User Guidelines for Advanced Model Diagnostics with ss3diags

Hennig Winker (JRC-EC)   Felipe Carvalho (NOAA)
Massimiliano Cardinale (SLU)   Laurence Kell (Sea++)

14 April, 2022

Contents

1 Getting started

This vignette introduces you to the `ss3diags` R package, which accompanies the paper "A cookbook for using model diagnostics in integrated stock assessments" by Carvalho, Winker et al. (2021).

The `ss3diags` comprises a set of functions for applying advanced model diagnostics to Stock Synthesis models. The package builds on the widely used R package `r4ss` (Taylor et al. 2021), which is designed to support the use of the Stock Synthesis software modeling framework (Methot and Wetzel, 2013). This vignette is divided into four sections. Section 1 consists of installing `ss3diags` and loading the example data from a simulated, cod-like stock that is included with the package. Section 2 describes the plotting of various model diagnostics as described in the Cookbook. Section 3 provides a detailed explanation on how to assess model uncertainty using `ss3diags`. In Section 4 we provide a series of "cookbook recipes" on how to implement selected model diagnostics on Stock Synthesis models.

1.1 Installation

Both `ss3diags` and `r4ss` can be installed from gihtub using the `remotes` package:

```
install.packages("remotes")

remotes::install_github("r4ss/r4ss")

remotes::install_github("PIFSCstockassessments/ss3diags")
```

Once the packages are installed they can be loaded by:

```r
library(r4ss)
library(ss3diags)
```

1.2 Loading built-in example data

The package contains output from a simple, cod-like SS model that was simulated using ss3sim. The model includes 2 fleets, one fishery and one survey. Catch data is available from year 26 to year 100 (final year of model). An index of abundance is available from the survey fleet for years 62 - 100. No discard data was simulated. Simulated composition data includes length (fleets 1 and 2), age (fleets 1 and 2), and conditional age-at-length (fleet 1). Examples of the the output of a single run (as read by `r4ss::SS_output()`) of the model as well as the output from a retrospective analysis with 5 year peels (as read by `r4ss::SSgetout()`) are available with the package.

1.2.1 "Simple" model

The example outputs can be loaded into R by:

```r
# Single run output
data("simple")

# retrospective analysis output
data("retroSimple")
```

- `simple`: list of stock synthesis objects created with `r4ss::SS_output()`
- `retroSimple`: list of retrospective runs created with `r4ss:SS_doRetro()` and read by `r4ss::SSgetoutput()`.
    - The first object in the list is the reference run and the other 5 objects are the 5 1-year peels.

1.2.1.0.1 WE DONT HAVE AN ASPM MODEL EXAMPLE ANYMORE

- `aspm.phk`: Comprises of two runs the "Reference_Run" and the "ASPM", which can be loaded together using `r4ss::SSgetoutput()` and then summarized with `r4ss::SSsummarize()`

```r
asem.phk <- r4ss::SSgetoutput(dirvec = file.path(dir.path, paste0("Reference_Run",
    "APSM")))
asem.phk <- r4ss::SSsummarize(asem.phk)
```

2 Model Diagnostics with ss3diags

The plotting options are kept mainly to those provided by r4ss. Like with r4ss, if, for example, 'SSplotRunstest() called with no further specifications several windows will open, the number of windows depends on the number abundance indices.

2.1 Residual diagnostics

The runs test is a nonparametric hypothesis test for randomness in a data sequence that calculates the 2-sided p-value to estimate the number of runs (i.e., sequences of values of the same sign) above and below a reference value. The runs test can diagnose model misspecification using residuals from fits to abundance indices Carvalho et al. 2017 by testing if there are non-random patterns in the residuals. It can also be applied to other data components in assessment models such as the mean-length residuals and mean-age residuals. In addition, the three-sigma limits can be considered to identify potential outliers as any data point would be unlikely given a random process error in the observed residual distribution if it is further than three standard deviations away from the expected residual process average of zero.

The output for `SSplotRunstest()` includes a plot of the residuals by fleet and a table with the results from the runs test and 'three-sigma limit' values. In the plots below, the shaded area represents the 'three-sigma limit', or three residual standard deviations from zero. If any of the individual residual points fall outside of the three-sigma limit, they are colored red as in the fishery length-composition. Green shaded area indicates the residuals are randomly distributed (p-value >= 0.05) and red shaded area indicates the residuals are not randomly distributed and there is some misspecification with the indices or composition data (p-value < 0.05). To visualize the runs test for multiple indices, it is recommended to use the function `r4ss::sspar()` to specify row and column layout and any other plotting parameters. The option `add=TRUE` included in any of the ss3diags plotting functions prevents the functions from over-writing `sspar()`.

```
r4ss::sspar(mfrow = c(2, 2))
SSplotRunstest(simple, subplots = "cpue", add = TRUE)
   Running Runs Test Diagnostics w/ plots forIndex
   Plotting Residual Runs Tests
   Residual Runs Test (/w plot) stats by Index:
      Index runs.p   test  sigma3.lo sigma3.hi type
   1 Survey  0.033 Failed -0.4320694 0.4320694 cpue
SSplotRunstest(simple, subplots = "len", add = TRUE)
   Running Runs Test Diagnostics w/ plots forMean length
   Plotting Residual Runs Tests
   Residual Runs Test (/w plot) 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
SSplotRunstest(simple, subplots = "con", add = TRUE)
   Running Runs Test Diagnostics w/ plots forConditional age-at-length
   Plotting Residual Runs Tests
```

```
   Residual Runs Test (/w plot) stats by Conditional age-at-length:
       Index runs.p   test  sigma3.lo sigma3.hi type
   1 Fishery    0.5 Passed -0.1491212 0.1491212  con
```

It is also possible to select the indices that should be plotted. For example, if we only want to plot the fishery length composition residuals, we can specify this with the `indexselect` argument.

```
r4ss::sspar()
SSplotRunstest(simple, subplots = "len", indexselect = 1, add = TRUE)
   Running Runs Test Diagnostics w/ plots forMean length
   Plotting Residual Runs Tests
```

```
   Residual Runs Test (/w plot) stats by Mean length:
       Index runs.p   test  sigma3.lo sigma3.hi type
   1 Fishery  0.724 Passed -0.1454301 0.1454301  len
```

In addition to the residual plots, `SSplotRunstest()` produces a summary table of the runs test output values, including:

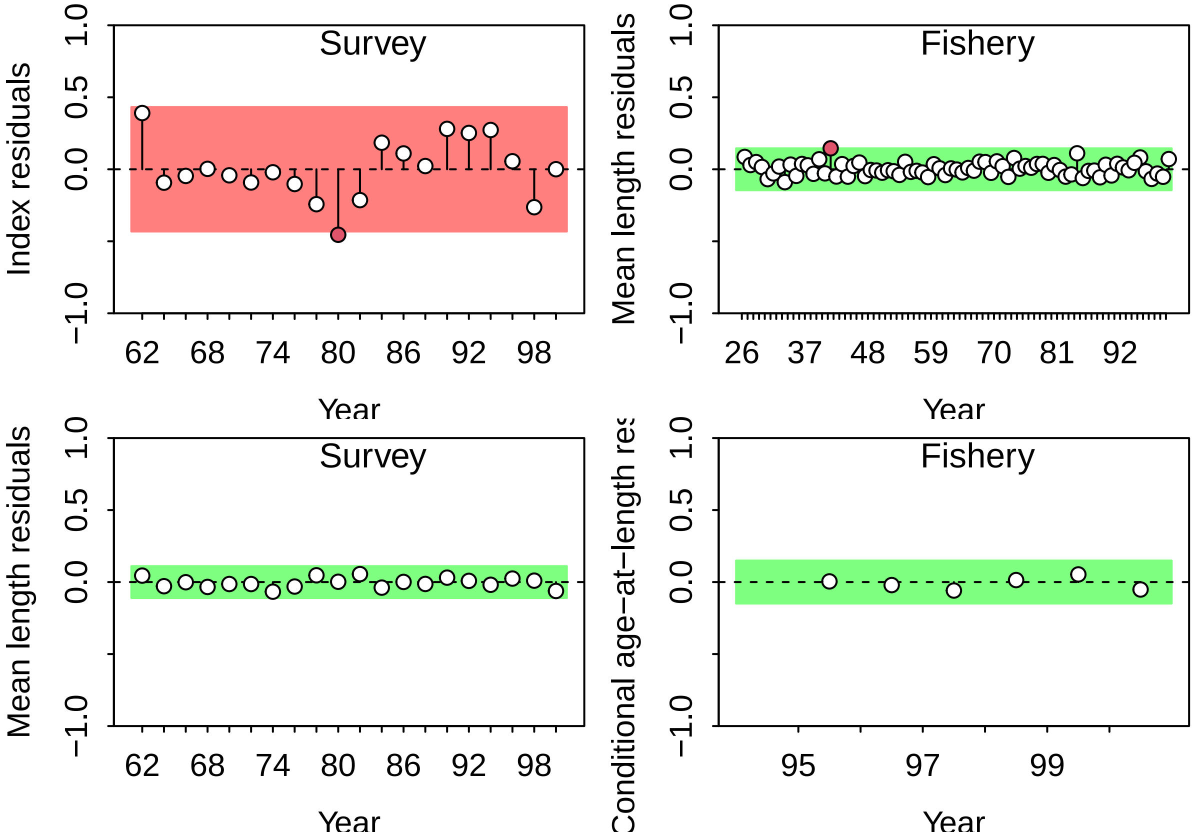

Figure 1: Runs test plots for CPUE index fits. Green shading indicates no evidence (p = 0.05) and red shading evidence ($p < 0.05$) to reject the hypothesis of a randomly distributed time-series of residuals. The shaded (green/red) area spans three residual standard deviations to either side from zero, and the red points outside of the shading violate the 'three-sigma limit' for that series.

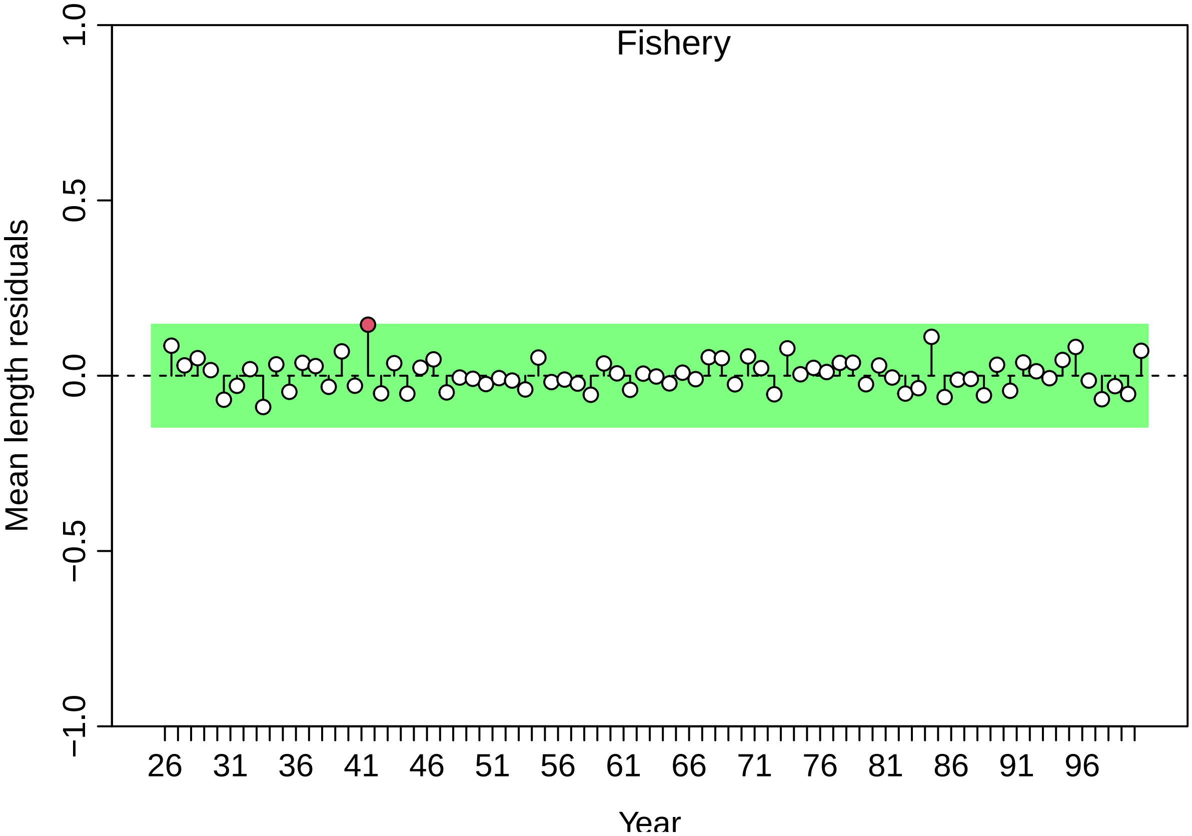

Figure 2: Runs test plot for fits to fishery length composition data.

- p-value for the runs test

- if the test passed or failed (indicated by green or red shading in the plot)
- upper and lower limits for the 3-sigma interval
- type of data tested (CPUE, length-composition, age-composition, or conditional age-at-length)

To only produce the summary table and skip the plot, e.g. to faciliate automated processing, use `SSrunstest()`.

```
rcpue <- SSrunstest(simple, quants = "cpue")
   Running Runs Test Diagnosics for Index
   Computing Residual Runs Tests
   Residual Runs Test stats by Index:
rlen <- SSrunstest(simple, quants = "len")
   Running Runs Test Diagnosics for Mean length
   Computing Residual Runs Tests
   Residual Runs Test stats by Mean length:
rbind(rcpue, rlen)
       Index runs.p   test  sigma3.lo sigma3.hi type
   1  Survey  0.033 Failed -0.4320694 0.4320694 cpue
   2 Fishery  0.724 Passed -0.1454301 0.1454301  len
   3  Survey  0.338 Passed -0.1105796 0.1105796  len
```

The second function for residual diagnostics is the function `SSplotJABBAres()`. This function is from the R package JABBA and plots a time series of residuals for all fleets of the indicated data (CPUE or composition). In the example below, we plot the residuals for the mean age (age-composition) and mean length (length-composition) for both fleets.

```
r4ss::sspar(mfrow = c(1, 2), plot.cex = 0.8)
SSplotJABBAres(simple, subplots = "age", add = TRUE)
   RMSE stats by Index:
      indices RMSE.perc nobs
   1  Fishery       9.3   69
   2   Survey       5.1   20
   3 Combined       8.5   89
SSplotJABBAres(simple, subplots = "len", add = TRUE)
```

```
   RMSE stats by Index:
      indices RMSE.perc nobs
   1  Fishery       4.5   75
   2   Survey       3.4   20
   3 Combined       4.3   95
```

2.2 Retrospective and Forecast bias

Retrospective analysis is commonly used to check the consistency of model estimates, i.e., the invariance in spawning stock biomass (SSB) and fishing mortality (F) as the model is updated with new data in retrospect. The retrospective analysis involves sequentially removing observations from the terminal year (i.e., peels), fitting the model to the truncated series, and then comparing the relative difference between model estimates from the full-time series with the truncated time-series.

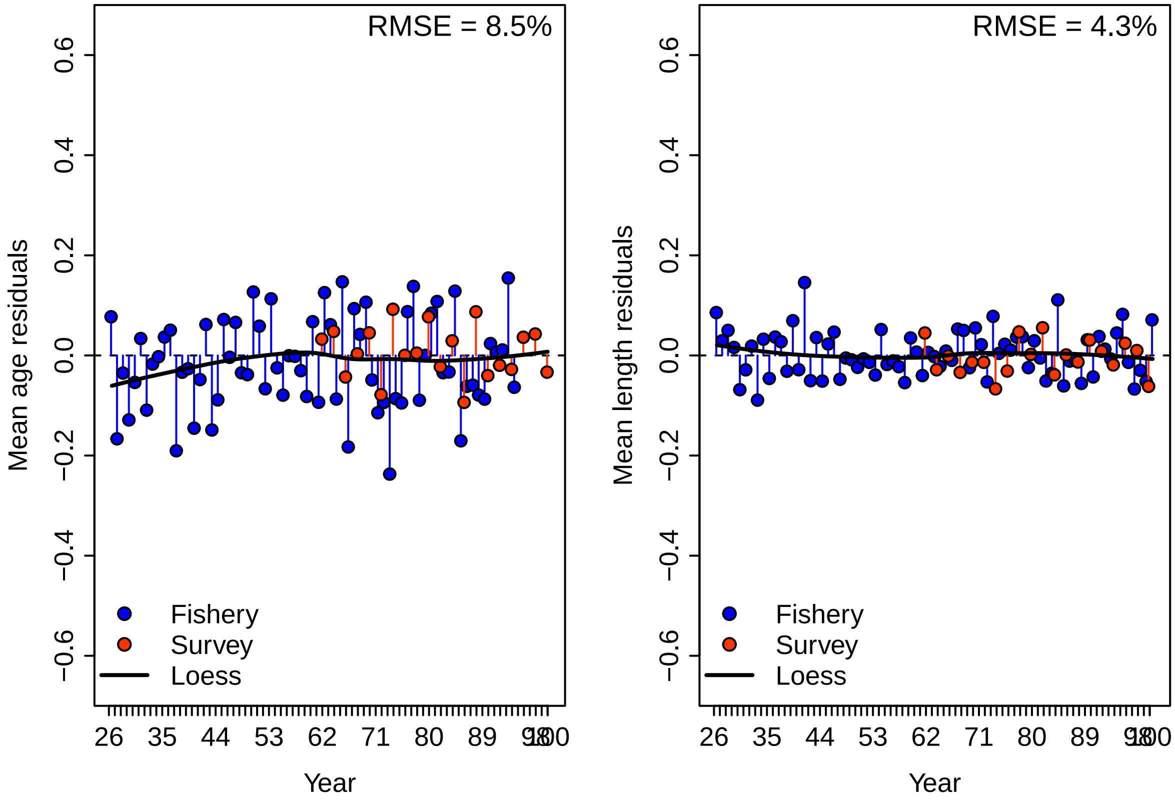

Figure 3: Joint residual plots for fits to age and length compositions, where the vertical lines with points show the residuals, and solid black lines show loess smoother through all residuals. Boxplots indicate the median and quantiles in cases where residuals from the multiple indices are available for any given year. Root-mean squared errors (RMSE) are included in the upper right-hand corner of each plot.

In Stock Synthesis, retrospective analysis can be routinely implemented using `r4ss:SS_doRetro()` (see Section 3.1). `ss3diags` provides the function `SSplotRetro()` to visualize the retrospective patterns of SBB and F and compute the associated Mohn's rho value (i.e. retrospective bias). This first requires loading the retrospective runs (Section 1.2]), which are already inbuilt into `ss3diags` in this case. The next step is to summarize the list of retrospective runs using `r4ss::SSsummarize()`.

```r
retroI.simple <- r4ss::SSsummarize(retroSimple, verbose = F)
```

We use notation "retroI" because `r4ss::SSsummarize()` summarizes the modeled quantities and abundance indices, but not length or age composition data. Using `retroI.phk` it is possible to produce some basic retrospective plots.