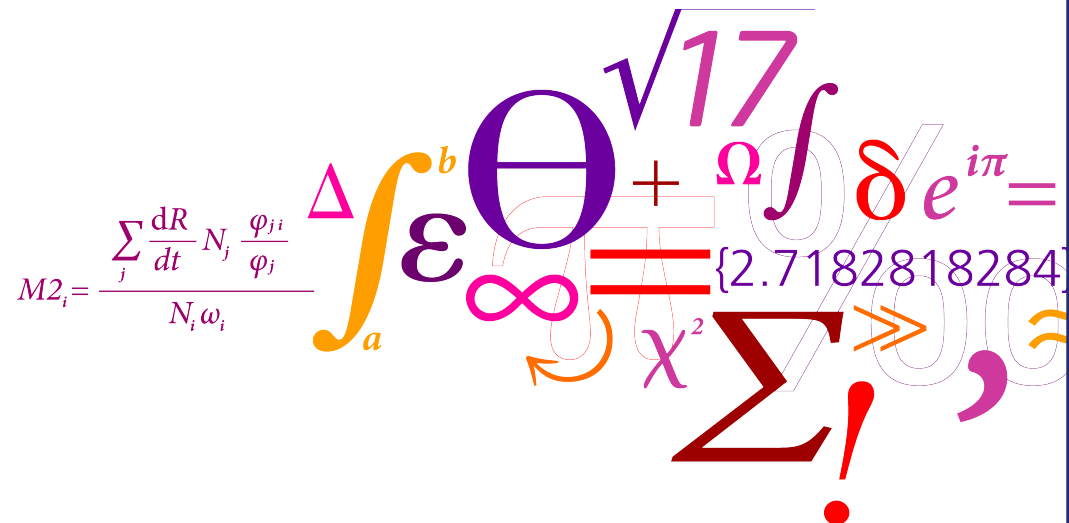


Model validation in state-space fish stock assessments models

Anders Nielsen & Olav Nikolai Breivik
an@aqua.dtu.dk

DTU Aqua
National Institute of Aquatic Resources



$$M2_i = \frac{\sum_j \frac{dR}{dt} N_j \frac{\varphi_{ji}}{\varphi_j}}{N_i \omega_i}$$

$$\Delta \int_a^b \epsilon \Theta^{\sqrt{17}} + \Omega \int \delta e^{i\pi} = \{2.7182818284\}$$

$$\chi^2 \sum !$$

Fish stock assessment models have evolved



- The obvious tool for time series data
- Quantification of observation errors
- Quantification of process errors
- Process formulation of time-varying quantities
- Reasonable (low) number of model parameters
- Prediction as part of model formulation

So possibly our model diagnostics needs updating

Current practice include:

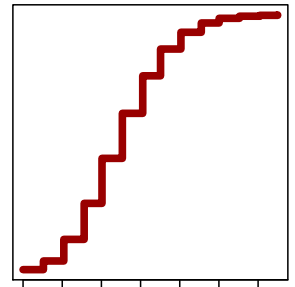
- Residuals (raw or standardized)
- Retrospective patterns of key outputs
- Leave-fleet-out runs (to check consistency between data sources)
- Jittered starting point analysis

Residuals

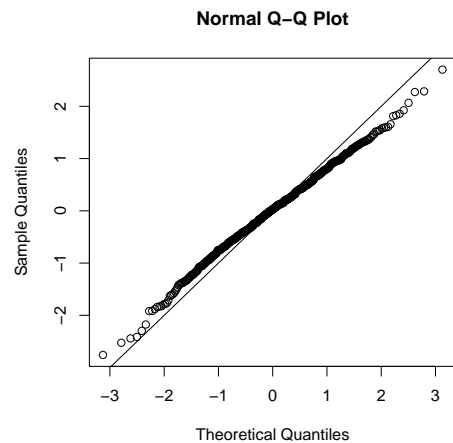
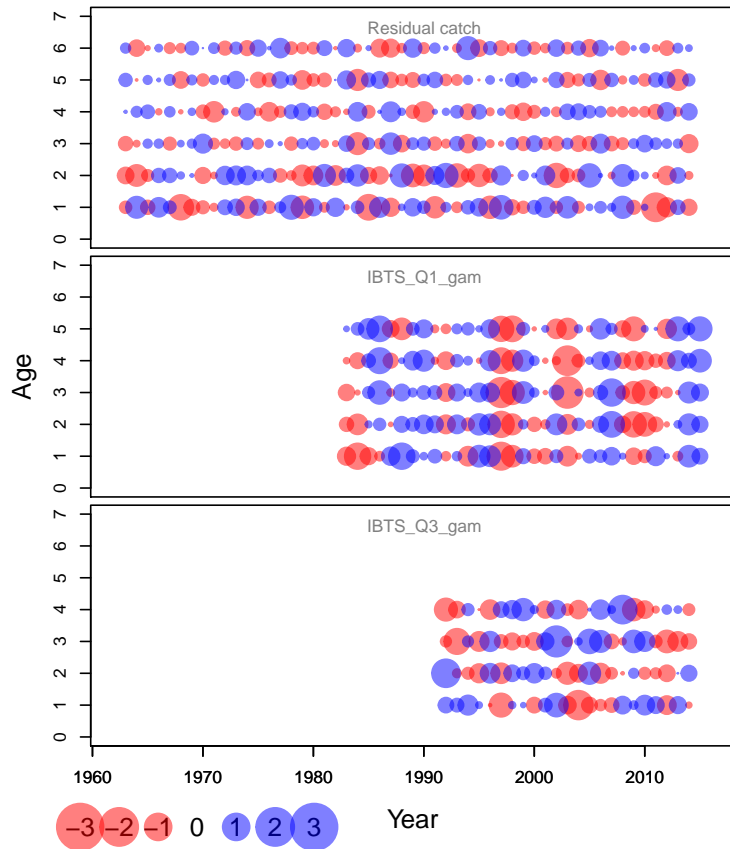
- In state-space assessment models residuals calculated as $r_i = (y_i - \hat{y}_i)/\hat{\sigma}_i$ are not supposed to be independent $N(0, 1)$ even in perfectly correct models.
- A safer alternative is the **one-observation-ahead** residuals $(y_i - \hat{y}_{i|i-1})/\hat{\sigma}_{i|i-1}$.
- More generally the **one-observation-ahead-quantile-residuals**

$$\Phi^{-1}(P(Y_i \leq y_i | Y_{i-1} = y_{i-1} \dots Y_1 = y_1))$$

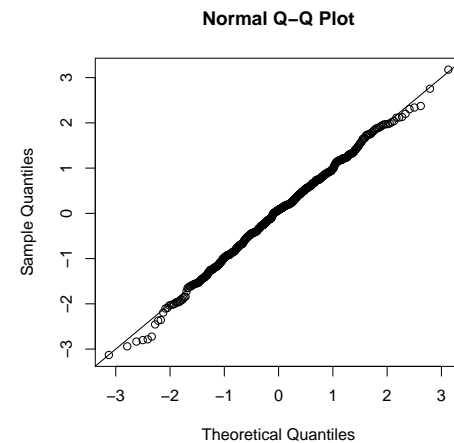
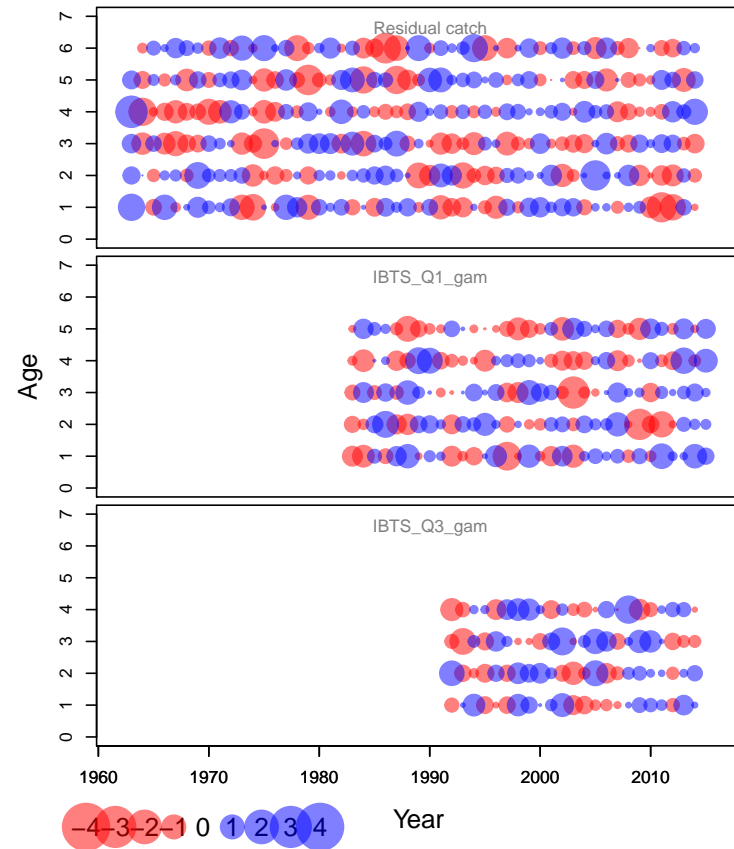
- Randomized if originating from a discrete distribution
- Requires extra work when the model is solved via Laplace approximation
- But it does matter — the residuals are different.



Wrong $(y_i - \hat{y}_i)/\hat{\sigma}_i$



Right $\Phi^{-1}(P(Y_i \leq y_i | Y_{i-1} = y_{i-1} \dots Y_1 = y_1))$



One-obs-ahead is now build into TMB

```
#include <TMB.hpp>

template<class Type>
Type objective_function<Type>::operator() (){
  DATA_VECTOR(y);
  DATA_VECTOR_INDICATOR(keep, y);  // NOTICE "keep"
  PARAMETER(logSdRw);
  PARAMETER(logSdObs);
  PARAMETER_VECTOR(lam);
  Type sdRw=exp(logSdRw);
  Type sd0=exp(logSdObs);
  int timeSteps=y.size();
  Type ans=Type(0);

  for(int i=1;i<timeSteps;i++){
    ans += -dnorm(lam(i),lam(i-1),sdRw,true);
  }

  for(int i=0;i<timeSteps;i++){
    ans += -keep(i)*dnorm(y(i),lam(i),sd0,true);
  }

  return ans;
}
```

res1/rw.cpp

- Notice the use of `keep` in both files

```
library(TMB)
compile("rw.cpp")
dyn.load(dynlib("rw"))
data <- list(y=scan("rw.dat"))
para <- list()
para$logSdRw=0
para$logSdObs=0
para$lam=rep(0,length(data$y))
obj <- MakeADFun(data,para,random="lam",DLL="rw")

opt <- nlminb(obj$par,obj$fn,obj$gr)

pred <- oneStepPredict(obj, observation.name="y",
                       data.term.indicator="keep")
```

res1/rw.R

For discrete observations (randomization)

- Let $x_i \sim \text{pois}(\lambda)$ (with a c.d.f. P)
- Define $u_i \sim \text{unif}(P(x_i - 1), P(x_i))$
- Define $z_i = \Phi^{-1}(u_i)$
- Now $z_i \sim \mathcal{N}(0, 1)$

```
# observations
x <- rpois(1000,3)
ppois.u <- function(x, lambda){
  runif(length(x), ppois(x-1,lambda), ppois(x,lambda)) #uses the fact that ppois(-1,lambda)=0
}
U <- ppois.u(x,3)
Z <- qnorm(U)
```

rand.R

Mini exercise: Repeat this example with a different distribution (e.g. Negative binomial) to see that you can in fact get perfect $N(0, 1)$ residuals using this approach.

Exercise: AR1-Poisson example

Recall the example from earlier where:

$$y_i \sim \mathcal{P}(e^{\gamma_i})$$

and

$$\gamma_i \sim \mathcal{N}(\phi\gamma_{i-1}, \sigma^2)$$

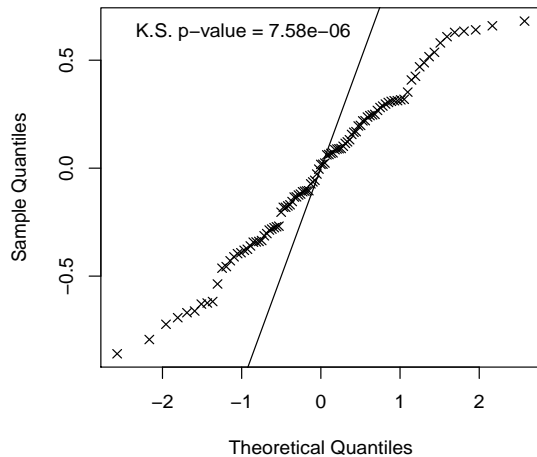
The data is available in the file `cpue.RData`, and an implementation is in the files `ar1.cpp` and `ar1.R`.

Add the parts to compute one-obs-ahead residuals for this model

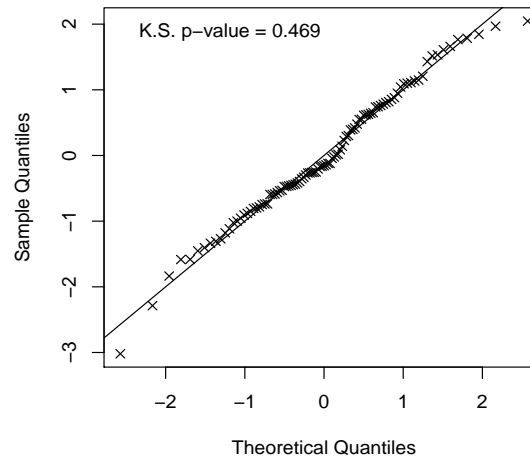
Process residuals

- Cannot just use the predicted random effects U , but
- If (Y, U) is distributed according to joint pdf. $L(y, u)$
- Observed y is then a sample from marginal distribution with pdf. $\int L(y, u)du$
- Generate one sample u^* from conditional distribution of $U|Y = y$
- Then the set (y, u^*) is a sample from joint distribution of (Y, U)
- Assumed distribution of u^* can be validated by standard tests

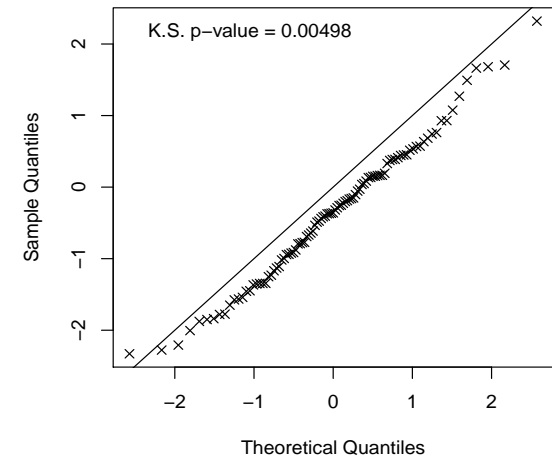
Wrong (using est. RE)

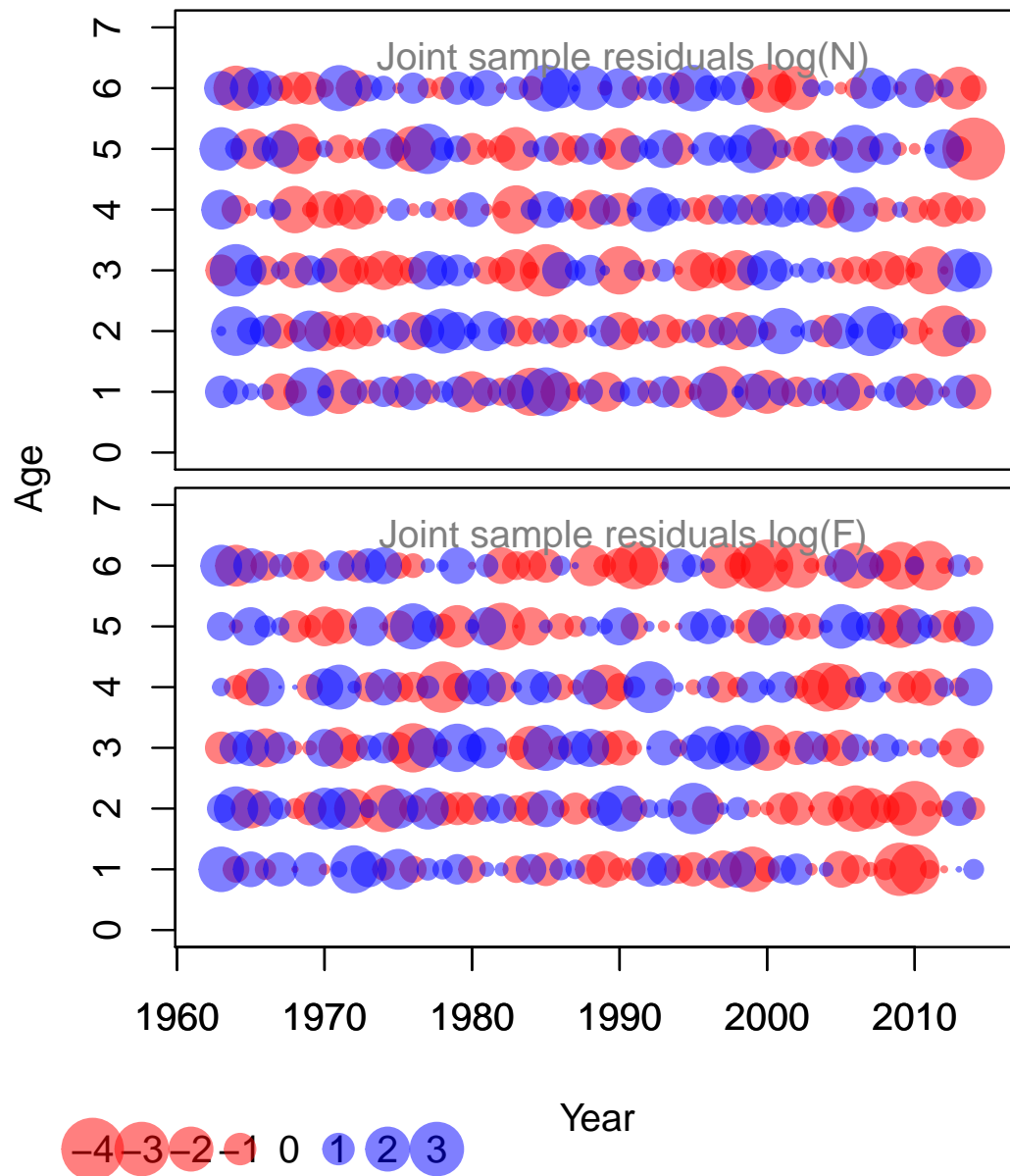


Right (joint sample)



Right. Model wrong





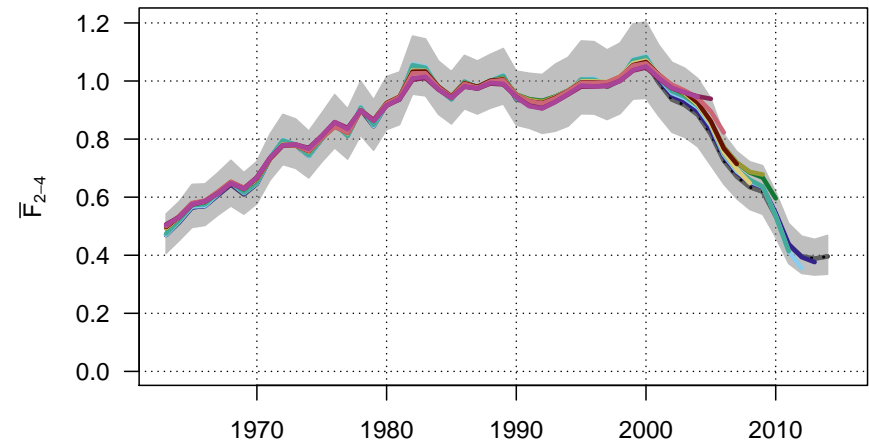
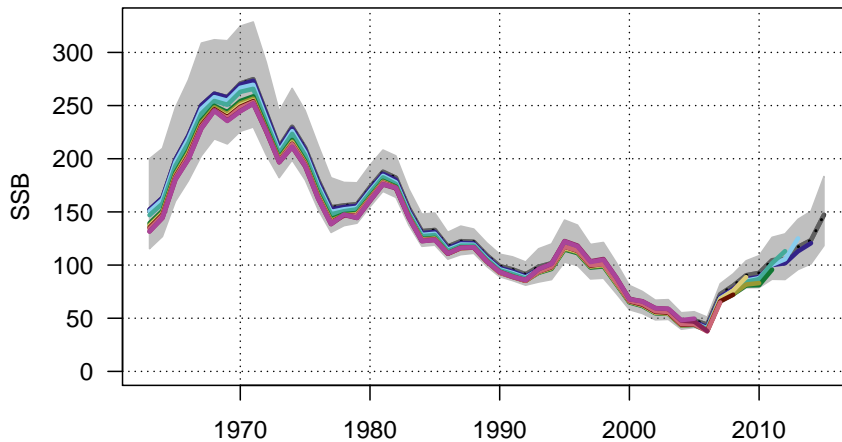
Code for the joint sample approach

```
sdr <- sdreport(obj)
estX <- summary(sdr,"random")
C <- solve(obj$env$spHess(obj$env$last.par.best, random=TRUE))
Xr <- MASS::mvrnorm(1,estX[,1],C)
```

Exercise: Calculate process residuals for the AR1-Poisson example. What distribution should we expect?

Retrospective pattern

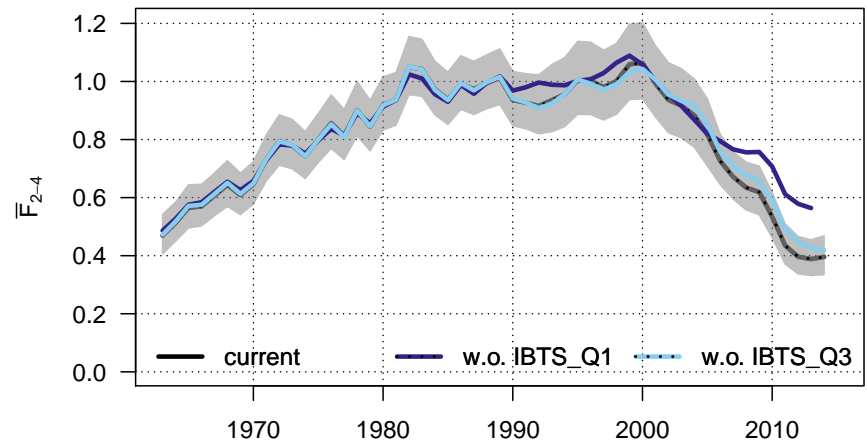
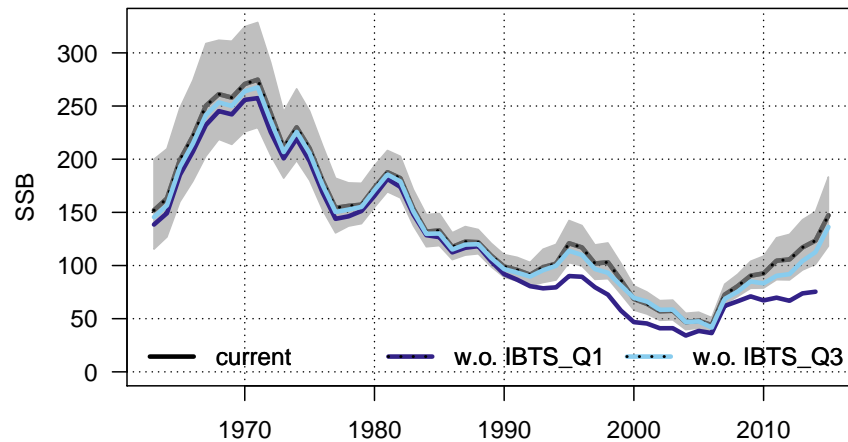
- Retrospective analysis (as done for fish stock assessments) are not predictions
- Possibly came about because prediction was not possible (within model)
- The procedure is:
 - Run model without last 1, 2, 3, \dots , n years of data
 - Compare key estimates to model run with all data.



- Final year's estimate is of special interest in fish stock assessment applications
- Retrospective is still valuable (even if the models now can predict)
- Only relevant after model verified via other model diagnostics

Leave-out-fleet runs

- Leaving out individual data sources one at a time
- Useful to see if one fleet is having an undue influence



Jitter analysis

- Use a number of random (widely scattered) initial values
- Verify that the same solution is obtained

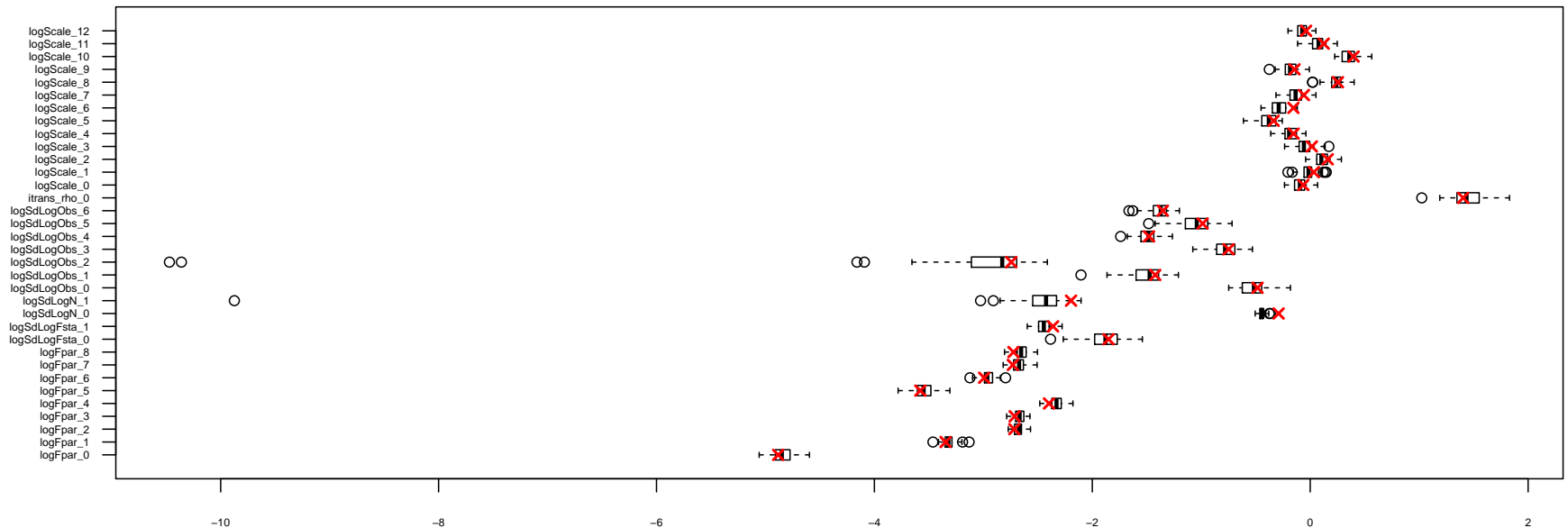
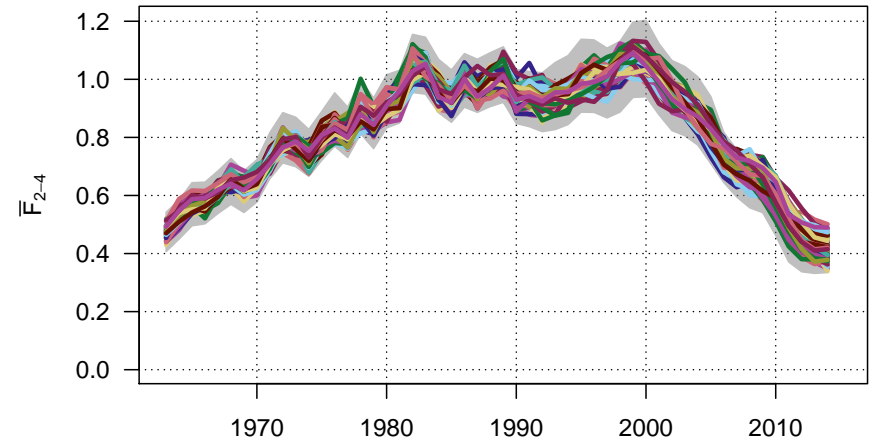
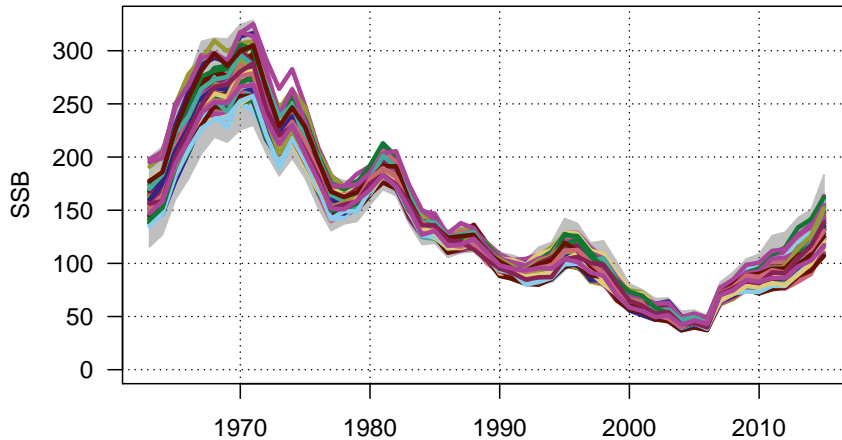
```
> fit.jit <- jit(fit, nojit=100)
> fit.jit
```

	max(delta)
logFpar	3.141487e-12
logSdLogFsta	4.359180e-12
logSdLogN	1.002443e-11
logSdLogObs	1.441203e-11
itrans_rho	3.332890e-11
logScale	2.292500e-12
logF	1.112670e-10
logN	8.915474e-11
ssb	5.637389e-09
fbar	5.945799e-12
rec	2.247407e-08
catch	2.957563e-09
logLik	2.924025e-10

- Possibly even more important for state-space models

Simulation validation

- Simulate from model and re-estimate (self test, parametric bootstrap)
- No (assessment) model should be accepted without it.



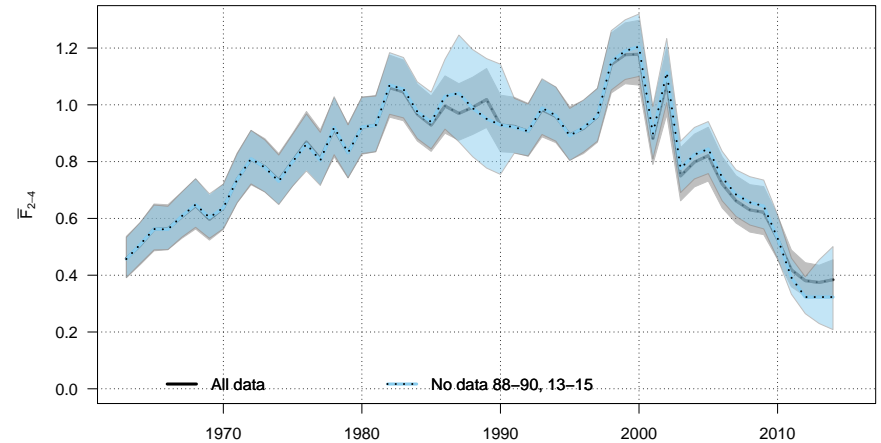
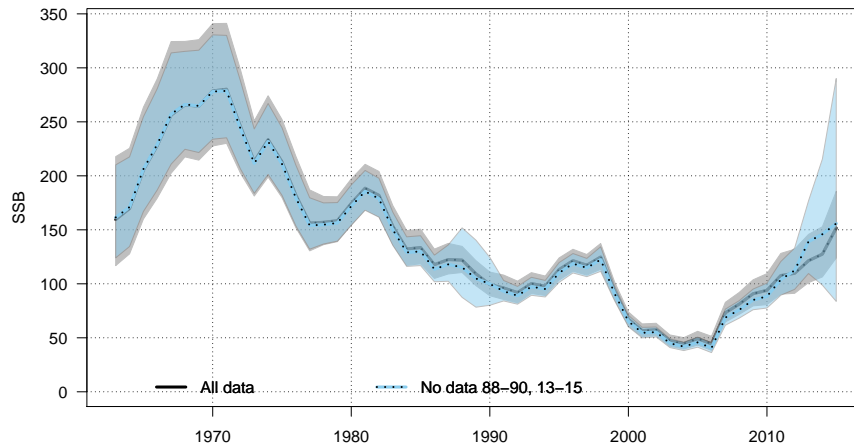
Exercise: Try it in SAM

- Pick a stock (e.g. one from online, or the build-in example).
- Try the functions:

```
residuals  
procrustes  
retro  
leaveout  
jit  
simstudy
```

Prediction and cross-validation

- Validate if the model is realistic w.r.t. coverage of confidence intervals
- Of special interest is the 2-3 year ahead predictions



- The only thing that is real is the observations
- when evaluating (and comparing) models we should look at their ability to predict observations.
- With state-space models we can (difficult to compare to other model types).

Implementing such cross validations

- Remember the NA trick? That makes this implementation simple:

```
# function for cross-validation
xval <- function(fit, year=NULL, fleet=NULL, age=NULL, ...){
  data <- fit$data
  nam <- c("year", "fleet", "age")[c(length(year)>0,length(fleet)>0,length(age)>0)]
  if((length(year)==0) & (length(fleet)==0) & (length(age)==0)){
    idx <- rep(TRUE,nrow(data$aux))
  }else{
    idx <- !do.call(paste, as.data.frame(data$aux[,nam,drop=FALSE])) %in% do.call(paste, as.data.frame(cbind(
      year=year, fleet=fleet, age=age)))
  }
  idx <- !idx
  data$logobs[idx] <- NA
  idx2 <- which(is.na(data$logobs))
  conf <- fit$conf
  par <- defpar(data,conf)
  thisfit <- sam.fit(data, conf, par, rm.unidentified = TRUE, newtonsteps=0, silent=TRUE,...)
  ret <- as.data.frame(cbind(data$aux[idx2,], obs=fit$data$logobs[idx2], pred=thisfit$pl$missing, predSd=
    thisfit$plsd$missing))
  ret <- ret[complete.cases(ret),]
  attr(ret, "fit") <- thisfit
  return(ret)
}

pred <- xval(fit, year=c(1988:1990,2013:2015))
```

- Exercise: Try it!

Check Laplace via simulation

- TMB offers a very neat approach
- The expectation of the gradient of the negative log-likelihood is 0.

$$E_{\theta} \nabla \ell(\theta; X) = 0$$

- This means if we simulate from the model, then the average gradient should be zero.
- But this only holds for the real likelihood.
- So if the approximation is wrong, then the average gradient will not be zero
- We can simulate as many data sets as we wish, so we can test this.
- Notice: that even the smallest bias will be detected if we simulate enough
- Notice: Models with a modest bias can still be useful

```
$joint$p.value  
[1] 0.4690289  
...  
$marginal$p.value  
[1] 0.7745296  
$marginal$bias  
...
```

Check Laplace in practice

- Build into TMB, so easy to use:

```
library(stockassessment)
fit <- sam.fit(nscodData, nscodConf, nscodParameters, sim.condRE=FALSE)
cc<-TMB::checkConsistency(fit$obj, n=200)
summary(cc)
```

Summary

Current	Updated
Standard residuals	One-observation-ahead residuals
	Process residuals
	Prediction diagnostics
(Simulation testing)	Simulation testing
(Jitter analysis)	Jitter analysis
	Laplace checker
Retrospective pattern	Retrospective pattern
Leave-one-out	Leave-one-out

Appendix: The math for the Laplace checker

$$\begin{aligned} E_{\theta} (\nabla \ell(\theta; X)) &= \int P_{\theta}(x) \nabla \ell(\theta; x) dx \\ &= - \int P_{\theta}(x) \frac{1}{P_{\theta}(x)} \nabla P_{\theta}(X) dx \\ &= - \nabla \int P_{\theta}(x) dx \\ &= 0 \end{aligned}$$