The role of random effects in next-generation stock assessment models

Anders Nielsen (an@aqua.dtu.dk)

Assessment models
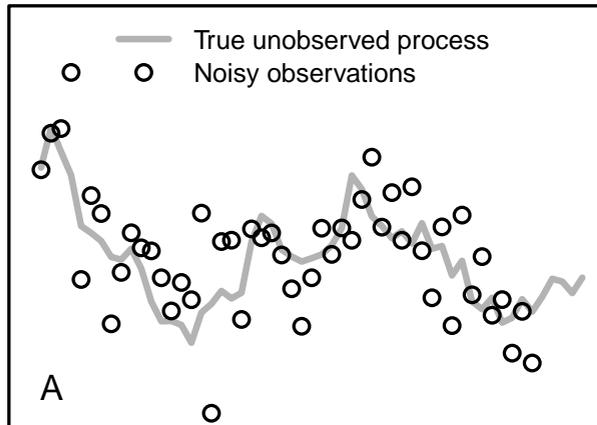
- This is a simplified illustration
- Simplest real versions have hundreds of non-linear model parameters
- Fast runtimes are often important
- The results are needed to make expensive decisions
- This problem has inspired efficient non-standard modelling tools
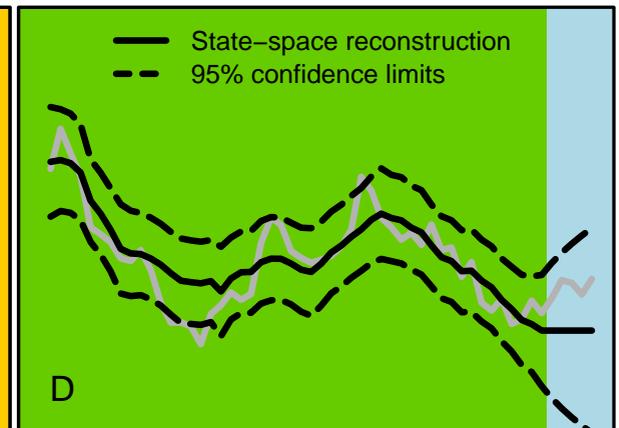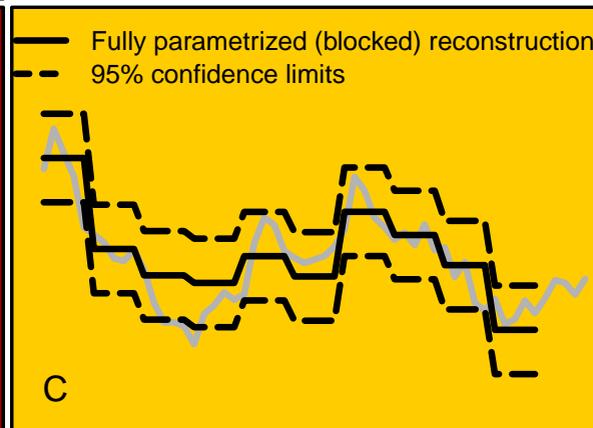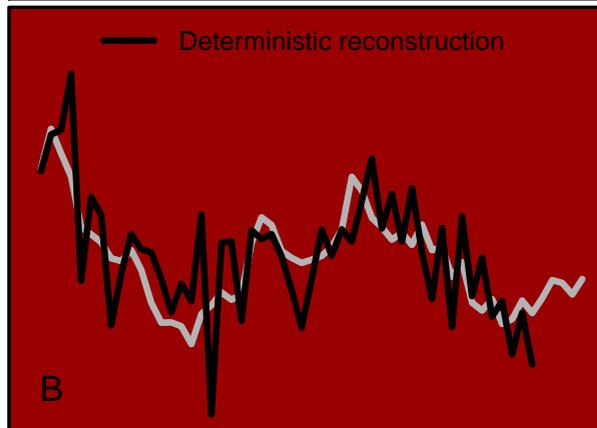
- The obvious tool for time series data
- Quantification of observation errors
- Quantification of process errors
- Process formulation of time-varying
- Reasonable (low) number of parameters
- Prediction as part of model formulation

Random effects

- Unobserved quantities with a distribution

- Often used when something unobserved gives extra variation or correlated observations

- May sound strange, but often it is easier to formulate joint distribution of unobserved and observed

- To estimate the model parameters we need to integrate

- From estimated model parameters we describe the random effect's distribution

Summary: Estimation with purely fixed effects

- We have:

  Observations: $y = (y_1, y_2, \ldots, y_n)$

  Parameters (μ, σ) in model: $y_i \sim N(\mu, \sigma^2)$
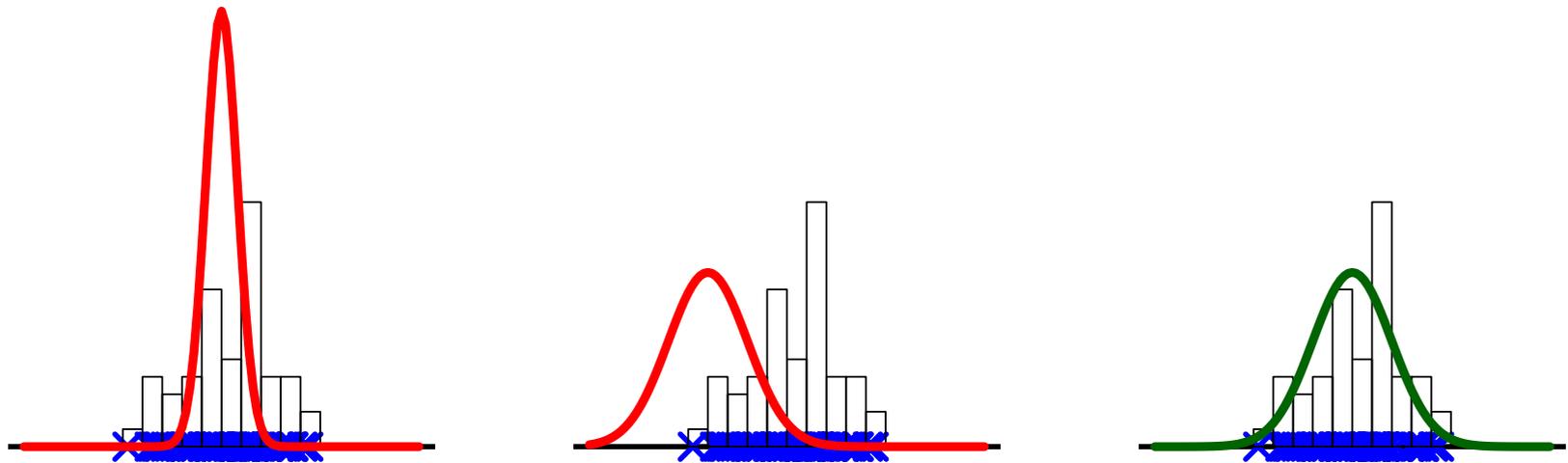
- Choose parameters which makes our model best match the data (optimize likelihood).

Estimation with random effects

- We have:

  Observations: y

  Unobserved random effects: u

  Parameters (θ) in model: $(y, u) \sim D(\theta)$
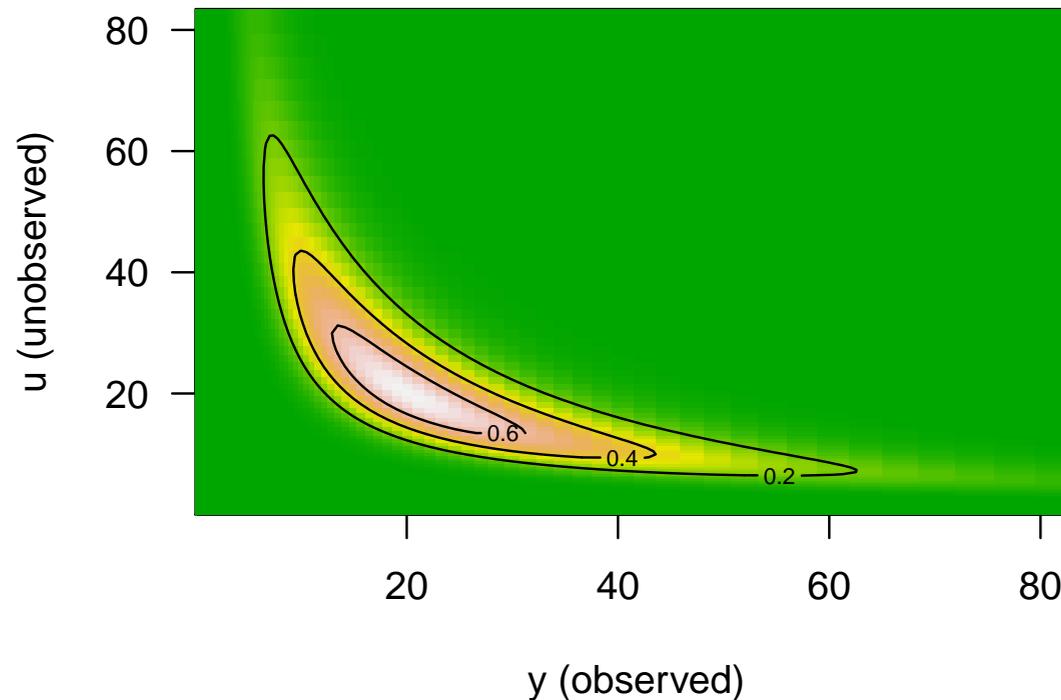
- How do we estimate our parameters when some of our observations are not observed?

Estimation with random effects — 2

- The banana is only an intermediate calculation

  1: Joint model (banana) is determined from model parameters θ

  2: Marginal model is calculated from joint by integration

  3: Marginal is matched to data as always

- Imagine the distribution $D(\theta)$ is described by a likelihood function $L(y, u, \theta)$, then:

$$L_M(y, \theta) = \int L(y, u, \theta) du$$

  is the marginal likelihood.
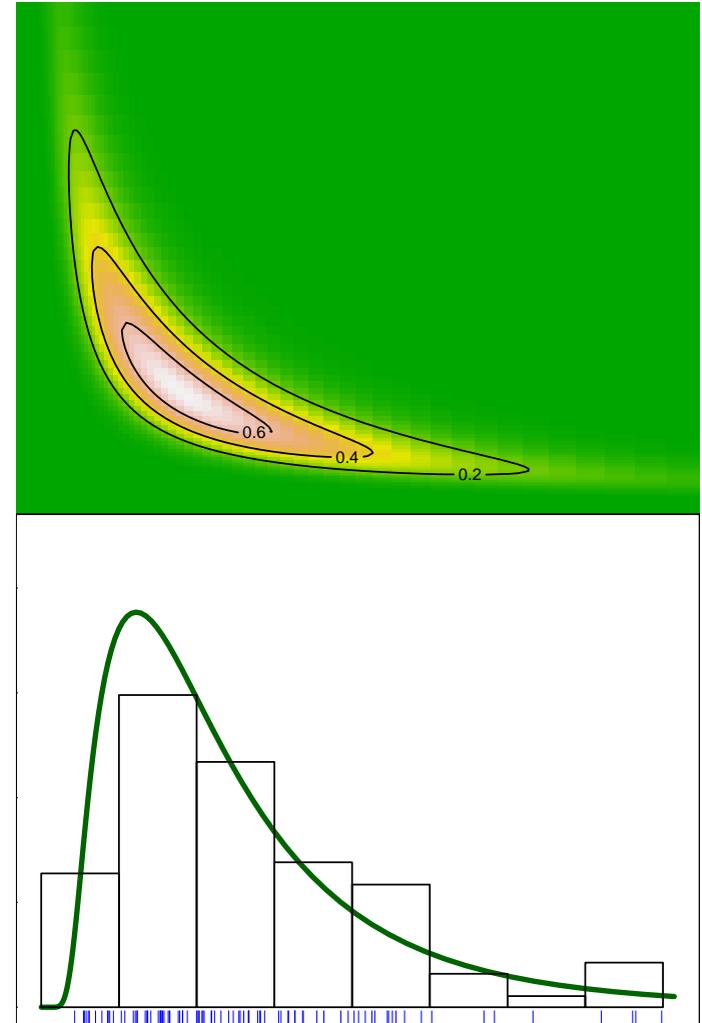
Laplace approximation of the needed integral

- We can <u>approximate</u> the needed integral by:

$$\int L(y, u, \theta) du \approx \int \exp(\text{``2. order Taylor} \log L\text{''}) du = L(y, \hat{u}_\theta, \theta) \sqrt{\frac{(2\pi)^q}{|(-\ell''_{uu}(y, u, \theta)|_{u=\hat{u}_\theta})|}}$$

- This is obtained from:
  - 2. order Taylor approximation of $\log L$ around $\hat{u}_\theta$ (so 1. order term is 0)
  - recognizing the multivariate Gaussian integral

- Notice that to calculate this we only need to:
  - Optimize
  - Take derivatives
  - Very very efficiently and accurately!

- In practice via TMB:
  - Code up the joint negative log likelihood
  - From R identify which quantities are considered to be random effects, as e.g:
    ```
    obj <- MakeADFun(data, parameters, random="u")
    ```

(There are other approaches)

Kalman filter

- Clever sequential algorithm
- For linear Gaussian systems (in its pure form)
- Requires initial assumptions

MCMC approximation

- Simulate Markov chain with posterior as its equilibrium
- Slow in complex models
- Difficult to judge convergence
- As precise as our patience allow

Utilization in assessment models

Missing observation

- Imagine we have a data set:

$$x_1 = \begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \end{pmatrix}, x_2 = \begin{pmatrix} x_{21} \\ x_{22} \\ x_{23} \end{pmatrix}, \ldots, x_7 = \begin{pmatrix} x_{71} \\ \\ x_{73} \end{pmatrix}$$

- (Notice x_{72} is missing)

- Want to run model like:

$$x_i \sim N(\mu, \Sigma) \quad, \quad i = 1 \ldots 7$$

- With random effects: Define the missing as a random effect - keep model code simple

$$x_1 = \begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \end{pmatrix}, x_2 = \begin{pmatrix} x_{21} \\ x_{22} \\ x_{23} \end{pmatrix}, \ldots, x_7 = \begin{pmatrix} x_{71} \\ U \\ x_{73} \end{pmatrix}$$

- Without random effects: specialized code to find marginal distribution

- Simple in this case — but not in general $f(x_i) \sim N(\mu, \Sigma)$

Missing observations (code ex)

- Some of the observations in the `obs` vector are missing `NA`

- On the R-side we can add the random effects as:

```r
par$missing <- numeric(sum(is.na(mydata$obs)))  ## count them
obj <- MakeADFun(mydata, par, random="missing")
```

- Then in the C-code we can use them where observations are missing

```c
int idxmis=0;
for(int i=0;i<nobs;i++){
  if(isNA(obs(i))){
    obs(i)=missing(idxmis++);
  }
}
```

- The rest of the program is unchanged.

- Then the model can work where observations were missing and even produce predictions of the missing (if we should need it).

- Remember we are using the Laplace approximation

- (`isNA` is a small helper function)

Introduce correlations

- For a mackerel stock we need to use tags

- The expected number of tags returned $r_{a,y}^{(j)}$ from the $j'th$ release is:

$$\mu_{a,y}^{(j)} = n_{scan} \frac{R^{(j)} p_{surv}}{N^{(j)}}$$

- where $N^{(j)}$ is the number in the cohort at release time

- Model could be:

$$r_{a,y}^{(j)} \sim NB(\mu_{a,y}^{(j)}, \phi) \quad , \quad \text{independent}$$

- Possibly we can expect the recaptures from same release event (j) to be correlated

- Can use random effects to introduce such correlations:

$$\mu_{a,y}^{(j)} = n_{scan} \frac{R^{(j)} p_{surv}}{N^{(j)}} e^{U_j}$$

- Here $U_j \sim \mathcal{N}(0, \sigma_U^2)$

- Getting correlation structure correct is the key part in getting correct weighting

Flexible processes over time

- A big challenge in parametric models - many problematic options available, e.g:

  – Constant

  – Constant in blocks (how many and where to cut)

  – Splines (how flexible e.g. no nodes)

  – Mean plus deviances (what should dev-sd be fixed at)

- One of the best features of random effects is that you can use random processes, e.g:

$$\log F_{y+1} = \log F_y + \varepsilon_y \quad , \quad \text{where } \varepsilon_y \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$

  – Flexible

  – Few parameters

  – No artificial choices

  – Natural framework for timeseries

- "Anything with a y-subscript should be a random effect" — Noel Cadigan

Processes in assessment

- The logical unobserved ones
  - Recruitment
  - Survival
  - Fishing mortality
- But should really also apply to:
  - Mean weights (stock, catch, landed, discard)
  - Maturity
  - Landing fraction
  - Natural mortality
- Should be inside the model because all of these processes are likely not independent

We need to predict

- Too much smoothing will bias the signal

- Too little smoothing will drown the signal in noise

- Correct amount will help you look ahead
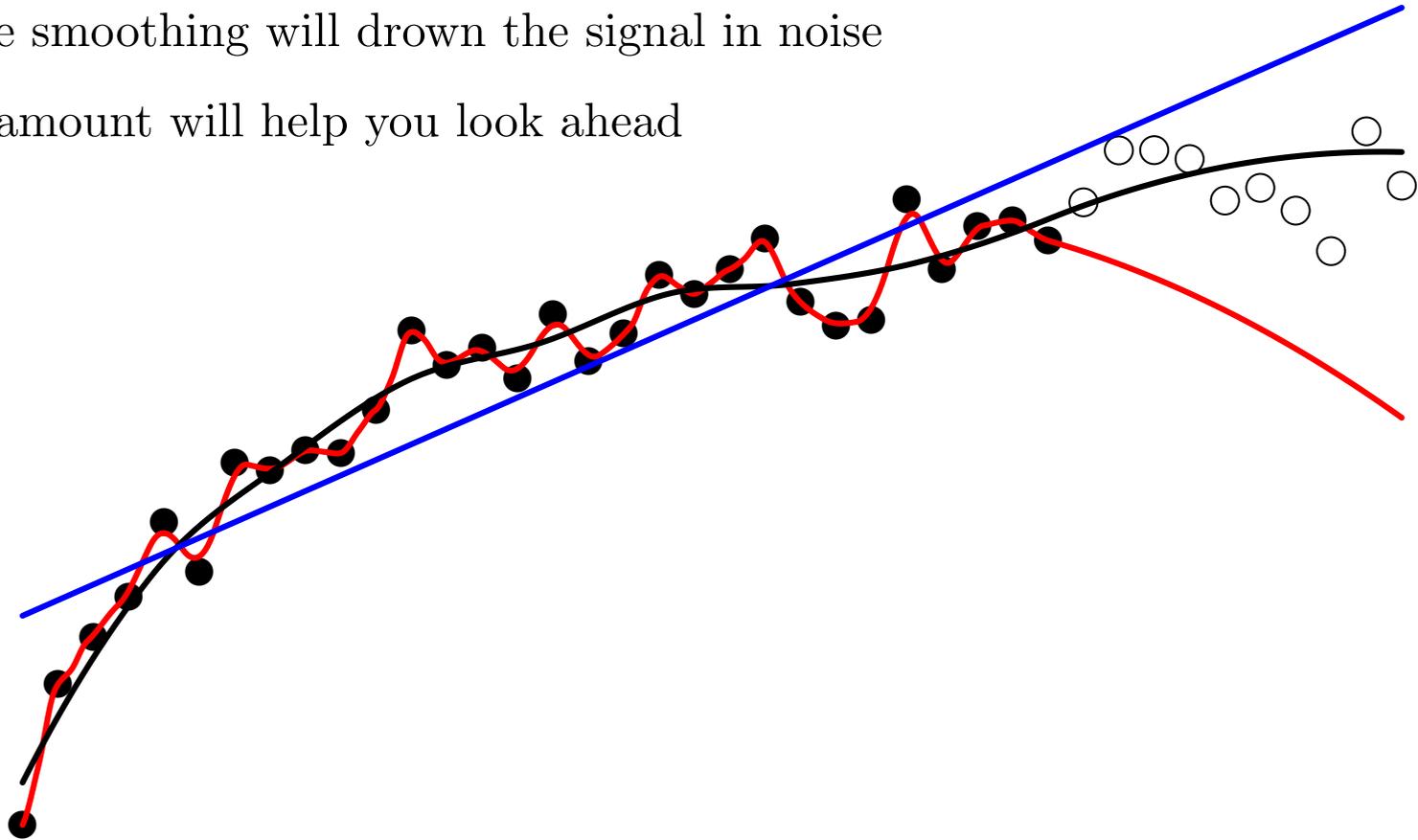
- Correct amount should not be subjective (e.g: 5-year average, fixed dev-sd, or spline nodes)

- Correct amount should be estimated

Predictions

- Random effects models are designed to predict

- Prediction is obvious when you have a model like this:

$$
\begin{pmatrix} \log N_{1,y} \\ \log N_{2,y} \\ \vdots \\ \log F_{1,y} \\ \log F_{2,y} \\ \vdots \end{pmatrix} = T(\begin{pmatrix} \log N_{1,y-1} \\ \log N_{2,y-1} \\ \vdots \\ \log F_{1,y-1} \\ \log F_{2,y-1} \\ \vdots \end{pmatrix}) + \varepsilon_{y-1}
$$

- and you know T and the distribution of ε, and have estimated all parameters.

- What is estimated is exactly how to move to the next time-step

- Noise is split into observation noise and process noise

- The estimated process noise express how well we can expect to predict

- Smoothness should be part of what we estimate

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
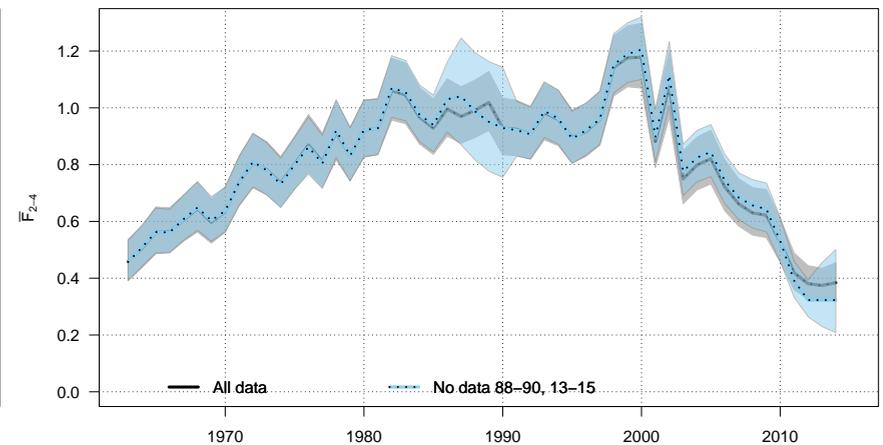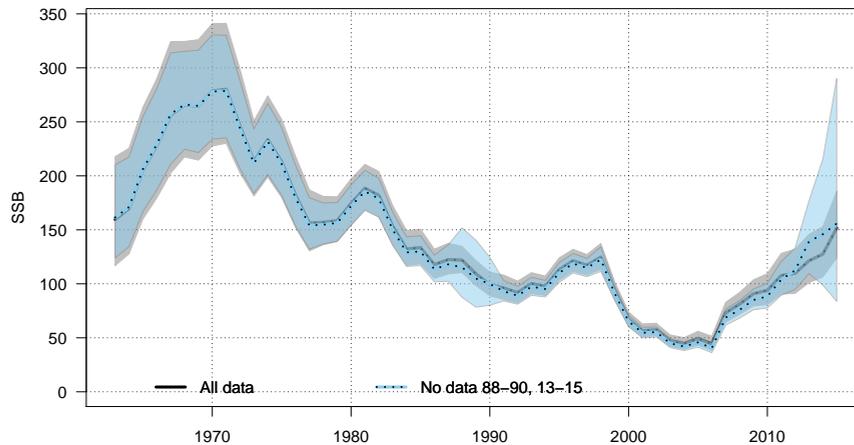
- The only thing that is real is the observations

- when evaluating (and comparing) models we should look at their ability to predict observations.

- Prediction should be part of the model — not left to an afterthought

Spatial extensions

- Random processes in 2D is a logical tool for spatial models

- Often we don't really want 5 boxes

- We need a flexible structure

- We need to predict to patches of space where we have less data

- So all said for the time varying quantities can be repeated for space

- In practice it may still make the models (too) slow

- Many efficient approaches (e.g. AR×AR, GMRF, Matérn) have recently become available in TMB

Correlations between stocks

- Stock development of multiple stocks can be linked in many ways
  - One stock can eat another
  - They can compete for same resources
  - They can depend on the same environmental conditions
  - They can be targeted by the same fishing fleets
- Most likely stocks are linked by a combination of many effects
- Multi-species models become hugely complex and data hungry
- However if we have stocks as random processes, then it is fairly simple to model them as correlated
- These estimated correlations can for instance help us:
  - get more realistic short-term predictions
  - if data is poor or missing sometimes for one of the stocks
  - get more realistic uncertainties
- The added model complexity is minimal

Linked stocks
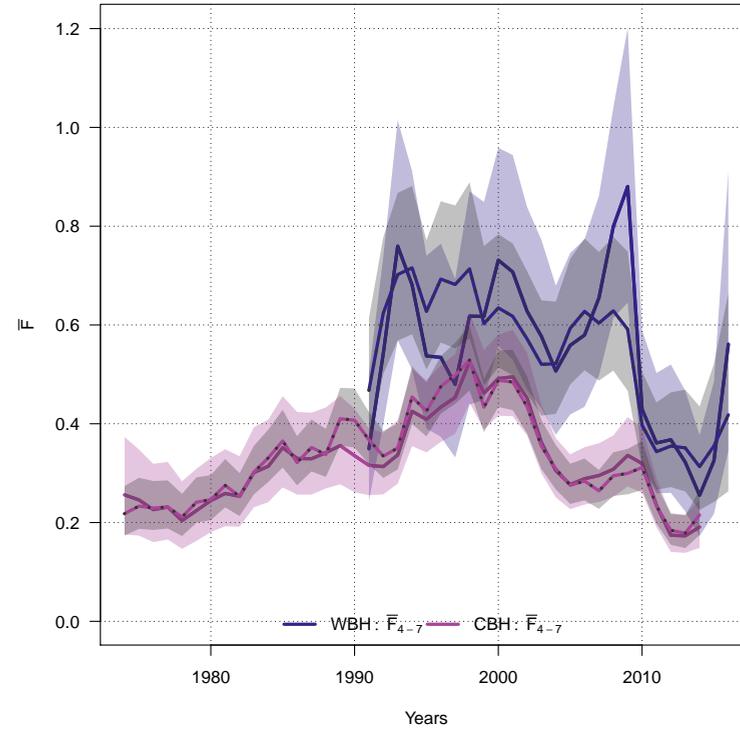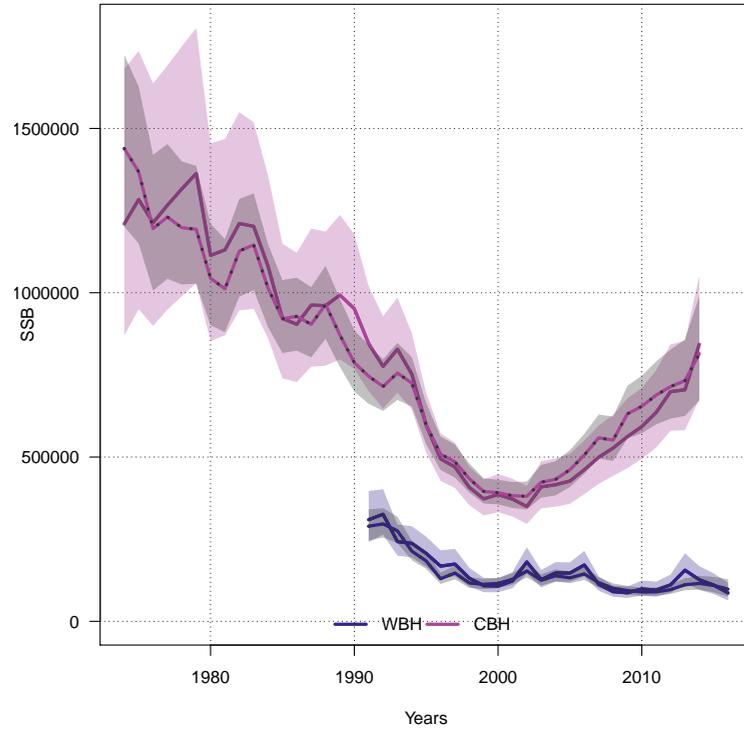

```
library(stockassessment)
cbh <- fitfromweb("cbh2015_tmb")
wbh <- fitfromweb("wbss_herring_2017_tmb")
library(multiStockassessment)
stocks <- c(WBH=wbh,CBH=cbh)
cs<-suggestCorStructure(stocks,nAgeClose=2)
mfitS<-multisam.fit(stocks,cs)
```

Simulation

- Fairly recent addition to TMB makes it simple to simulate from your models

- Simply add a simulation block <u>everywhere</u> you add to your likelihood function

```
for(int i=1;i<y.size();i++){
   ans+=-dnorm(lam(i),lam(i-1),sdRw,true);
   SIMULATE{
     lam(i)=rnorm(lam(i-1),sdRw);
   }
}

for(int i=0;i<y.size();i++){
   ans+=-dnorm(y(i),lam(i),sdObs,true);
   SIMULATE{
     y(i)=rnorm(lam(i),sdObs);
   }
}
SIMULATE{
   REPORT(lam)
   REPORT(y)
}
```

- Then from the R-side you can simply do:

```
simdata <- obj$simulate(complete=TRUE)
```

Check Laplace via simulation

- We can compare results to mcmc, but then we have to deal with all the issues with mcmc.

- 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

Final remarks

- Random effects are useful in many parts of assessment modelling

- Modelling tool should provide efficient ways to estimate models with random effects

- Especially for next-generation models

- More details in:

– DA Fournier, HJ Skaug, J Ancheta, J Ianelli, A Magnusson, MN Maunder, A Nielsen, J Sibert 2012. AD Model Builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. Optimization Methods and Software 27 (2), 233-249

– K Kristensen, A Nielsen, CW Berg, HJ Skaug, B Bell 2016. TMB: Automatic differentiation and Laplace approximation. Journal of Statistical Software 70 (5), 1-21

– A Nielsen, and CW Berg 2014. Estimation of time-varying selectivity in stock assessments using state-space models. Fisheries Research 158, 96-101

– CM Albertsen, A Nielsen, UH Thygesen 2017. Connecting single-stock assessment models through correlated survival. ICES Journal of Marine Science 75 (1), 235-244

– UH Thygesen, CM Albertsen, CW Berg, K Kristensen, and A Nielsen 2017. Validation of state space models fitted as mixed effects models. Environmental and Ecological Statistics 24 (2), 317-339

– http://tmb-project.org

– https://github.com/fishfollower/SAM

Thank you for listening

Appendix (random stuff someone may ask for)

Approx. REML inference for variance parameters

- Restricted/residual maximum likelihood reduces bias for variance parameters

- Especially important for small sample sizes

- Difficult to derive for non-linear models

- Can be approximated by defining mean-value parameters as random effects

The Laplace approximation

- We need to calculate the difficult integral

$$L_M(\theta, y) = \int_{\mathbb{R}^q} L(\theta, u, y) du$$

- So we set up an approximation of $\ell(\theta, u, y) = \log L(\theta, u, y)$

$$\ell(\theta, u, y) \approx \ell(\theta, \hat{u}_\theta, y) - \frac{1}{2}(u - \hat{u}_\theta)^t \left(-\ell''_{uu}(\theta, u, y)|_{u=\hat{u}_\theta}\right)(u - \hat{u}_\theta)$$

- Which (for given θ) is the 2. order Taylor approximation around:

$$\hat{u}_\theta = \operatorname*{argmax}_{u} L(\theta, u, y)$$

- With this approximation we can calculate:

$$L_M(\theta, y) = \int_{\mathbb{R}^q} L(\theta, u, y) du$$

$$\approx \int_{\mathbb{R}^q} e^{\ell(\theta, \hat{u}_\theta, y) - \frac{1}{2}(u - \hat{u}_\theta)^t \left(-\ell''_{uu}(\theta, u, y)|_{u = \hat{u}_\theta}\right)(u - \hat{u}_\theta)} du$$

$$= L(\theta, \hat{u}_\theta, y) \int_{\mathbb{R}^q} e^{-\frac{1}{2}(u - \hat{u}_\theta)^t \left(-\ell''_{uu}(\theta, u, y)|_{u = \hat{u}_\theta}\right)(u - \hat{u}_\theta)} du$$

$$= L(\theta, \hat{u}_\theta, y) \sqrt{\frac{(2\pi)^q}{|\left(-\ell''_{uu}(\theta, u, y)|_{u = \hat{u}_\theta}\right)|}}$$

- In the last step we remember the normalizing constant for a multivariate normal, and that $|A^{-1}| = 1/|A|$.

- Taking the logarithm we get:

$$\ell_M(\theta, y) \approx \ell(\theta, \hat{u}_\theta, y) - \frac{1}{2}\log(|\left(-\ell''_{uu}(\theta, u, y)|_{u = \hat{u}_\theta}\right)|) + \frac{q}{2}\log(2\pi)$$

The math

$$E_\theta\left(\nabla\ell(\theta; X)\right) = \int P_\theta(x)\nabla\ell(\theta; x)dx$$

$$= -\int P_\theta(x)\frac{1}{P_\theta(x)}\nabla P_\theta(X)dx$$

$$= -\nabla\int P_\theta(x)dx$$

$$= 0$$

Notice we have already seen this

- In the Poisson distribution the variance is equal to the mean, which is an assumption that is not always valid.

- Consider the model:

$$Y \sim \text{Pois}(\lambda), \quad \text{where} \quad \lambda \sim \Gamma\left(n, \frac{1-\phi}{\phi}\right) \quad 0 < \phi < 1$$

- It can be shown that:

$$Y \sim \text{Nbinom}(n, \phi)$$

- Notice:

  - No λ in marginal likelihood for Y

  - Analytical integration is not the typical case