.

- `C_t`: catch data, in numbers or weight (to be specified later)
- `I_t`: abundance index.

```
## years with catch data, make sure they are numbers and not characters
catch_years <- as.numeric(names(C_t))

## years with index data, make sure they are numbers and not characters
index_years <- as.numeric(names(I_t))

## find minimum year across all data types
min_yr <- min(c(length_years, catch_years, index_years))

## find maximum year across all data types
max_yr <- max(c(length_years, catch_years, index_years))

total_years <- min_yr:max_yr
```

```
data_list <- list("years"=total_years, "LF"=LF, "C_t"=C_t, "I_t"=I_t)
data_list
```

```
## $years
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##
## $LF
##    1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
## 1  0 0 0 0 0 0 0 0 0  0  1  2  1  1  1  0  0  1  0  4  6  6  4  7  6  5  2
## 2  0 0 0 0 0 0 0 0 0  0  1  2  2  0  0  0  0  3  2  5 10  5  9 10  5  4  4
## 3  0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  2  2  3  7 11 14 10  5  4  5
## 4  0 0 0 0 0 0 0 0 0  0  0  0  3  1  0  0  0  0  0  2  3  2  4  3  5  7  5
## 5  0 0 0 0 0 0 0 0 0  0  0  1  0  1  0  0  0  1  1  6  5  7 11  6  5  1  3
## 6  0 0 0 0 0 0 0 0 0  0  1  0  3  0  1  0  0  1  2  0  4  3  4  6  6  3  4
## 7  0 0 0 0 0 0 0 0 0  0  2  5  3  5  1  0  1  1  1  4  9 13 14 11  8  8  4
## 8  0 0 0 0 0 0 0 0 0  1  0  1  0  0  0  0  1  2  3 13 13 15 19 16 23 12 10
## 9  0 0 0 0 0 0 0 0 0  0  0  0  0  2  1  0  0  0  0  8  7  7  7  7  5  7  7
## 10 0 0 0 0 0 0 0 0 0  0  0  0  0  2  0  0  0  0  0  4  4  4  3  5  6  3  5
## 11 0 0 0 0 0 0 0 0 0  0  1  1  2  2  0  0  0  1  2  3 10 11 11  5  8  4  2
## 12 0 0 0 0 0 0 0 0 0  0  0  2  9  5  1  0  0  1  2  7  2  6  9  4 10  9  7
## 13 0 0 0 0 0 0 0 0 0  0  0  0  2  0  0  0  1  1  8 10 15 26 26 24 12  4  7
## 14 0 0 0 0 0 0 0 0 0  0  0  1  0  1  0  0  1  1  1  4  7  9 12 12  7  8 10
## 15 0 0 0 0 0 0 0 0 0  0  0  2  0  2  0  1  0  3  5  4  7 10 13  8  6  6  1
## 16 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  1  2  7  5  9 12 13  8  5  2
## 17 0 0 0 0 0 0 0 0 0  0  0  0  0  0  1  0  0  1  0  1  4  2  6  5  7  5  3
## 18 0 0 0 0 0 0 0 0 0  0  1  0  0  0  0  1  0  1  1  5  6  8  6  6  4  5  6
## 19 0 0 0 0 0 0 0 0 0  0  0  1  0  1  3  0  1  2  1  1  2  1  2  4  6  3  1
## 20 0 0 0 0 0 0 0 0 0  0  0  1  3  0  0  0  0  2  2  5 12  5 19 15  8  9  2
##    28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
## 1   2  6  5  7  7  7  7  4  7  3  5  4  6  1  2  3  3  3  0  3  7  8  3  1
## 2   5  3  3  5  9 12  6  6  9  7  4  4  4  3  2  4  3  1  2  5  2  2  3  0
## 3   5  4  5  5  6  6  3  7  4  8  4  4  4  3 11  4  4  6  1  5  4  2  2  0
## 4   7  2  7 11  6  7  8  6 11 13  8  8  6  4  4  4  6  2  5  2  4  5  1  3
## 5   3  5  4  3  3  1  6  6  4 10  8  6  6  8  6  4  6  3  3  6  4  3  5  5
## 6   5  5  8  5 11  8  5  1  9  4  2  2  2  1  4  7  4  6  5  2  5  5  3  6
## 7   3  1  3  1  4  4  1  6  5  4  4  4  3  3  1  7  2  6  2  6  6  4  4  3
## 8   6  3  4 10  3  2  3  5  3  2  1  0  1  0  4  2  2  2  3  0  3  2  0  2
## 9   8 12  9 16 17 12 10  8  6  7  1  7  1  2  3  2  0  2  2  1  2  0  2  2
## 10  4  6  5  7  5  7  9 11 11 12 17 11 12  8  5  7  5  3  1  0  2  1  2  2
## 11  5  4  5  5  0  2  4  3  7  9  5  6  8 11  9  8  7  3  5  4  2  2  4  3
## 12  5  4  5 10  4  9  3  5  5  6  2  5  5  4  3  1  7  6  3  4  7  2  2  3
## 13  4  2  1  7  3  3  3  3  1  0  1  2  2  4  2  2  0  1  1  2  4  2  1  1
## 14 10 13  9 11 12 13 11  6  6  4  2  1  2  2  1  2  0  2  2  1  2  0  2  0
## 15  7  6 10 10  7  8  4  7 11  9 11  7 11  2  2  6  2  1  2  0  0  0  3  1
## 16  7  8  9 10 10  4  6  5  8  9  7  6  6  7  3  8  3  3  2  3  1  3  1  1
## 17  7  8 10 11  9 13  7  6  7  6  6  9  4  6  2  6  6  8  3  6  5  3  4  0
## 18  5  7  7  3  6  5  5  7  5  2  7  5  8  9  9  6  7  4  5  3  4  6  5  6
## 19  2  8  5  8  9  9  3  9  3  5 10  5  6  0  9  5  2  8  7 12  4  3  6  5
## 20  4  2  5  3  3  2  1  5  3  4  5  2  2  6  4  2  3  7  6  5  4  7  3  3
##    52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
## 1   3  5  8  2  5  1  2  3  4  6  0  4  1  1  2  0  2  0  0  0  0  0  0  0
## 2   4  1  0  2  1  7  3  2  0  0  4  1  2  0  1  0  0  1  1  0  2  2  0  0
## 3   5  1  2  4  4  3  1  0  1  2  3  0  0  3  0  0  0  0  1  0  0  0  0  0
## 4   0  3  1  2  4  3  1  2  1  1  0  1  0  0  0  0  1  0  1  0  1  1  1  0
```

```
## 5   2  5  4  2  1  3  2  3  2  3  0  1  2  0  2  0  0  1  0  0  0  0  1  0
## 6   4  6 10  4  4  3  2  2  5  1  2  0  2  0  0  0  0  1  1  0  0  0  0  0
## 7   4  3  2  2  1  0  1  4  1  0  2  1  0  0  0  0  0  0  0  0  1  0  0  0
## 8   1  0  1  0  1  0  0  0  1  0  0  2  0  0  0  1  0  0  1  0  0  0  0  0
## 9   0  3  1  1  0  1  1  0  1  0  0  0  0  0  1  0  1  0  0  0  0  0  0  0
## 10  0  0  2  2  0  0  0  2  0  1  0  1  0  0  1  0  0  0  0  1  0  1  0  0
## 11  1  0  1  3  2  1  2  2  1  0  1  1  0  0  0  0  1  0  0  0  0  0  0  0
## 12  4  1  3  3  1  0  1  0  0  0  0  0  0  0  0  0  0  0  2  0  0  1  0  0
## 13  2  1  1  3  0  0  1  0  1  0  1  0  0  0  0  1  0  0  0  1  0  0  0  0
## 14  2  1  1  0  1  3  0  1  1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0
## 15  0  2  0  1  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 16  3  1  0  0  0  0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
## 17  6  1  2  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
## 18  1  0  1  2  2  4  1  1  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
## 19  7  3  3  5  4  1  1  1  2  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
## 20  6  1  4  1  2  4  2  1  2  0  2  0  0  0  0  0  1  0  0  0  0  0  0  0
##    76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98
## 1   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 2   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 3   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 4   0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 5   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 6   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 7   0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 8   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 9   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 10  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 11  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 12  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 13  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 14  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 15  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 17  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 18  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 19  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 20  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##
## $C_t
##         1         2         3         4         5         6         7
##  34.57798  97.74775 105.75883 135.62760 156.50493 187.77404 160.05154
##         8         9        10        11        12        13        14
## 194.60334 207.19326 227.12236 205.13366 151.69859 201.73241 154.20927
##        15        16        17        18        19        20
## 232.43909  82.61458 110.14214 101.91499  92.10815  97.01780
##
## $I_t
##           1           2           3           4           5           6
## 0.013383592 0.013783633 0.013592323 0.013149325 0.012132628 0.010773542
##           7           8           9          10          11          12
## 0.009859794 0.010495202 0.010661281 0.009949099 0.008489889 0.007572050
##          13          14          15          16          17          18
## 0.008358553 0.008707140 0.009396856 0.008562653 0.008969735 0.008605467
##          19          20
```

```
## 0.008048952 0.007748832
```

Remember to plot your data to make sure they look right. See above section "Plot generated data" where we already checked the generated data used for the vignette.

Run LIME

Now we can use `run_LIME` to run the LIME assessment. LIME uses Template Model Builder (TMB) to calculate recruitment deviations as random effects and standard errors (Kristensen et al. 2016). However, the LIME package has the model pre-compiled, so all you really need are the TMB and TMBhelper packages downloaded and the `run_LIME` function.

The function `run_LIME` has several required inputs:

- `modpath`: model path for saving LIME output; can set as `NULL` to run within the R environment only without saving locally.
- `lh`: life history list, output from `create_lh_list`.
- `input_data`: list of data inputs, output above called `data_list`.
- `est_sigma`: vector of variance parameters to estimate, must match parameter names. Estimate recruitment standard deviation with `log_sigma_R`. Another desired option may be including the coefficient of variation around the growth curve, `log_CV_L`. Other options include the catch or index standard deviations: `log_sigma_C`, `log_sigma_I`. Concatenate to estimate multiple variance parameters: e.g. c(`log_sigma_R`,`log_CV_L`). If not included in `est_sigma`, parameter will be fixed at the starting value set in `create_lh_list`.
- `data_avail`: Types of data included, must at least include "LC"" as length data is the minimum data input. May also include "Catch" or "Index" if included, separated by an underscore, e.g. "LC", "Catch_LC", "Index_LC", or "Index_Catch_LC".

There are many other settings and tools included as arguments to `run_LIME`:

Fixing parameters

- `fix_param`: default=FALSE, or can specify a non-time-varying parameter name (must match name from model). Many parameters are automatically fixed unless specific data scenarios arise. Non-time-varying parameters that could be fixed using this argument include:
- `log_q_I`: log catchability coefficient, automatically fixed to starting value from `create_lh_list` unless using an abundance index.
- `beta`: log equilibrium recruitment, automatically fixed to starting value from `create_lh_list` unless using catch data.
- `logS50`: log length at 50% selectivity, automatically estimated unless specified in `fix_param`.
- `logSdelta`: log difference between length at 95% and 50% selectivity, automatically estimated unless specified in `fix_param`.
- `log_theta`: log Dirichlet-multinomial parameter related to effective sample size, automatically estimated unless specified in `fix_param`.
- `fix_param_t`: default=FALSE, or can fix some time-varying values with the name as the first item in the list and the years as a vector in the second item of the list.
- to fix years 1-10 of fishing mortality rates: `fix_param_t`=list(param="log_F_t_input", years=1:10)
- to fix years 1-10 of recruitment deviations: `fix_param_t`=list(param="Nu_input", years=1:10)
- `S_l_input`: default=-1 to estimate selectivity parameters or use input values. Fix the selectivity curve to specific values by inputting a vector of selectivity-at-length. The objective of this setting is to explore the impact of dome-shaped selectivity or exploring other shapes besides the one- or two-parameter logistic curves.
- `randomR`: default=TRUE to derive recruitment deviations as random effects. Change to FALSE to turn off random effect on recruitment.
- `param_adjust`: vector of parameter names, as a simple way to change the starting or input values for some parameters than creating a new list with `create_lh_list`

- full list includes: `SigmaR`, `SigmaF`, `SigmaC`, `SigmaI`, `CVlen`, `M`, `linf`, `vbk`, `ML50`.
- Other parameters require new life history list with `create_lh_list`
- `val_adjust`: adjusted values in the order listed by parameter names in `param_adjust`

Other settings

- `rewrite`: default=TRUE to run LIME assessment, change to FALSE to avoid re-running LIME if output already exists in the directory.
- `C_opt`: default=0 when not including catch data. 1= input catch in numbers, 2= input catch in biomass.
- `newtonsteps`: specify number of newton steps to take after optimization; default=3 to help with optimization; 0 will turn this feature off.
- `F_up`: upper bound of fishing mortality parameters, default=10
- `S50_up`: upper bound of length at 50% selectivity, default=NULL
- `LFdist`: default=1 to use dirichlet-multinomial likelihood function to fit length comp data, other option 0= multinomial likelihood using nominal sample size of length data.
- `derive_quants`: default=FALSE due to longer run time, can set to TRUE to output additional derived quantities, including MSY-based reference points when catch data included.
- `theta_type`: default=1 to estimate single theta related to effective sample size for all years of length comp. 0=estimate annual theta for each year of length data.
- `Fpen`: penalty on fishing mortality; default=1 ON, 0=OFF
- `SigRpen`: penalty on recruitment standard deviation; default=1 ON, 0=OFF

Settings related to simulation

- `f_true`: default=FALSE for starting values of fishing mortality rates at 1.0. change to TRUE to start at the true values from the simulation.
- `itervec`: default=NULL, for use in simulation study to run multiple iterations within the same set of model run settings.
- `simulation`: default=FALSE, change to TRUE if using LIME in the context of a simulation study, to require `itervec` specification.

Here is an example to run LIME with default settings, with a "data-rich" scenario including 20 years of length, catch, and abundance index data.

```
res_rich <- run_LIME(modpath=NULL,
                     lh=lh,
                     input_data=data_list,
                     est_sigma="log_sigma_R",
                     data_avail="Index_Catch_LC",
                     C_opt=2)
```

Let's compare results from the data-rich scenario when we exclude catch data by changing the `data_avail` argument:

```
res_nocatch <- run_LIME(modpath=NULL,
                        lh=lh,
                        input_data=data_list,
                        est_sigma="log_sigma_R",
                        data_avail="Index_LC")
```

Now let's run LIME with length data only:

```
res_length <- run_LIME(modpath=NULL,
                       lh=lh,
                       input_data=data_list,
                       est_sigma="log_sigma_R",
                       data_avail="LC")
```

Finally, let's run LIME with only 10 years of length data, but continue modeling 20 years:

```
## new data list
data_list_short <- list("years"=total_years, "LF"=LF_short)

res_length_short <- run_LIME(modpath=NULL,
                        lh=lh,
                        input_data=data_list_short,
                        est_sigma="log_sigma_R",
                        data_avail="LC")
```

Check convergence and plot output

All models we ran estimate 20 years of annual fishing mortality (`log_F_t_input`), recruitment standard deviation (`log_sigma_R`), selectivity parameters (`logS50` and `logSdelta`), and the Dirichlet-multinomial theta parameter (`log_theta`).

Some other parameters are estimated if an abundance index or catch data are included.

Before we use any of the above output, we need to check the model convergence. We assess model convergence based on if the absolute value of the final gradient of each parameter is less than 0.001.

Data-rich scenario, includes default estimated parameters as well as catchability coefficient (`log_q_I`) and equilibrium recruitment (`beta`):

```
res_rich$df[,1:2]
```

```
##    Final gradient     Parameter
## 1   -1.642559e-05 log_F_t_input
## 2    1.745647e-07 log_F_t_input
## 3   -5.523737e-08 log_F_t_input
## 4    1.622661e-08 log_F_t_input
## 5    1.424245e-07 log_F_t_input
## 6    2.268518e-07 log_F_t_input
## 7   -8.181393e-06 log_F_t_input
## 8    5.176697e-07 log_F_t_input
## 9   -7.997324e-06 log_F_t_input
## 10   4.494931e-07 log_F_t_input
## 11   2.792490e-07 log_F_t_input
## 12   8.164304e-06 log_F_t_input
## 13   3.028769e-07 log_F_t_input
## 14  -8.064583e-06 log_F_t_input
## 15   9.060119e-06 log_F_t_input
## 16  -7.842030e-06 log_F_t_input
## 17   9.117125e-06 log_F_t_input
## 18   6.180100e-07 log_F_t_input
## 19   7.031541e-07 log_F_t_input
## 20  -6.971701e-07 log_F_t_input
## 21   5.667669e-07       log_q_I
## 22   2.045032e-06          beta
## 23  -2.655983e-05   log_sigma_R
## 24  -4.399494e-06        logS50
## 25  -1.358464e-05     logSdelta
## 26  -1.910559e-04     log_theta
```

Index and length scenario, includes default estimated parameters as well as catchability coefficient:

```
res_nocatch$df[,1:2]
```

```
##    Final gradient     Parameter
## 1   -1.074296e-04 log_F_t_input
## 2    1.082637e-05 log_F_t_input
## 3    2.087347e-05 log_F_t_input
## 4    2.881413e-06 log_F_t_input
## 5    7.427266e-06 log_F_t_input
## 6    1.172242e-05 log_F_t_input
## 7    1.109069e-05 log_F_t_input
## 8    9.269421e-06 log_F_t_input
## 9    2.521750e-05 log_F_t_input
## 10   3.132836e-05 log_F_t_input
## 11   1.036268e-05 log_F_t_input
## 12   1.112484e-05 log_F_t_input
## 13   1.011125e-05 log_F_t_input
## 14   1.268066e-05 log_F_t_input
## 15   1.204588e-05 log_F_t_input
## 16   2.747869e-05 log_F_t_input
## 17   1.124615e-05 log_F_t_input
## 18   2.430731e-05 log_F_t_input
## 19   7.997274e-06 log_F_t_input
## 20  -1.254684e-11 log_F_t_input
## 21   8.767184e-05       log_q_I
## 22  -1.870621e-04   log_sigma_R
## 23  -2.262153e-04        logS50
## 24  -4.536713e-04     logSdelta
## 25  -8.401927e-04     log_theta
```

Length only scenarios (20 years and 10 years), include default estimated parameters only.

```
res_length$df[,1:2]
```

```
##    Final gradient     Parameter
## 1   -4.030707e-09 log_F_t_input
## 2   -6.243325e-09 log_F_t_input
## 3    4.724236e-09 log_F_t_input
## 4    7.667820e-09 log_F_t_input
## 5    7.622251e-09 log_F_t_input
## 6    7.621586e-10 log_F_t_input
## 7    2.420244e-09 log_F_t_input
## 8    3.209552e-09 log_F_t_input
## 9   -4.812128e-10 log_F_t_input
## 10   8.865206e-10 log_F_t_input
## 11   1.292298e-10 log_F_t_input
## 12  -9.421138e-09 log_F_t_input
## 13   2.921274e-09 log_F_t_input
## 14  -2.599970e-06 log_F_t_input
## 15  -1.842834e-09 log_F_t_input
## 16   1.791704e-10 log_F_t_input
## 17  -1.055677e-08 log_F_t_input
## 18   2.018534e-09 log_F_t_input
## 19  -1.418480e-09 log_F_t_input
## 20  -1.678304e-13 log_F_t_input
## 21  -2.567493e-09   log_sigma_R
```

```
## 22   5.402348e-08        logS50
## 23   3.113626e-08     logSdelta
## 24  -1.043663e-04     log_theta
```

10 years of length:

```
res_length_short$df[,1:2]
```

```
##    opt_save.final_gradient names.obj_save.par.
## 1            -3.641175e-07       log_F_t_input
## 2            -1.643148e-05       log_F_t_input
## 3            -6.053072e-07       log_F_t_input
## 4             1.564117e-05       log_F_t_input
## 5            -4.638418e-06       log_F_t_input
## 6            -1.763603e-05       log_F_t_input
## 7             1.428154e-05       log_F_t_input
## 8            -8.847326e-07       log_F_t_input
## 9            -1.000576e-06       log_F_t_input
## 10           -4.591598e-07       log_F_t_input
## 11           -1.711913e-05       log_F_t_input
## 12           -7.972557e-07       log_F_t_input
## 13            1.568714e-05       log_F_t_input
## 14           -5.964907e-07       log_F_t_input
## 15           -1.645468e-07       log_F_t_input
## 16           -1.598052e-05       log_F_t_input
## 17           -1.608654e-05       log_F_t_input
## 18            1.904840e-07       log_F_t_input
## 19           -3.159419e-05       log_F_t_input
## 20            3.795921e-11       log_F_t_input
## 21            1.576108e-05         log_sigma_R
## 22            1.263351e-05              logS50
## 23           -1.155368e-06           logSdelta
## 24            9.690644e-05           log_theta
```

Once we have confirmed each model has converged, we can explore model output. The black lines indicate the true value of fishing mortality, recruitment, SPR, and selectivity. Blue lines indicate the maximum likelihood estimates, blue points indicate a year with data, and blue shading indicates 95% confidence intervals.

```
plot_output(all_years=total_years, true_years=total_years, lc_years=length_years, Inputs=res_rich$Inputs
```

```
plot_output(all_years=total_years, true_years=total_years, lc_years=length_years, Inputs=res_nocatch$Inp
```

```
plot_output(all_years=total_years, true_years=total_years, lc_years=length_years, Inputs=res_length$Inpu
```

```
plot_output(all_years=total_years, true_years=total_years, lc_years=11:20, Inputs=res_length_short$Input
```

Check sensitivites

Biological inputs

The most straightforward way to test LIME sensitivity to input parameters is to create alternate life history/input lists using `create_lh_list`. For example, if we want to test the impact of a 25% larger asympototic length, we could create a new list:

```
lh_alt <- create_lh_list(vbk=0.21,
                         linf=65*1.25,
                         t0=-0.01,
```

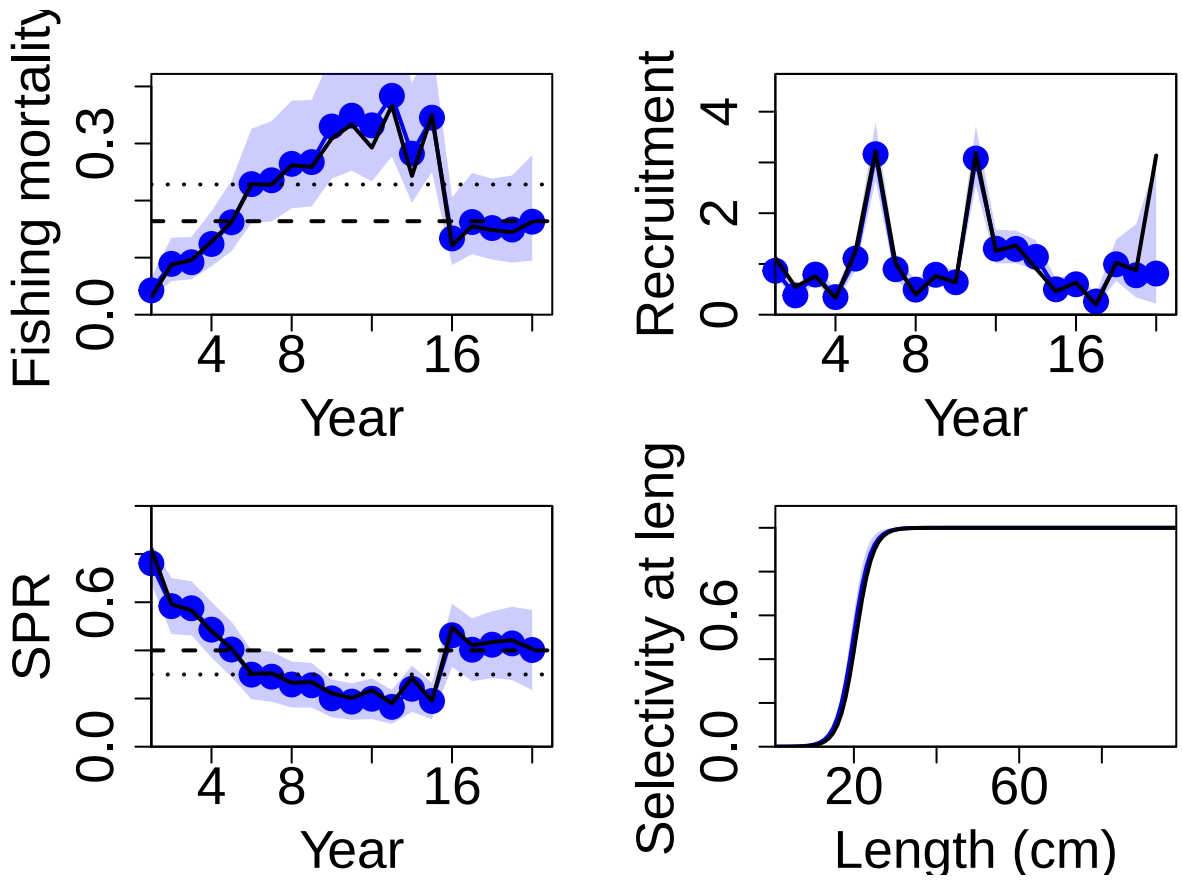


Figure 5: Results from data-rich scenario

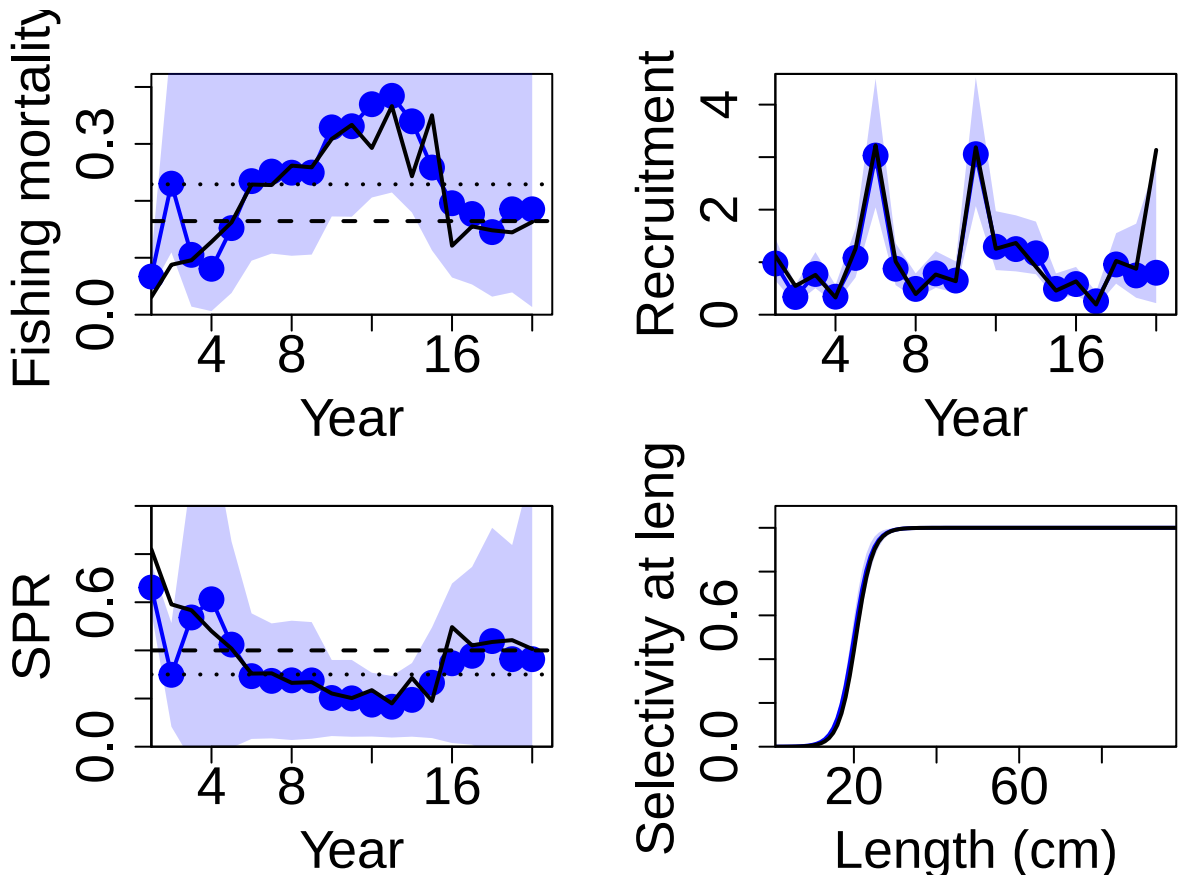


Figure 6: Results from including index and length

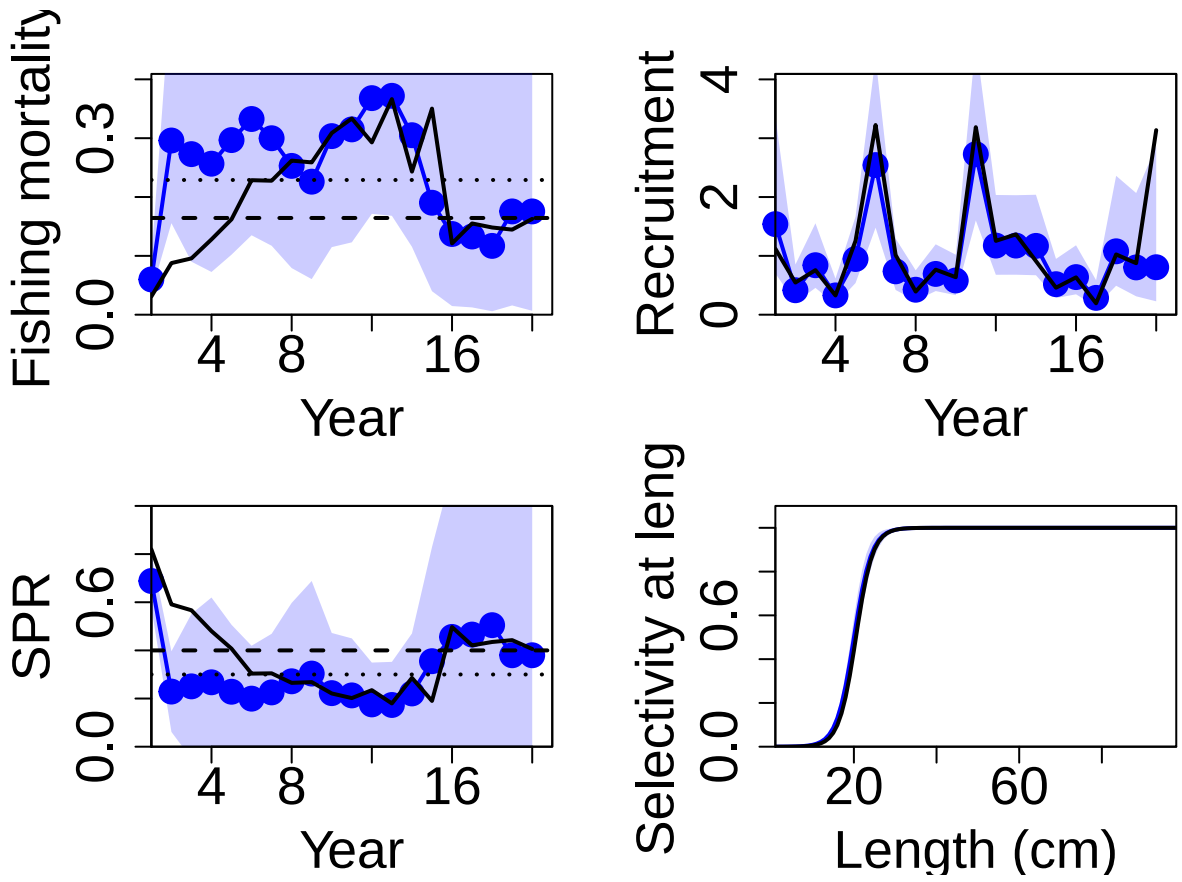


Figure 7: Results including length only

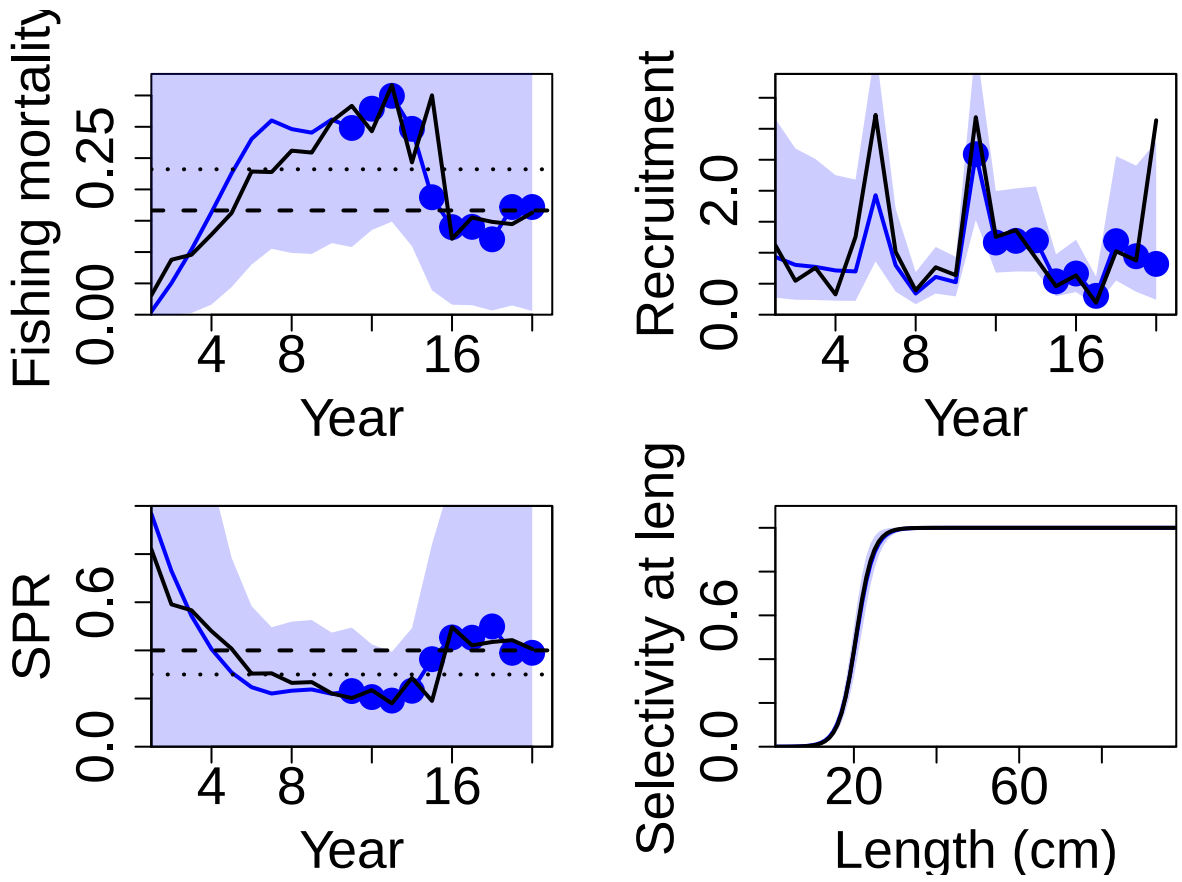


Figure 8: Results including 10 years of length data only

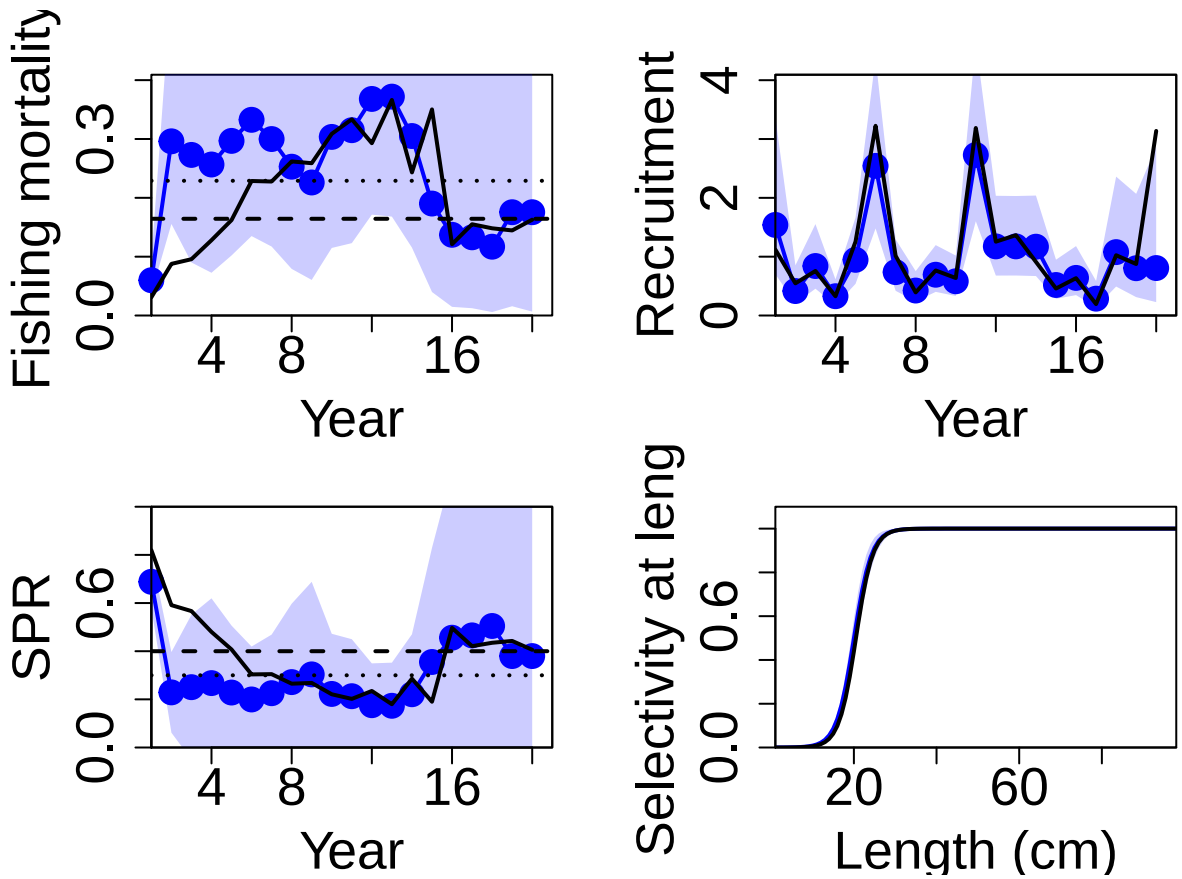
```
                    lwa=0.0245,
                    lwb=2.79,
                    M=0.27,
                    M50=34,
                    M95=NULL,
                    maturity_input="length",
                    S50=20,
                    S95=26,
                    selex_input="length",
                    binwidth=1,
                    CVlen=0.1,
                    SigmaR=0.737,
                    SigmaF=0.2,
                    R0=1,
                    rho=0.43,
                    nseasons=1)
```

Then re-run the assessment with the new life history list, for example with length data only:

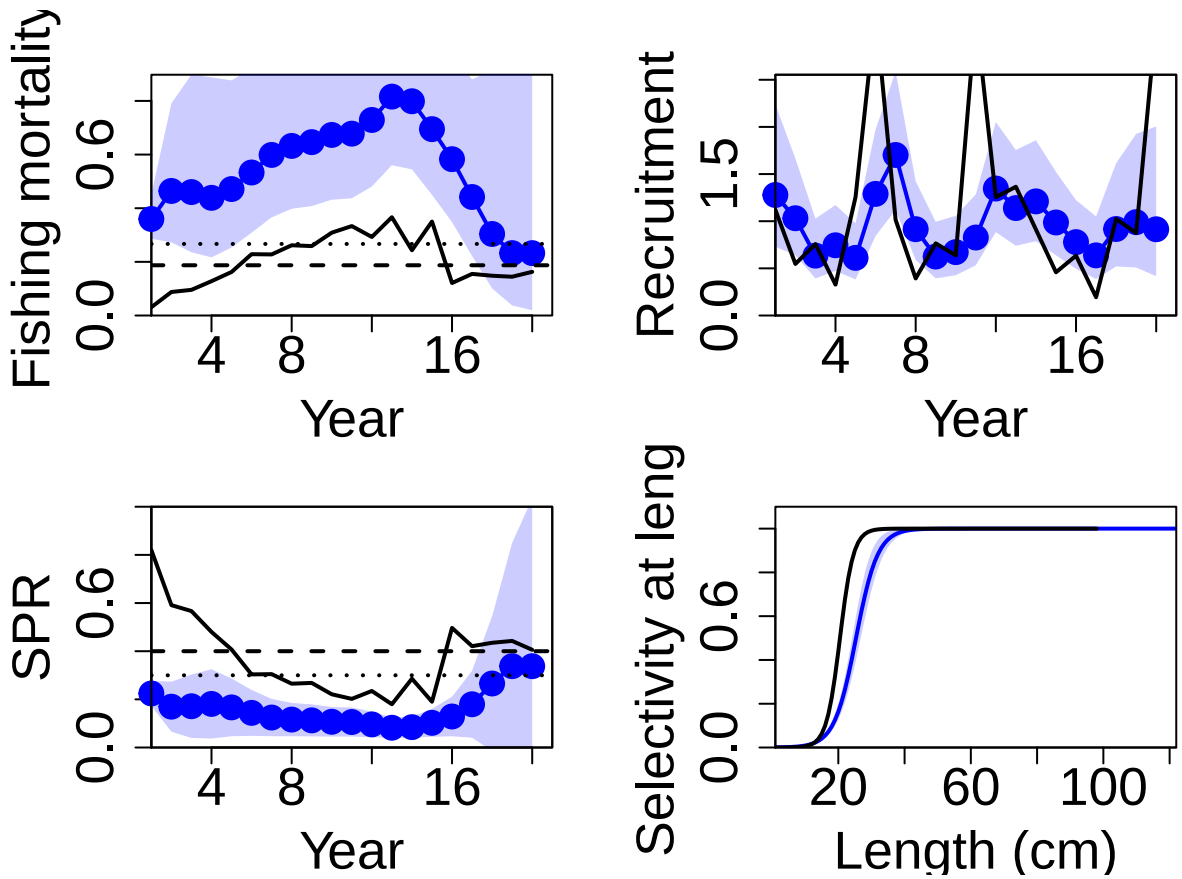
```
res_alt <- run_LIME(modpath=NULL,
                    lh=lh_alt,
                    input_data=data_list,
                    est_sigma="log_sigma_R",
                    data_avail="LC")
```

Then check how the larger asymptotic length impacts the assessment results:

```
plot_output(all_years=total_years, true_years=total_years, lc_years=length_years, Inputs=res_length$Inpu
```



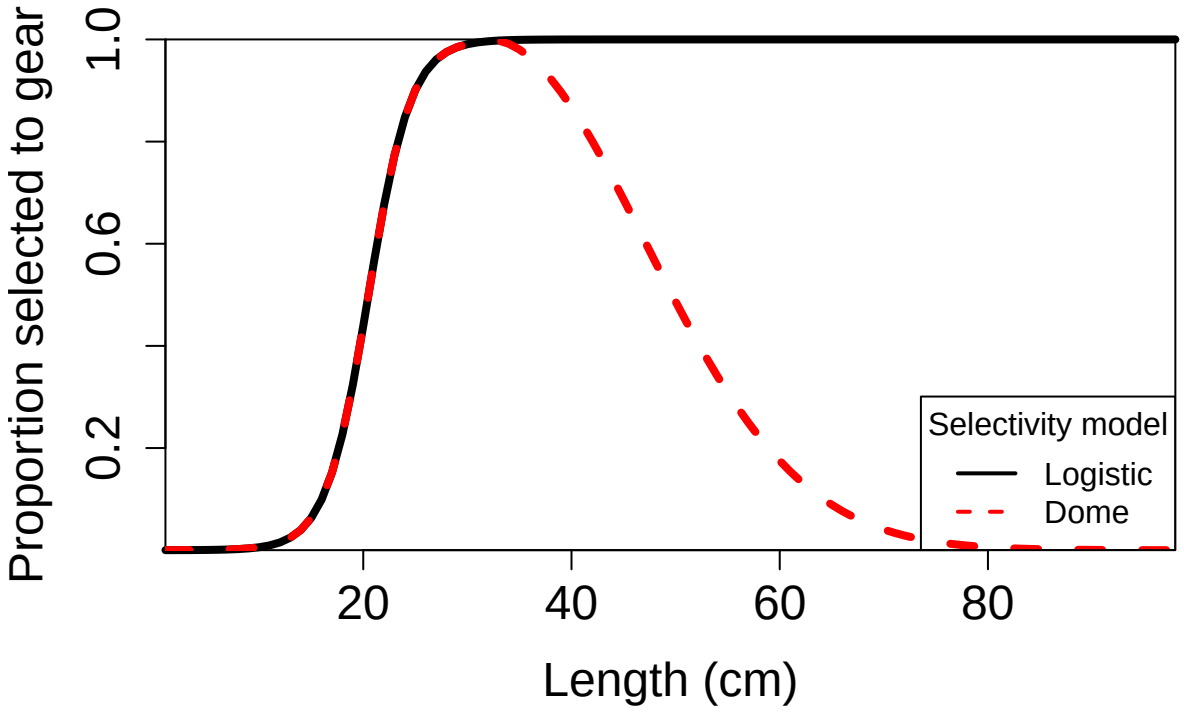

```
plot_output(all_years=total_years, true_years=total_years, lc_years=length_years, Inputs=res_alt$Inputs
```



Selectivity model

Users may also want to fix the selectivity curve, either based on previous studies or to examine the impact of dome-shaped selectivity. Here is an example where we create an input list with an assumed function for dome-shaped selectivity:

```
lh_dome <- create_lh_list(vbk=0.21,
                          linf=65,
                          t0=-0.01,
                          lwa=0.0245,
                          lwb=2.79,
                          M=0.27,
                          M50=34,
                          M95=NULL,
                          maturity_input="length",
                          S50=20,
                          S95=26,
                          selex_input="length",
                          selex_type="dome",
                          dome_sd=15,
                          binwidth=1,
                          CVlen=0.1,
                          SigmaR=0.737,
                          SigmaF=0.2,
                          R0=1,
                          rho=0.43,
                          nseasons=1)
```

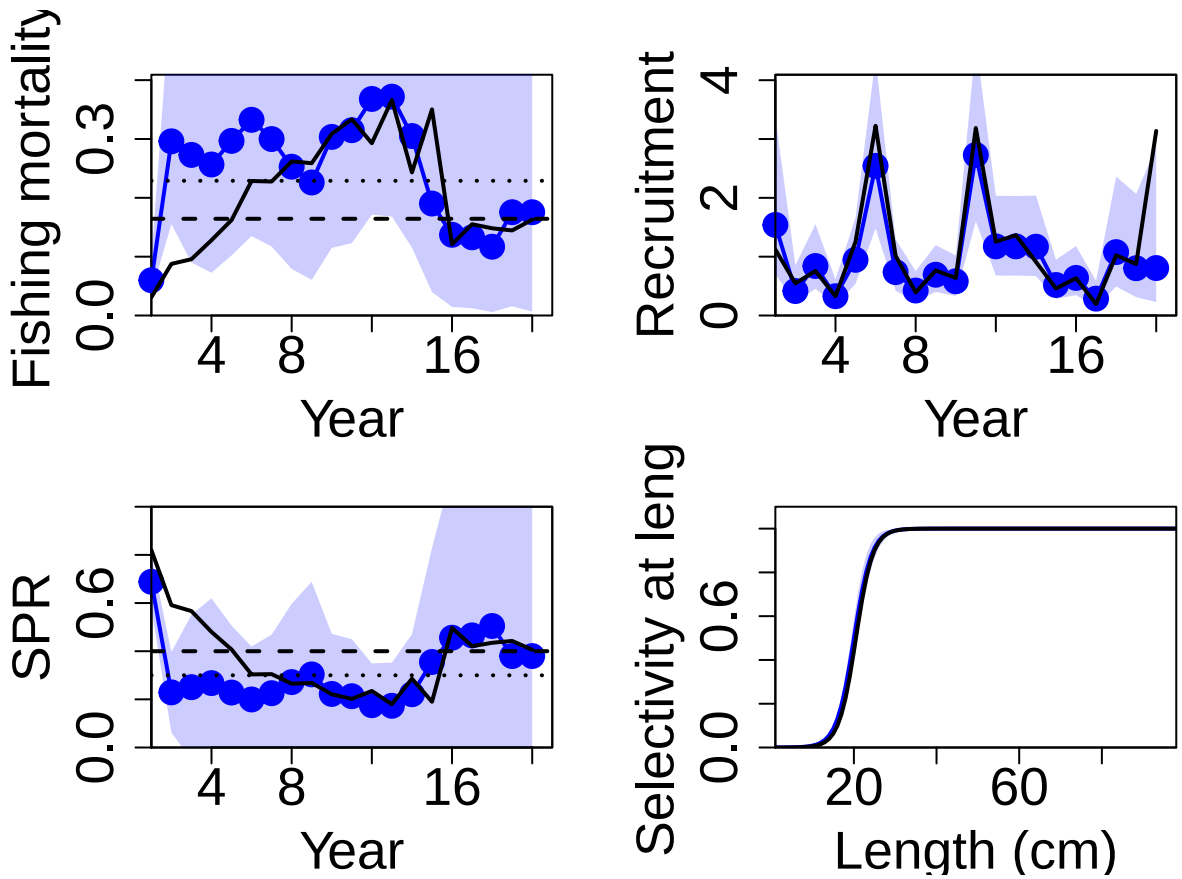


Then use the `S_l_input` argument to re-run the assessment with the fixed dome-shaped selectivity curve. This assessment will not estimate selectivity parameters.

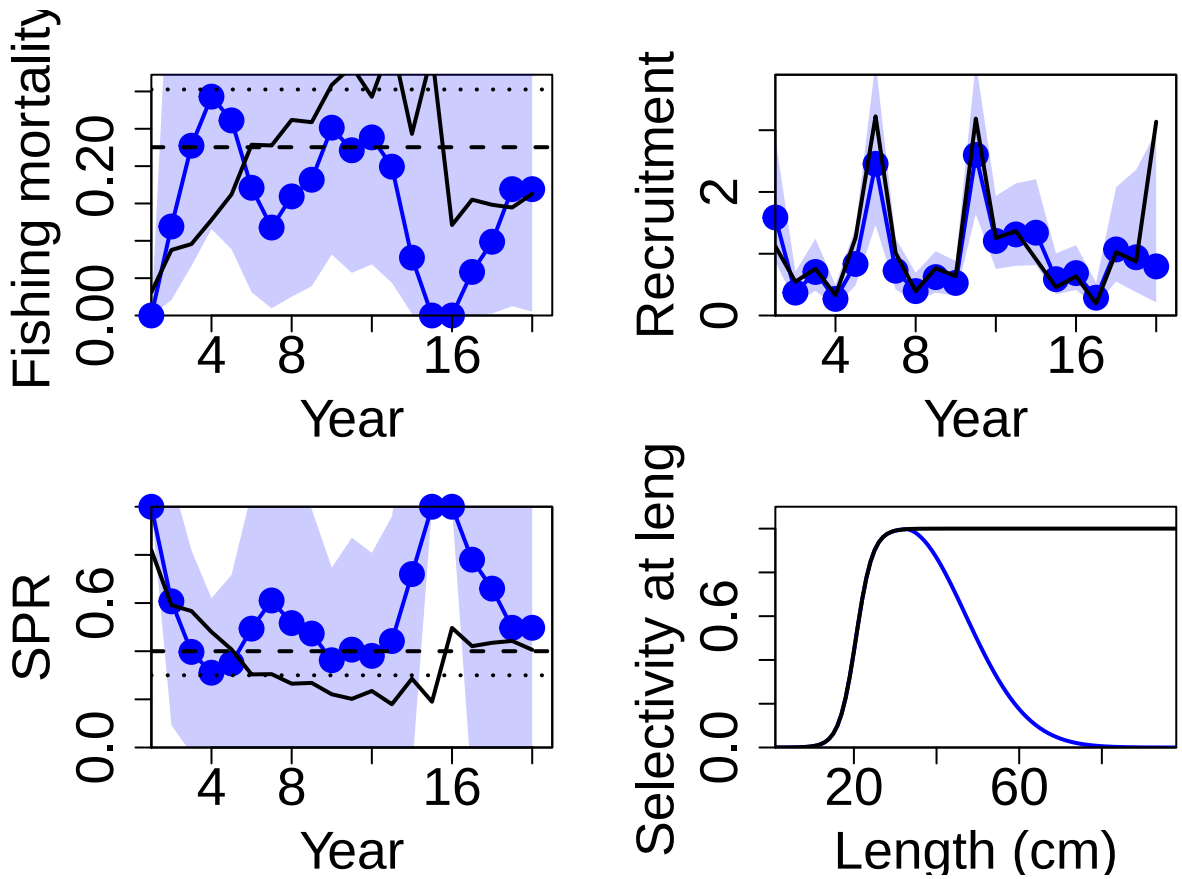
```
res_dome <- run_LIME(modpath=NULL,
                     lh=lh,
                     input_data=data_list,
                     est_sigma="log_sigma_R",
                     data_avail="LC",
                     S_l_input=lh_dome$S_l)
```

Then check how the assumed dome-shaped selectivity curve impacts assessment results:

```
plot_output(all_years=total_years, true_years=total_years, lc_years=length_years, Inputs=res_length$Inp
```




```
plot_output(all_years=total_years, true_years=total_years, lc_years=length_years, Inputs=res_dome$Inputs
```



References

Kristensen, K., Nielsen, A., Berg, C.W., Skaug, H., and Bell, B. 2016. TMB: Automatic Differentiation and Laplace Approximation. Journal of Statistical Software.

Rudd, M.B. and Thorson, J.T. 2017. Accounting for variable recruitment and fishing mortality in length-based stock assessments for data-limited fisheries. Canadian Journal of Fisheries and Aquatic Sciences. https://doi.org/10.1139/cjfas-2017-0143

Thorson, J.T., Jensen, O.P, and Zipkin, E.F. 2014. How variable is recruitment for exploited marine fishes? A hierarchical model for testing life history theory. Canadian Journal of Fisheries and Aquatic Sciences 71(7):973-983.

Thorson, J.T. Johnson, K.F., Methot, R.D., and Taylor, I.G. 2017. Model-based estimates of effective sample size in stock assessment models using the Dirichlet-multinomial distribution. Fisheries Research 192:84-93.