A Guide to Using ss3diags for Model Evaluation

Megumi Oshima[1], Eric Fletcher[2], Henning Winker[3], Massimiliano Cardinale[4], Laurence Kell[5], and Felipe Carvalho[1]

[1] NOAA Fisheries, Pacific Islands Fisheries Science Center

[2] Cooperative Institute for Marine and Atmospheric Research, University of Hawaii at Manoa

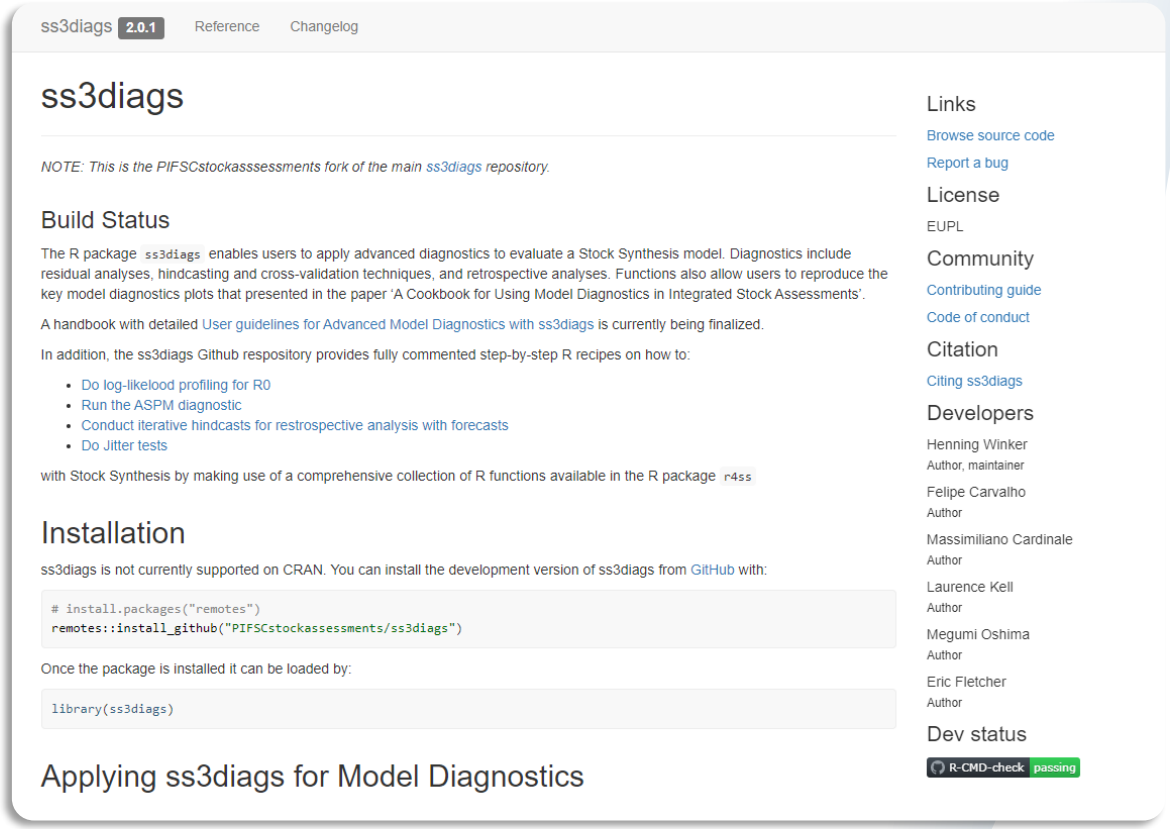
[3] Joint Research Center, European Commission

[4] Swedish University of Agricultural Sciences, Department of Aquatic Resources, Institute of Marine Research

[5] Center for Environmental Policy, Imperial College London

What is ss3diags?

- Functions for applying advanced model diagnostics to SS models
- R package
- Works with SS models of varying degrees of complexity



ss3diags 2.0.1 Reference Changelog

ss3diags

NOTE: This is the PIFSCstockasssessments fork of the main ss3diags repository.

Build Status

The R package `ss3diags` enables users to apply advanced diagnostics to evaluate a Stock Synthesis model. Diagnostics include residual analyses, hindcasting and cross-validation techniques, and retrospective analyses. Functions also allow users to reproduce the key model diagnostics plots that presented in the paper 'A Cookbook for Using Model Diagnostics in Integrated Stock Assessments'.

A handbook with detailed User guidelines for Advanced Model Diagnostics with ss3diags is currently being finalized.

In addition, the ss3diags Github respository provides fully commented step-by-step R recipes on how to:

- Do log-likelood profiling for R0
- Run the ASPM diagnostic
- Conduct iterative hindcasts for restrospective analysis with forecasts
- Do Jitter tests

with Stock Synthesis by making use of a comprehensive collection of R functions available in the R package `r4ss`

Installation

ss3diags is not currently supported on CRAN. You can install the development version of ss3diags from GitHub with:

```
# install.packages("remotes")
remotes::install_github("PIFSCstockassessments/ss3diags")
```

Once the package is installed it can be loaded by:

```
library(ss3diags)
```

Applying ss3diags for Model Diagnostics

Links
Browse source code
Report a bug
License
EUPL
Community
Contributing guide
Code of conduct
Citation
Citing ss3diags
Developers
Henning Winker
Author, maintainer
Felipe Carvalho
Author
Massimiliano Cardinale
Author
Laurence Kell
Author
Megumi Oshima
Author
Eric Fletcher
Author
Dev status
R-CMD-check passing

Motivation for ss3diags

Fisheries Research 240 (2021) 105959

Contents lists available at ScienceDirect

Fisheries Research

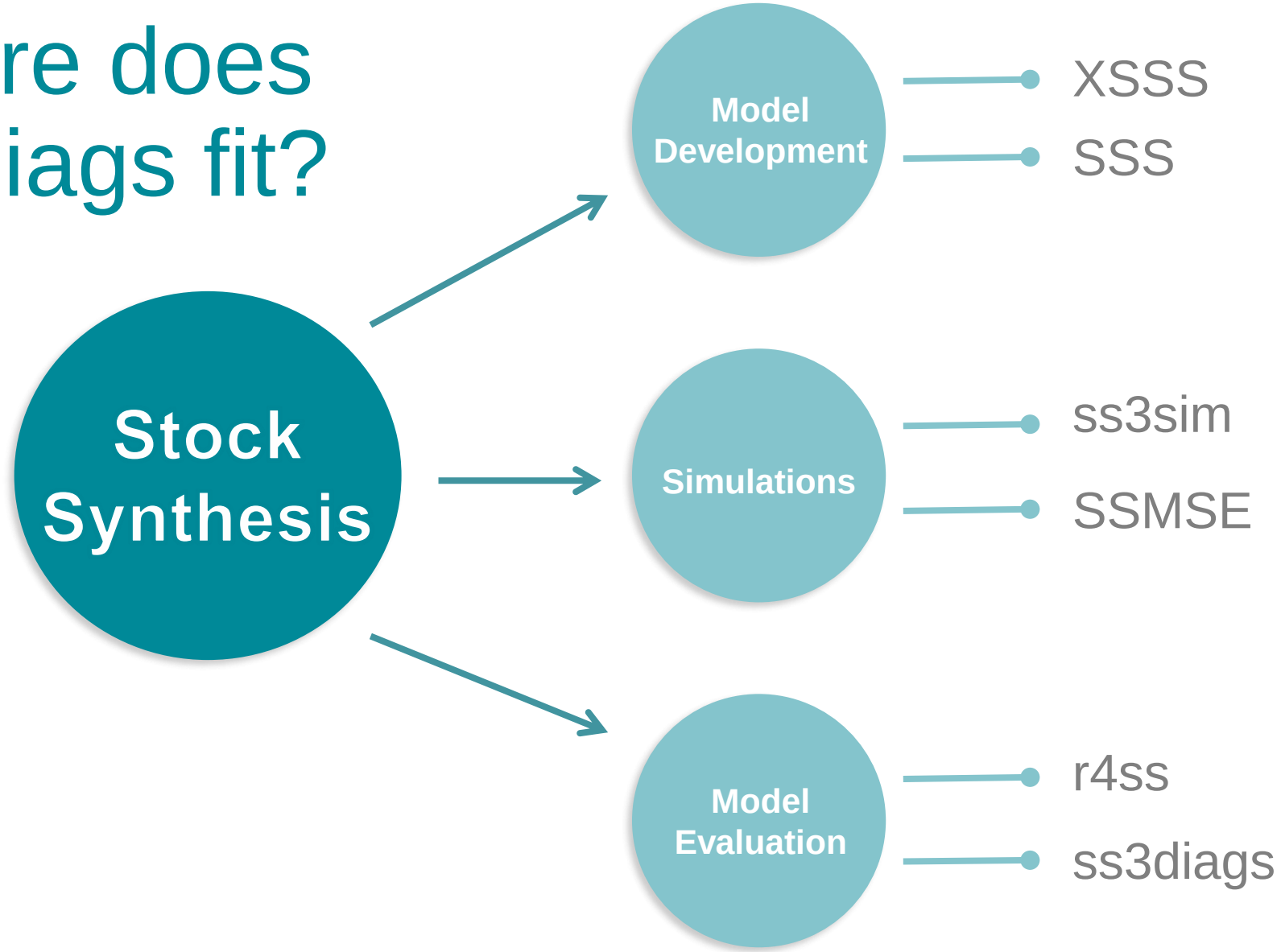
journal homepage: www.elsevier.com/locate/fishres

A cookbook for using model diagnostics in integrated stock assessments

Felipe Carvalho [a,*,1], Henning Winker [b,1], Dean Courtney [c], Maia Kapur [d], Laurence Kell [e], Massimiliano Cardinale [f], Michael Schirripa [g], Toshihide Kitakado [h], Dawit Yemane [i], Kevin R. Piner [j], Mark N. Maunder [k,l], Ian Taylor [m], Chantel R. Wetzel [m], Kathryn Doering [n], Kelli F. Johnson [m], Richard D. Methot [m]

"Enable efficient implementation of the presented diagnostics with Stock Synthesis"

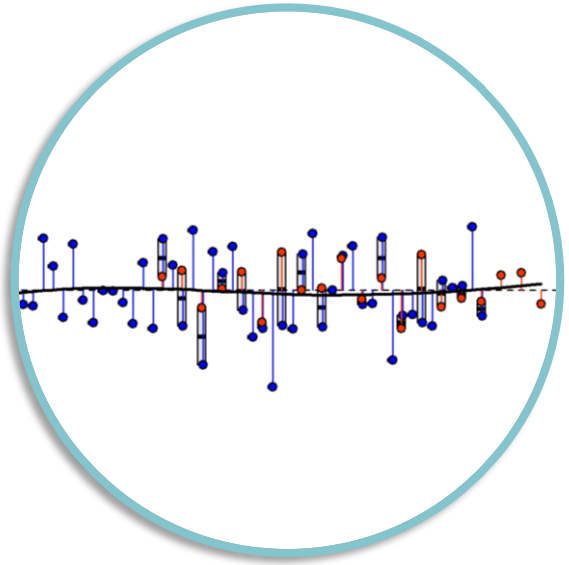
Where does ss3diags fit?



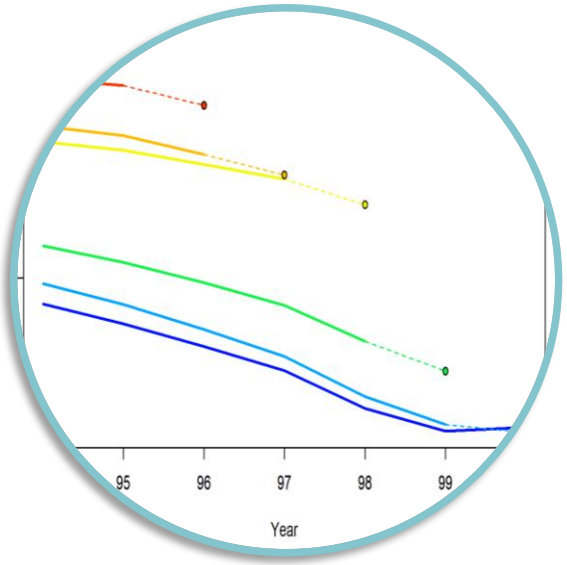
NOAA FISHERIES

ss3diags Assists With:

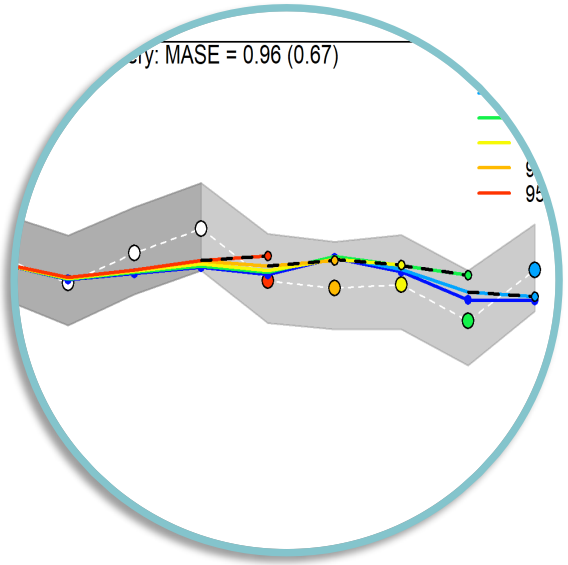
Goodness-of-Fit



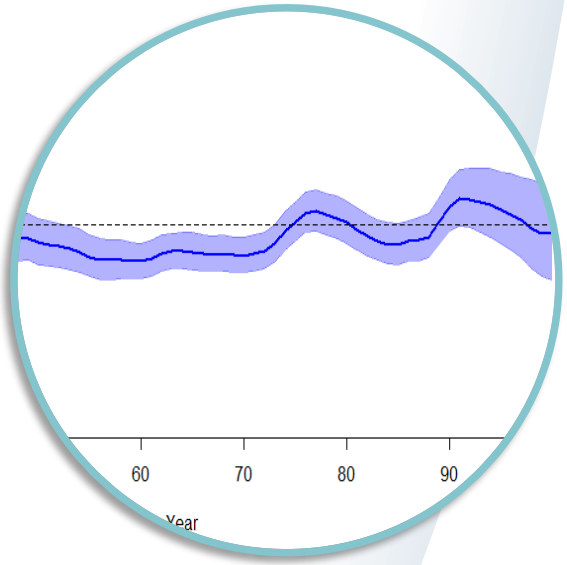
Retrospective and Forecast Bias



Prediction Skill



Model Uncertainty



NOAA FISHERIES

Example with ss3diags

Quick guide for key functions and their outputs

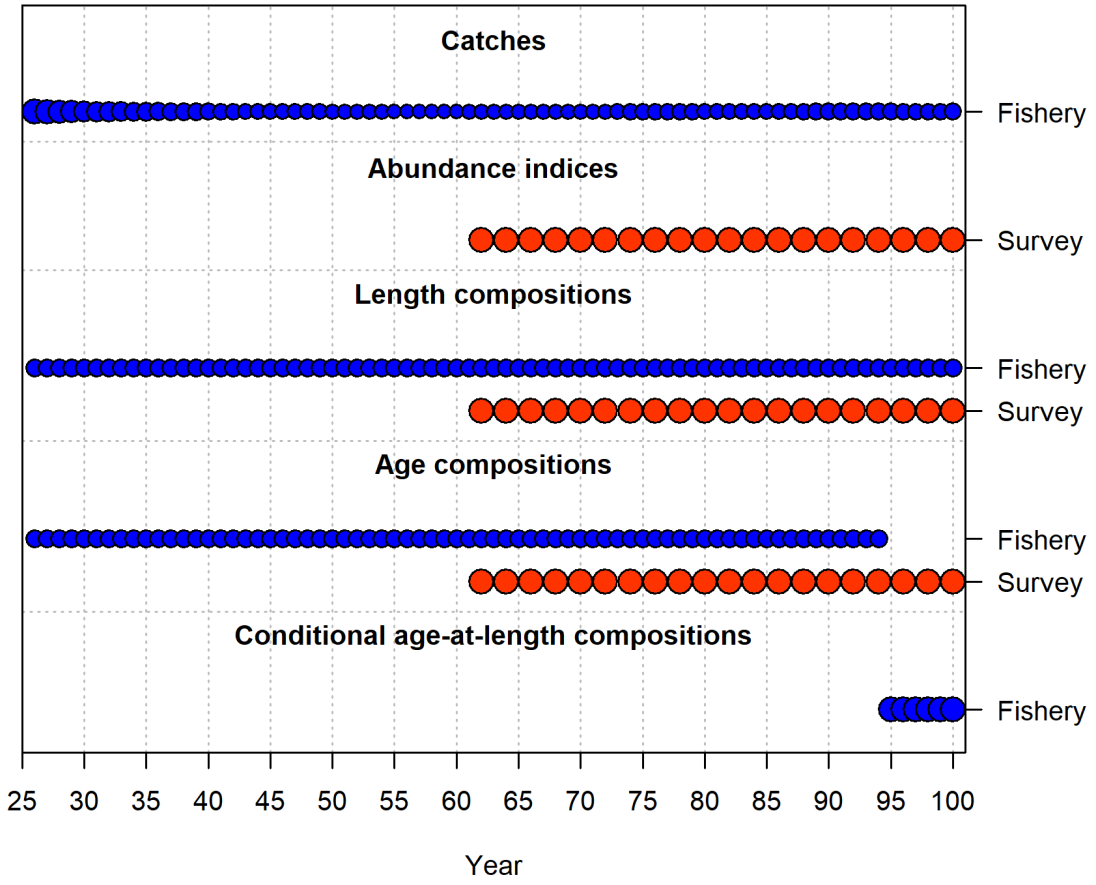
NOAA FISHERIES

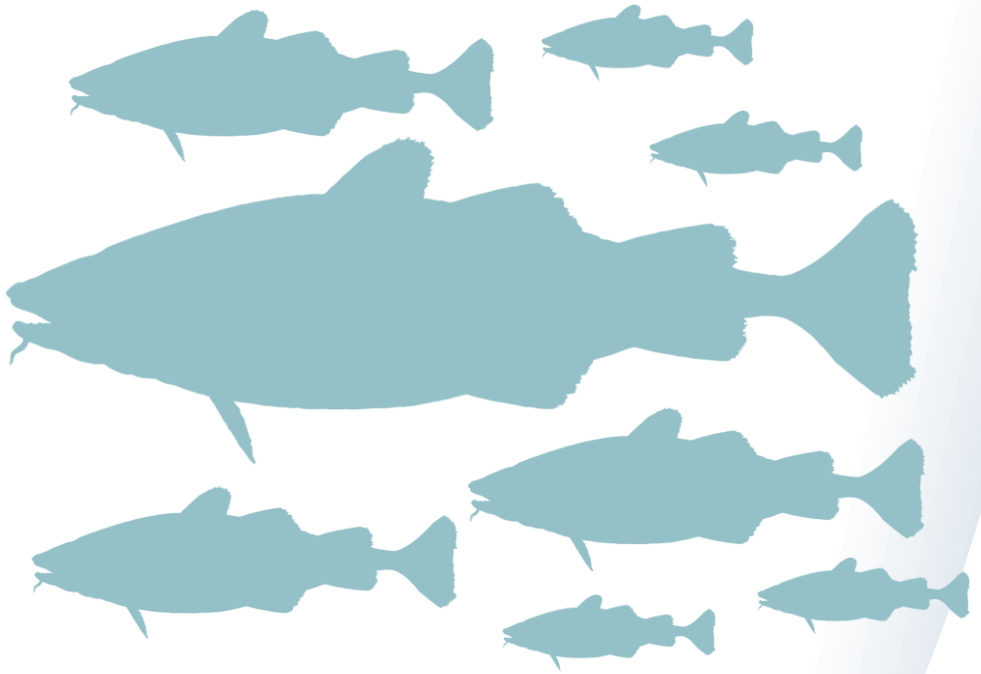
Setup

- Install and load package
- Load example SS model

```
# install.packages("remotes")
remotes::install_github("PIFSCstockassessments/ss3diags")
library(ss3diags)

data("simple")
```

Example with ss3diags



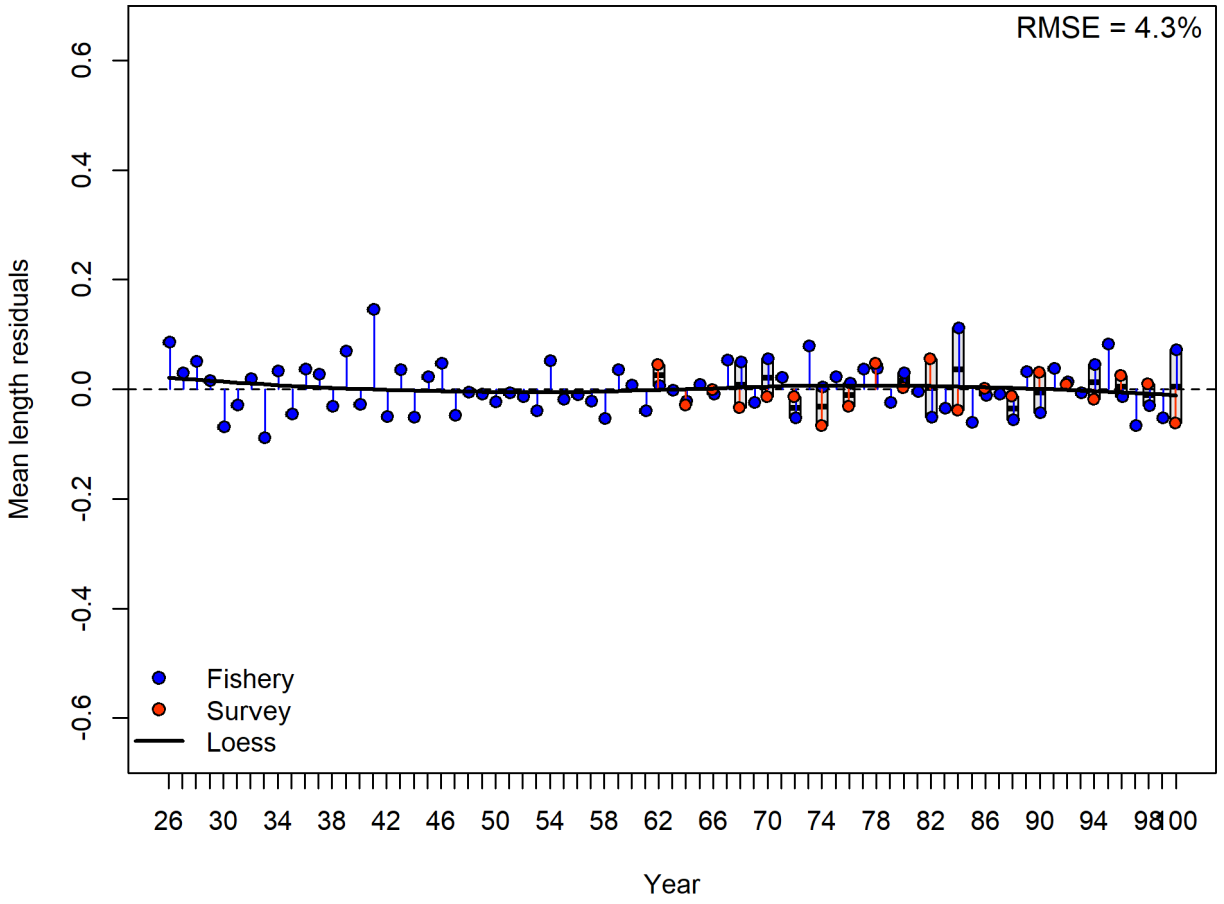


Residual Analysis: Goodness-of-Fit

```
1  SSplotJABBAres(simple, subplots = "len")
2  SSplotJABBAres(simple, subplots = "age")
```

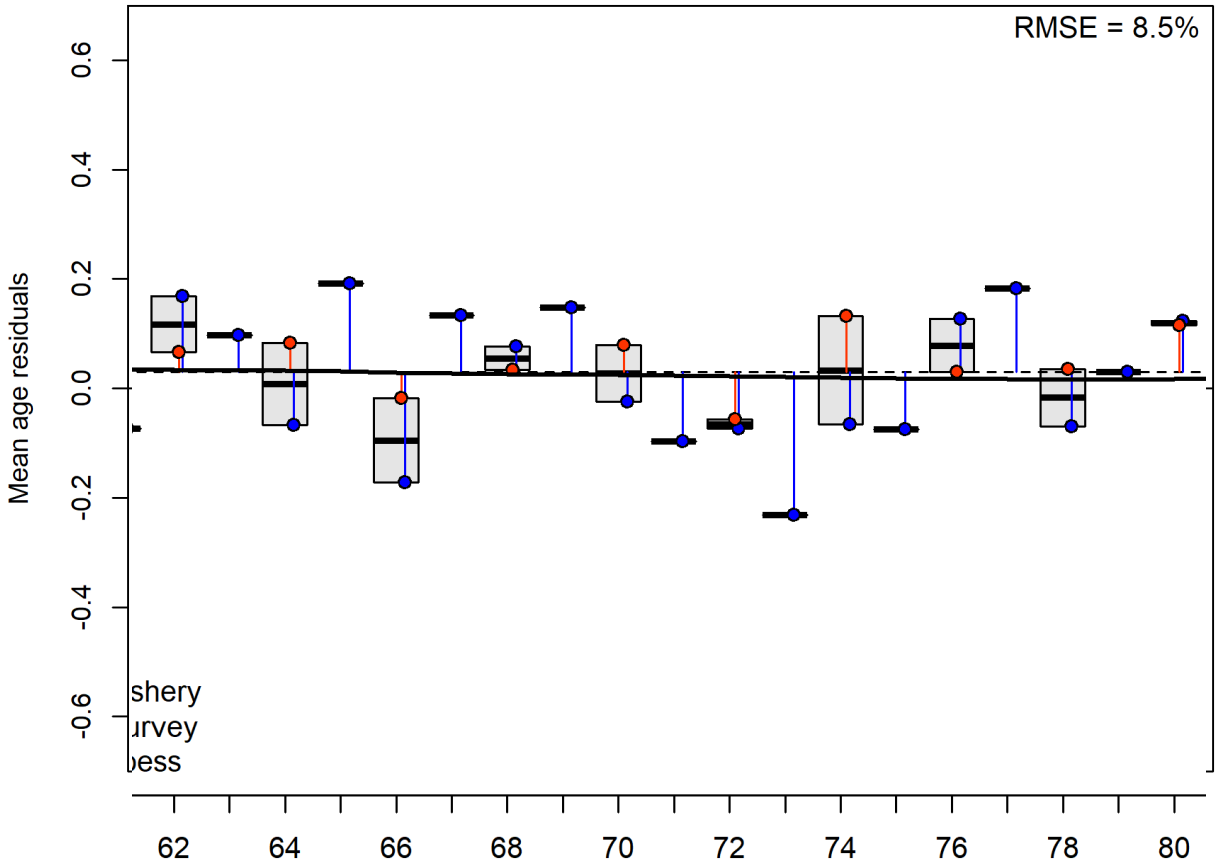

NOAA FISHERIES

SSplotJABBAres() Output




```
> SSplotJABBAres(simple, subplots = "len", add = TRUE)
Plotting JABBA residual plot

RMSE stats by Index:
   indices RMSE.perc nobs
1  Fishery       4.5   75
2   Survey       8.8   20
3 Combined       4.3   95
```

SSplotJABBAres() Output




```
> SSplotJABBAres(simple, subplots = "age", add = TRUE)
Plotting JABBA residual plot

RMSE stats by Index:
   indices RMSE.perc nobs
1  Fishery       9.3   69
2   Survey      17.3   20
3 Combined       8.5   89
```

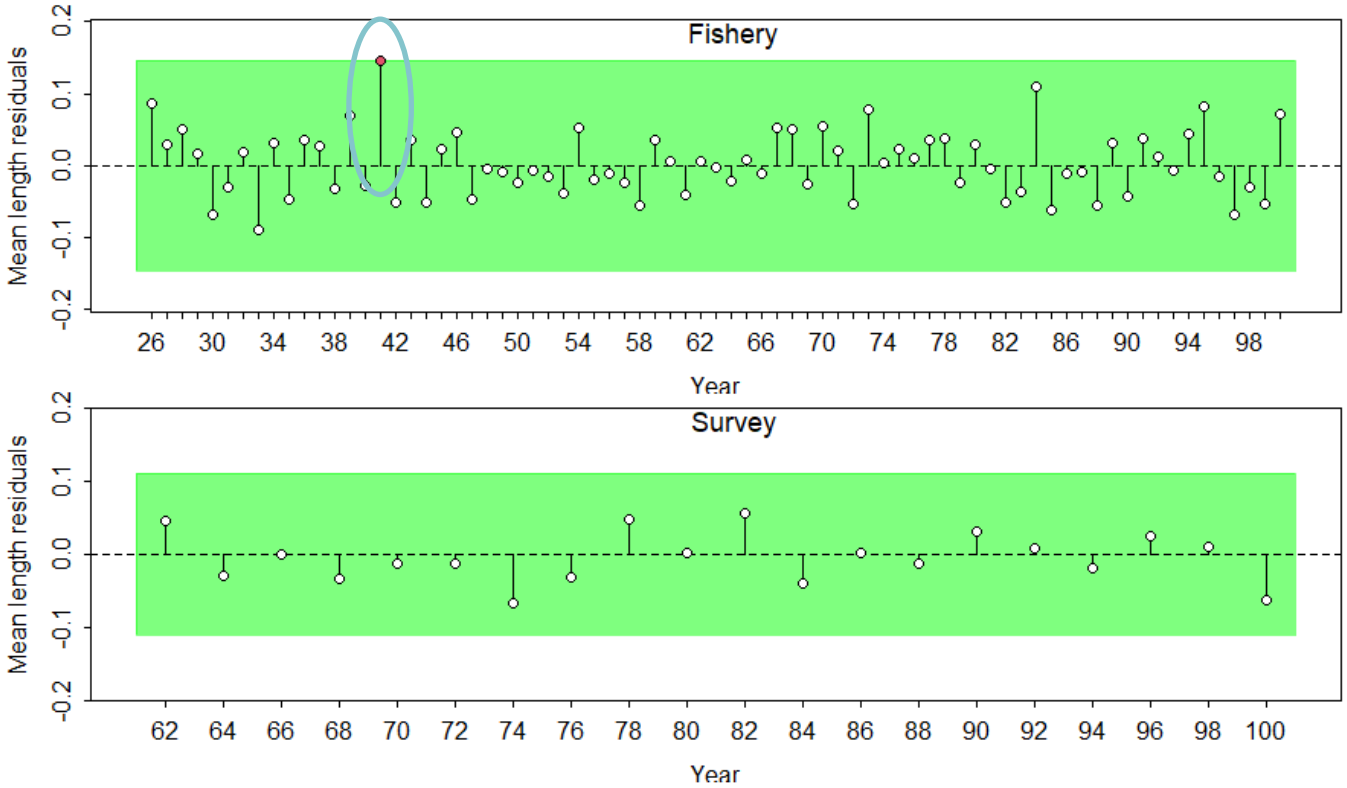

NOAA FISHERIES

Residual Analysis: Runs Test

```
# Length Composition
SSplotRunstest(simple, subplots = "len", add = TRUE)
# CPUE
SSplotRunstest(simple, subplots = "cpue")
```


NOAA FISHERIES

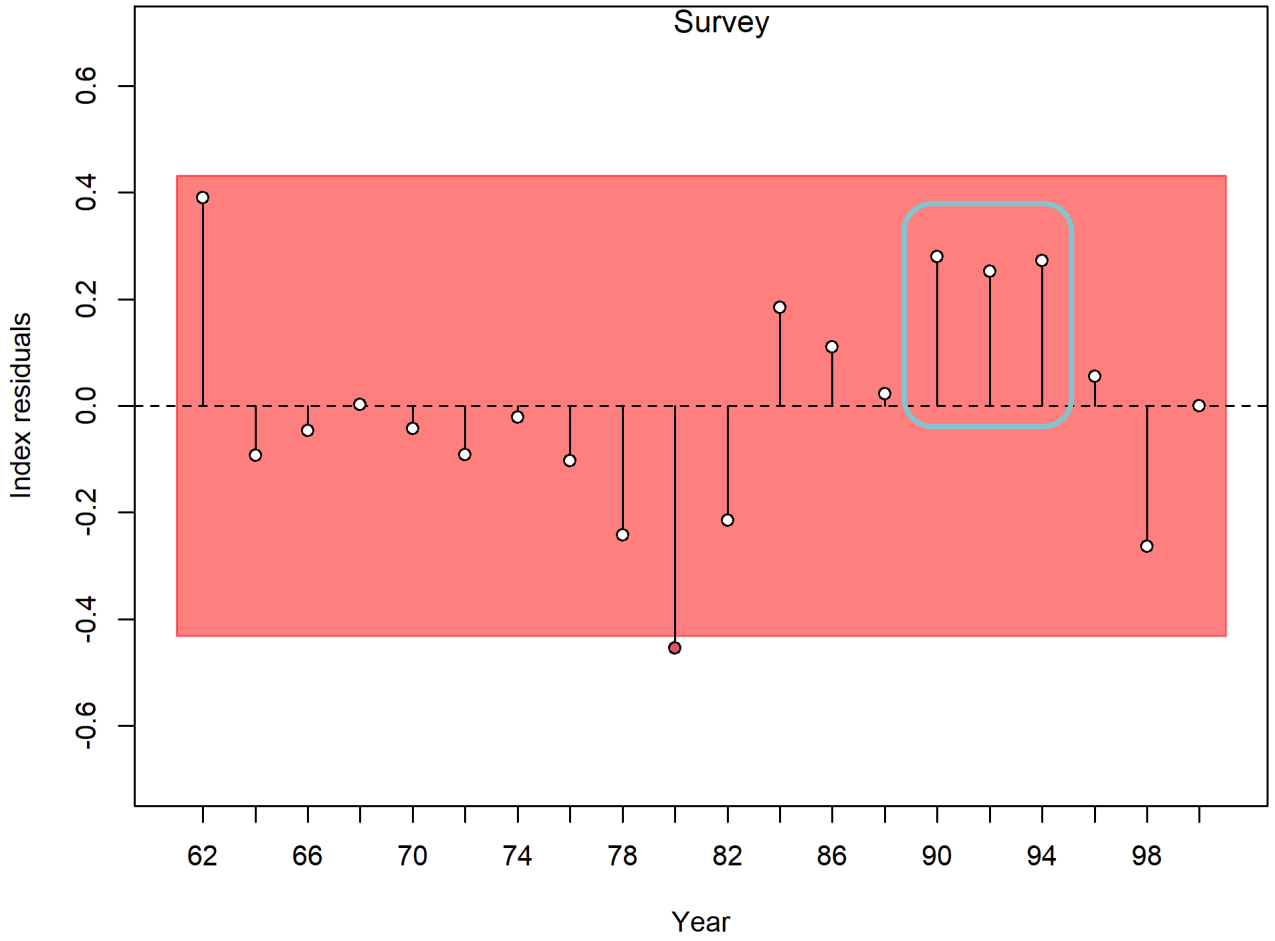
SSplotRunstest() Output




```
> SSplotRunstest(simple, subplots = "len", add = TRUE)

 Running Runs Test Diagnosics for Mean length
Plotting Residual Runs Tests

Runs Test stats by Mean length:
     Index runs.p   test  sigma3.lo sigma3.hi type
1 Fishery  0.724 Passed -0.1454301 0.1454301  len
2  Survey  0.338 Passed -0.1105796 0.1105796  len
```


NOAA FISHERIES

SSplotRunstest() Output

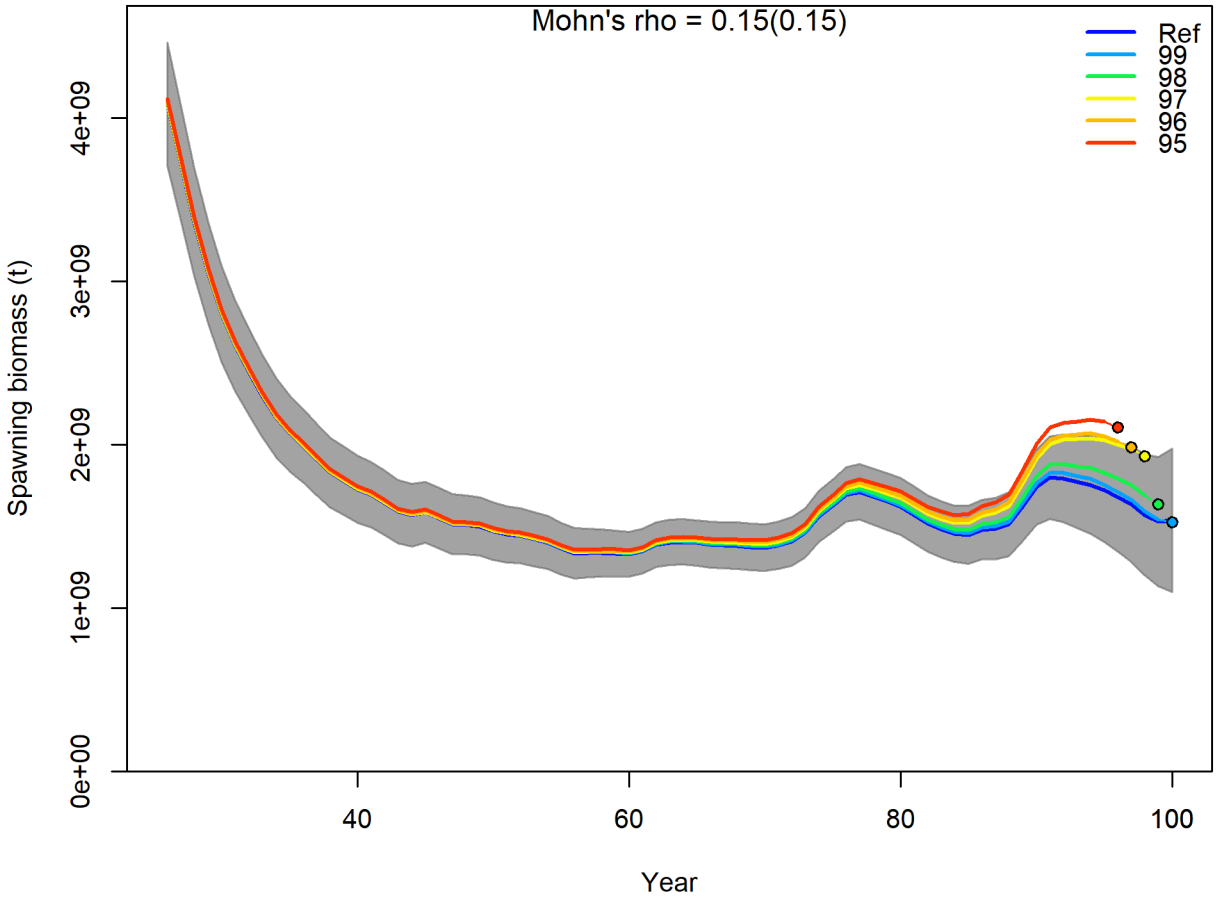



```
> SSplotRunstest(simple, subplots = "cpue")

 Running Runs Test Diagnosics for Index
Plotting Residual Runs Tests

Runs Test stats by Index:
    Index runs.p   test  sigma3.lo sigma3.hi type
1 Survey  0.033 Failed -0.4320694 0.4320694 cpue
```

Retrospective Bias

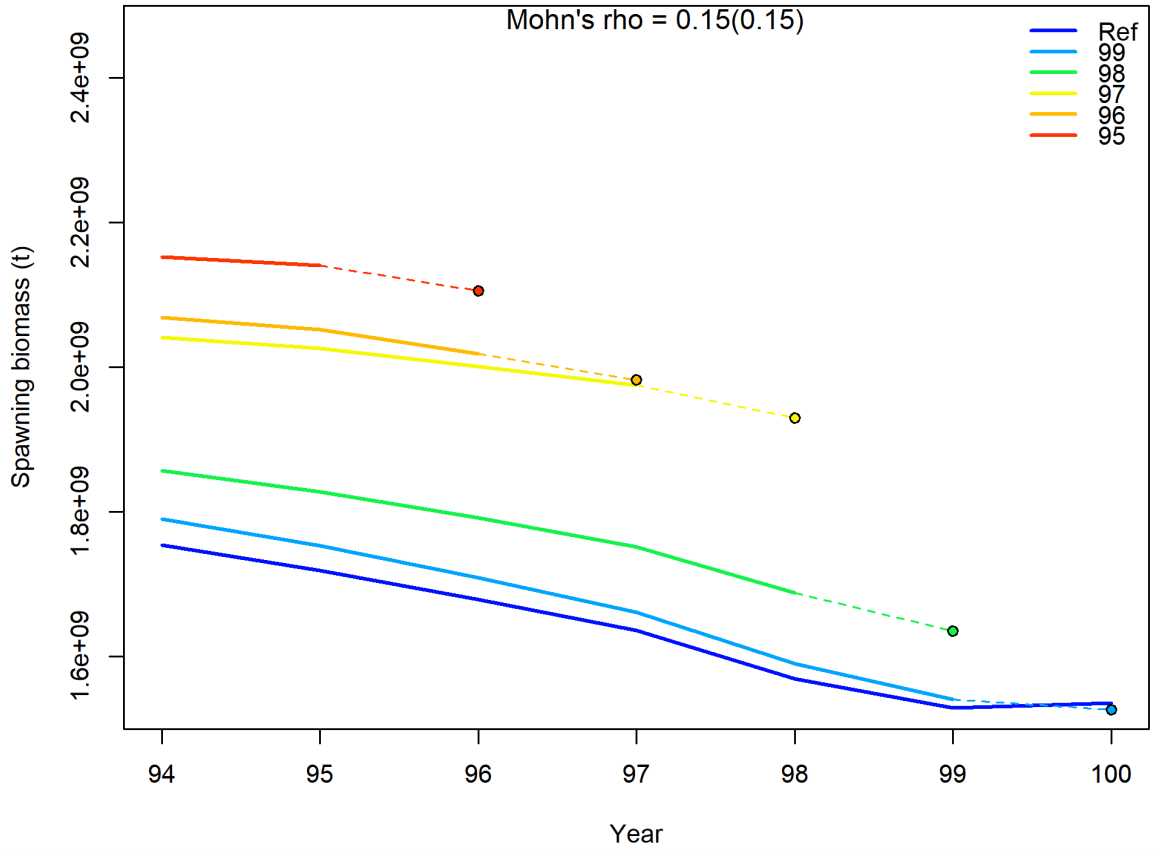
```
1  data("retroSimple")
2
3  sumSimple <- r4ss::SSsummarize(retroSimple)
4
5  SSplotRetro(sumSimple, subplots = "SSB")
6  SShcbias(sumSimple, quants = "SSB")
7
```

SSplotRetro() Output




```
> SSplotRetro(sumSimple, subplots = "SSB")
Plotting Retrospective pattern

Mohn's Rho stats, including one step ahead forecasts:
  type     peel         Rho  ForecastRho
1  SSB       99 0.007769174 -0.006152424
2  SSB       98 0.075590953  0.069386314
3  SSB       97 0.207121898  0.229780185
4  SSB       96 0.202493492  0.211816848
5  SSB       95 0.245173711  0.254376716
6  SSB Combined 0.147629846  0.151841528
```

SSplotRetro(forecast = TRUE)



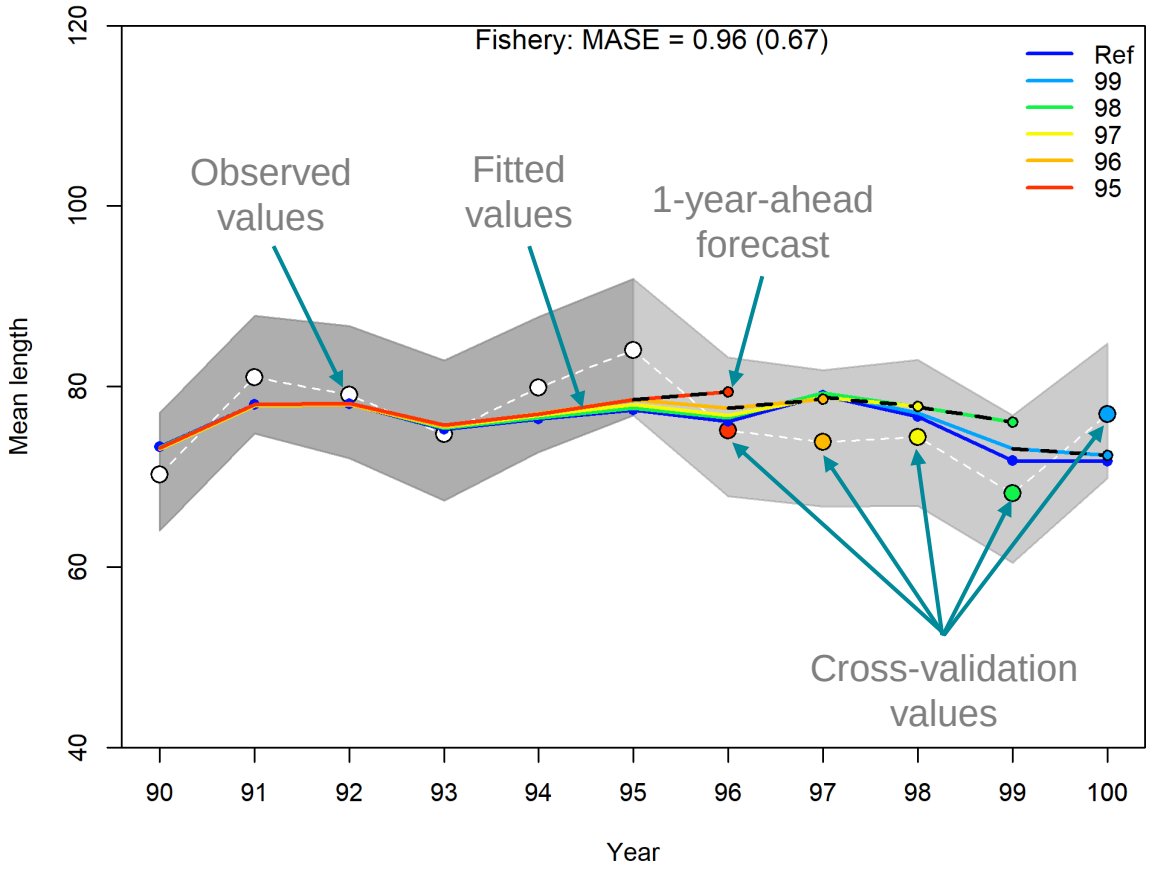

```
1  SSplotRetro(sumSimple, subplots = "SSB", forecast = TRUE, uncertainty = FALSE, xlim = c(94,100))
```

Hindcast Cross-Validation

```r
# Composition data
retroC.simple <- SSretroComps(retroSimple)
SSplotHCxval(retroC.simple, subplots = "age")

# Index data
retroI.simple <- r4ss::SSsummarize(retroSimple)
SSplotHCxval(retroI.simple, subplots = "cpue")

SSmase(retroSimple, quant="len", indexselect = c(1:2))
```


NOAA FISHERIES

SSplotHCxval() Output




```
> SSplotHCxval(retroI.simple, subplots = "len")
Plotting Hindcast Cross-Validation (one-step-ahead)

MASE stats by Index:
     Index Season      MASE    MAE.PR   MAE.base MASE.adj n.eval
1 Fishery      1 0.9635032 0.0666456 0.06917009 0.666456      5
2  Survey      1 0.4246802 0.0420929 0.09911671 0.420929      2
```

SSmase()

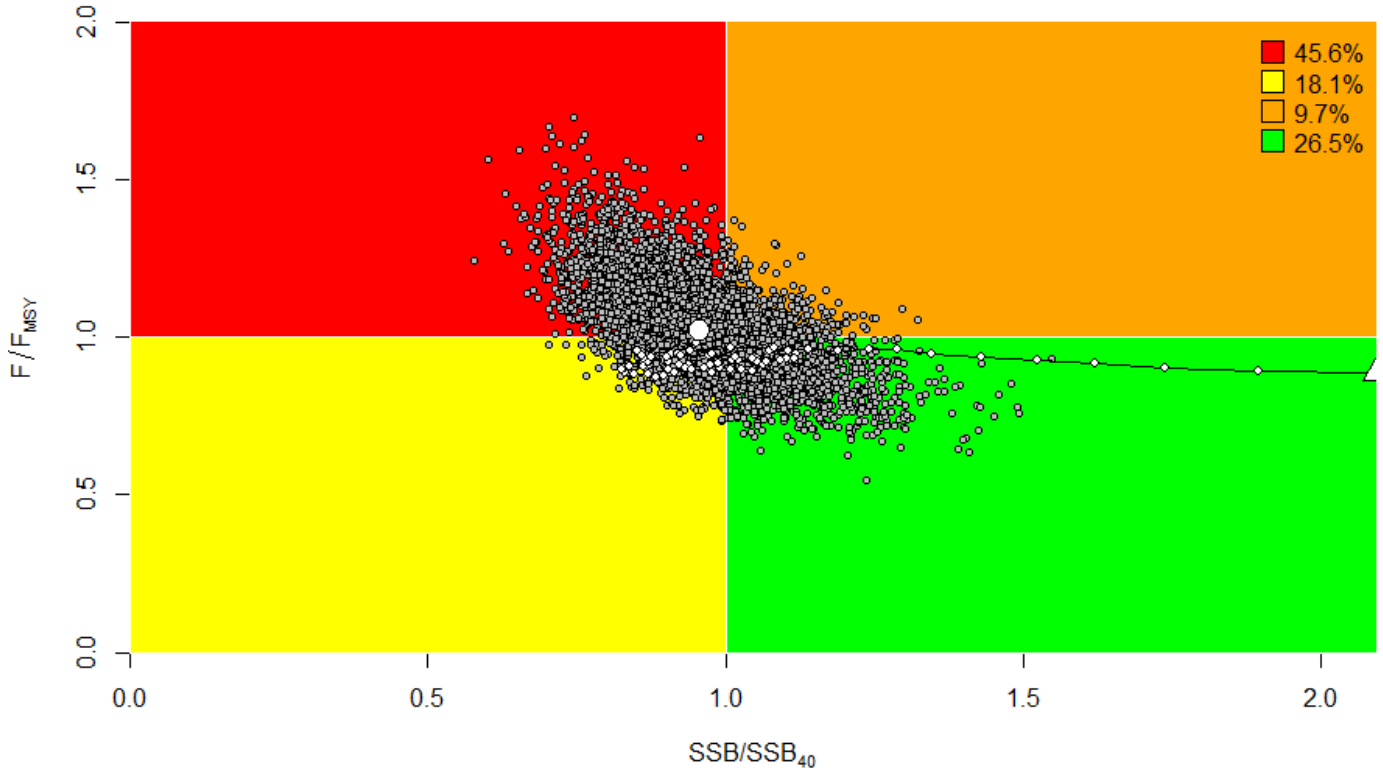
```
> SSmase(retroSimple,quant="len",MAE.base.adj = 0.15,indexselect = c(1:2))
Converting retroSummary to summarized list using ss3diags::SSretroComps()
     Index Season      MASE     MAE.PR   MAE.base  MASE.adj n.eval
1 Fishery      1 0.9635032 0.06664560 0.06917009 0.4443040      5
2  Survey      1 0.2433708 0.02412211 0.09911671 0.1608141      2
3   joint        0.7011276 0.05449603 0.07772627 0.3633069      7
```


NOAA FISHERIES

Model Uncertainty

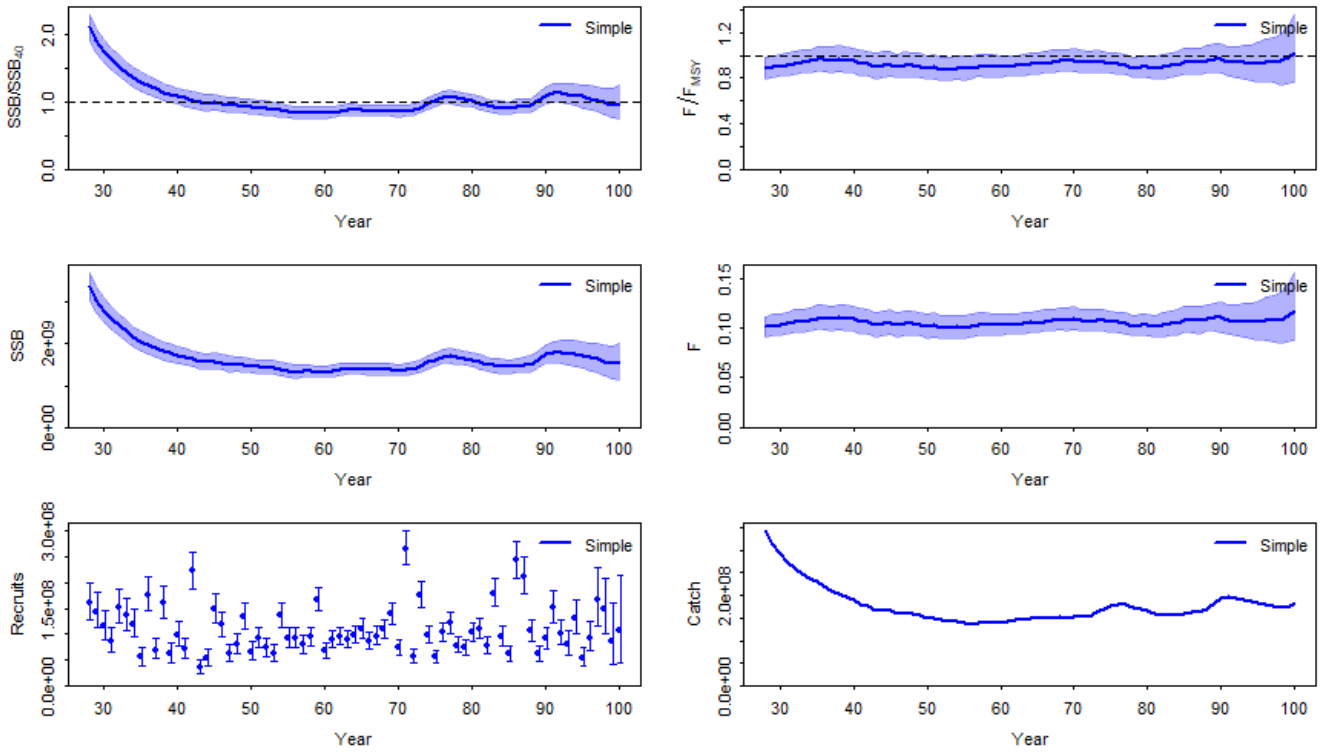
```r
mvln <- SSdeltaMVLN(simple, run = "Simple")

sspar(mfrow = c(3, 2), plot.cex = 0.7)
SSplotEnsemble(mvln$kb, ylabs = mvln$labels, add = T)
```

SSdeltaMVLN() Output




```
> str(mvln)
List of 4
$ kb    :'data.frame': 365000 obs. of  10 variables:
  ..$ year   : num [1:365000] 28 28 28 28 28 28 28 28 28 28 ...
  ..$ run    : chr [1:365000] "Simple" "Simple" "Simple" "Simple" ...
  ..$ type   : chr [1:365000] "fit" "fit" "fit" "fit" ...
  ..$ iter   : int [1:365000] 1 2 3 4 5 6 7 8 9 10 ...
  ..$ stock  : num [1:365000] 2.1 2 2.16 2.06 2.23 ...
  ..$ harvest: num [1:365000] 0.889 0.956 0.85 0.874 0.867 ...
  ..$ SSB    : num [1:365000] 3.21e+09 3.53e+09 3.26e+09 3.22e+09 3.44e+09 ...
  ..$ F      : num [1:365000] 0.1005 0.1089 0.0967 0.1004 0.0984 ...
  ..$ Recr   : num [1:365000] 1.50e+08 1.56e+08 1.31e+08 1.43e+08 1.80e+08 ...
  ..$ Catch  : num [1:365000] 3.42e+08 3.42e+08 3.42e+08 3.42e+08 3.42e+08 ...
$ mle   :'data.frame': 73 obs. of  9 variables:
  ..$ year   : num [1:73] 28 29 30 31 32 33 34 35 36 37 ...
  ..$ run    : chr [1:73] "Simple" "Simple" "Simple" "Simple" ...
  ..$ type   : chr [1:73] "fit" "fit" "fit" "fit" ...
  ..$ stock  : num [1:73] 2.09 1.9 1.74 1.62 1.52 ...
  ..$ harvest: num [1:73] 0.885 0.894 0.903 0.915 0.929 ...
  ..$ SSB    : num [1:73] 3.36e+09 3.05e+09 2.80e+09 2.60e+09 2.45e+09 ...
  ..$ F      : num [1:73] 0.101 0.102 0.103 0.104 0.106 ...
  ..$ Recr   : num [1:73] 1.60e+08 1.44e+08 1.15e+08 8.75e+07 1.53e+08 ...
  ..$ Catch  : num [1:73] 3.42e+08 3.14e+08 2.91e+08 2.75e+08 2.62e+08 ...
$ quants: chr [1:6] "stock" "harvest" "SSB" "F" ...
$ labels:  expression("SSB/SSB"[40], F/F[MSY], "SSB") ...
```

SSplotEnsemble() Output




```
> head(mvln$kb)
  year    run type iter    stock   harvest       SSB          F      Recr     Catch
1   28 Simple  fit    1 2.102307 0.8890874 3212569527 0.10048893 149761715 342373000
2   28 Simple  fit    2 2.000164 0.9558910 3532394198 0.10894144 156495955 342373000
3   28 Simple  fit    3 2.159903 0.8495990 3262712456 0.09670561 131017940 342373000
4   28 Simple  fit    4 2.063483 0.8736431 3217280078 0.10036551 142594947 342373000
5   28 Simple  fit    5 2.229065 0.8666970 3441154638 0.09844169 179595741 342373000
6   28 Simple  fit    6 2.021941 0.9625931 3297950980 0.10989031 148291848 342373000
```

Future Implementations

- RMSE for recruitment predictions
- Include option for conditional age-at-length data for the hindcast cross-validation

Thank you!

If you have any suggestions or would like to contribute to ss3diags you can submit an issue in the repository or email Megumi.Oshima@noaa.gov

https://github.com/PIFSCstockassessments/ss3diags

NOAA FISHERIES