User guidelines for Advanced Model Diagnostics with ss3diags

Hennig Winker, Felipe Carvalho, Massimiliano Cardinale & Laurence Kell

30 April, 2021

Contents

1 Getting started (UNDER CONSTRUCTION)

This vignette explains functions for applying advanced model diagnostics for Stock Synthesis model with the `ss3diags` R package for model. The `ss3diags` package builds on R package `r4ss` (Taylor et al. 2021), which is designed to support the use of the Stock Synthesis software (Methot and Wetzel, 2013).

1.1 Installation

Both `ss3diags` and `r4ss` can be installed from gihtub using library(devtools):

```
installed.packages("devtools")

devtools::install_github("JABBAmodel/r4ss")

devtools::install_github("JABBAmodel/ss3diags")
```

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

1.2 Loading built-in example data

The package contains two examples of Stock Synthesis assessments as presented in Carvalho, Winker et (2021).

1.2.1 The Pacif hake (*Merluccius productus*) base case model

2020 Pacific hake/whiting Stock Synthesis assessment. Joint Technical Committee of the U.S. and Canada Pacific Hake/Whiting Agreement, National Marine Fisheries Service and Fisheries and Oceans Canada

```
data("pac.hke")
```

Individual list objects generated using R package r4ss:

- `ss3phk`: output from Stock Synthesis as read by `r4ss::SS_output()`

```
dir.path = "C:/Users/henni/Dropbox/ss3diags_demo/PacificHake"
ss3pke = r4ss::SS_output(file.path(dir.path,"Reference_Run"))
```

- `retro.phk`: list of retrospective runs with `r4ss:SS_doRetro()` as read by `r4ss::SSgetoutput()`

In this case the retrospective runs, produced with `r4ss:SS_doRetro()`(see recipe below), were located in the subfolders `/Retro_Reference_Run` and named "retro0","retro-1", "retro-2" and so on. To load those at once we use the `r4ss::SSgetoutput()`. In this case we conducted eight retrospective runs, where "retro0" corresponds to the "Reference_Run" and "retro-1" to "retro-7" are "peels". To assign the model names, we there specify `start.retro = 0` and `end.retro = 7` below.

```
start.retro = 0
end.retro = 7
retro.runs = "Retro_Reference_Run"
# load models
retro.phk <- r4ss::SSgetoutput(dirvec=file.path(dir.path,
retro.runs,paste0("retro",start.retro:-end.retro)))
```

Again, note that list object `retro.phk[[1]]` ("retro0") corresponds to the reference run ss3phk

`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()`

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

1.2.2 The ICCAT North Atlantic shortfin mako (*Isurus oxyrinchus*) reference model

```
data("natl.sma")
```

Individual list objects generated using R package r4ss:

- `ss3sma`: output from Stock Synthesis as read by `r4ss::SS_output()`
- `retro.sma`: list of retrospective runs with `r4ss:SS_doRetro()` as read by `r4ss::SSgetoutput()`
- `aspm.sma`: list of Stock Synthesis referece and aspm created with `r4ss::SSsummarize()`

2 Model Diagnostics with ss3diags

2.1 Plotting residual diagnostic with `ss3diags`

The plotting options are kept mostly to those provided by r4ss. Just like with r4ss, if, for example, 'SSplotRunstest() called with no further specifications several windows will open, which depends in this case on the number abundance indices.

```
SSplotRunstest(ss3sma)
```

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). 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.

To visiualized the runs test for multiple indices, it is suggested to make use of the function `sspar()` and the option plot `add=TRUE`. `sspar()` facilitates setting the graphic parameters so that they are suitable for `ss3diags` plots and option `add=TRUE` prevents the plotting functions from over-writing `sspar()`.

```
sspar(mfrow=c(3,2),plot.cex=0.8)
rt = SSplotRunstest(ss3sma,add=T,verbose=F)
```

It is also possible to select the indices that should be plotted. For example, in this case we may want to exclude CPUE2 as it was not fitted (zero weight to the likelihood). This also creates space to add another plot, such as joint-residual `SSplotJABBAres` to summarize all selected indices.

```
sspar(mfrow=c(3,2),plot.cex=0.8)
rt= SSplotRunstest(ss3sma,add=T,indexselect = c(1,3:6),legendcex = 0.8,verbose=F)
jr = SSplotJABBAres(ss3sma,add=T,indexselect = c(1,3:6),legendcex = 0.55,verbose=F)
```

The default for `SSplotRunstest()` and `SSplotJABBAres()` is plot the residual runs for the abundance indices, but it is also possible to do the plot plot for the composition data by specifying `subplots="len"` (or "age")

```
sspar(mfrow=c(3,2),plot.cex=0.8)
rt= SSplotRunstest(ss3sma,add=T,legendcex = 0.8,subplot="len",verbose=F)
jr = SSplotJABBAres(ss3sma,add=T,
                    legendcex = 0.55,legendloc="bottomright",subplot="len",verbose=F)
```

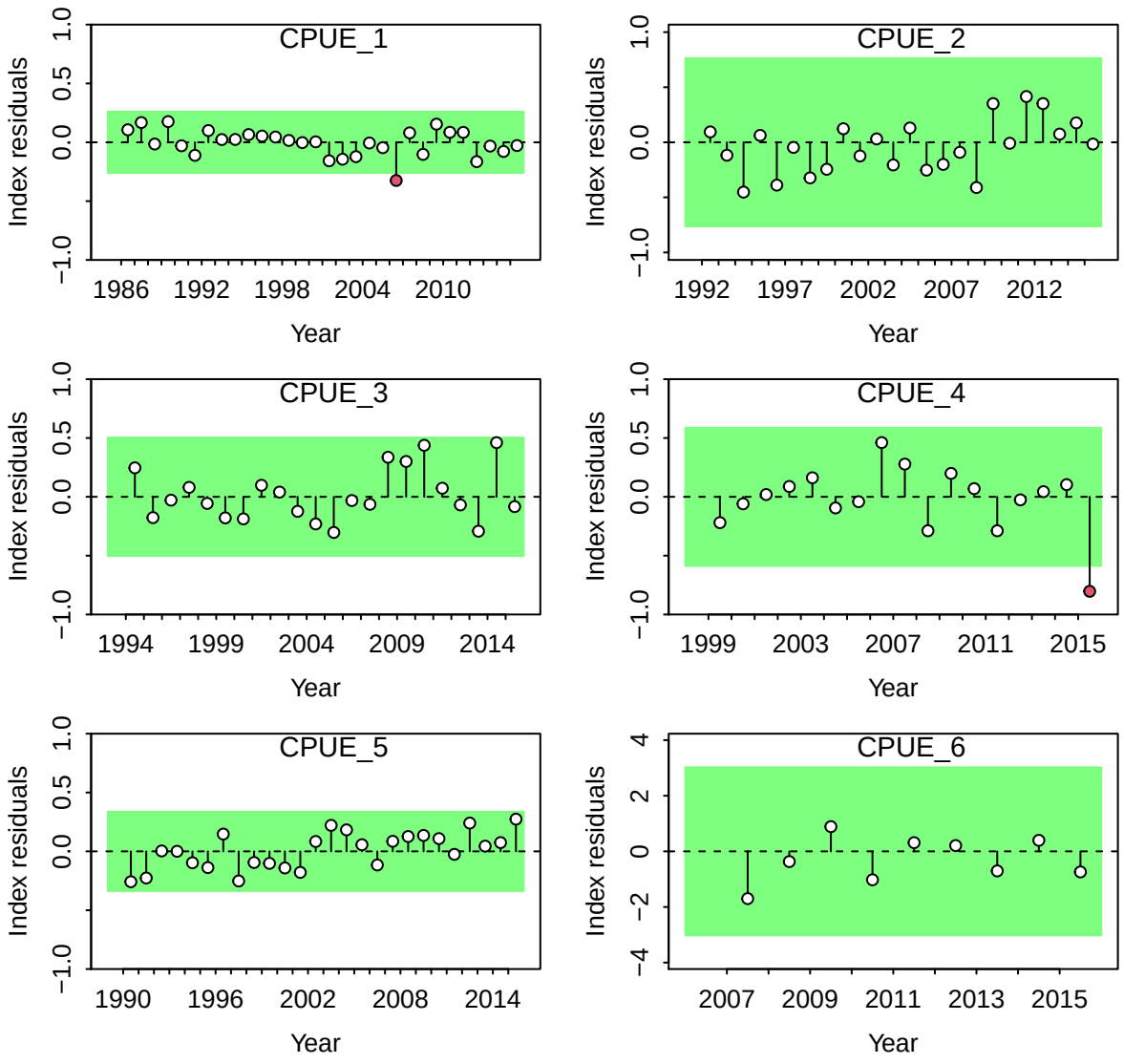


Figure 1: Runs test plots for CPUE index fits. Green shading indicates no evidence ($p = 0.05$) and red shading evidence ($p < 0.05$) to reject the hypothesis of a randomly distributed time-series of residuals, respectively. The shaded (green/red) area spans three residual standard deviations to either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

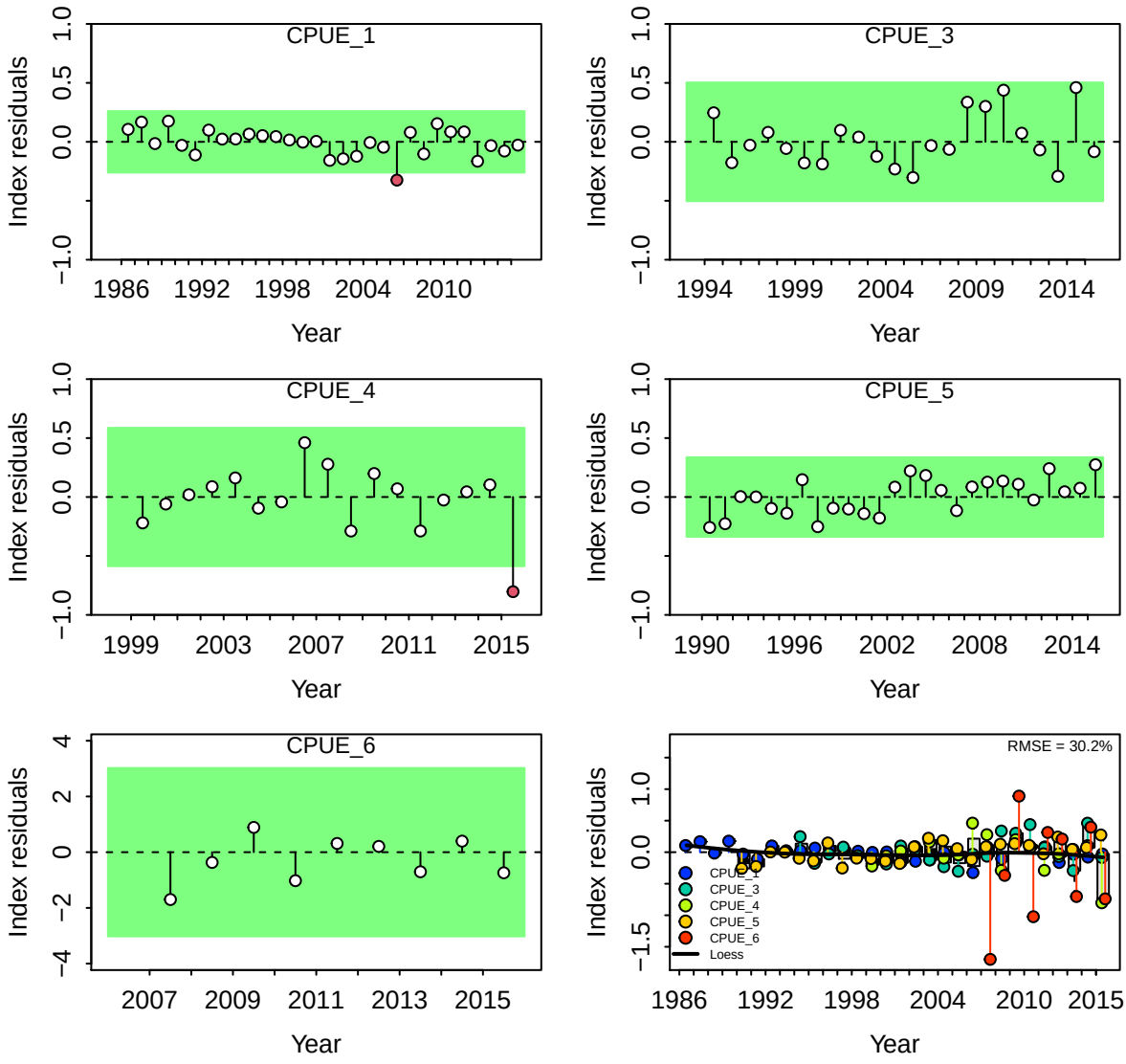


Figure 2: Runs test plot and Joint residual plot for fits to CPUE indices, where the vertical lines with points show the residuals, and solid black lines show loess smoother through all residuals. Boxplots indicate the median and quantiles in cases where residuals from the multiple indices are available for any given year. Root-mean squared errors (RMSE) are included in the upper right-hand corner of each plot.

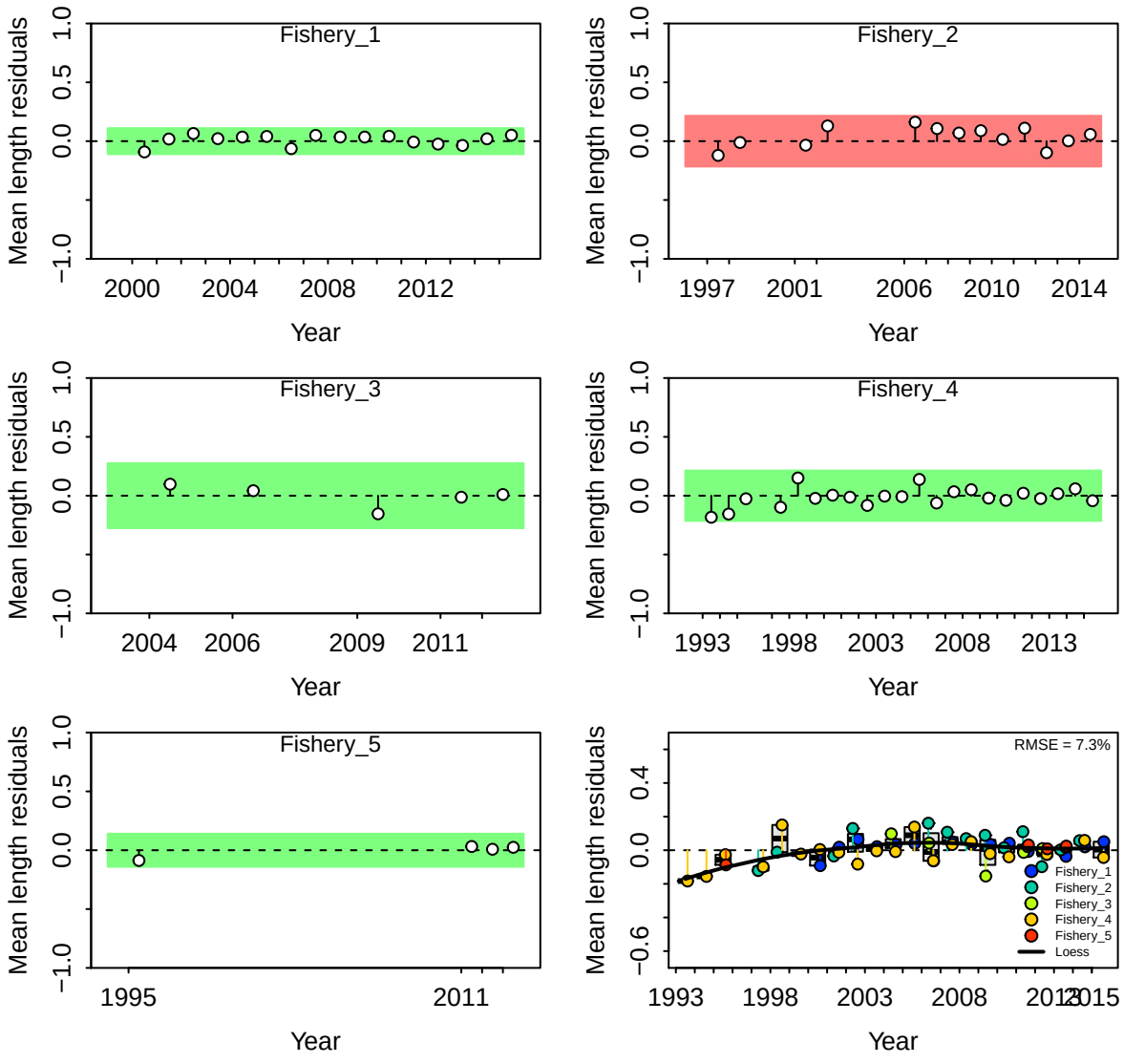


Figure 3: Runs test plot and Joint residual plot for mean lengths from fits length composition data

Several diagnostic tests can also be called without plotting, e.g. to facilitate automated processing

```
rti= SSrunstest(ss3sma,quant="cpue",verbose=F)
rtl= SSrunstest(ss3sma,quant="len",verbose=F)
rbind(rti,rtl)
          Index runs.p   test  sigma3.lo sigma3.hi type
   1     CPUE_1  0.069 Passed -0.2605783 0.2605783 cpue
   2     CPUE_2  0.717 Passed -0.7630777 0.7630777 cpue
   3     CPUE_3  0.229 Passed -0.5033328 0.5033328 cpue
   4     CPUE_4  0.406 Passed -0.5868380 0.5868380 cpue
   5     CPUE_5  0.065 Passed -0.3369695 0.3369695 cpue
   6     CPUE_6  0.870 Passed -3.0226356 3.0226356 cpue
   7  Fishery_1  0.127 Passed -0.1103741 0.1103741  len
   8  Fishery_2  0.040 Failed -0.2156036 0.2156036  len
   9  Fishery_3  0.331 Passed -0.2759992 0.2759992  len
   10 Fishery_4  0.806 Passed -0.2141490 0.2141490  len
   11 Fishery_5  0.159 Passed -0.1407228 0.1407228  len
```

The pacific hake assessment provides an example of fits to age composition instead of length composition data, which can plotted by specifying `subplots="age"`

```
sspar(mfrow=c(2,2),plot.cex=0.8)
rti = SSplotRunstest(ss3phk,add=T,legendcex = 0.8,subplot="cpue",verbose=F)
rta = SSplotRunstest(ss3phk,add=T,legendcex = 0.8,subplot="age",verbose=F)
jra = SSplotJABBAres(ss3phk,add=T,legendcex = 0.7,subplot="age",verbose=F)
```

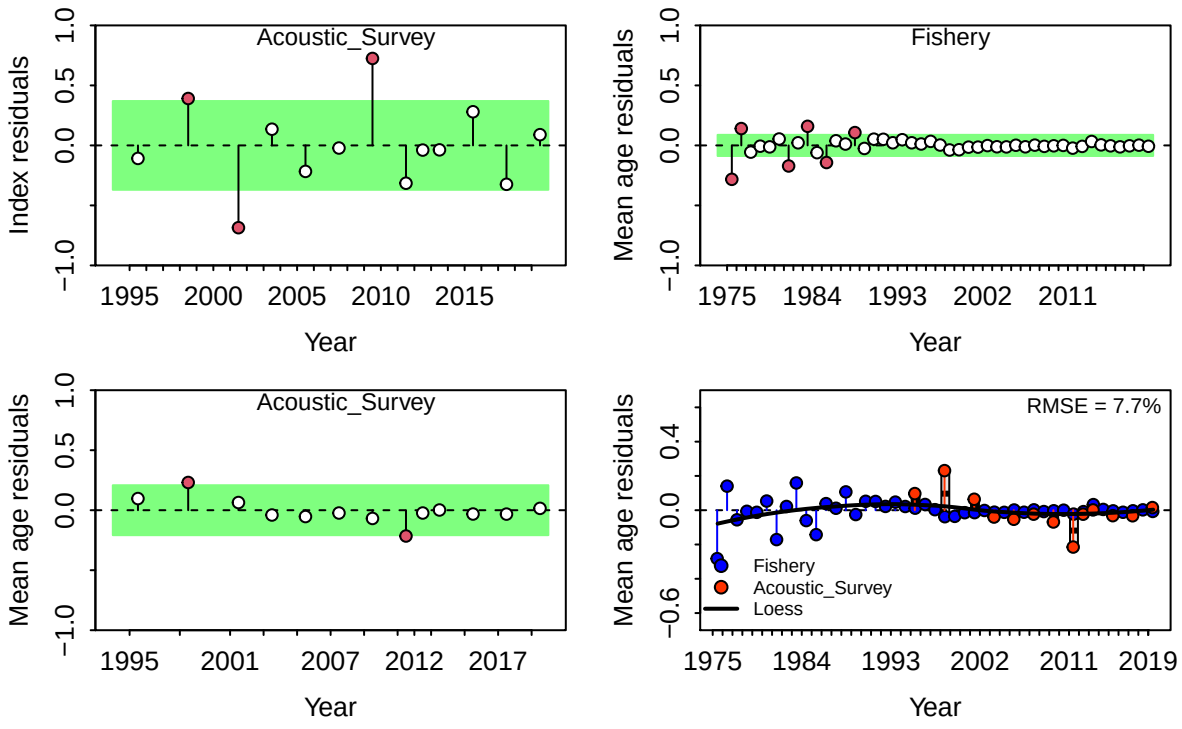


Figure 4: Runs test plot and Joint residual plot for a survey abudance index and mean ages from fits to survey and fisheries dependent age-composition data

2.2 Ploting 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.

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

```
retroI.phk <- r4ss::SSsummarize(retro.phk,verbose=F)
```

We use notation "retroI" because `r4ss::SSsummarize()` summarizes the modelled quantaties and abudances indices, but length or age composition data, which becomes relevant again in the next Section. With summarized object `retroI.phk` it is now possible to produce some basic retrospective plots.

```
sspar(mfrow=c(1,2),plot.cex=0.8)
rb = SSplotRetro(retroI.phk,add=T,forecast = F,legend = F,verbose=F)
rf = SSplotRetro(retroI.phk,add=T,subplots="F", ylim=c(0,0.4),
                 forecast = F,legendloc="topleft",legendcex = 0.8,verbose=F)
```

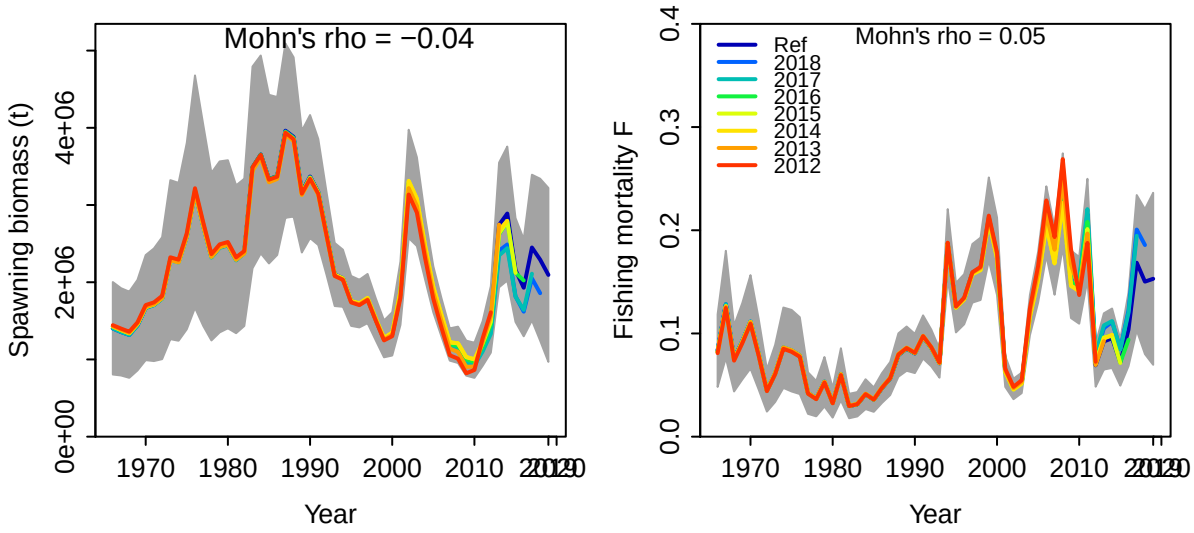


Figure 5: Retrospective analysis of spawning stock biomass (SSB) and fishing mortality estimates for Pacific hake conducted by re-fitting the reference model (Ref) after seven years, one year at a time sequentially. Mohn's rho statistic are denoted on top of the panels. Grey shaded areas are the 95 % confidence intervals from the reference model in cases where the analysis was run with Hessiean

Retrospective analysis is useful to evaluate how consistent the modeled quantities are in retrospect. However, providing fisheries management advice requires predicting a stock's response to management and checking that predictions are consistent when updated by new data in the future. A first, intuitive extension of the retrospective analysis is to assess potential forecast bias by adding the additional step of forward projecting quantities, such as SSB, over the truncated years. A desirable feature of Stock Synthesis is that forecasts are automatically done when using `r4ss:SS_doRetro()`.The forecasts are based on the settings in

'forecast.ss', which are also evoked when conducting future projections with the same model, only that the observed catches are used for the retrospective forecasts. Retrospective forecasts (i.e. hindcasts) with Stock Synthesis is therefore only a matter of visualization, which can be done by setting the `SSplotRetro()` option `forecast=TRUE`.

```
sspar(mfrow=c(1,2),plot.cex=0.8)
rb = SSplotRetro(retroI.phk,add=T,forecast = T,legend = F,verbose=F,xmin=2000)
rf = SSplotRetro(retroI.phk,add=T,subplots="F", ylim=c(0,0.4),
                 forecast = T,legendloc="topleft",legendcex = 0.8,verbose=F,xmin=2000)
```

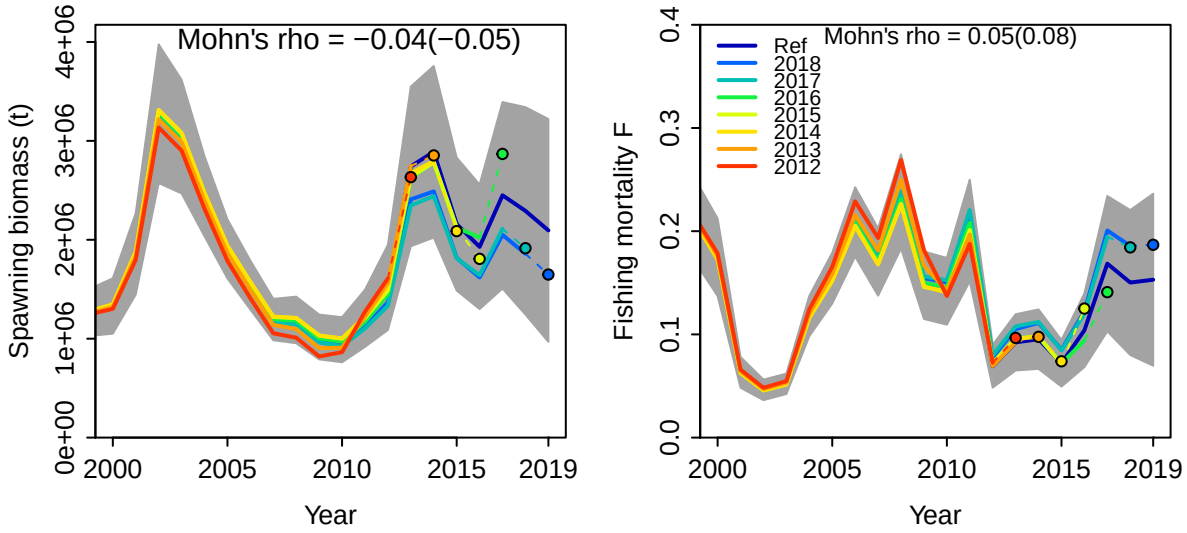


Figure 6: Retrospective results shown for the most recent years only. Mohn's rho statistic and the corresponding 'hindcast rho' values (in brackets) are now printed at the top of the panels. One-year-ahead projections denoted by color-coded dashed lines with terminal points are shown for each model.

The statistics in from the retrospective analysis with forecasting, including mohn's rho and forecast bias, can also be called without plotting using the function `SShcbias()`

```
SShcbias(retroI.phk,quant="SSB",verbose=F)
     type     peel          Rho  ForcastRho
   1  SSB     2018 -0.189511890 -0.21290187
   2  SSB     2017 -0.138534518 -0.16468977
   3  SSB     2016  0.048946154  0.17079324
   4  SSB     2015 -0.017403026 -0.06301299
   5  SSB     2014 -0.032761073 -0.03298098
   6  SSB     2013  0.001623057 -0.01285596
   7  SSB     2012  0.059944948 -0.03946034
   8  SSB Combined -0.038242335 -0.05072981

SShcbias(retroI.phk,quant="F",verbose=F)
     type     peel          Rho  ForcastRho
   1    F     2018  0.235928805  0.22117119
   2    F     2017  0.154620273  0.22678319
   3    F     2016 -0.096502898 -0.16382344
   4    F     2015 -0.008902238  0.20182834
   5    F     2014  0.034769554  0.02664839
   6    F     2013  0.003690925  0.02342468
   7    F     2012  0.049728943  0.04693738
   8    F Combined  0.053333338  0.08328139
```

2.3 Hindcast Cross-Validation and prediction skill

Retrospective forecasting provides an indication of the expected bias of in modelled quantities when updated with new data, but is not suitable for model validation, which should based on actual observations that are unknown to the model.To address this, Kell et al. (2016) proposed the used of hindcasting cross-validation techniques (HCXval) for stock assessment model validation, where observations (e.g. CPUE, length comps) are compared to their predicted future values. The key concept behind the HCXval approach is 'prediction skill', which is defined as any measure of the accuracy of a forecasted value to the actual observed value that is not known by the model (Kell et al., 2021).

Implementing the Hindcast Cross-Validation (HCxval) diagnostic in Stock Synthesis requires the same model outputs that are already produced with generated by `r4ss:SS_doRetro()` as described in Section 3.1. Therefore, there is no additional computationally intensive step needed for HCxval if conducted in conjunction with retrospective analysis. As robust measure of prediction skill, we implemented the mean absolute scaled error (MASE). In brief, the MASE score scales the mean absolute error (MAE) of forecasts (i.e., prediction residuals) to MAE of a naïve in-sample prediction, which is realized in the form of a simple 'persistence algorithm', i.e. tomorrow's weather will be the same as today's (see Eq. 3, p.5 in Carvalho and Winker et al. 2021). A MASE score > 1 indicates that the average model forecasts are worse than a random walk. Conversely, a MASE score of 0.5 indicates that the model forecasts twice as accurately as a naïve baseline prediction; thus, the model has prediction skill.

HCxval is implemented for Stock Synthesis using function `SSplotHCxval()`to produce the novel HCxval diagnostic plot and computes the MASE scores for CPUE indices, mean lengths or mean ages that have observations falling within the hindcast evaluation period.

Plotting HCxval for abudance indices requires the same step of summarizing the list of retrospective runs as for the retrospective analysis, which, of course, only needs be done once. Here, we summarize the retro runs for shortfin mako.

```
retroI.sma <- r4ss::SSsummarize(retro.sma,verbose=F)
```

```
sspar(mfrow=c(3,2),plot.cex=0.8)
hci = SSplotHCxval(retroI.sma,add=T,verbose=F,ylimAdj = 1.3,legendcex = 0.7)
```

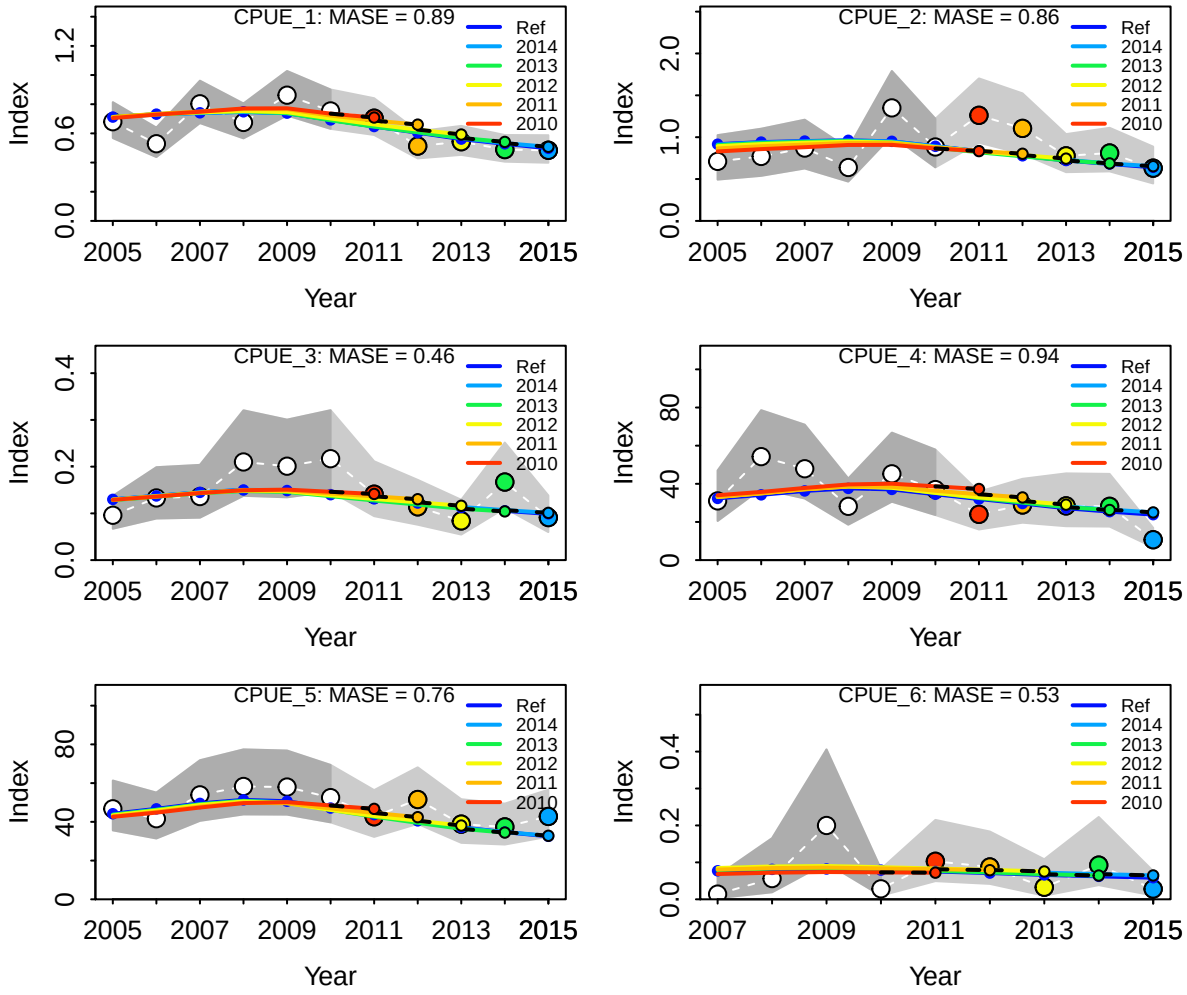


Figure 7: Hindcasting cross-validation (HCxval) results CPUE fits, showing observed (large points connected with dashed line), fitted (solid lines) and one-yearahead forecast values (small terminal points).HCxval was performed using one reference model (Ref) and five hindcast model runs (solid lines) relative to the expected CPUE. The observations used for crossvalidation are highlighted as color-coded solid circles with associated 95 % confidence intervals. The model reference year refers to the endpoints of each one-year-ahead forecast and the corresponding observation (i.e., year of peel + 1). The mean absolute scaled error (MASE) score associated with each CPUE

By comparison, the forecast length- and age-composition are somewhat hidden as "ghost files" of Stock Synthesis report.sso. To extract and summarize the composition in the form of oberved and expected mean lenghts and mean ages, respectively, `ss3diags` provides the function `SSretroComps()`.

```
retroC.sma = SSretroComps(retro.sma)
```

```
sspar(mfrow=c(1,2),plot.cex=0.8)
hcl = SSplotHCxval(retroC.sma,subplots="len",add=T,verbose=F,
ylimAdj = 1.3,legendcex = 0.7,indexselect = c(1,2))
```

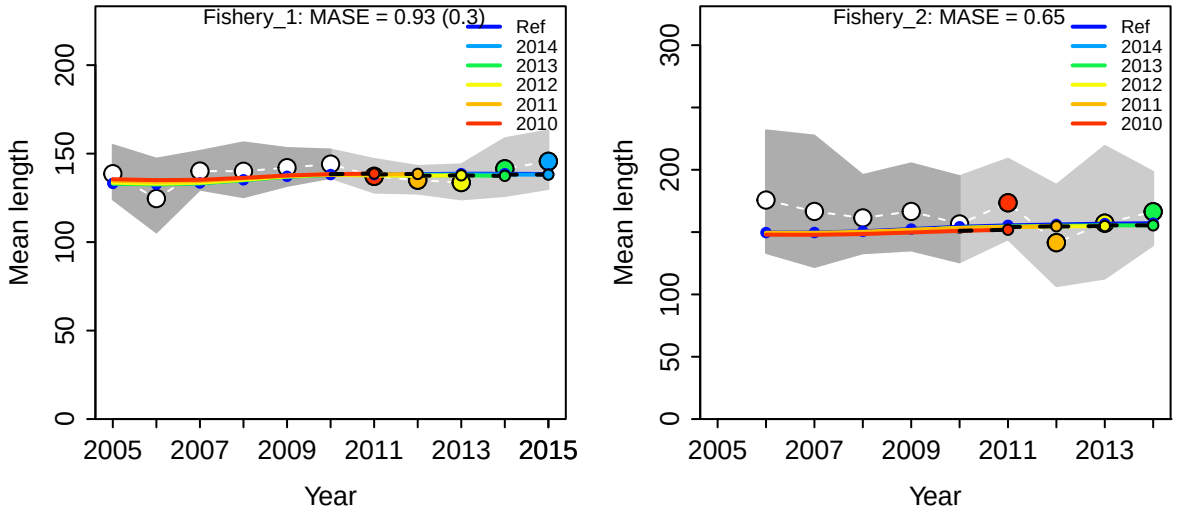


Figure 8: Hindcasting cross-validation (HCxval) results for mean lengths. Note that MASE values in breakets are adjusted MASE values for cases where naive predictions have a Mean-Absolute-Error below 0.1

Note that Figure 8, provides some additional, so called adjusted MASE values, in brackets. This gets invoked in cases where the inter-annual variation in the observed values is very small (default MAE < 0.1 for naive predictions log(y[t+1])-log(y[t])). The reasoning is that prediction residuals must be already very accurate to fall below this threshold. The adjusted MASE essential keep the naive prediction MAE denominator of the MASE to a maximum. Below we show the effect of changing adjustment threshold from the default `MAE.base.adj = 0.1`

```
mase1 = SSmase(retroC.sma,quant="len",MAE.base.adj = 0.1,indexselect = c(1:2))

    Computing MASE with all 5 of 5  prediction residuals for Index Fishery_1

    Computing MASE with only 4 of 5  prediction residuals for Index Fishery_2

    Warning:  Unequal spacing of naive predictions residuals may influence the interpretation of MASE

   MASE stats by Index:
mase1
         Index Season      MASE     MAE.PR   MAE.base  MASE.adj n.eval
   1 Fishery_1      1 0.9265301 0.02981727 0.03218165 0.2981727      5
   2 Fishery_2      1 0.6504563 0.07615571 0.11708045 0.6504563      4
```

to a larger value `MAE.base.adj = 0.15`

```
SSmase(retroC.sma,quant="len",MAE.base.adj = 0.15,indexselect = c(1:2))

    Computing MASE with all 5 of 5  prediction residuals for Index Fishery_1

    Computing MASE with only 4 of 5  prediction residuals for Index Fishery_2

    Warning:  Unequal spacing of naive predictions residuals may influence the interpretation of MASE

   MASE stats by Index:
         Index Season      MASE     MAE.PR   MAE.base  MASE.adj n.eval
   1 Fishery_1      1 0.9265301 0.02981727 0.03218165 0.1987818      5
   2 Fishery_2      1 0.6504563 0.07615571 0.11708045 0.5077048      4
```

where `MASE` is the ratio of the mean absolute error of the prediction residuals `MAE.PR` to the residuals of the naive predictions `MAE.base`

```
mase1$MAE.PR/mase1$MAE.base
   [1] 0.9265301 0.6504563
mase1$MASE
   [1] 0.9265301 0.6504563
```

and MASE.adj

```
mase1$MAE.PR/pmax(mase1$MAE.base,0.1)
   [1] 0.2981727 0.6504563
mase1$MASE.adj
   [1] 0.2981727 0.6504563
```

Note that applying HCxval for composition data requires to correctly specify the composition data type that was fitted in the model. For example, age composition data need be specified as "age" in `SSplotHCxval` and `SSmase`, as shown below for the Pacific hake model.

```
retroC.phk = SSretroComps(retro.phk) # summarize comps

sspar(mfrow=c(1,2),plot.cex=0.8)
hcl = SSplotHCxval(retroC.phk,subplots="age",add=T,
verbose=F,ylimAdj = 1.3,legendcex = 0.7,indexselect = c(1,2))
```

```
SSmase(retroC.phk,quants="age")

    Computing MASE with all 7 of 7  prediction residuals for Index Fishery

    Computing MASE with only 4 of 7  prediction residuals for Index Acoustic_Survey

    Warning:  Unequal spacing of naive predictions residuals may influence the interpretation of MASE

   MASE stats by Index:
               Index Season      MASE     MAE.PR  MAE.base  MASE.adj n.eval
   1         Fishery      1 0.6319721 0.09063054 0.1434091 0.6319721      7
   2 Acoustic_Survey      1 0.3556384 0.05181084 0.1456841 0.3556384      4
```

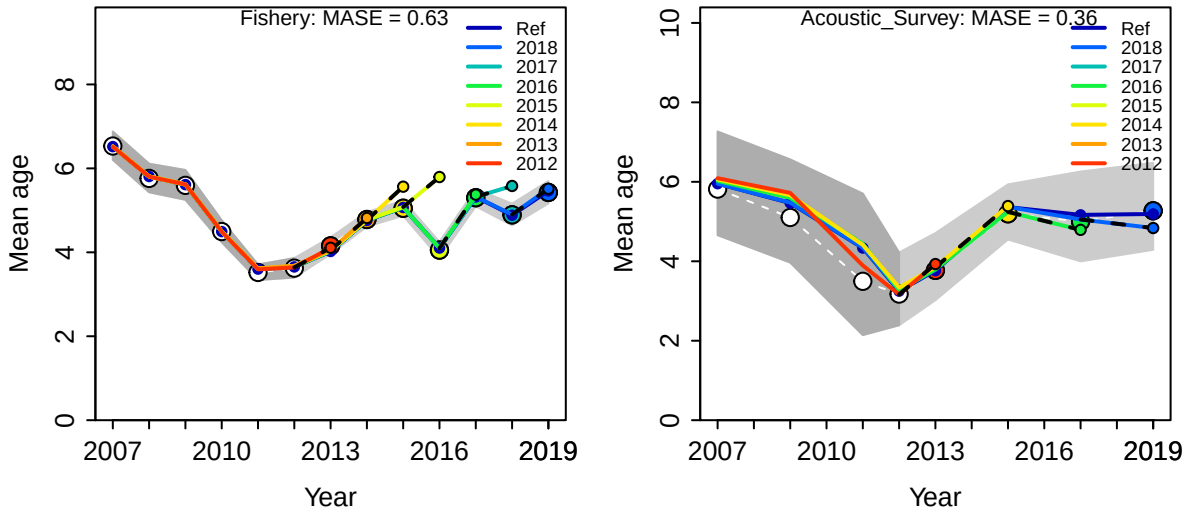


Figure 9: Hindcasting cross-validation (HCxval) results for mean lengths. Note that MASE values in breakets are adjusted MASE values for cases where naive predictions have a Mean-Absolute-Error below 0.1

2.4 Uncertainty with `ss3diags`

2.5 Model Ensembles with `ss3diags`

3 Cookbook Recipies

3.1 Retrospetive with hindcasts

Retrospective analysis can be run for Stock Synthesis using the function `r4ss::SS_doRetro()` available in r4ss. This setup of the retrospective analysis has the advantage that forecasts are conducted automatically given the catch. This makes in possible to apply retrospective forecasting and hindcast cross-validations of observations based on the same output.

```
library(r4ss)
```

In the following we provide a step-by-step cookbook recipe for retrospective analysis in Stock Synthesis.

3.1.1 Identify restrospective period

The first step is specifying the range pf peels that will then determine the `end.yr.vec` of runs in `r4ss::SS_doRetro()`

```
start.retro <- 0    # end year of reference year
end.retro   <- 7    # number of years for retrospective e.g.,
```

3.1.2 Identify the base directory

The second step is to specify the path directory that holds the folder with the base case run. In this case the Pacific Hake folder with a model folder 'Reference_Run"

```
dirname.base = "C:/Users/henni/Dropbox/ss3diags_demo/PacificHake"
run = "Reference_Run"

model.run <- file.path(dirname.base,run)
```

```
model.run
```

DAT and CONTROL files

The third step is to specify the names of the data and control files. Note these files are named differently from the `DATA.ss` and `CONTROL.ss`. In this case

```
DAT = "phk.dat"
CTL =  "phk.ctl"
```

The names of the DAT and CONTROL are declared on the top of the 'starter.ss', e.g.

```
#C Hake starter file phk.dat phk.ctl
```

3.1.3 Create a subdirectory for the Retrospectives

There are several ways to organise the retrospective output sturcture. Here we simpy create a new subfolder for the retrospective runs output

```
dir.retro <- paste0(dirname.base,'/Retro_',run)

dir.create(path=dir.retro, showWarnings = F)
```

We also create a subdirectory for the retrospective model folders

```
dir.create(path=file.path(dirname.Retrospective,"retros"), showWarnings = F)
```

3.1.4 Copy model run files to new retrospective folder

```
file.copy(file.path(model.run,"starter.ss_new"),  file.path(dir.retro,"starter.ss"))

file.copy(file.path(model.run,"control.ss_new"),
file.path(dir.retro, CTL))

file.copy(file.path(model.run,"data.ss_new"),
file.path(dir.retro, DAT))

file.copy(file.path(model.run,"forecast.ss"),
file.path(dir.retro, "forecast.ss"))

file.copy(file.path(model.run,"SS.exe"),
file.path(dir.retro, "SS.exe"))

# Automatically ignored for models without wtatage.ss
file.copy(file.path(model.run,"wtatage.ss"),
file.path(dir.retro, "wtatage.ss"))
```

3.1.5 Modify Starter.ss file

Modifying the Starter File helps to speed up model runs

```
starter <- readLines(paste0(dir.retro, "/starter.ss"))

#[8] "2 # run display detail (0,1,2)"
linen <- grep("# run display detail", starter)
```

```
starter[linen] <- paste0( 1 , " # run display detail (0,1,2)" )
# write modified starter.ss
write(starter, file.path(dir.retro, "starter.ss"))
```

Execute retrospective runs

Now all is readt to run the retrospective analyses with r4SS function `SS_doRetro` Ideally, the runs should be done with Hessian to allow evaluating the retrospective trajectories with respect to confidence interval coverage of the reference model.

```
r4ss::SS_doRetro(masterdir=dir.retro, oldsubdir="",
newsubdir="", years=start.retro:-end.retro)
```

However, for larger models it might be desirable to shorten run times, by not inverting the hessian, using the option `extras = "-nohess"` (much faster)

```
r4ss::SS_doRetro(masterdir=dir.retro,oldsubdir="",
newsubdir="", years=start.retro:-end.retro,extras = "-nohess")
```

3.1.6 Read `SS_doRetro()` output

```
retro.phk <- r4ss::SSgetoutput(dirvec=file.path(dir.retro,
             paste0("retro",start.retro:-end.retro)))
```

It is often useful to save the retro model runs as `.rdata` file for further processing with `ss3diags`, considering that reading the models with `r4ss::SSgetoutput()` can be quite time-consuming for more complex models.

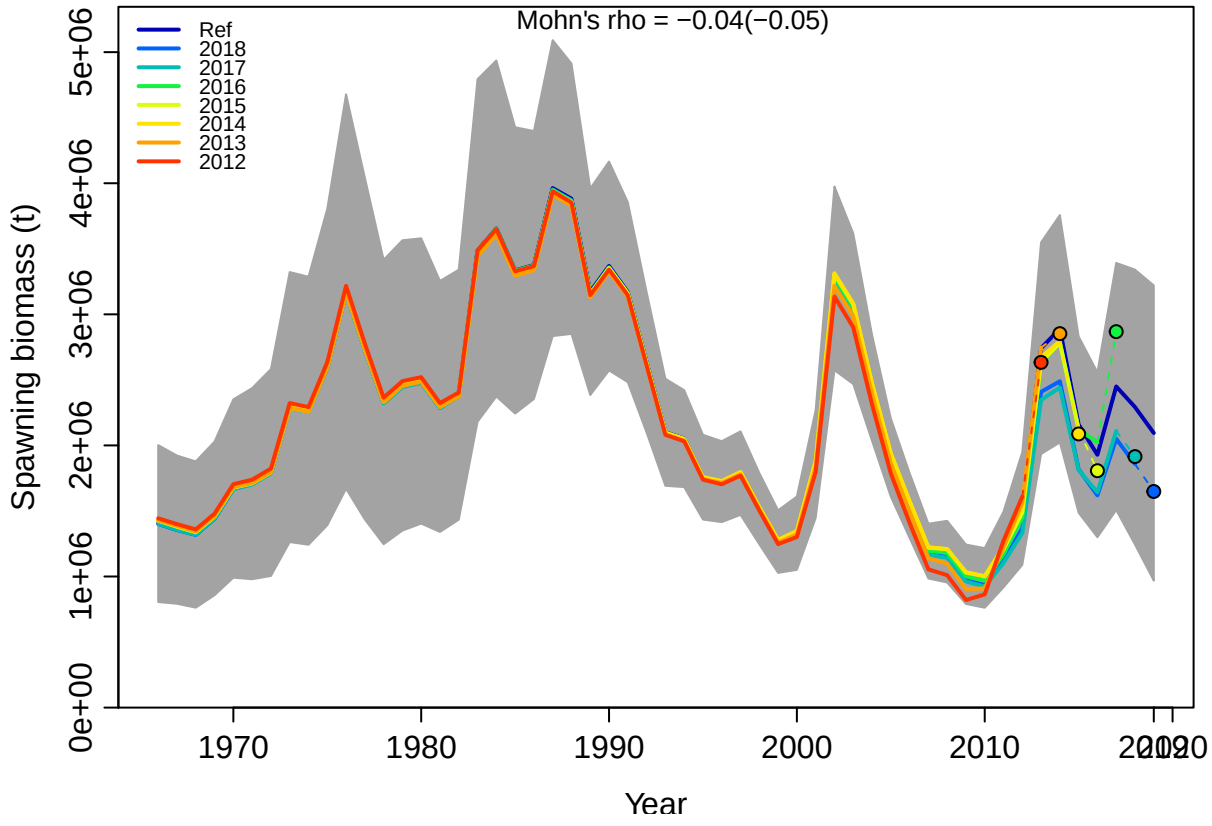
```
save(retro.phk, file=file.path(dir.retro,"retro.phk.rdata"))
```

3.1.7 Quick Check

```
library(ss3diags)

check.retro =r4ss::SSsummarize(retro.phk)
   Summarizing 8 models:
   imodel=1/8
     N active pars = 237
   imodel=2/8
     N active pars = 237
   imodel=3/8
     N active pars = 237
   imodel=4/8
     N active pars = 237
   imodel=5/8
     N active pars = 237
   imodel=6/8
     N active pars = 237
   imodel=7/8
     N active pars = 237
   imodel=8/8
     N active pars = 237
   Summary finished. To avoid printing details above, use 'verbose = FALSE'.
sspar(mfrow=c(1,1))
```

```
SSplotRetro(check.retro,forecast = T,add=T,legendcex = 0.7,legendloc = "topleft")
   Plotting Retrospective pattern
```




```
Mohn's Rho stats, including one step ahead forecasts:
   type      peel          Rho ForecastRho
1   SSB      2018 -0.189511890 -0.21290187
2   SSB      2017 -0.138534518 -0.16468977
3   SSB      2016  0.048946154  0.17079324
4   SSB      2015 -0.017403026 -0.06301299
5   SSB      2014 -0.032761073 -0.03298098
6   SSB      2013  0.001623057 -0.01285596
7   SSB      2012  0.059944948 -0.03946034
8   SSB Combined -0.038242335 -0.05072981
```

3.2 Profiling

3.3 Jittering