Fisheries data integration: a spatio-temporal SDM framework with the LGNB model

Marie-Christine Rufener

macrufener@gmail.com

19 April 2021

Course agenda

- **Fisheries data integration**
  - Theoretical concepts
  - Developing the LGNB-SDM
  - Case study application

  BREAK (10 min?)

- **Introduction to Template Model Builder (TMB)**
  - Basics of MLE
  - Toy exercise
  - Shifting gears to TMB

  BREAK (10 min?)

- **LGNB-SDM tutorial**

1

Fisheries data integration

Theoretical concepts

Fisheries data integration

How do we know where and how many fish are out there?

Scientific surveys
(Fishery-independent)

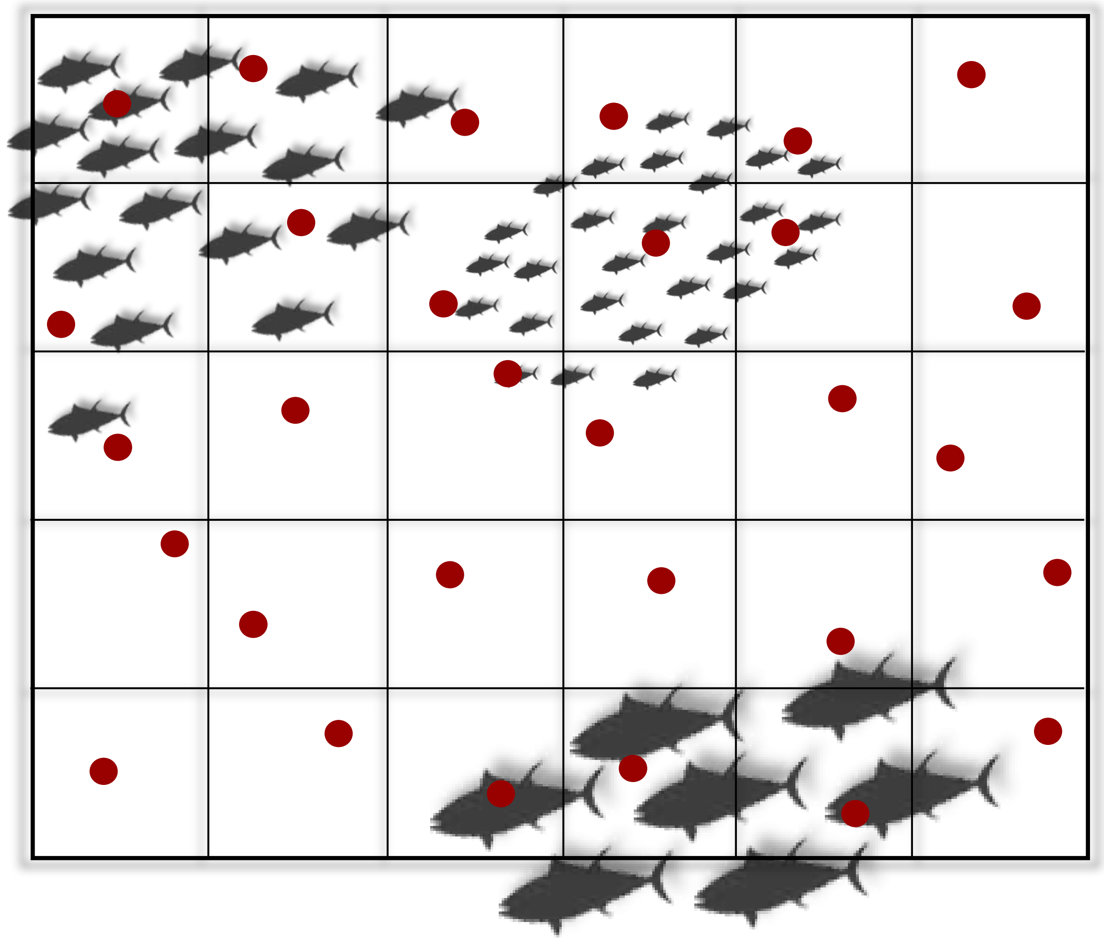
Commercial fisheries
(Fishery-dependent)

Fisheries data integration

How do we know where and how many fish are out there?

Scientific surveys (Fishery-independent)

- **<u>Sampling design</u>**: systematic and/or stratified random sampling design; standardized fishing gear and fishing effort
- **<u>Spatial coverage</u>**: wide
- **<u>Temporal coverage</u>**: narrow
- **<u>Costs per unit observation:</u>** $$$

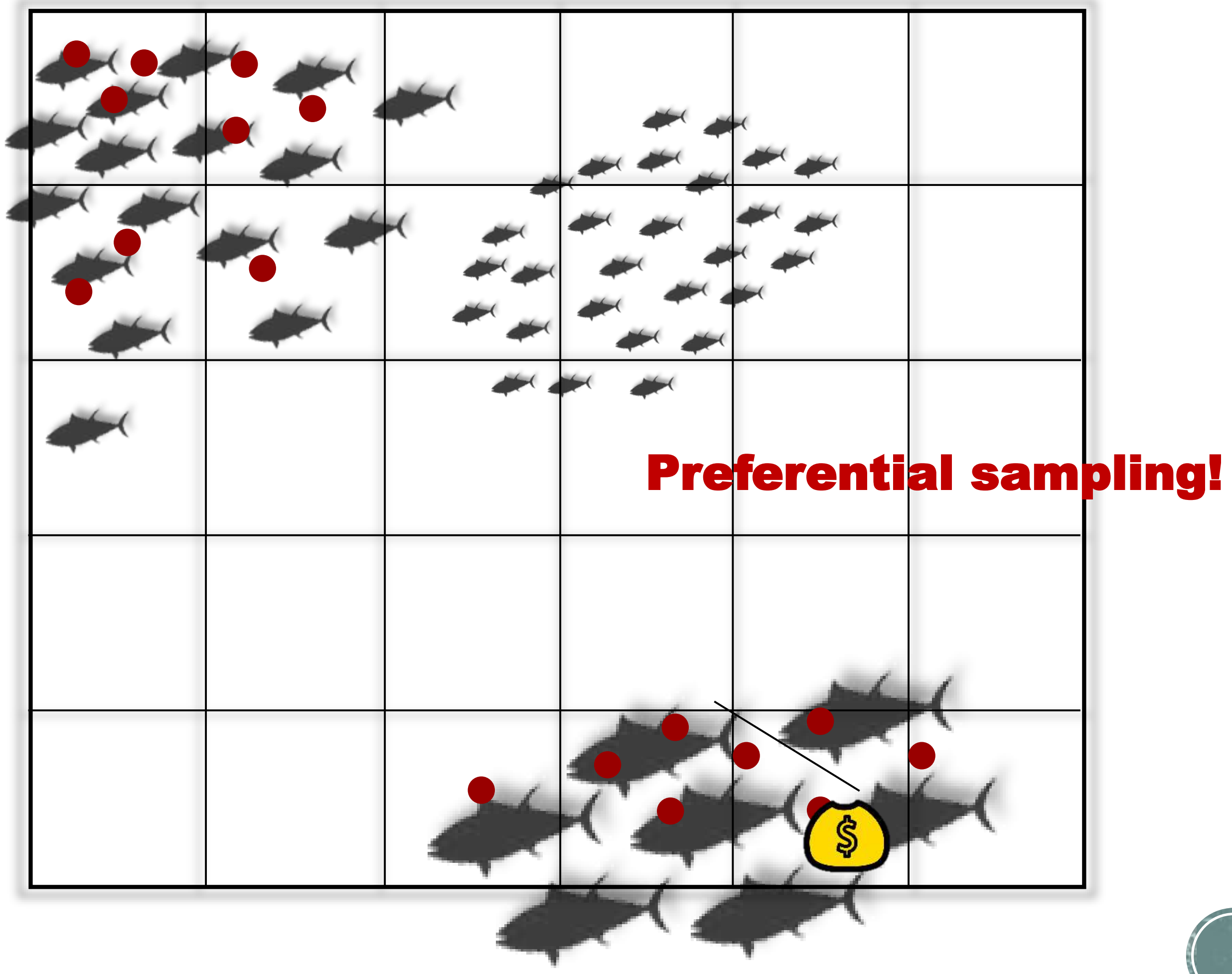


Fisheries data integration

How do we know where and how many fish are out there?

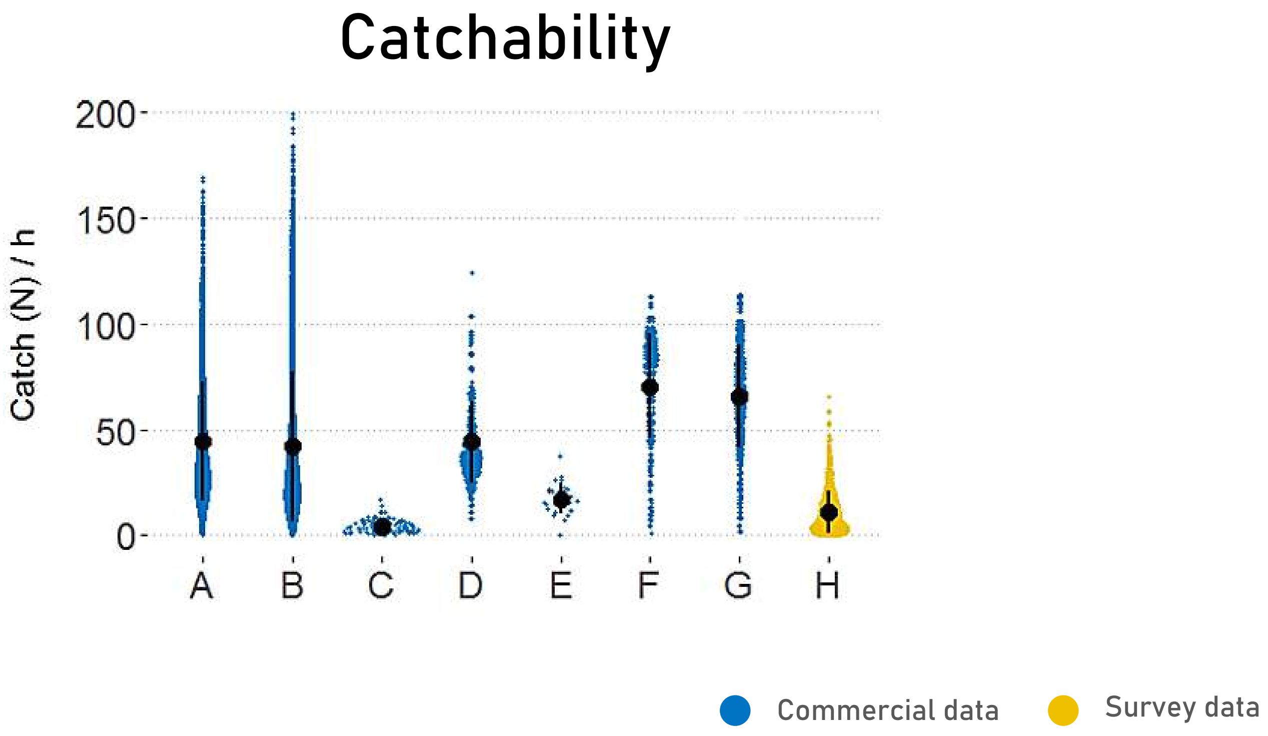
Commercial fisheries (Fishery-dependent)

- **Sampling design**: none; multiple fishing gears and fishing efforts
- **Spatial coverage**: narrow
- **Temporal coverage**: wide
- **Costs per unit observation:** $



Fisheries data integration

How do we know where and how many fish are out there?



Métiers

A = OTB_DEF_>=105_1_110

B = OTB_DEF_>=105_1_120

C = OTB_DEF_90-104_0_0

D = PTB_DEF_>=105_1_110

E = PTB_DEF_>=105_1_120

F = SDN_DEF_>=105_1_110

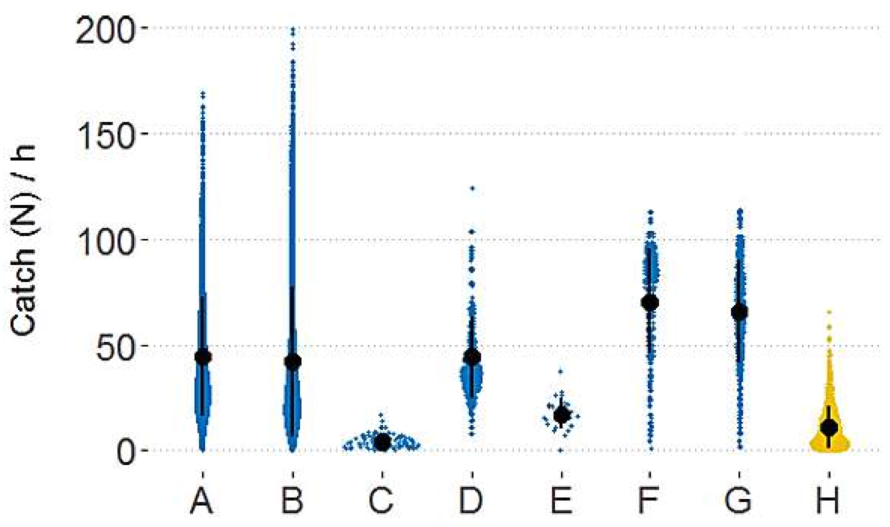
G = SDN_DEF_>=105_1_120

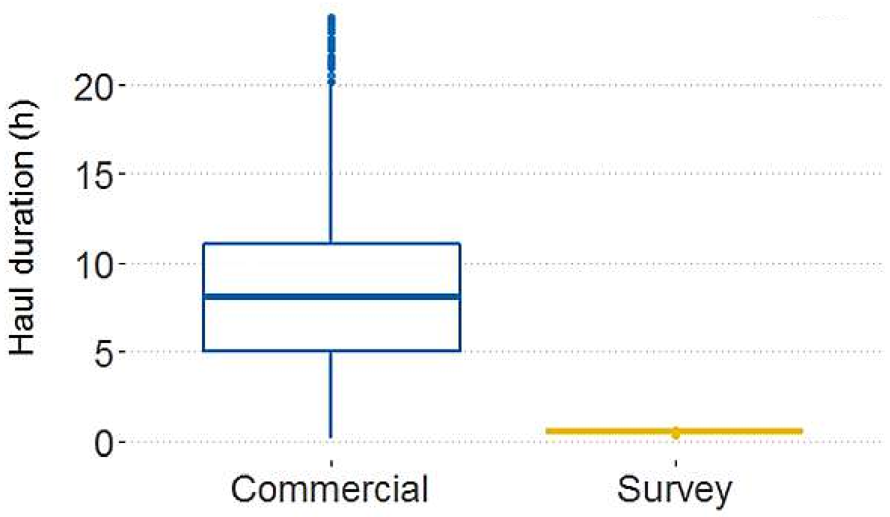
H = TVS

Examples from the Kattegat & western Baltic cod trawl fishery and Baltic International Trawl Surveys (BITS)

Fisheries data integration

How do we know where and how many fish are out there?





Commercial data Survey data

Métiers

A = OTB_DEF_>=105_1_110

B = OTB_DEF_>=105_1_120

C = OTB_DEF_90-104_0_0

D = PTB_DEF_>=105_1_110

E = PTB_DEF_>=105_1_120

F = SDN_DEF_>=105_1_110

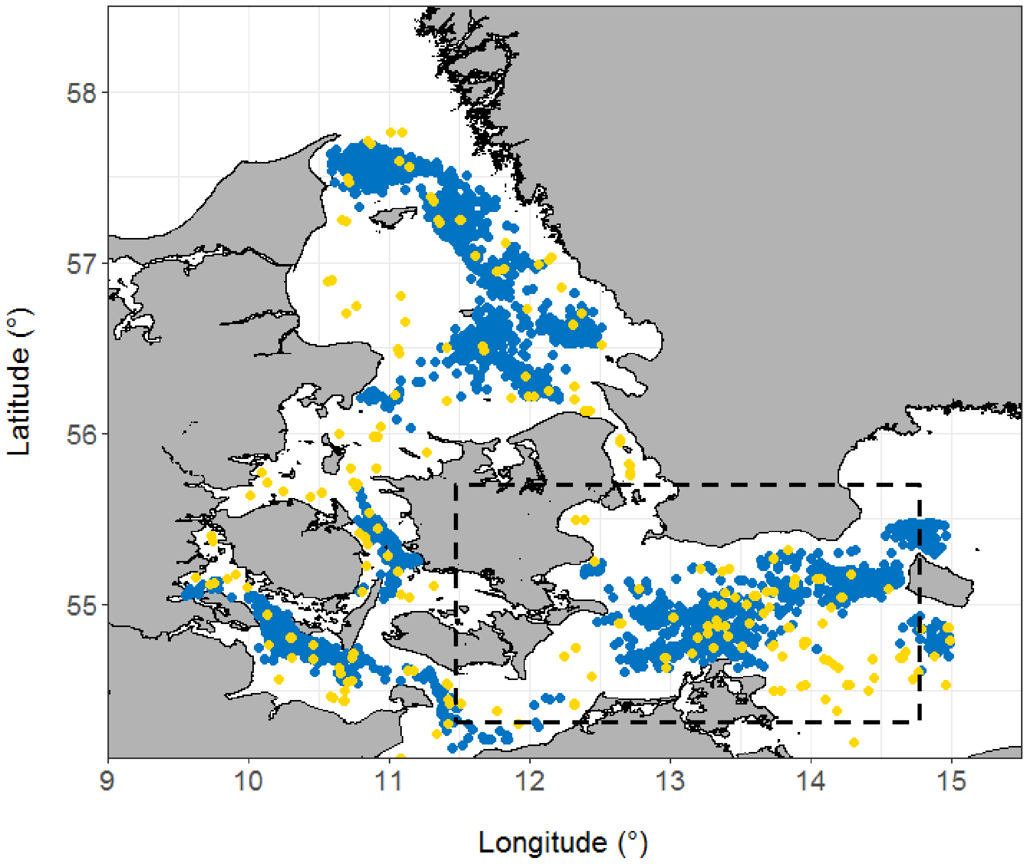
G = SDN_DEF_>=105_1_120

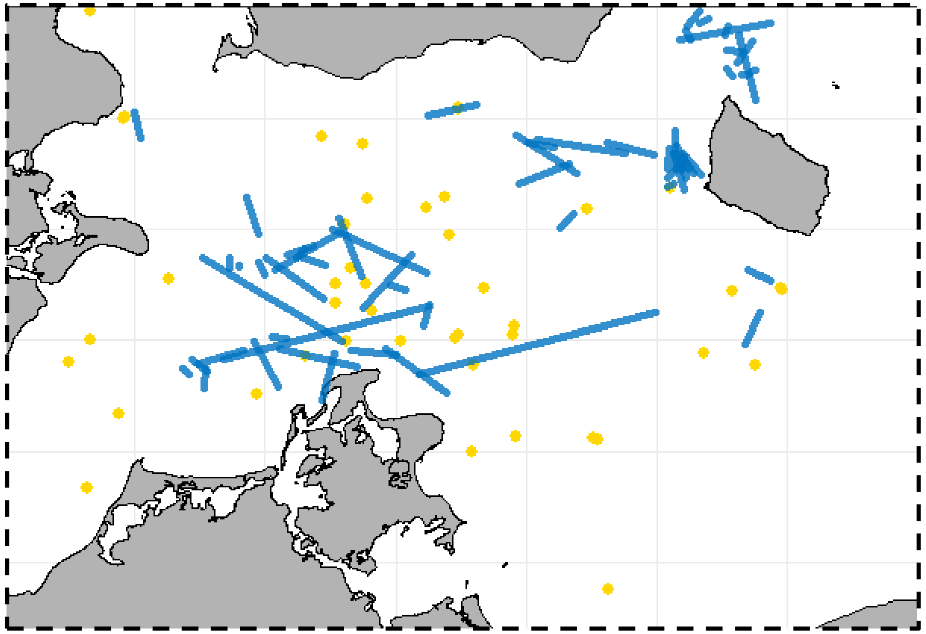
H = TVS

Examples from the Kattegat & western Baltic cod trawl fishery and Baltic International Trawl Surveys (BITS)

Fisheries data integration

How do we know where and how many fish are out there?

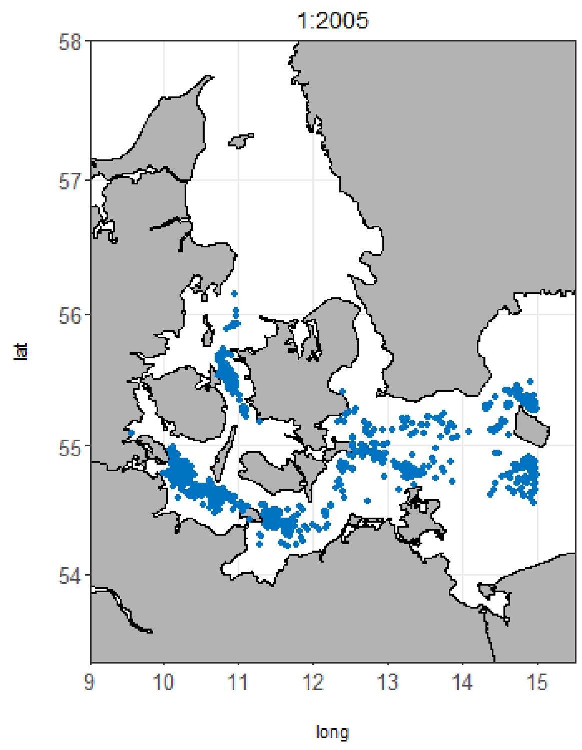


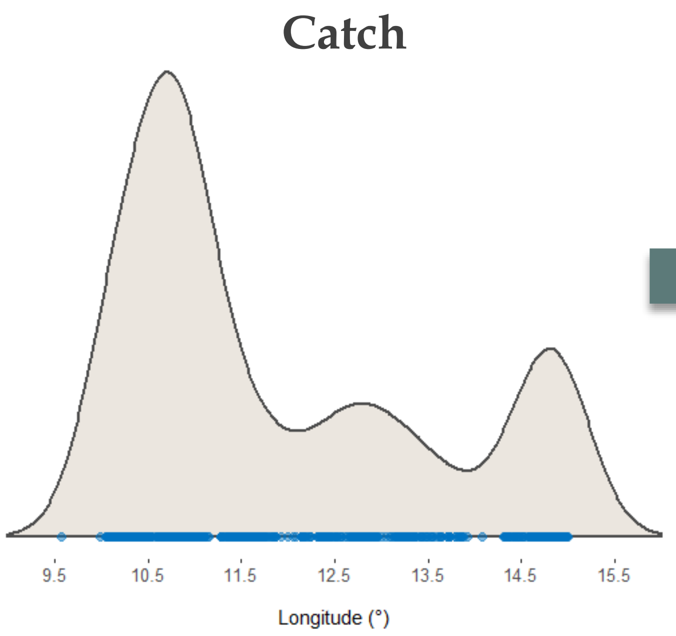


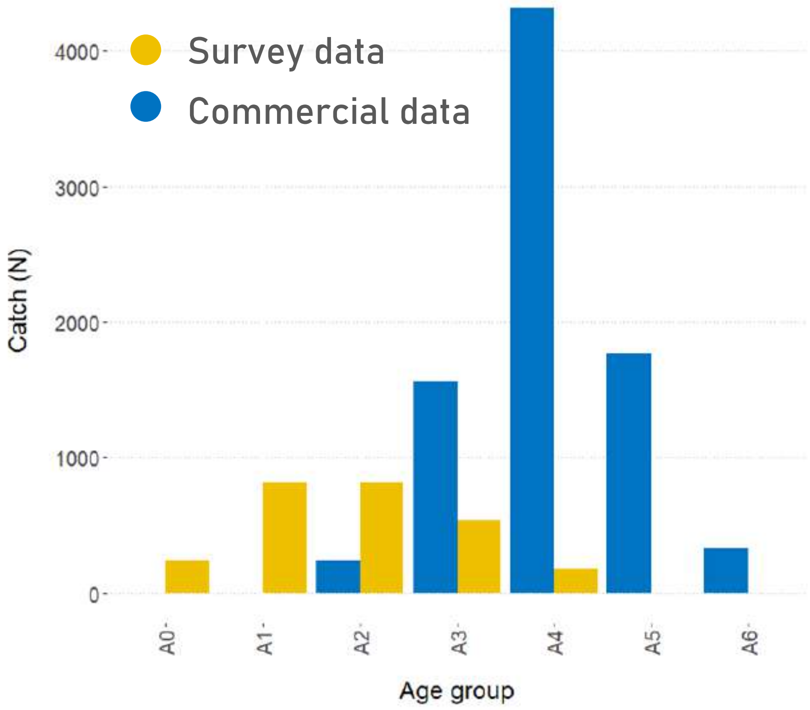
Examples from the Kattegat & western Baltic cod trawl fishery and Baltic International Trawl Surveys (BITS)

Fisheries data integration

How do we know where and how many fish are out there?







Examples from the Kattegat & western Baltic cod trawl fishery and Baltic International Trawl Surveys (BITS)

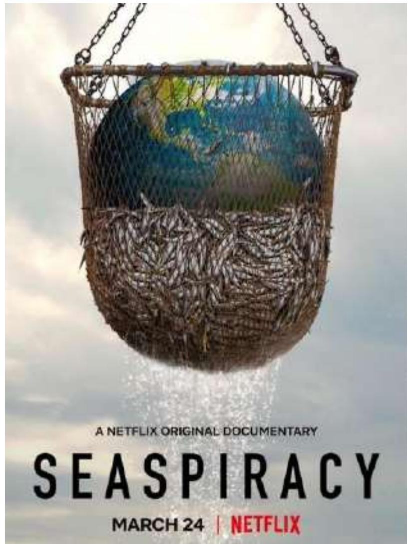
Fisheries data integration

How do we know where and how many fish are out there?

Rapid worldwide depletion of predatory fish communities

Ransom A. Myers & Boris Worm

Nature **423**, 280–283(2003) | Cite this article



RESEARCH ARTICLE

Impacts of Biodiversity Loss on Ocean Ecosystem Services

Boris Worm[1,*], Edward B. Barbier[2], Nicola Beaumont[3], J. Emmett Duffy[4], Carl Folke[5,6], Benjamin S. Halpern[7], Jeremy ...
+ See all authors and affiliations

Science 03 Nov 2006:
Vol. 314, Issue 5800, pp. 787-790
DOI: 10.1126/science.1132294

Fisheries data integration

How do we know where and how many fish are out there?

The Pauly-Hilborn dilemma

Pauly

Hilborn

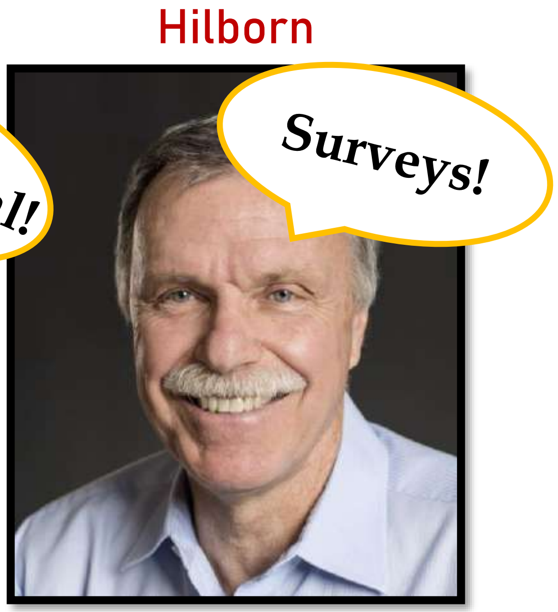
Hilborn: https://sustainablefisheries-uw.org/staff/

Pauly: https://europe.oceana.org/en/about-oceana/people-partners/board-directors/dr-daniel-pauly

Fisheries data integration

How do we know where and how many fish are out there?

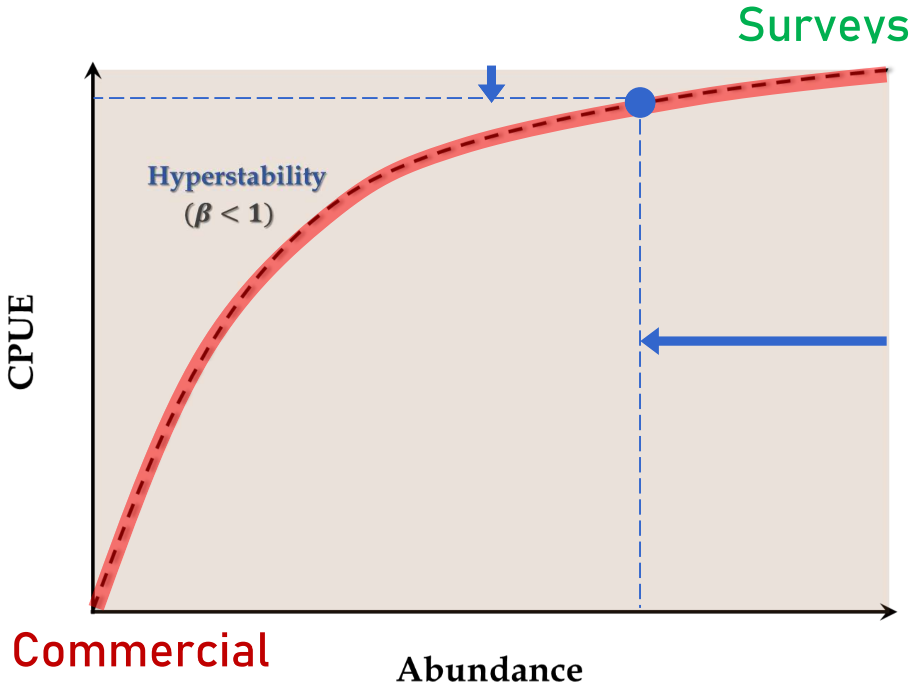
The Pauly-Hilborn dilemma



Hilborn: https://sustainablefisheries-uw.org/staff/

Pauly: https://europe.oceana.org/en/about-oceana/people-partners/board-directors/dr-daniel-pauly

$$C = qNE^{\beta}$$



$C = catch$
$q = catchability$
$N = No. ind.$
$E = Effort$

Developing the LGNB model

Fisheries data integration

How can we tackle these differences?

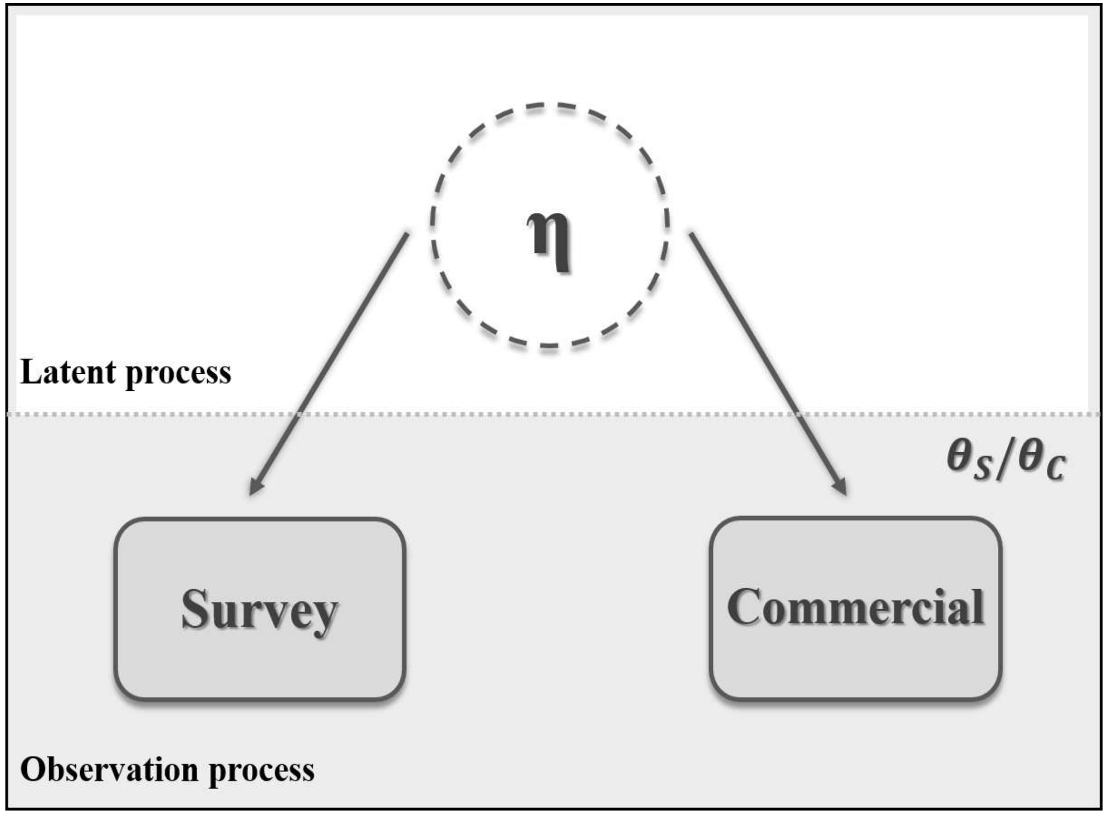
Hierarchical models (point-process models)

Observation process ($Y | \lambda$)

(e.g., Catch)

Latent process (λ)

(e.g., Abundance field)



* Figure extracted from Rufener et al. – Manuscript under review

Fisheries data integration

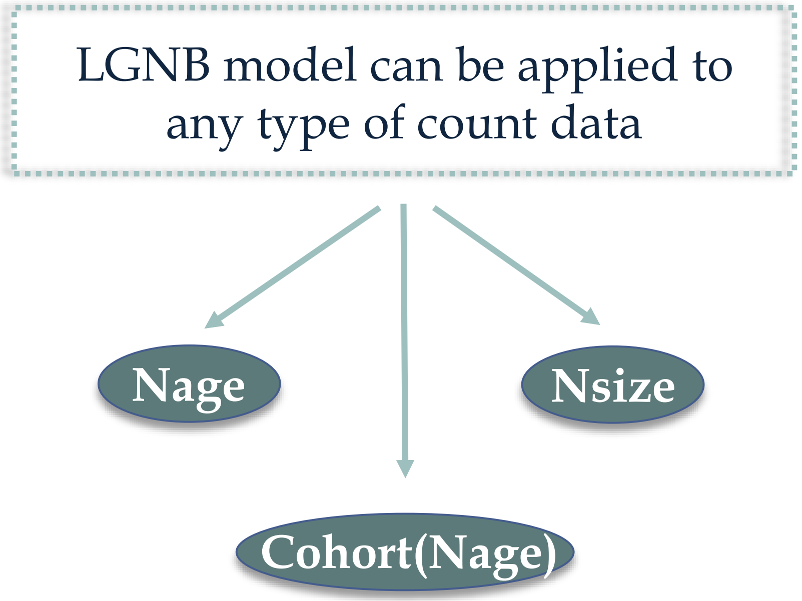
How can we tackle these differences?

The LGNB model

Observation process (Y | λ)

$$Y(s,t) \sim NB(\lambda(s,t), \phi)$$

(Log-Gaussian Negative Binomial process – LGNB)



Fisheries data integration

How can we tackle these differences?

The LGNB model

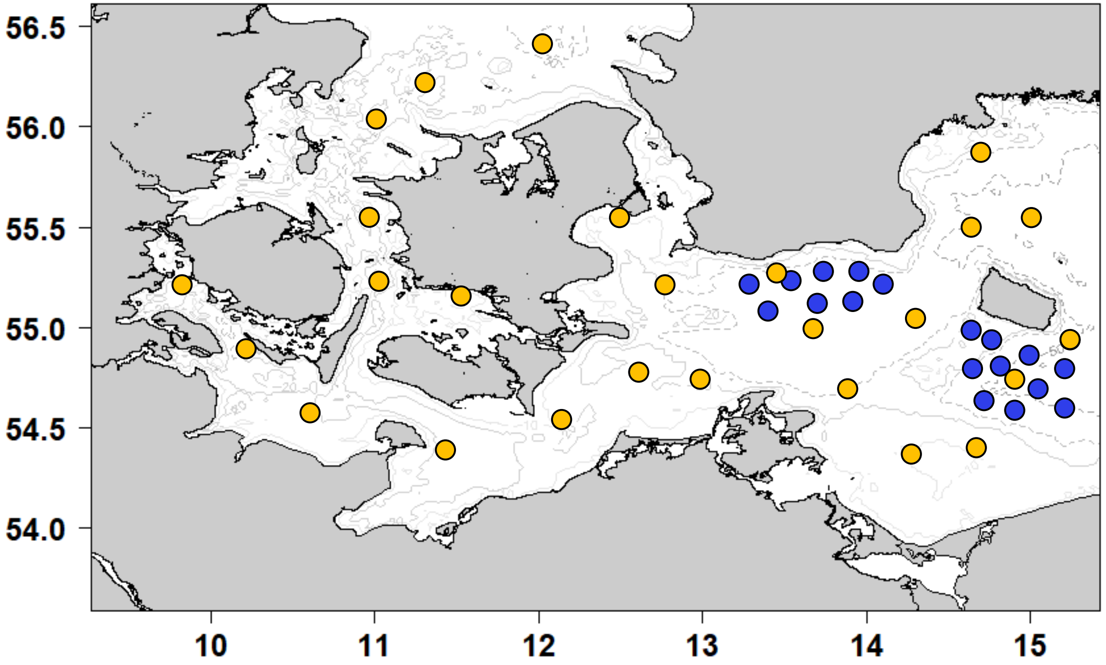
Observation process ($Y \mid \lambda$)

Survey catch ($Y_{sur} \mid \lambda$)

Commercial catch ($Y_{com} \mid \lambda$)

$$log(\mu_i^{\mathrm{SUR}}) = \log(\lambda(s_i, t_i)) + \sum_{\mathrm{k=1}}^{K_{\mathrm{SUR}}} \beta_k^{\mathrm{SUR}} X_{k,i}^{\mathrm{SUR}} + \gamma_i$$

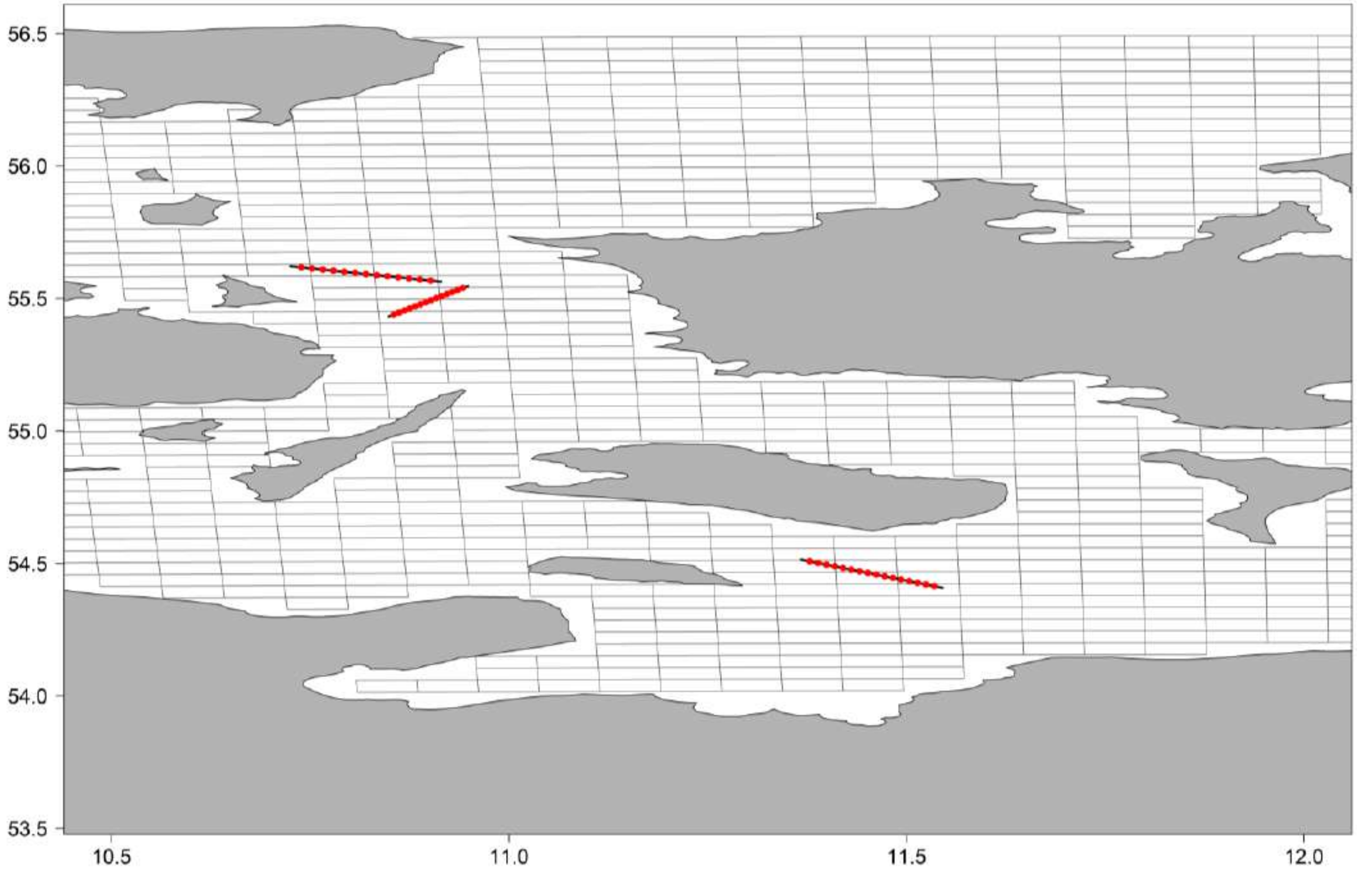
$$log(\mu_i^{\mathrm{COM}}) = \log(\lambda(s_i, t_i)) + \sum_{\mathrm{k=1}}^{K_{\mathrm{COM}}} \beta_k^{\mathrm{COM}} X_{k,i}^{\mathrm{COM}}$$



Fisheries data integration

How can we tackle these differences?

The LGNB model



Haul discretization

$$log(\mu_i^{\mathrm{COM}}) = \log\big(\lambda(s_i, t_i)\big) + \sum_{\mathrm{k=1}}^{K_{\mathrm{COM}}} \beta_k^{\mathrm{COM}} X_{k,i}^{\mathrm{COM}}$$

$$E\big(Y_j^{COM}|\lambda\big) = \sum\nolimits_{i\in I_j} \mu_i^{COM}$$

$j = transect$

Fisheries data integration

How can we tackle these differences?

The LGNB model

$$log(\mu_i^{\text{COM}}) = \log\big(\lambda(s_i, t_i)\big) + \sum_{\text{k}=1}^{K_{\text{COM}}} \beta_k^{\text{COM}} X_{k,i}^{\text{COM}}$$

Preferential sampling

$$P(V_i = v_i | \eta) = \frac{\lambda(s_i, t_i)^{\alpha_{f_i}}}{\sum_{s \in G_{f_i}} \lambda(s, t_i)^{\alpha_{f_i}}}$$

v = sampling position

G = spatial grid

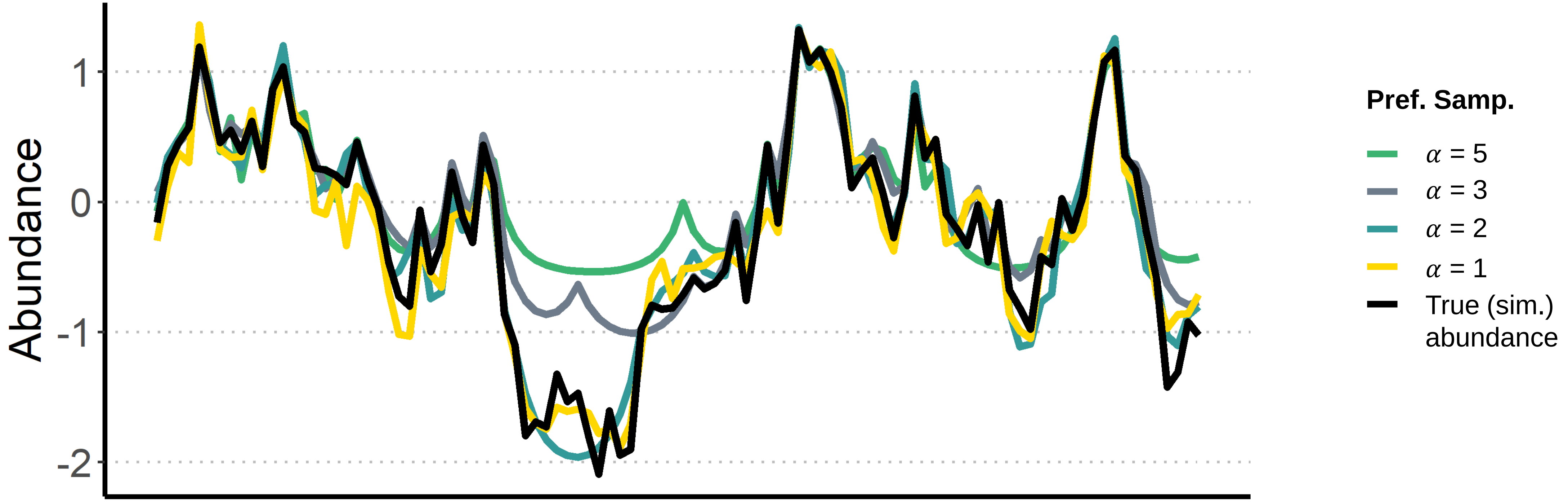
f = sampling support area

α = 0 : No PS
α > 0 : + PS (high density areas)
α < 0 : - PS (low density areas)

Fisheries data integration

How can we tackle these differences?

The LGNB model



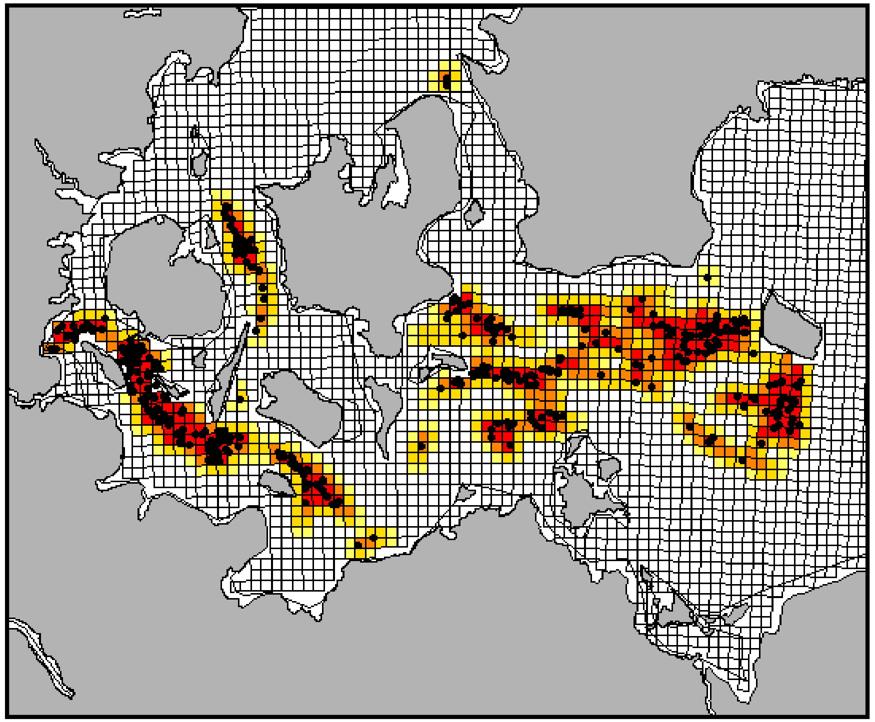
Fisheries data integration

How can we tackle these differences?

The LGNB model

$$P(V_i = v_i | \eta) = \frac{\lambda(s_i, t_i)^{\alpha_{f_i}}}{\sum_{s \in G_{f_i}} \lambda(s, t_i)^{\alpha_{f_i}}}$$

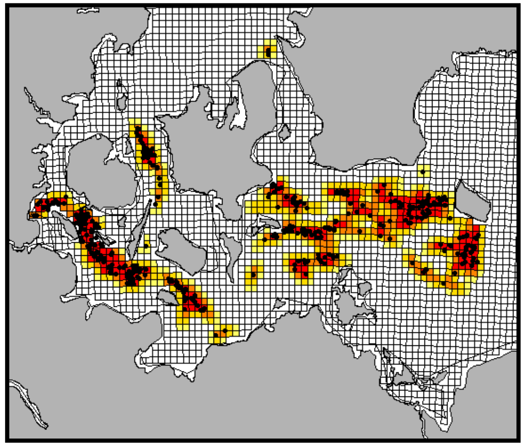
Sampling support area (*f*)



Fisheries data integration

How can we tackle these differences?

The LGNB model

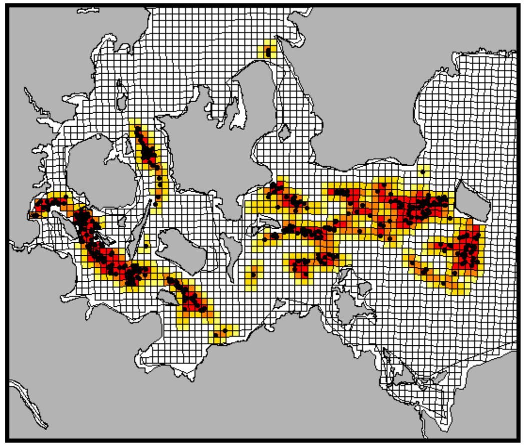


- **Cases where no PS occur**
  - No α-parameter is estimated (e.g., survey data) -> No sampling support area

Fisheries data integration

How can we tackle these differences?

The LGNB model

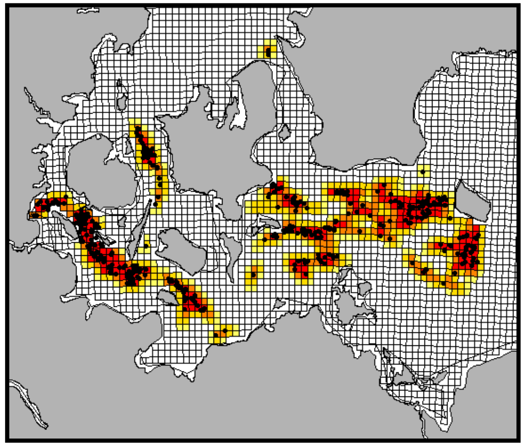


- **Cases where no PS occur**
  - No α-parameter is estimated (e.g., survey data) -> No sampling support area

- **Cases where PS occur without temporally varying fishing effort**
  - Only one α-parameter is estimated -> Single sampling support area

Fisheries data integration

How can we tackle these differences?
The LGNB model



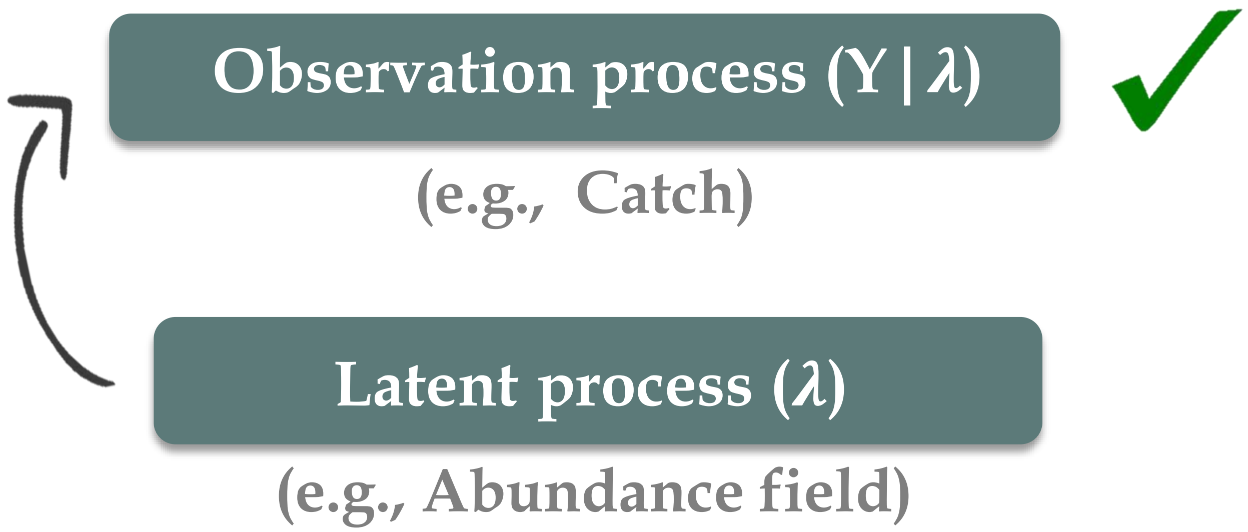
- **Cases where no PS occur**
  - No α-parameter is estimated (e.g., survey data) -> No sampling support area
- **Cases where PS occur without temporally varying fishing effort**
  - Only one α-parameter is estimated -> Single sampling support area
- **Cases where PS occur with temporally varying fishing effort***
  - Multiple α-parameters are estimated -> Multiple sampling support areas

* Code implemented, but not yet tested

Fisheries data integration

How can we tackle these differences?

The LGNB model



Fisheries data integration

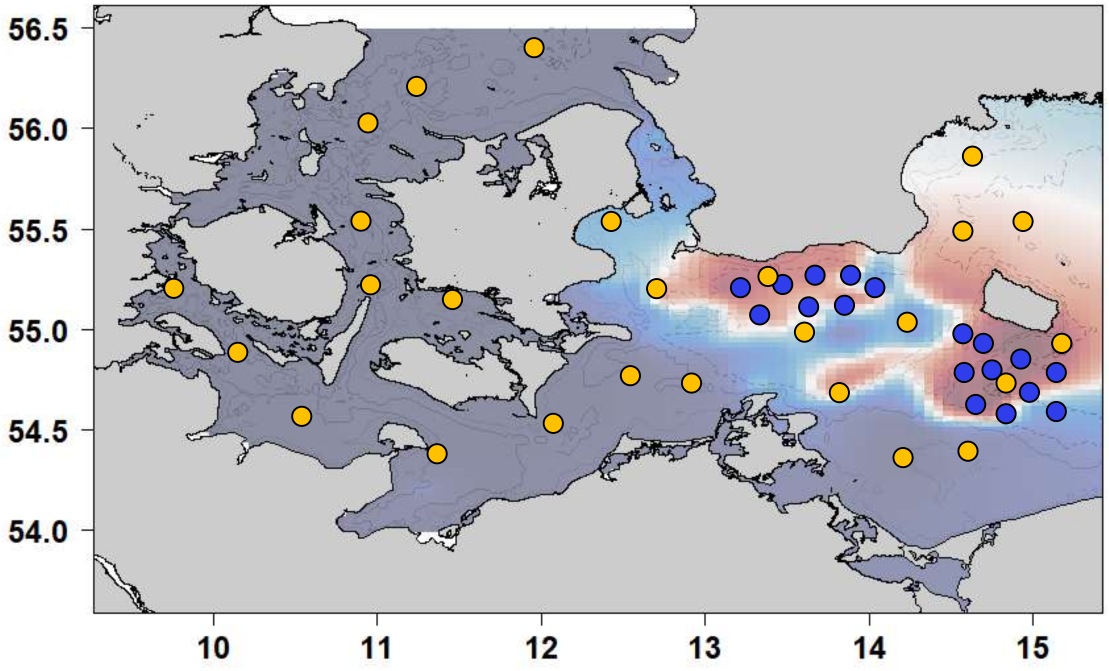
How can we tackle these differences?

The LGNB model

Latent process (λ)

(Abundance field)

$$\lambda(s,t) = \exp\left(\sum_{k=1}^{K} \beta_k X_k(s,t) + \xi(s,t)\right)$$



Fisheries data integration

How can we tackle these differences?

The LGNB model

$$\left(\xi(\mathrm{s},\mathrm{t})\right) \sim MVN(\mathbf{0},\boldsymbol{\Sigma})$$

$$\sum = \boldsymbol{\Sigma_S} \otimes \boldsymbol{\Sigma_T} = \begin{pmatrix} s_{11}\boldsymbol{\Sigma_T} & \cdots & s_{1n}\boldsymbol{\Sigma_T} \\ \vdots & \ddots & \vdots \\ s_{n1}\boldsymbol{\Sigma_T} & \cdots & s_{nn}\boldsymbol{\Sigma_T} \end{pmatrix}$$

Separable covariance matrix

Fisheries data integration

How can we tackle these differences?

The LGNB model

Temporal correlation (AR1)

$$\Sigma_T(t_1, t_2) = \rho^{|t_1 - t_2|*}$$

$*$ t can be described by any desired time resolution (monthly, quarterly, yearly …)

Fisheries data integration

How can we tackle these differences?

The LGNB model

Spatial correlation (CAR)

$$\boldsymbol{\Sigma_S} = \boldsymbol{\Sigma^{-1}} = \boldsymbol{Q_{ij}} = \begin{cases} -\kappa, & \text{if cell i neighbors cell j} \\ \kappa(m_i + \delta), & \text{if i = j} \\ 0, & \text{otherwise} \end{cases}$$

$$\sigma^2 = \frac{1}{M} tr(\boldsymbol{Q}^{-1}), \qquad H = \frac{h}{log(1 + \frac{\delta}{2} + \sqrt{\delta + \delta^2/4}}$$

$\kappa = scale$

$m_i = No. of\ neighbors\ of\ grid\ cell$

$\delta = decorrelation$

$M = No. grid\ cells$

$H = Decorrelation\ range$

$h = Grid\ cell\ zise$

$\sigma^2 = variance\ of\ GRF$

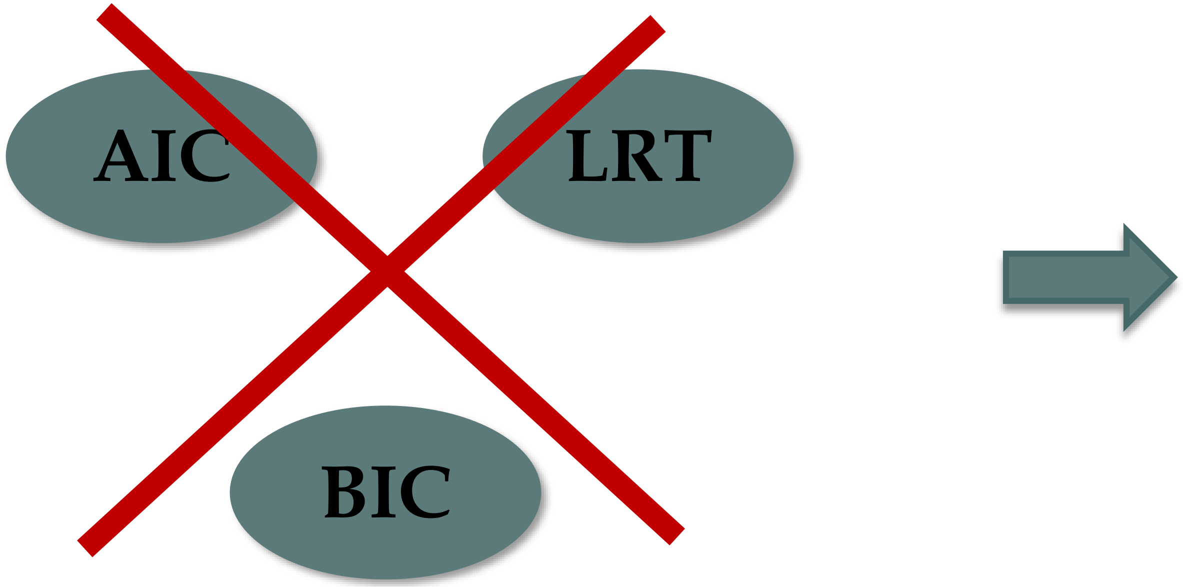
Kristensen et al., 2014; doi: 10.1139/cjfas-2013-0151

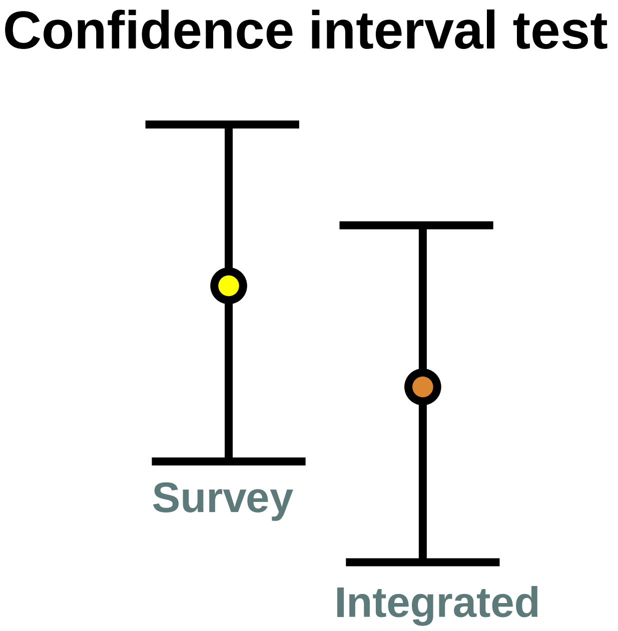
Fisheries data integration

How can we tackle these differences?

The LGNB model

Validating the integrated LGNB-SDM





Fisheries data integration

How can we tackle these differences?

The LGNB model

Validating the integrated LGNB-SDM

Parameters

1.a	Fit the survey model using the $\boldsymbol{L_{SUR}}$ equation and denote the estimates $\hat{\boldsymbol{\theta}}^{(1)}$ and $\hat{\boldsymbol{\theta}}_{SUR}^{(1)}$.

Fisheries data integration

How can we tackle these differences?

The LGNB model

Validating the integrated LGNB-SDM

Parameters

1.a	Fit the survey model using the $\boldsymbol{L_{SUR}}$ equation and denote the estimates $\hat{\boldsymbol{\theta}}^{(1)}$ and $\hat{\boldsymbol{\theta}}_{SUR}^{(1)}$.
2.a	Fit the integrated model using the $\boldsymbol{L_{BOTH}}$ equation and denote the estimates $\hat{\boldsymbol{\theta}}^{(2)}$, $\hat{\boldsymbol{\theta}}_{SUR}^{(2)}$ and $\hat{\boldsymbol{\theta}}_{COM}^{(2)}$.

Fisheries data integration

How can we tackle these differences?

The LGNB model

Validating the integrated LGNB-SDM

Parameters

1.a Fit the survey model using the $\boldsymbol{L_{SUR}}$ equation and denote the estimates $\hat{\boldsymbol{\theta}}^{(1)}$ and $\hat{\boldsymbol{\theta}}_{SUR}^{(1)}$.

2.a Fit the integrated model using the $\boldsymbol{L_{BOTH}}$ equation and denote the estimates $\hat{\boldsymbol{\theta}}^{(2)}$, $\hat{\boldsymbol{\theta}}_{SUR}^{(2)}$ and $\hat{\boldsymbol{\theta}}_{COM}^{(2)}$.

3.a Check that the second estimates are within the multivariate confidence region based on the first estimates by reporting the p-value $Pr(X \geq x)$ of the statistic

$$x = 2\left(logL_{SUR}\left(\hat{\boldsymbol{\theta}}^{(1)}, \hat{\boldsymbol{\theta}}_{SUR}^{(1)}\right) - logL_{SUR}\left(\hat{\boldsymbol{\theta}}^{(2)}, \hat{\boldsymbol{\theta}}_{SUR}^{(2)}\right)\right)$$

where $X \sim \chi^2(\text{df})$ and $df = dim(\boldsymbol{\theta}) + dim(\boldsymbol{\theta}_{SUR})$.

Fisheries data integration

How can we tackle these differences?

The LGNB model

Validating the integrated LGNB-SDM

Random effects

1.b	Using the parameter estimates from (1a) obtain the most probable random effects $\hat{\boldsymbol{\lambda}}^{(1)}$ by maximizing the integrand of the $\boldsymbol{L_{SUR}}$ equation.
2.b	Using the parameter estimates from (2a) obtain the most probable random effects $\hat{\lambda}^{(2)}$ by maximizing the integrand of the $\boldsymbol{L_{BOTH}}$ equation.
3.b	Using the logarithm of the integrand of the $\boldsymbol{L_{COM}}$, report P of $\hat{\boldsymbol{\lambda}}^{(2)}$ being inside the confidence region of $\hat{\boldsymbol{\lambda}}^{(1)}$ using a χ^2-distribution with dim($\boldsymbol{\lambda}$) degrees of freedom.

Fisheries data integration

Summary

- Point-process model (state-space model w/ spatial component)
  - Any type of count-related data can be modelled
  - Hidden effects on the catch process can be accounted for through structured and unstructured random effects
  - Different sets of covariates can be included in both observation and latent processes
  - Can be applied to only one data source when the other is missing
  - Preferential sampling can be accounted for

Case study application

Fisheries data integration

Case study – WB cod fishery

Bridging the gap between commercial fisheries and survey data to model the spatiotemporal dynamics of marine species*

Marie-Christine Rufener, Kasper Kristensen, J. Rasmus Nielsen, François Bastardie

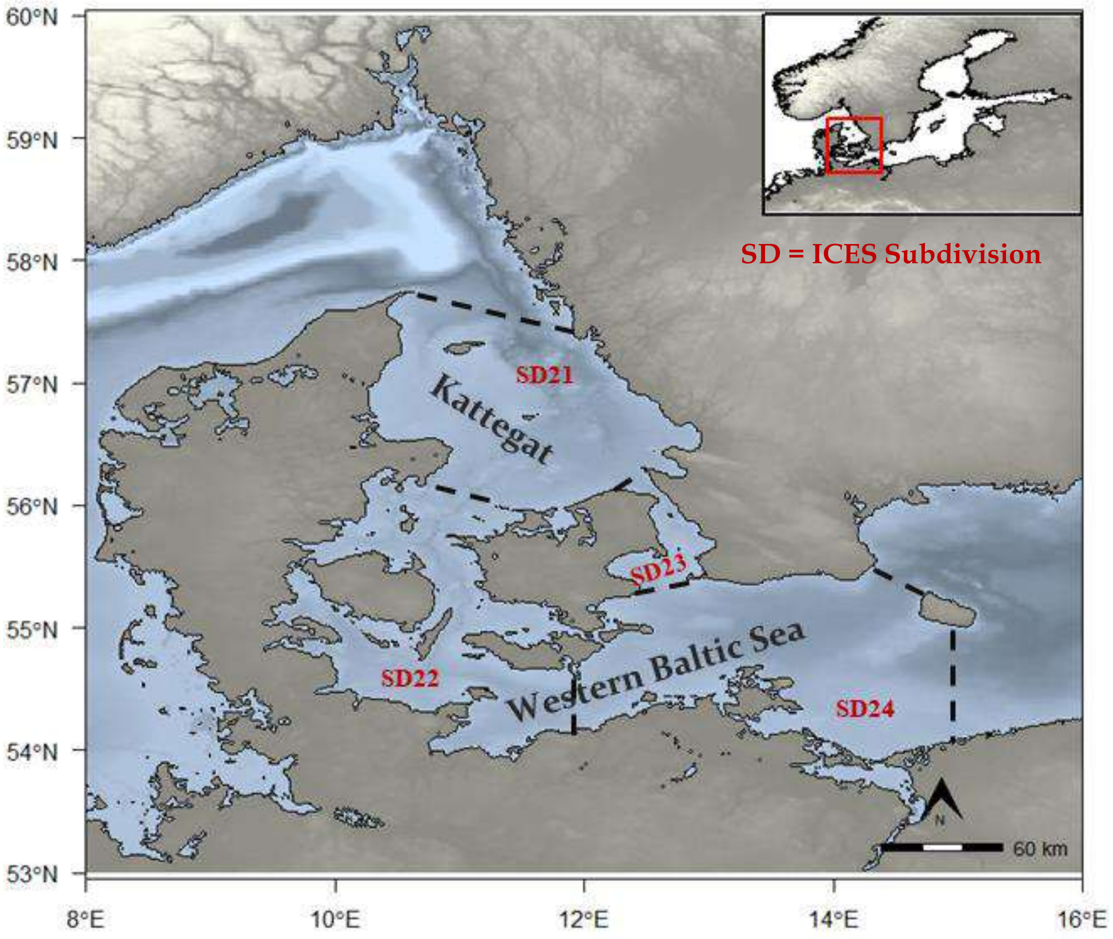
(Please do not share any figure related to this manuscript)

* Manuscript under review

Fisheries data integration

Case study – WB cod fishery (trawlers)

Background



2005-2016

Survey data

- BITS (1^{st} & 4^{th} Quarters)
- 1808 hauls

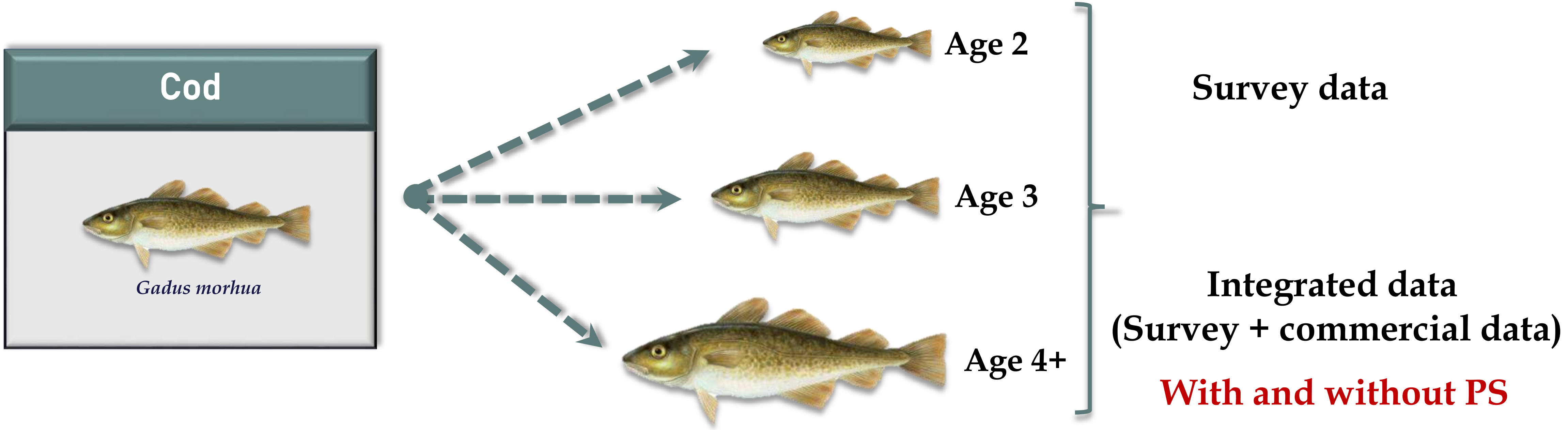
Commercial data

- On-board observers
- 432 hauls

Fisheries data integration

Case study – WB cod fishery (trawlers)

Background



Fisheries data integration

Case study – WB cod fishery (trawlers)

Background

Catch process

$$log(\mu_i^{\mathrm{SUR}}) = \log(\lambda(s_i, t_i)) + \sum_{\mathrm{k=1}}^{K_{\mathrm{SUR}}} \beta_k^{\mathrm{SUR}} X_{k,i}^{\mathrm{SUR}} + \gamma_i$$

- Time-period (Year-Quarter) – Fixed eff.
- Ship (2 levels) – Fixed. eff.
- Haul duration (h)

$$log(\mu_i^{\mathrm{COM}}) = \log(\lambda(s_i, t_i)) + \sum_{\mathrm{k=1}}^{K_{\mathrm{COM}}} \beta_k^{\mathrm{COM}} X_{k,i}^{\mathrm{COM}}$$

- Time-period (Year-Quarter) – Fixed eff.
- Métier (2 levels) – Fixed. eff.
- Vessel (80 levels) – Random eff.

Latent process

$$\lambda(\mathrm{s,t}) = \exp\left(\sum_{\mathrm{k=1}}^{\mathrm{K}} \beta_{\mathrm{k}} \mathrm{X}_{\mathrm{k}}(\mathrm{s,t}) + \xi(\mathrm{s,t})\right)$$

- Time-period (Year-Quarter)

Fisheries data integration

Case study – WB cod fishery (trawlers)

Results

Preferential sampling

Age group	Model	PS (α)	NLL	Npar	χ^2	Pr > χ^2
A2	M_A	-	-22570	110	-	-
	M_B	-0.08	-22566	111	7.87	0.005
A3	M_A	-	-21587	110	-	-
	M_B	-0.07	-21585	111	4.59	0.032
A4+	M_A	-	-20221	110	-	-
	M_B	1.5	-20080	111	281.41	0

M_A – model without preferential sampling (PS) correction term

M_B – model with preferential sampling (PS) correction term

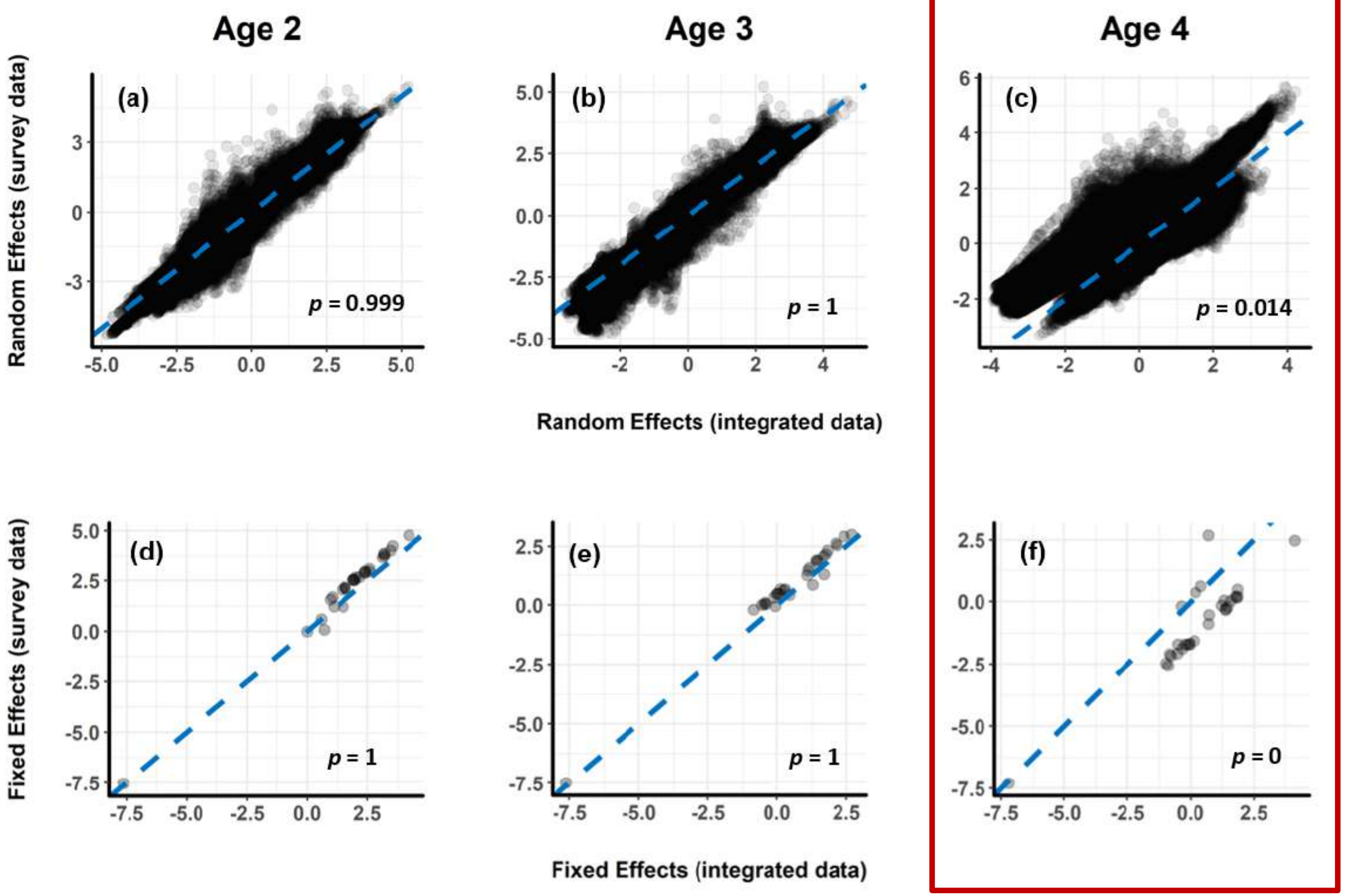
Fisheries data integration

Case study – WB cod fishery (trawlers)

Results

Validation

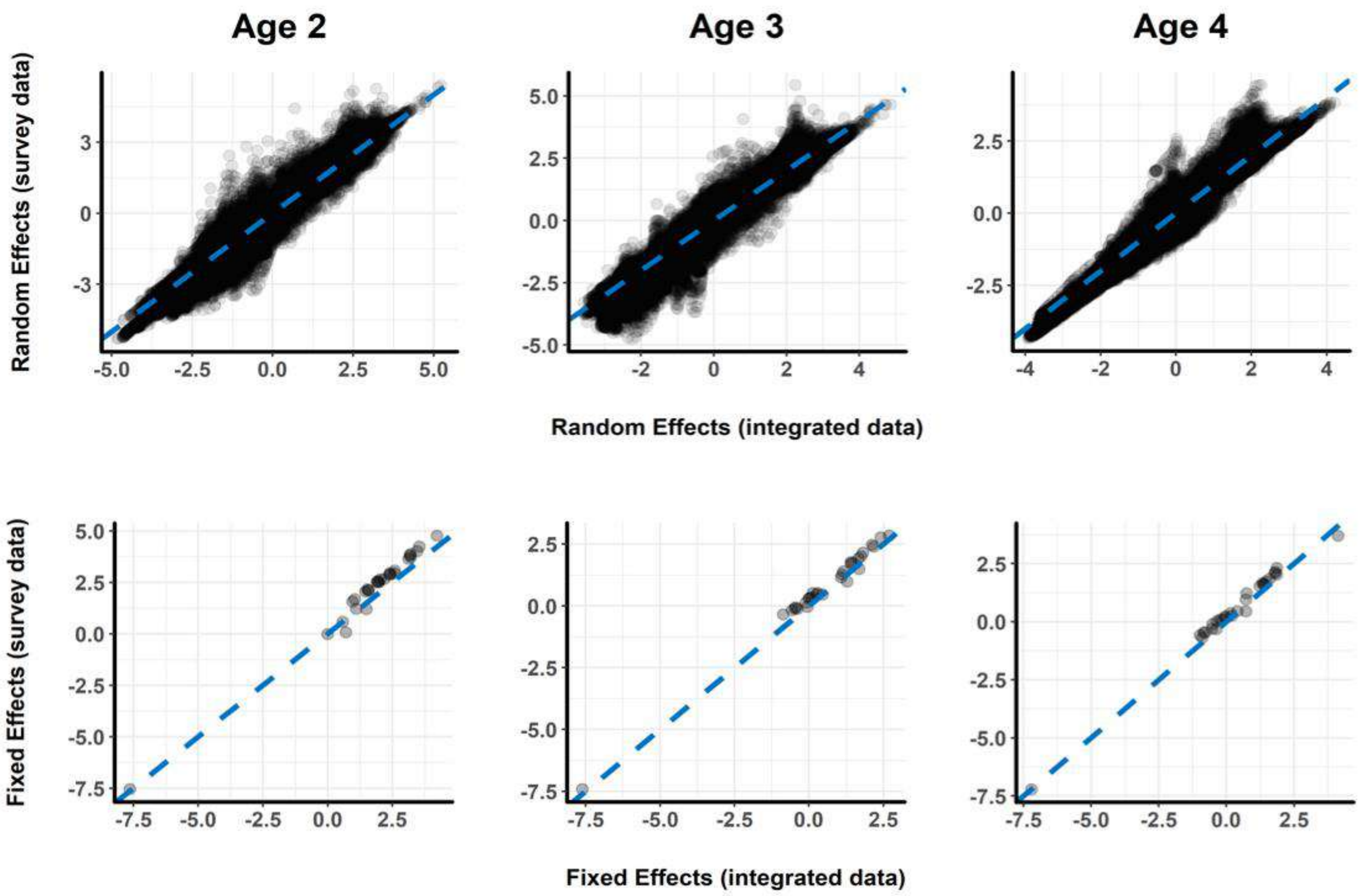
(w/ PS)



Fisheries data integration

Case study – WB cod fishery (trawlers)

Results



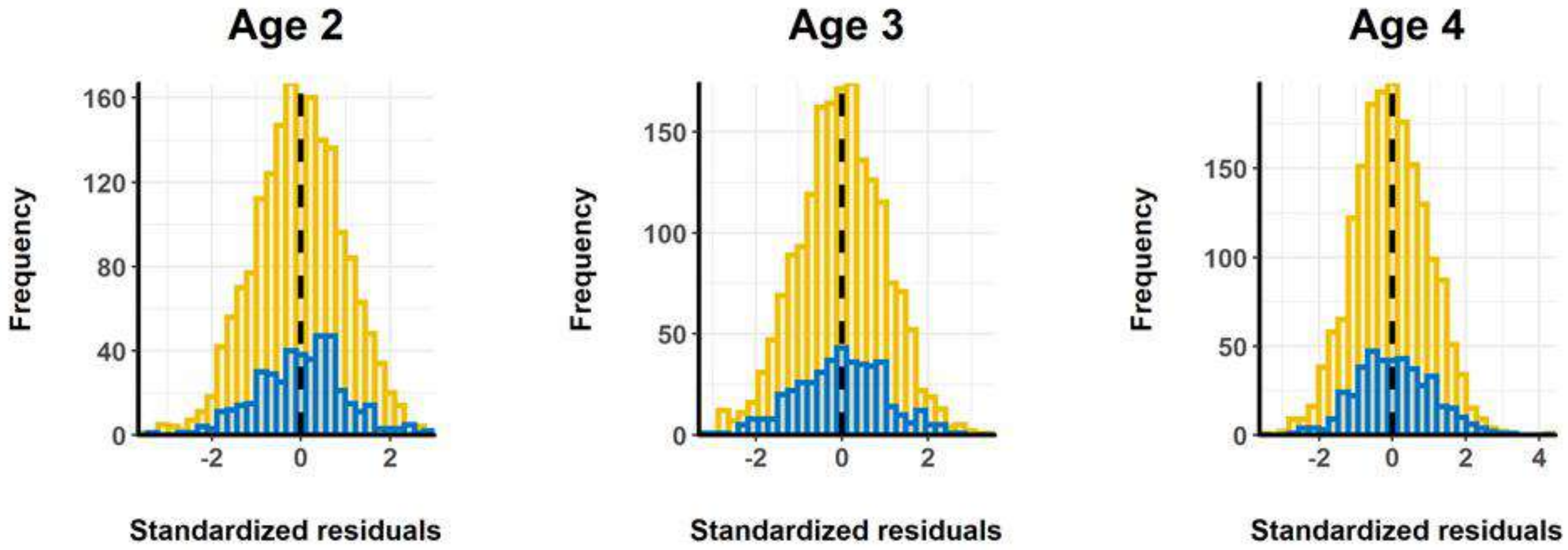
Validation

(no PS)

Fisheries data integration

Case study – WB cod fishery (trawlers)

Results



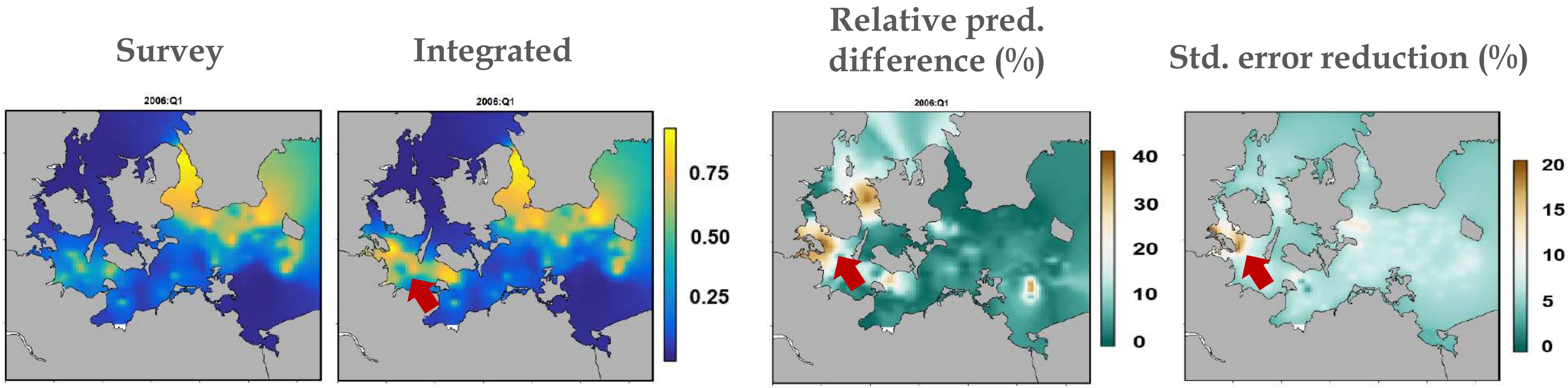
Fisheries data integration

Case study – WB cod fishery (trawlers)

Results

Age-3

Spatio-temporal abundance dynamics

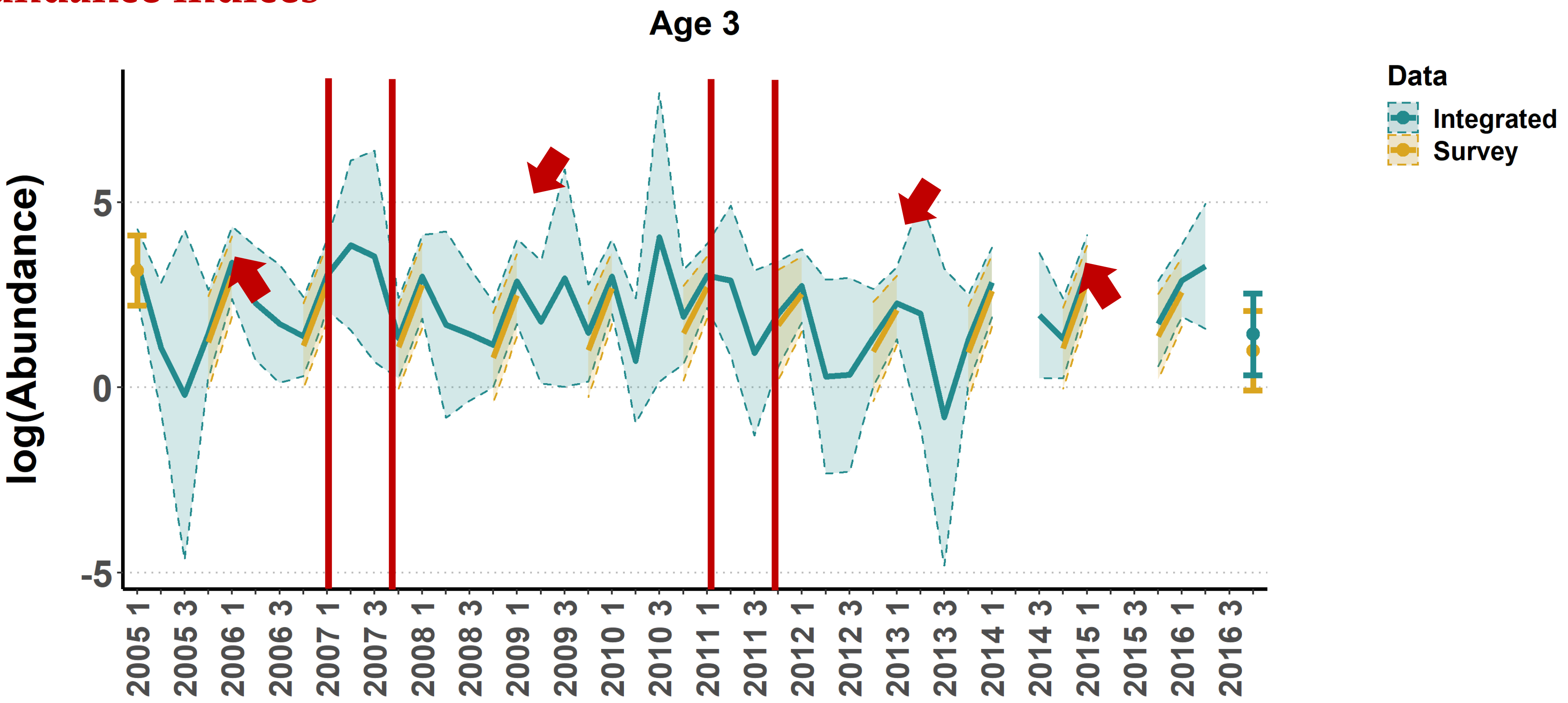


Fisheries data integration

Case study – WB cod fishery (trawlers)

Results

Seasonal abundance indices



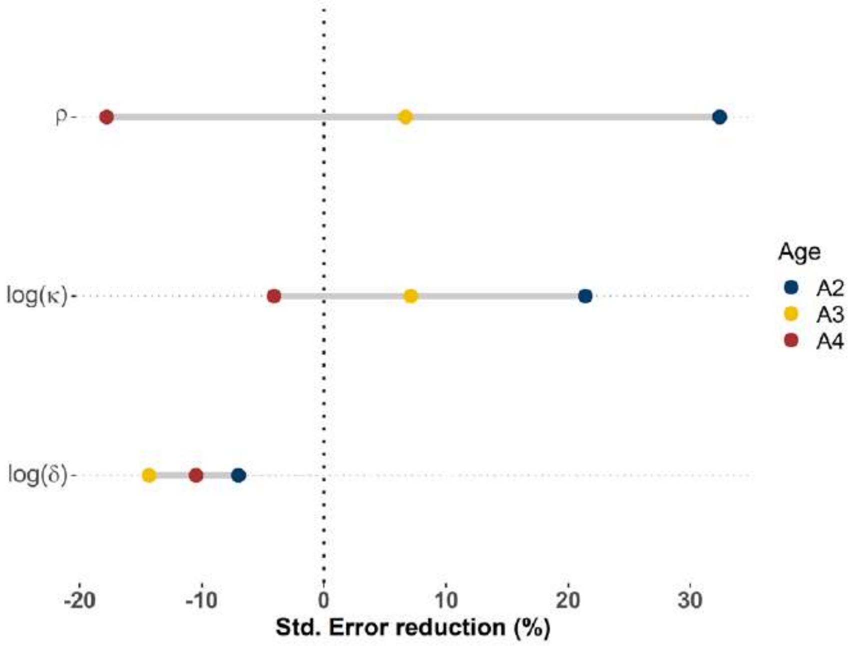
Fisheries data integration

Case study – WB cod fishery (trawlers)

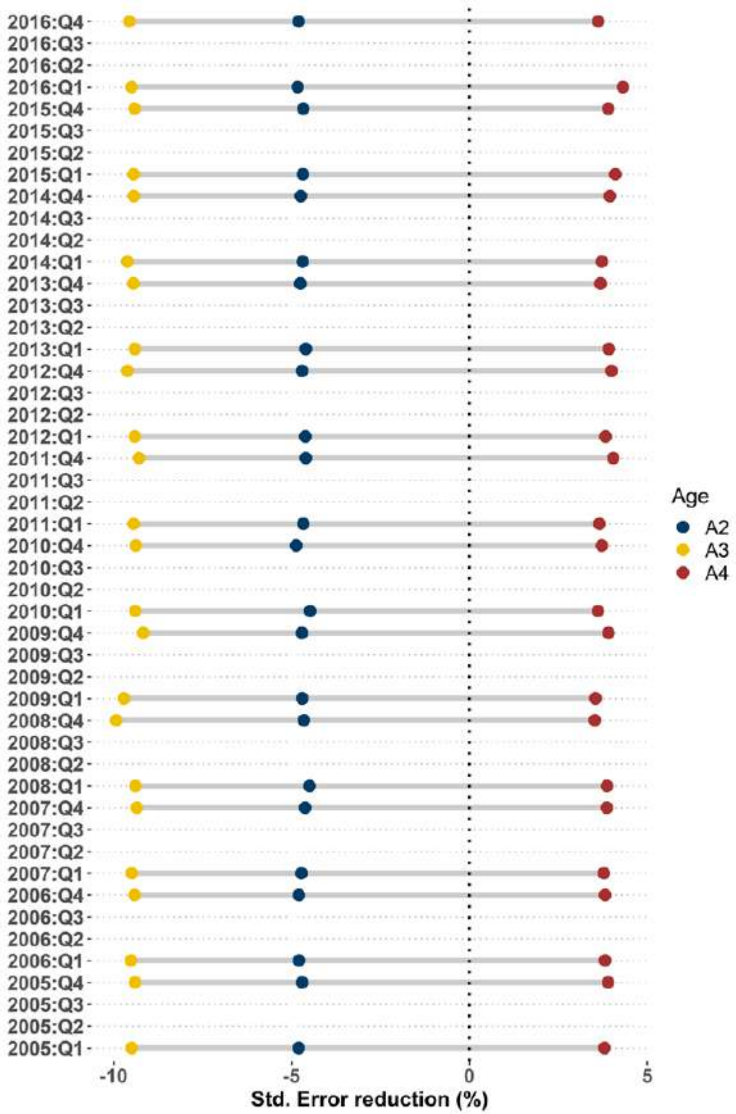
Results

Improvement in parameter estimates

Random effects (spatial & temporal corr.)



Fixed effects (Year-Quarter)



2 Introduction to Template Model Builder (TMB)

Basics of MLE

Template Model Builder (TMB)

How do we estimate things?

Template Model Builder (TMB)

How do we estimate things?

- **1)** Set-up a model that describes the system of interest (i.e., probability distribution)
  - Function to be used for predictions

Template Model Builder (TMB)

How do we estimate things?

- **1)** Set-up a model that describes the system of interest (i.e., probability distribution)
  - Function to be used for predictions

- **2)** Identify plausible values for the unknown parameters

Template Model Builder (TMB)

How do we estimate things?

- **1)** Set-up a model that describes the system of interest (i.e., probability distribution)
  - Function to be used for predictions
- **2)** Identify plausible values for the unknown parameters

  Different ways to estimate!
- **3)** Access the uncertainty around the estimated parameters
  - Explore the function around plausible parameter values (*parameter space*)

Template Model Builder (TMB)

The MLE approach

- Consists in finding the optimal parameter values (**θ**) by maximizing the likelihood (**L**) of the data (**D**)

$$L(D|\theta) \approx P(D|\theta)$$

The likelihood of the data given parameter(s) is the probability of the data given parameter(s)

- **Likelihood**: Given the data, we estimate the parameters (i.e., related to possible results)
- **Probability**: Given the parameters, we predict the data (i.e., related to hypotheses)

Template Model Builder (TMB)

The MLE approach

- When the data consists of n observations i, the likelihood is the product of the individual likelihoods

$$\mathrm{L}(\mathrm{D}|\theta) = \mathrm{L}(D_1|\theta)\,\mathrm{L}(D_2|\theta)\ldots\mathrm{L}(D_n|\theta)$$

Template Model Builder (TMB)

The MLE approach

- Problem with multiplication when *n*>>
  - log-transformation

$$L(D|\theta) = L(D_1|\theta)\, L(D_2|\theta) \ldots L(D_n|\theta)$$

$$\underbrace{\log\big(L(D|\theta)\big)}_{\text{Joint log Likelihood (JLL)}} = \log\big(L(D_1|\theta)\big) + \log\big(L(D_2|\theta)\big) + \cdots + L(D_n|\theta)$$

Joint log Likelihood (JLL)

Template Model Builder (TMB)

The MLE approach

- Choose parameter values that maximize the likelihood of the data

$$\hat{\theta} = argmax_{\theta}(L(y|\theta))$$

MLE of parameter(s)

Maximum value for L(y | θ) that can be achieved for any value of θ

- **argmax$_{\theta}$** is conducted with optimization algorithms
  - Computers like to optimize things by finding the *minimum*, rather the *maximum*

$$\hat{\theta} = argmin_{\theta}(L(y|\theta))$$

Template Model Builder (TMB)

The MLE approach

- The curvature of the JNLL provides an estimate of the maximum likelihood estimator

$$\widehat{\text{var}(\hat{\theta})} = \left(\left. \frac{\partial^2 L(y|\theta)}{\partial \theta^2} \right| \theta = \hat{\theta} \right)^{-1}$$

Hessian matrix

Template Model Builder (TMB)

The MLE approach

Profile likelihood

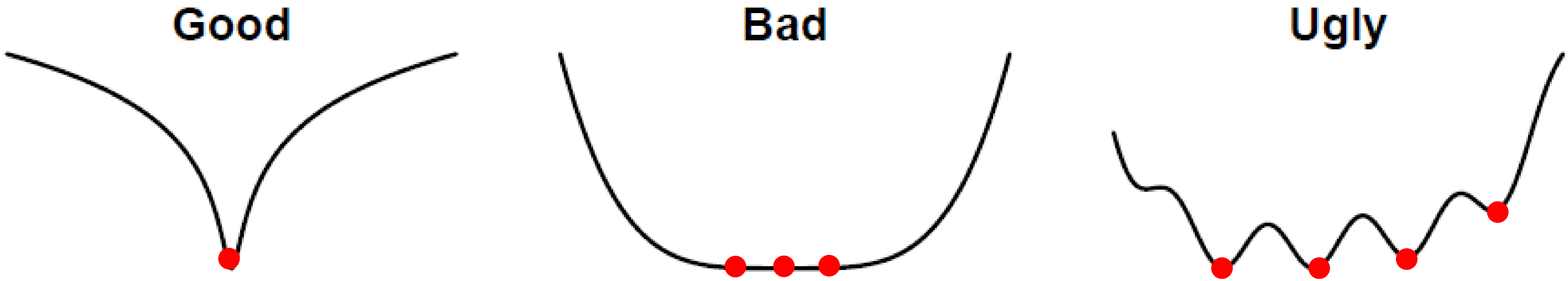


Figure extracted from Anders Nielsen (DTU Aqua)

Template Model Builder (TMB)

The MLE approach

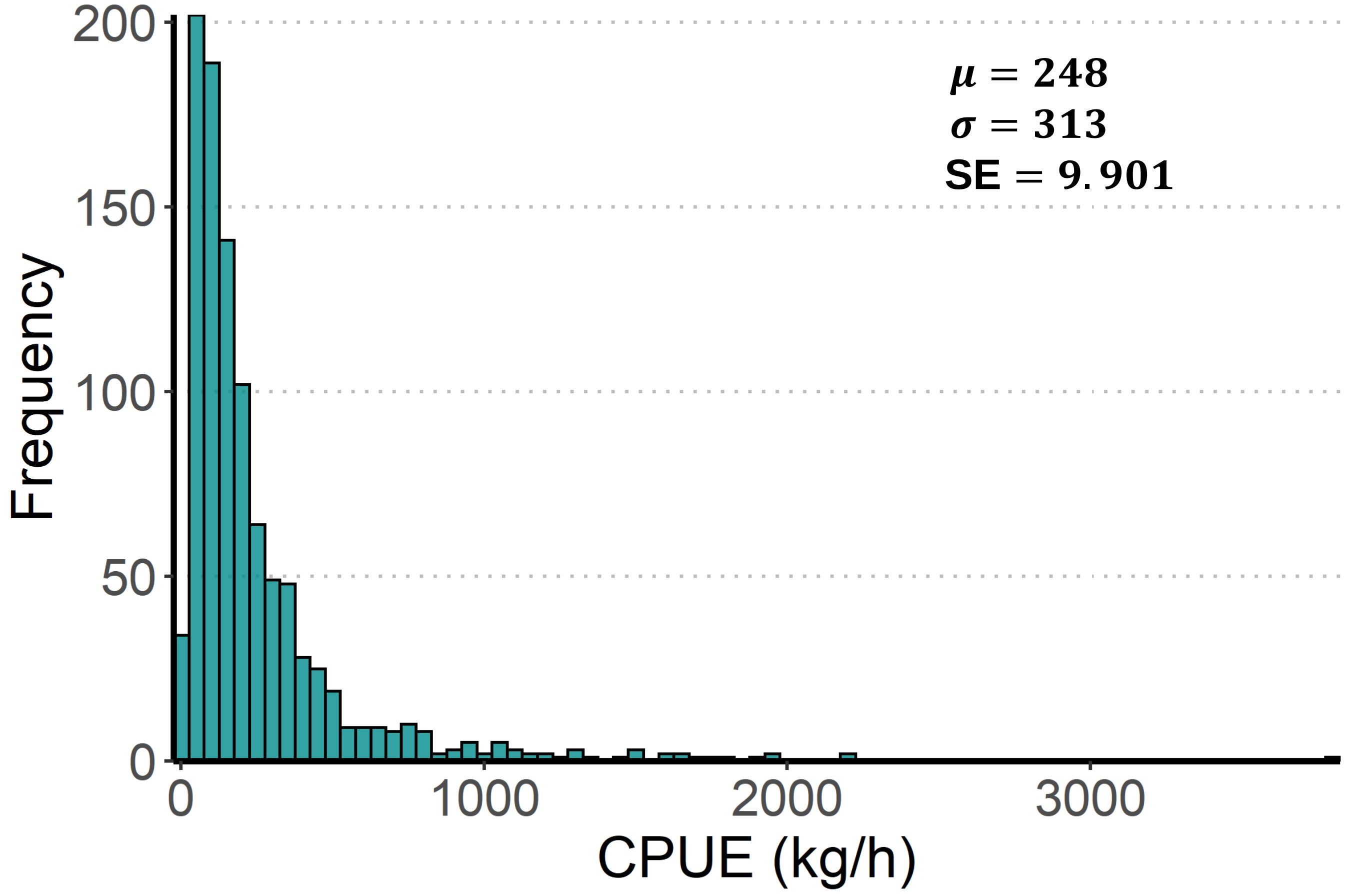
Profile likelihood & confidence interval

TMB_intro.R

Toy exercise

Template Model Builder (TMB)

Simulate data (e.g., CPUE)



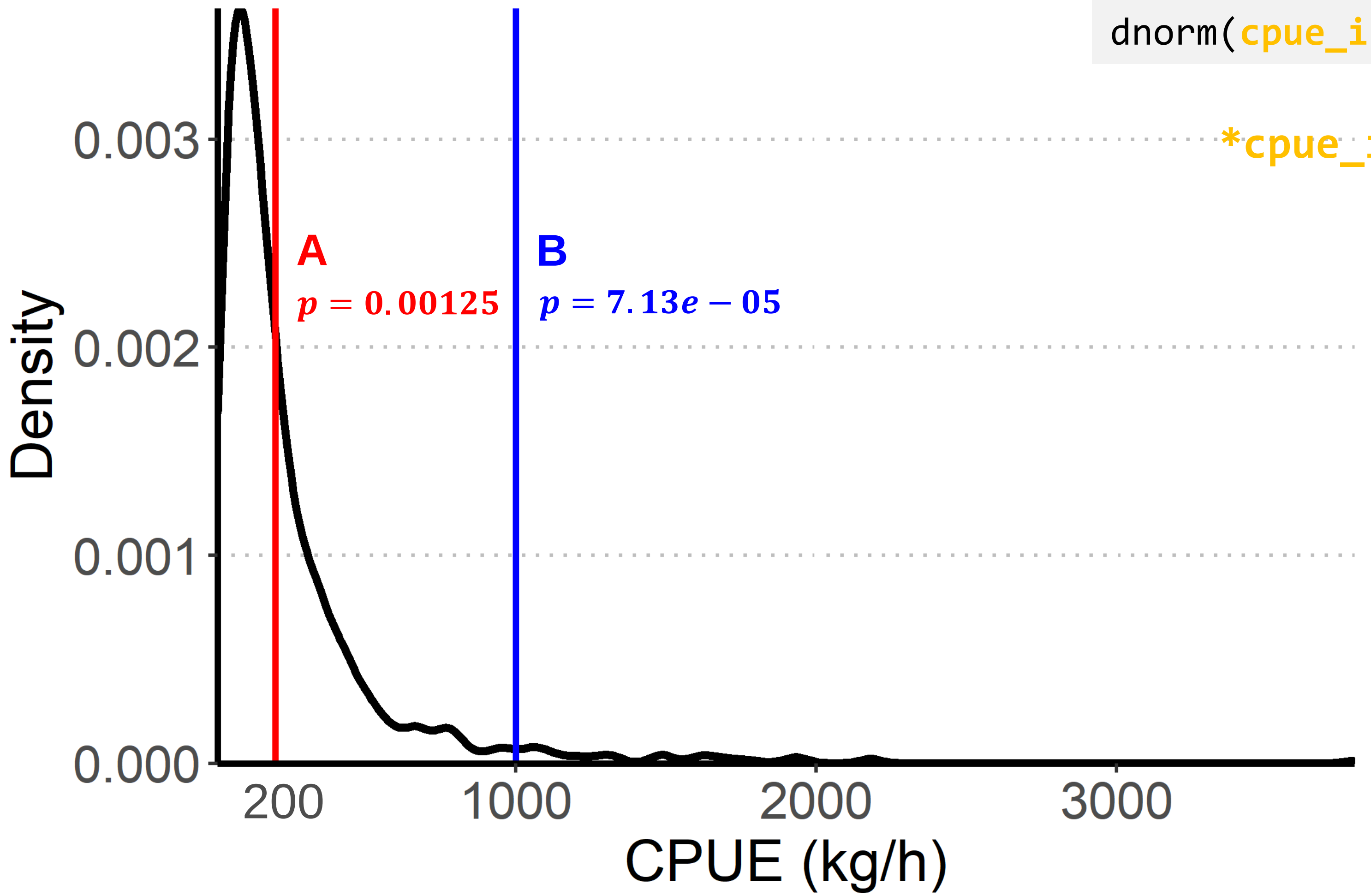

```
## Simulate a log-normal distribution

set.seed(123)
nsim <- 1000 #No of observations
mu_log <- 5
sd_log <- 1

df <- data.frame(cpue=rlnorm( n=nsim,
mean=mu_log, sd=sd_log))
```

Template Model Builder (TMB)

Likelihood of individual observations

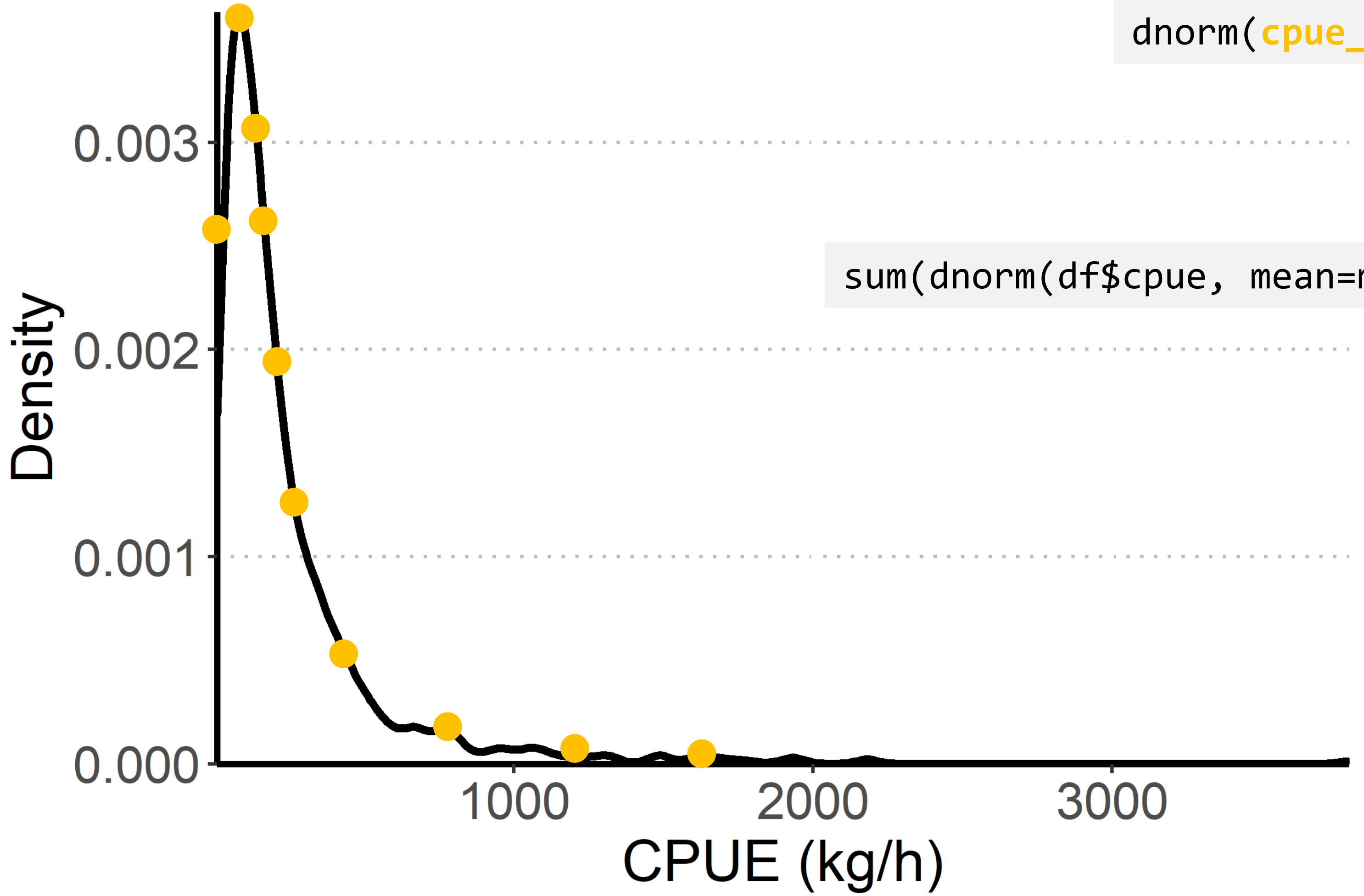



```
dnorm(cpue_i*, mean=mean(df$cpue), sd=sd(df$cpue))
```

*cpue_i = 200 or 1000

Template Model Builder (TMB)

Likelihood of the whole dataset



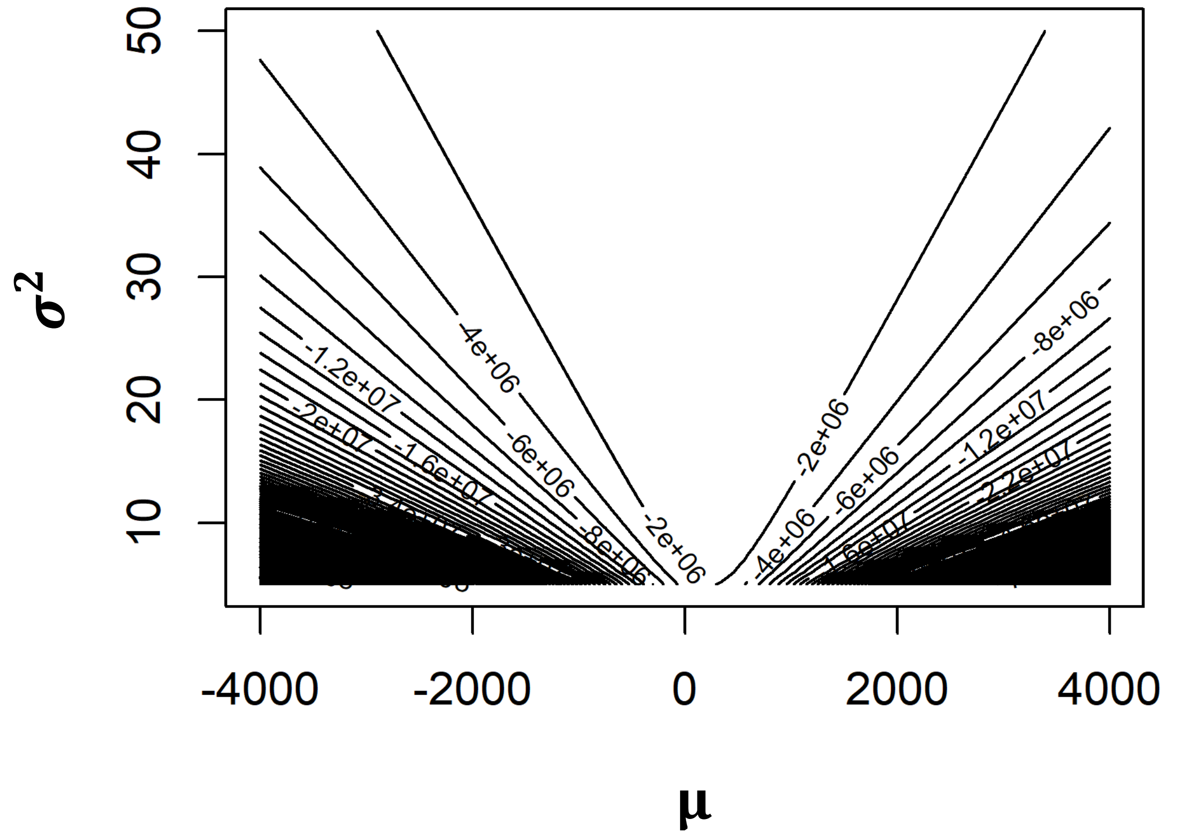

```
dnorm(cpue_i*, mean=mean(df$cpue), sd=sd(df$cpue))
```

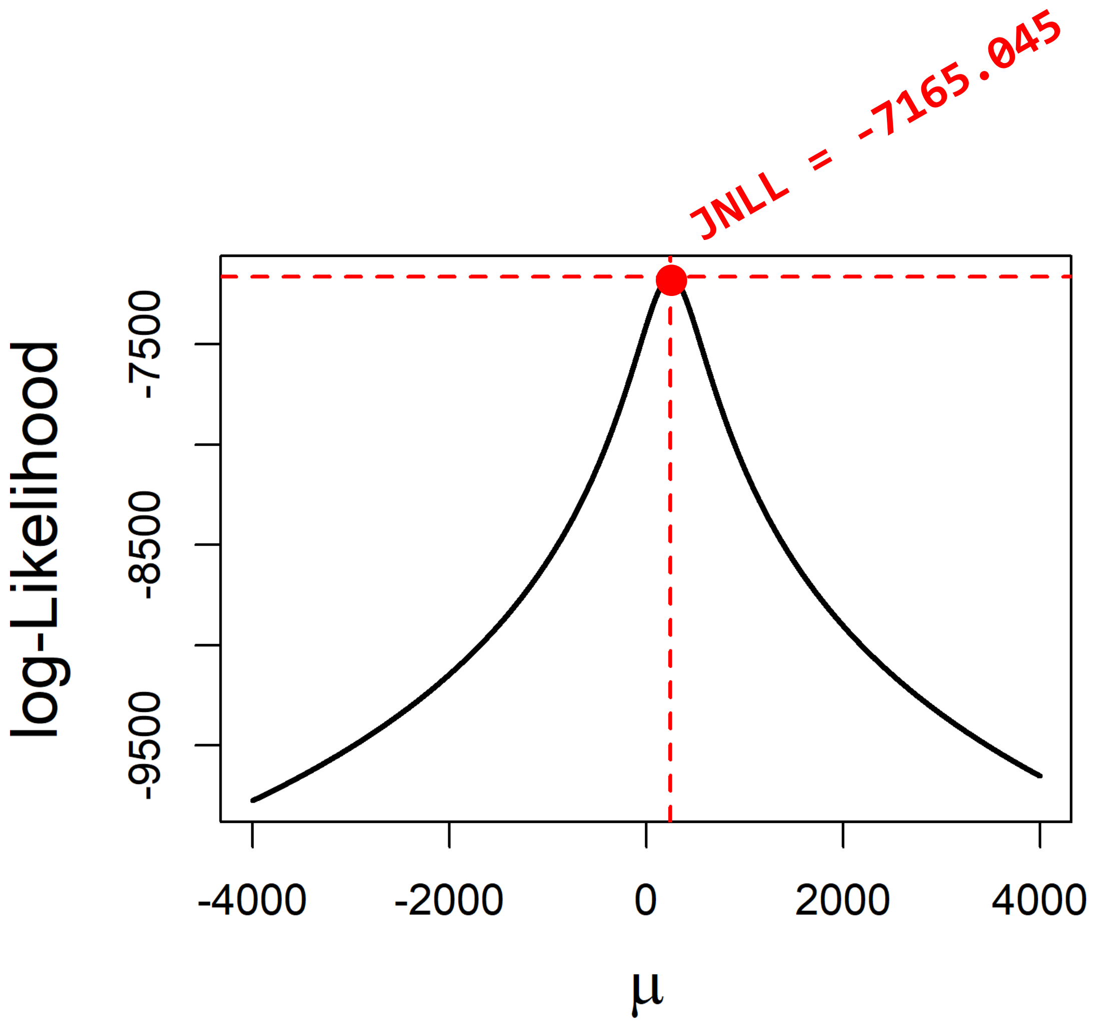
```
sum(dnorm(df$cpue, mean=mean(df$cpue), sd=sd(df$cpue), log=T))
```

JNLL = -7165.045

Template Model Builder (TMB)

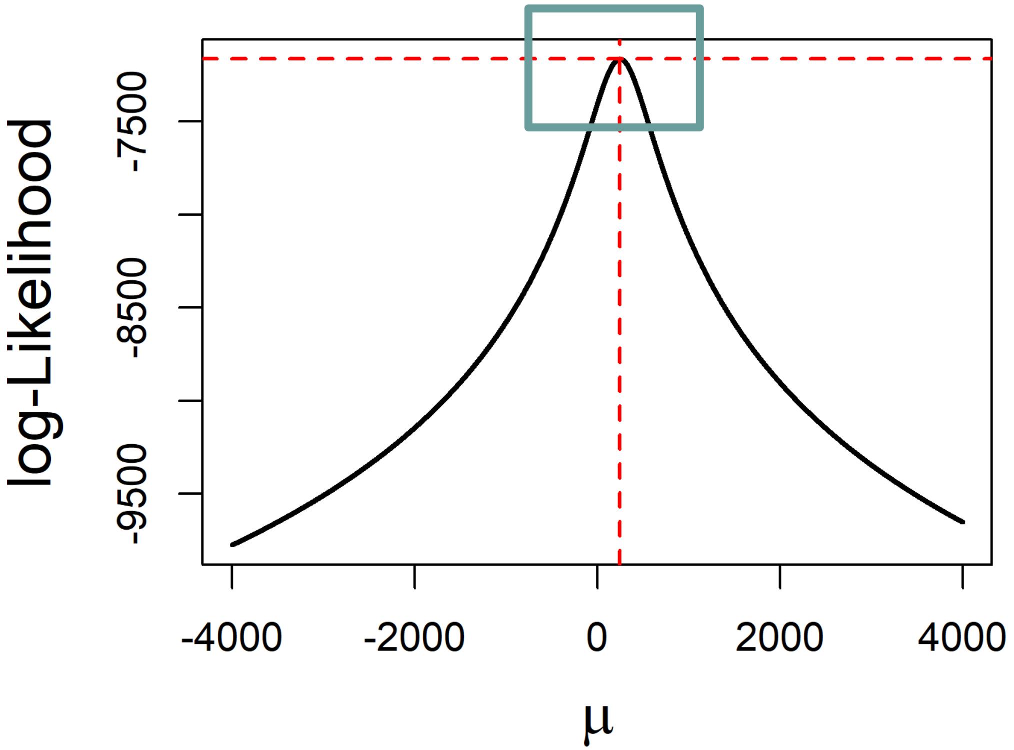
Profile likelihood

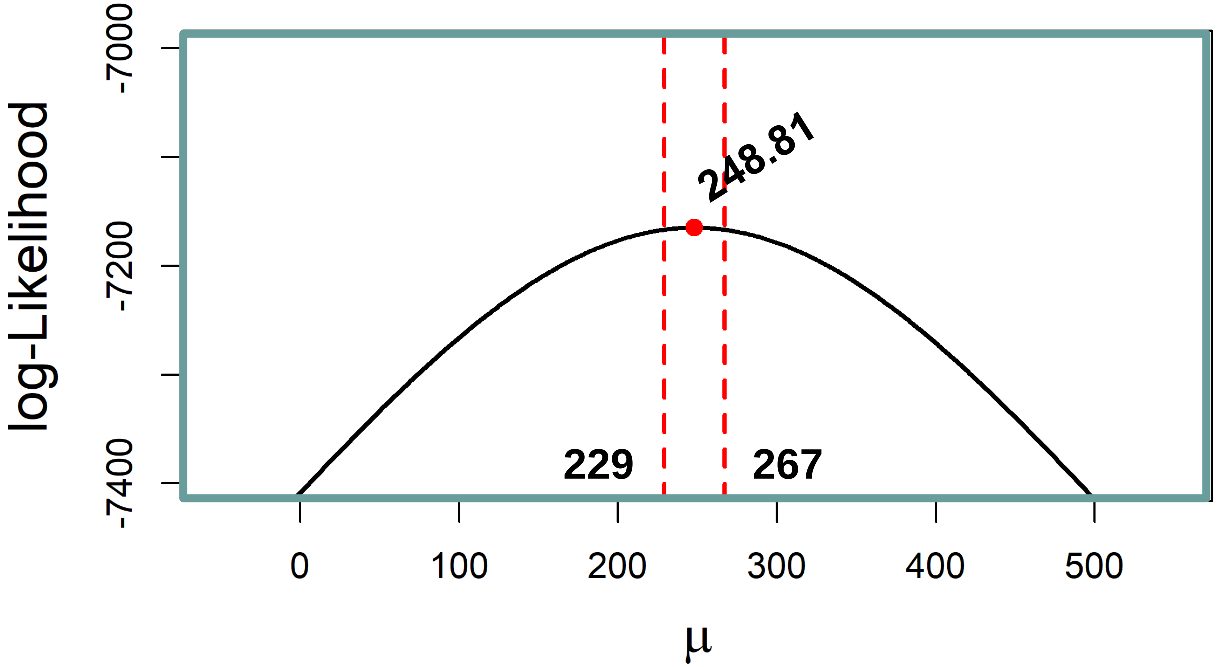




Template Model Builder (TMB)

Profile likelihood





Template Model Builder (TMB)

Estimate average CPUE

– Method 1: Linear model with standard R function

```
# Standard R functions
#~~~~~~~~~~~~~~~~~~~~~~~

LM = lm( df$cpue ~ 1 )
summary(LM)
```

Estimation method: Ordinary Least Squares

```
Call:
lm(formula = df$cpue ~ 1)

Residuals:
   Min      1Q Median      3Q     Max
-239.2 -168.9  -98.3    40.4 3545.4

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  248.100      9.902   25.06   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 313.1 on 999 degrees of freedom
```

$\mu = 248$
$\sigma = 313$
$\text{SE} = 9.901$

Template Model Builder (TMB)

Estimate average CPUE

- Method 2: R optimization function

- **Steps:**
  - **1)** Define the joint-negative log-likelihood (JNLL) function
  - **2)** Optimize the JNLL function
  - **3)** Analyze results

Template Model Builder (TMB)

Estimate average CPUE

– Method 2: R optimization function

1) Define the joint-negative log-likelihood (JNLL) function

```
#### Define JNLL function

JNLL <- function(data, parameters){

  mu <- parameters[1]
  sd <- parameters[2]

  jnll <- 0

  vector_of_likelihoods <- dnorm(data, mu, exp(sd), log=TRUE)
  jnll <- -1*sum(vector_of_likelihoods)

  return(jnll)
}
```

Template Model Builder (TMB)

Estimate average CPUE

– Method 2: R optimization function

2) Optimize the JNLL function

```
Opt_r <- optim(par = list("mu"=0,"log_sd"=0), #from {stats} package
               fn = JNLL,
               data = df$cpue,
               method = "BFGS",
               hessian=TRUE)
```

Template Model Builder (TMB)

Estimate average CPUE

– Method 2: R optimization function

3) Analyze results

```
> names(Opt_r) #What is contained in the Opt object?
[1] "par"            "value"          "counts"         "convergence" "message"      "hessian"
```

```
> Opt_r$par
        mu      log_sd
248.816195   5.746172
```

```
> Opt_r$value
[1] 7165.047
```

Template Model Builder (TMB)

Estimate average CPUE

– Method 2: R optimization function

3) Analyze results

```
> Opt_r$hessian
                   mu         log_sd
mu         0.01020817    -0.01461558
log_sd -0.01461558 1999.74940665
```

```
> print(se_optim_r <- sqrt(diag(solve(Opt_r$hessian))))
       mu    log_sd
9.8975649 0.0223622
```

Template Model Builder (TMB)

Estimate average CPUE

– Method 2: R optimization function

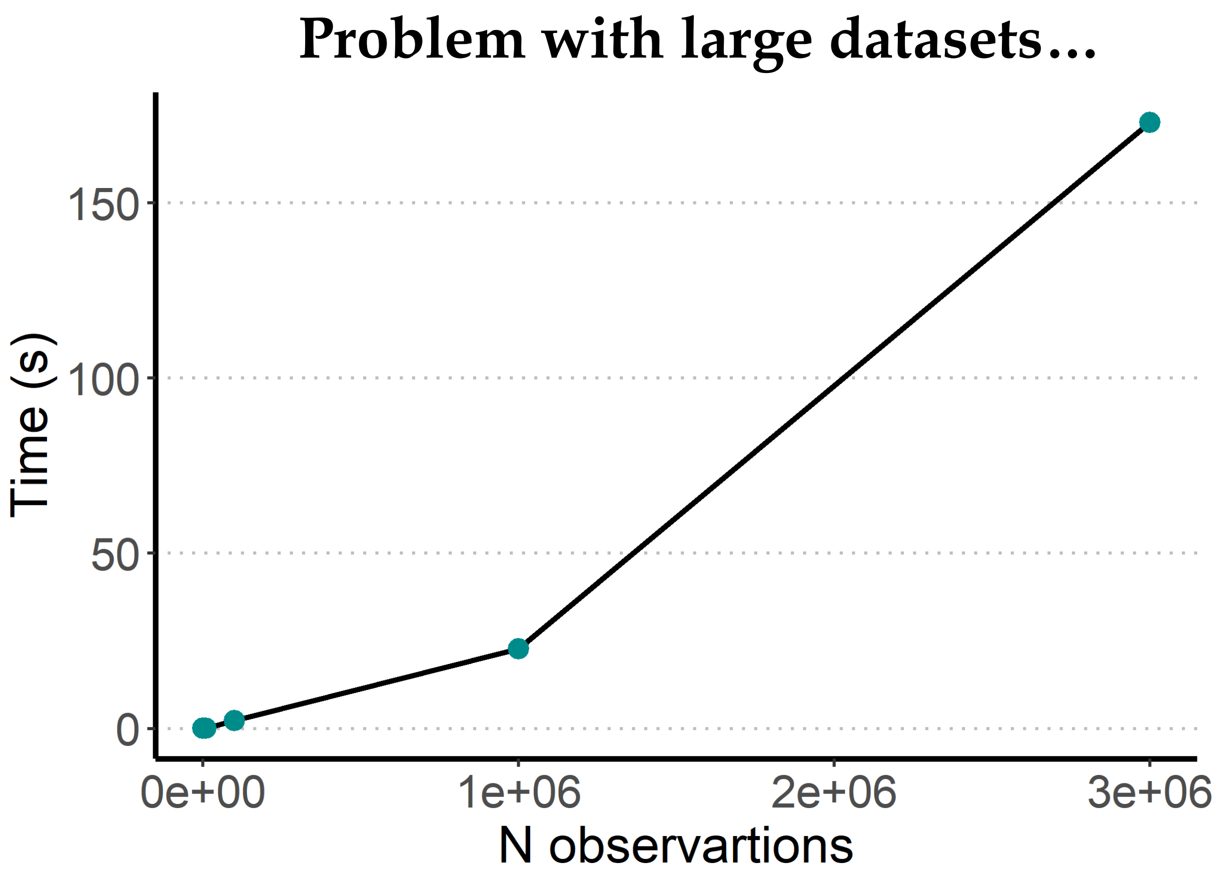
3) Analyze results – let's compare with the lm() estimates and raw data

```
> res_full
                   mu    log_sd
raw    248.1003 5.746606
lm     248.1003 5.746606
Roptim 248.8162 5.746172
```

Template Model Builder (TMB)

Estimate average CPUE

- Method 2: R optimization function



Shifting gears to TMB

Template Model Builder (TMB)

After all…what is TMB?

- R-package to fit statistical Latent Variable Models (LVMs)

Kasper Kristensen (DTU)

Anders Nielsen (DTU)

Casper Berg (DTU)

Hans Skaug (UiB)

Brad Bell (UW)

Kristensen et al., 2016; doi: 10.18637/jss.v070.i05

Template Model Builder (TMB)

After all…what is TMB?

- Programming language based in R that calculates the likelihood of complex statistical models with the computational speed of **C++**.

- Highly inspired on the Automatic Differentiation Model Builder (**ADMB**) **R**-package

- Uses **CppAD** (and other libraries) to evaluate 1^{st} and 2^{nd} (and possibly 3^{rd}) order derivatives of an objective function written in **C++**.

- Uses Laplace approximation to integrate out random effects

Template Model Builder (TMB)

Why use TMB?

- Rapid computational time
  - Based on C++ language
    - Minimization algorithm uses analytical derivatives

Template Model Builder (TMB)

Why use TMB?

- Rapid computational time
    - Based on C++ language
        - Minimization algorithm uses analytical derivatives

- No need to supply the derivatives to be minimized with regards to the parameters
    - Computed automatically using the reverse mode autodifferentiation

Template Model Builder (TMB)

Why use TMB?

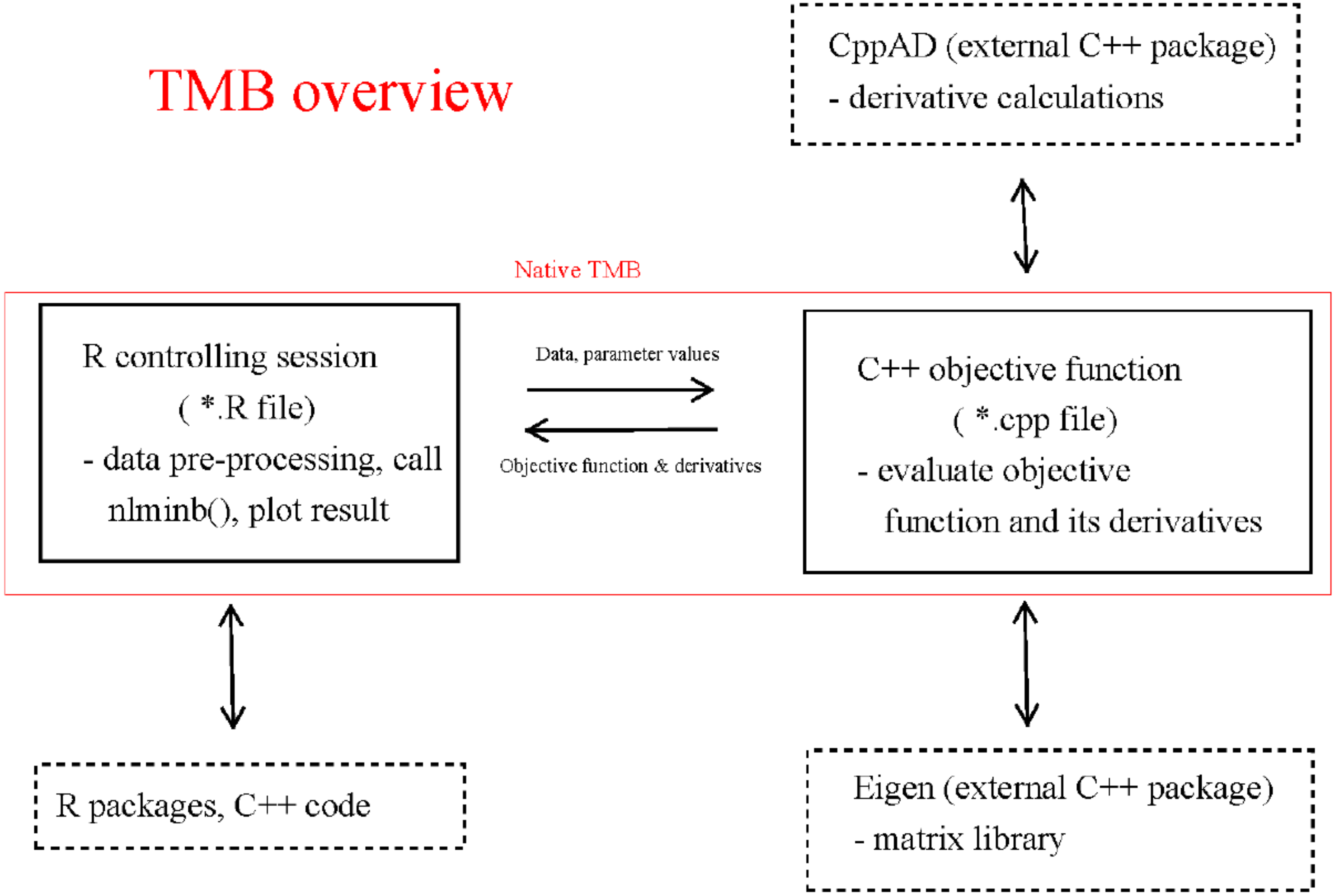
- Rapid computational time
  - Based on C++ language
    - Minimization algorithm uses analytical derivatives

- No need to supply the derivatives to be minimized with regards to the parameters
  - Computed automatically using the reverse mode autodifferentiation

- Flexibility to implement complex and non-standard models
  - Complex covariance structure
  - Non-linear relationship for parameters
  - Ease to reparametrize parameters
  - Multiple sources of observations, each described by different likelihoods….

Template Model Builder (TMB)

TMB workflow



Extracted from: http://kaskr.github.io/adcomp/_book/Structure-TMB.html

Template Model Builder (TMB)

TMB workflow

C++-side

- Set-up data, parameters and derived parameters
- Specify the model
- Define objective function that will be minimized

R-side

- Compile the C++ file
- Define data and parameter inputs
- Link the compiled C++ file to the optimizer
- Run the model
- Analyze model results

Toy exercise (cont.)

Template Model Builder (TMB)

Estimate average CPUE

– Method 3: TMB

- **C++-side:**

LM.cpp

```
#include <TMB.hpp>
template<class Type>
Type objective_function<Type>::operator() ()
{
  // Data
  DATA_VECTOR(y_i);

  // Parameters
  PARAMETER(mu);
  PARAMETER(log_sd);

  // Objective function & parameter transf.
  Type sd = exp(log_sd);
  Type jnll = 0;
  int n_data = y_i.size();

  // Probability of data conditional on fixed effect values
  for( int i=0; i<n_data; i++){
    jnll -= dnorm( y_i(i), mu,sd, true );
  }

  // Reporting
  return jnll;
}
```

Mandatory for every c++ file

Define input data & parameters

Calculate the JNLL

Template Model Builder (TMB)

TMB hints

- **C++-side:**

Data macros*	Parameter macros*
DATA_ARRAY()	PARAMETER()
DATA_FACTOR()	PARAMETER_VECTOR()
DATA_MATRIX()	PARAMETER_MATRIX()
DATA_INTEGER()	PARAMETER_ARRAY()
DATA_SCALAR()	
DATA_VECTOR()	
DATA_STRING()	

* For more options, refer to: http://kaskr.github.io/adcomp/_book/Tutorial.html

Template Model Builder (TMB)

TMB hints

- **C++-side:**

```
             R code             C++/TMB code

Comments      #                 //                         // Comment symbol
Constants     3.4               Type(3.4);                 // Explicit casting recommended in TMB
Scalar        x = 5.2           Type x = Type(5.2);        // Variables must have type
Arrays        x = numeric(10)   vector<Type> x(10);        // C++ code here does NOT initialize to 0
Indexing      x[1]+x[10]        x(0)+x(9);                 // C++ indexing is zero-based
Loops         for(i in 1:10)    for(int i=0;i<=10;i++)     // Integer i must be declared in C++
Increments    x[1] = x[1] + 3   x(0) += 3.0;               // += -= *= /= incremental operators in C++
```

* For more options, refer to: http://kaskr.github.io/adcomp/_book/Tutorial.html

Template Model Builder (TMB)

TMB workflow

C++-side

- Set-up data, parameters and derived parameters
- Specify the model
- Define objective function that will be minimized

R-side

- Compile the C++ file
- Define data and parameter inputs
- Link the compiled C++ file to the optimizer
- Run the model
- Analyze model results

Template Model Builder (TMB)

Estimate average CPUE

– Method 3: TMB

TMB_intro.R

- **R-side:**

```
#### Compile and load the C++ model
compile("LM.cpp")
dyn.load(dynlib("LM"))


#### Create Data and Parameter list
Data <- list("y_i" = df$cpue)
Parameters <- list("mu" = 0, "log_sd"=0) #set starting values for parameters


#### Construct objective function
Obj <- MakeADFun(data=Data, parameters=Parameters, DLL="LM")


#### Optimize objective function
Opt_tmb = nlminb(start=Obj$par, objective=Obj$fn, gradient=Obj$gr)
```

Template Model Builder (TMB)

Estimate average CPUE

– Method 3: TMB

TMB_intro.R

- **Results**

```
> Opt_tmb$objective #JNLL
[1] 7165.044
```

```
SD <- sdreport(Obj); SD #standard errors
outer mgc:  5.396742e-06
outer mgc:  1.024017e-05
outer mgc:  1.017842e-05
outer mgc:  1.998007
outer mgc:  2.001996
sdreport(.) result
         Estimate Std. Error
mu     248.100306 9.89696594
log_sd   5.746106 0.02236067
Maximum gradient component: 5.396742e-06
```

Template Model Builder (TMB)

Estimate average CPUE

- Method 3: TMB

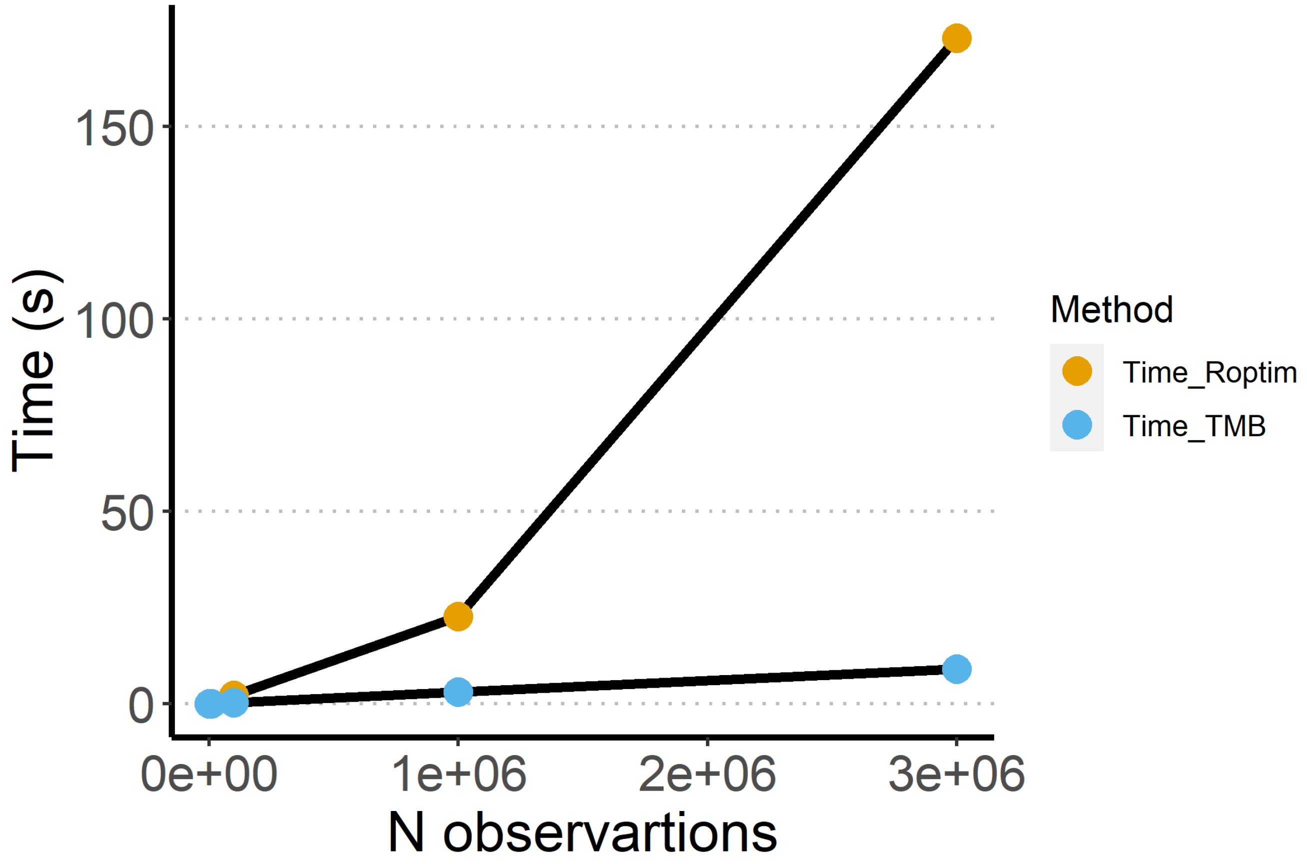
TMB_intro.R

- **Results**

```
> res_full2
             mu   log_sd
raw    248.1003 5.746606
lm     248.1003 5.746606
Roptim 248.8162 5.746172
TMB    248.1003 5.746106
```

Template Model Builder (TMB)

Computation time



Template Model Builder (TMB)

Additional stuff

- Additional examples in *TMB_intro.R* (lm with covariates, and LMM)
- Comprehensive TMB documentation: http://kaskr.github.io/adcomp/_book/Introduction.html

3

LGNB-SDM tutorial

LGNB tutorial

Structure

- *LGNB.cpp*
- *LGNB_Rmodel.R*
- *Validation_and_Residuals.R*
- *utilities.R*

github.com/mcruf/LGNB