Lecture 10: Virtual and Sequential Population Analysis

Noel Cadigan

$M = ?$

$M = .?$

$M = .2$

CFER

Centre for Fisheries Ecosystems Research

F6004 Lecture 10 Outline

Virtual/Sequential Population Analysis (Haddon Chapter 11)

1) Virtual Population Analysis
2) Sequential Population Analysis
3) Convergence
4) Estimation
5) ADAPT and other software
6) 3NO cod ADAPT

Sequential Population Analysis (SPA)

- <u>Was</u> a commonly used method to estimate the size of many fish stocks worldwide.
- I focus on the types of SPA's routinely used to assess Northwest Atlantic ground fish stocks.
- Recently the move is towards integrative models and state-space models.
- I describe these later
- However, there are some useful concepts to learn from SPA!

Sequential Population Analysis (SPA)

- SPA is a cohort model, based on a time series of annual fishery catches – the main focus.
- *Let* $C_{a,y}$ be the number of age a fish reported to be removed by the fishery in year y.
- For convenience, $y = 1,\ldots,Y$ and $a = 1,\ldots,A$

$$\begin{bmatrix} C_{11} & \cdots & C_{A1} \\ \vdots & \ddots & \vdots \\ C_{1Y} & \cdots & C_{AY} \end{bmatrix}$$

- **If a cohort is fished out then the catches can be added up back in time to estimate the minimum size of the cohort when first recruited to the fishery.**
- This is the **virtual population**.

Sequential Population Analysis (SPA)

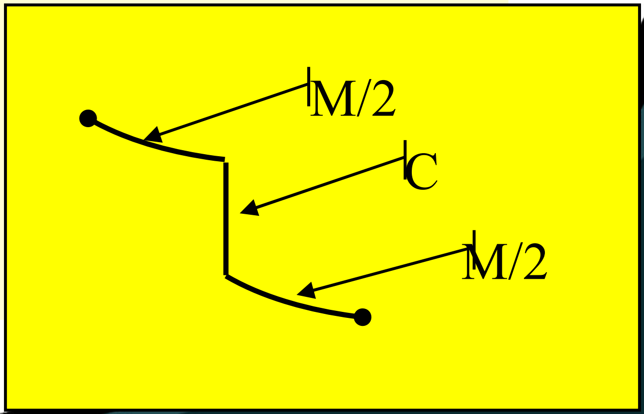
- When a time series of catches-at-age are available then this can be done for many consecutive cohorts, producing a time series of historical virtual stock sizes.
- This is called virtual population analysis (VPA).
- If other sources of mortality are negligible then the virtual stock size is almost the same as the recruited stock size,
- or the stock size at the first age exploited by the fishery.

Sequential Population Analysis (SPA)

- if information on other mortality is also available, then this can be added to fishing mortality to quantify absolute historic exploited stock size (N).
- Basic cohort model:
- $N_{a+1,y+1} = N_{a,y}\, e^{-Z_{a,y}}$, where $Z_{a,y} = F_{a,y} + M_{a,y}$ is the total mortality;
- F/Z is proportion of deaths due to fishing, and
- M/Z = proportion of deaths due to natural events.
- $C_{a,y} = N_{a,y}\,(1 - e^{-Z_{a,y}})F_{a,y}/Z_{a,y}$ ← the catch equation

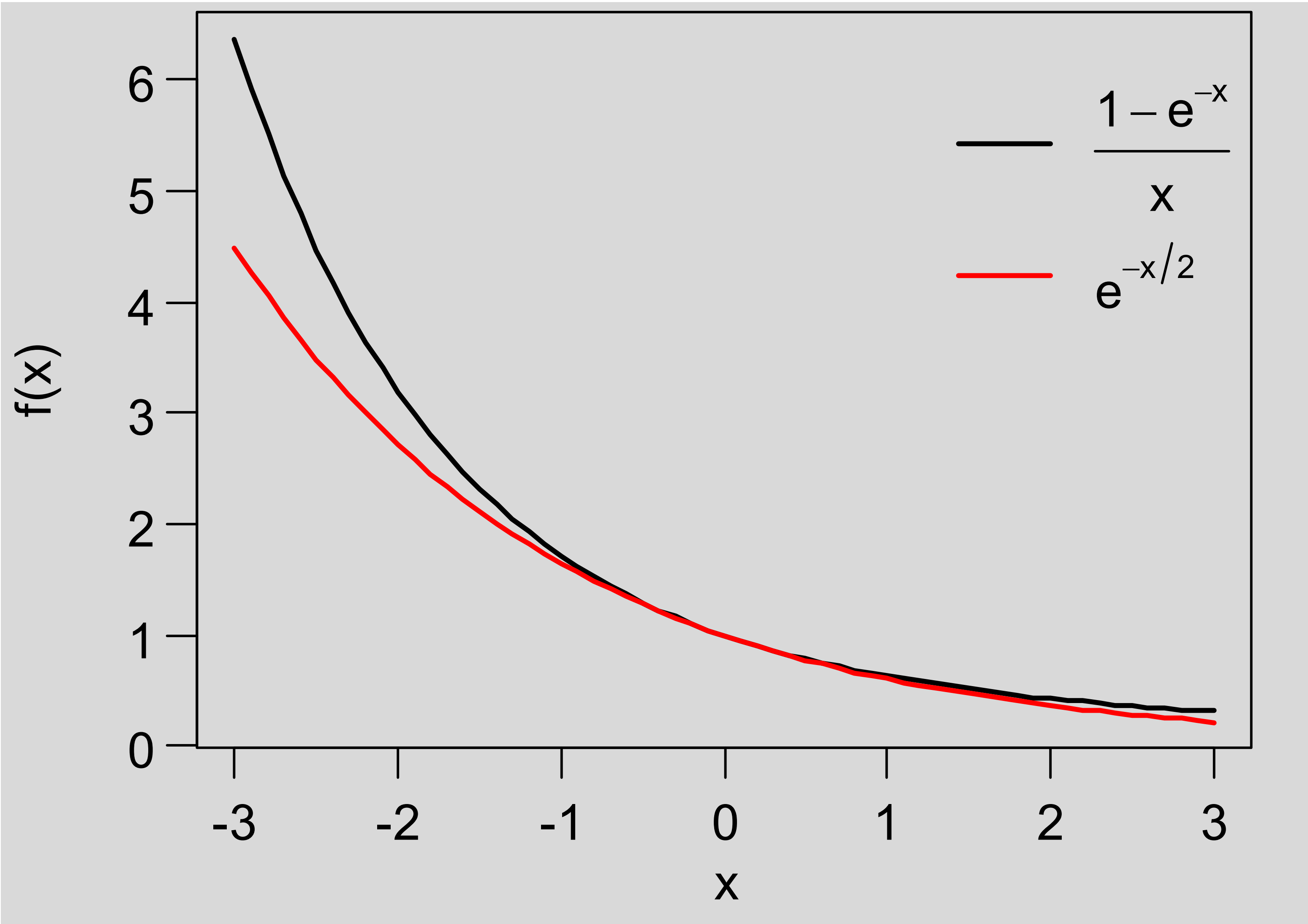
Sequential Population Analysis (SPA)

- Assumes the catches and M are known without error.
- Common value used: $M=0.2$; implies an approximately 82% annual survival rate.
- Pope's Approximation: $Z(1-e^{-F})/\{F(1-e^{-Z})\} \approx e^{M/2}$
- This leads to the following simple cohort model:
- forward: $N_{a+1,y+1} = (N_{a,y}\, e^{-M/2} - C_{a,y})\, e^{-M/2}$, or
- backward: $N_{a,y} = N_{a+1,y+1}\, e^{M} + C_{a,y}\, e^{M/2}$



Pope's Approximation

- A good approximation of $\frac{1-e^{-x}}{x} \approx e^{-x/2}$, when x is not large



Pope's Approximation

$$N_{a+1,y+1} = N_{a,y}e^{-Z_{a,y}} \Rightarrow N_{a+1,y+1}e^{M_{a,y}} = N_{a,y}e^{-F_{a,y}}$$

$$\Rightarrow N_{a+1,y+1}e^{M_{a,y}} = N_{a,y} - N_{a,y}\left(1 - e^{-F_{a,y}}\right)$$

From Baranov's catch equation,

$$C_{a,y} = N_{a,y}\left(1 - e^{-Z_{a,y}}\right)\frac{F_{a,y}}{Z_{a,y}} \Rightarrow N_{a,y} = \frac{C_{a,y}Z_{a,y}}{F_{a,y}\left(1 - e^{-Z_{a,y}}\right)}$$

therefore

$$N_{a+1,y+1}e^{M_{a,y}} = N_{a,y} - C_{a,y}\boxed{\frac{Z_{a,y}}{\left(1 - e^{-Z_{a,y}}\right)}\frac{\left(1 - e^{-F_{a,y}}\right)}{F_{a,y}}}$$

$$N_{a+1,y+1}e^{M_{a,y}} \approx N_{a,y} - C_{a,y}\frac{e^{-F_{a,y}/2}}{e^{-Z_{a,y}/2}} = N_{a,y} - C_{a,y}e^{M_{a,y}/2}$$

$$\boxed{\frac{1 - e^{-x}}{x} \approx e^{-x/2}}$$

Hence, Pope's approximation is

$$N_{a+1,y+1} = N_{a,y}e^{-M_{a,y}} - C_{a,y}e^{-M_{a,y}/2}$$

Numbers at Age Matrix-Backward Calculation

$$N_{a,y}=N_{a+1,y+1}\, e^{M} + C_{a,y}\, e^{M/2}$$

					Age						
Year	0	1	2	3	4	5	6	7	8	9	10
⋮											
1995											
1996											
1997											
1998											
1999											
2000											
2001											

Need to know the numbers in the blue blocks (survivors) and green blocks to reconstruct the population history.

The grey blocks indicate incomplete cohorts - haven't completely passed through fishery.

A+Y-1 unknowns

Numbers at Age Matrix - Backward Calculation

$$N_{a,y}=N_{a+1,y+1}\,e^{M}+C_{a,y}\,e^{M/2}$$

						Age					
Year	0	1	2	3	4	5	6	7	8	9	10
⋮											
1995											
1996											
1997											
1998											
1999											
2000											
2001											

usually estimate the survivors

usually approximate these

Need to know the numbers in the blue blocks (survivors) and green blocks to reconstruct the population history.

The grey blocks indicate incomplete cohorts - haven't completely passed through fishery.

F-constraints

- Unknowns in the cohort model: $N_{1,Y}, \ldots, N_{A,Y}$ (numbers in last year) and $N_{A,1}, \ldots, N_{A,Y-1}$ (numbers at oldest age) where *A* and *Y* are the oldest age and year in the model, respectively.

- The numbers at the oldest age are often approximated using constraints on their fishing mortalities.

- Eg. assume that the commercial fishery equally selects both $N_{A,y}$ and $N_{A-1,y}$ (i.e. $N_{A,y}/N_{A-1,y} = C_{A,y}/C_{A-1,y}$) then

- $N_{A,y} = N_{A-1,y} C_{A,y}/C_{A-1,y} = (N_{A,y+1} e^{M} + C_{A-1,y} e^{M/2}) C_{A,y}/C_{A-1,y}$

- i.e. $N_{A,y} = (N_{A,y+1} e^{M}/C_{A-1,y} + e^{M/2})\, C_{A,y}$

- this relationship can be used recursively to express the $N_{A,y}$'s in terms of survivors.

F-constraints

- This simple approximation is rarely used because it is sensitive to measurement errors in catches and these errors can be substantial.
- Methods to deal with measurement and other errors in the catches are for later.
- In practise the methods used to approximate $N_{A,y}$'s are more complicated but the effect is the same, and that is to express $N_{A,y}$'s as functions of survivors.
- Sometimes the $N_{A,y}$'s are estimated but constrained using a penalty function (e.g. shrinkage estimation).

SPA Convergence

- A consequence of using F constraints to compute N_{Ay}'s is SPA convergence
- i.e. historic population size hardly affected by the values of survivors (N_{aY}'s)

Converged block

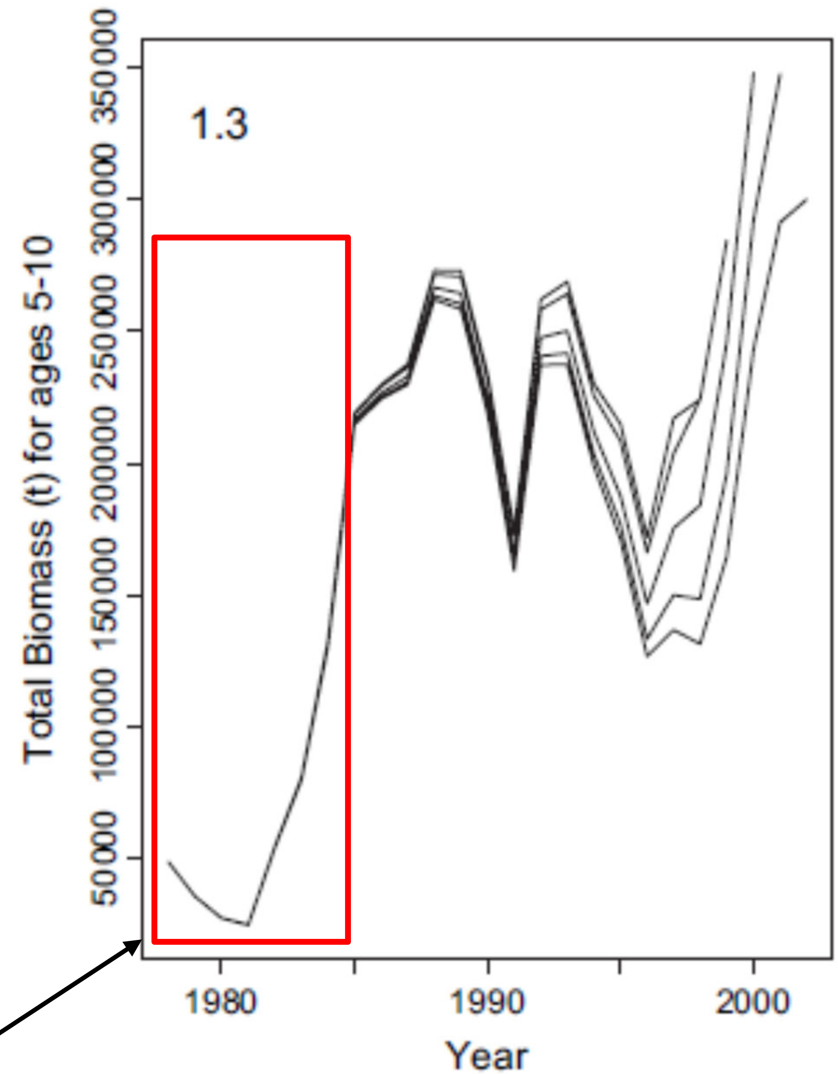
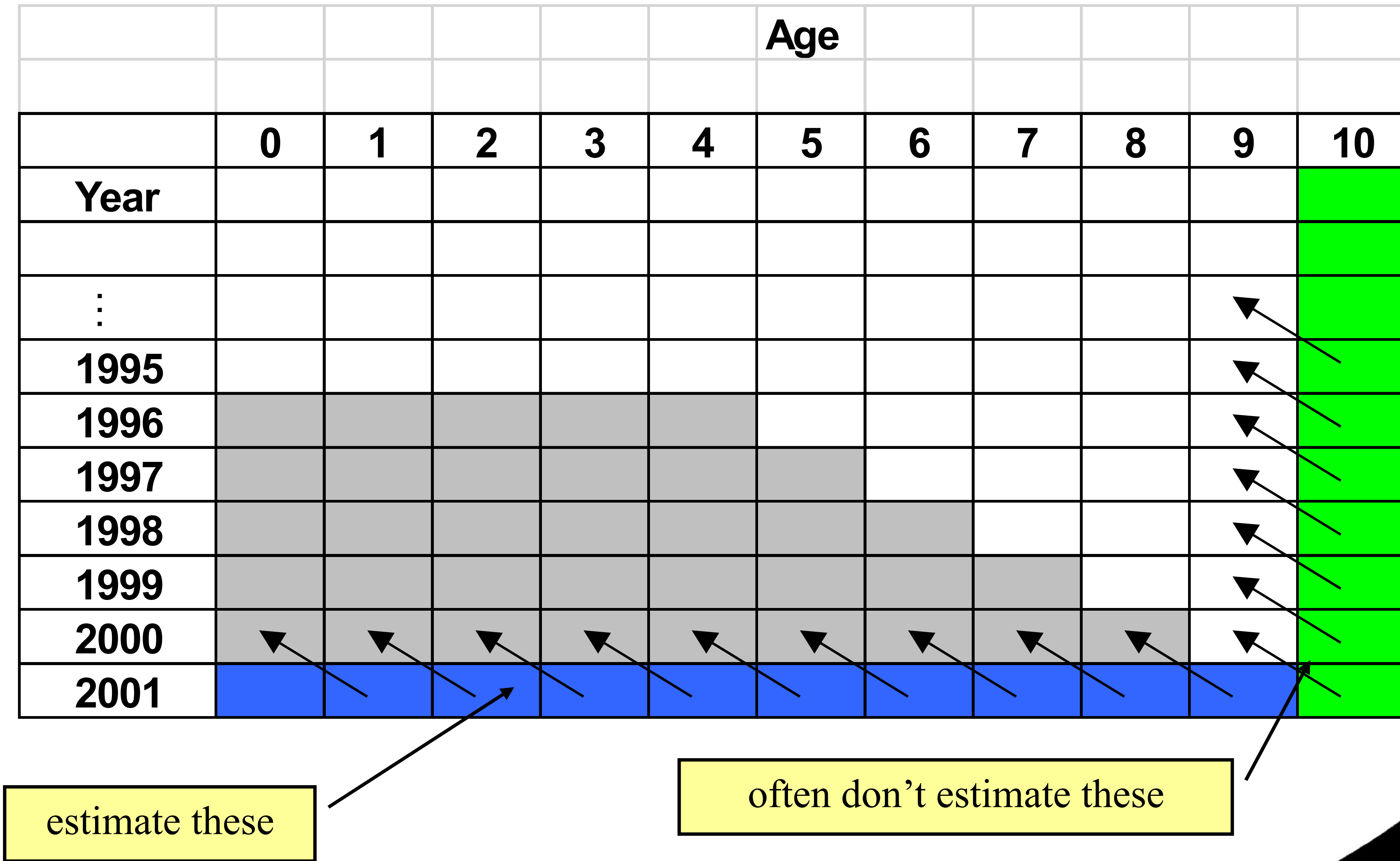


Figure 1. Retrospective estimates of total biomass (B_+) for the 4T fall herring stock. The retrospective ρ statistic is shown in the top left-hand corner.

Recall the model



Conceptual Parameter Estimation

- The convergence property is important to estimate the survivors (i.e. the blue parameters).
- survivor parameters are estimated essentially using the ratio method,
  - based on a long time-series of stock size indices, that cover some of the converged block.
- If $R_{a,y}$ is an index of stock size, then $R_{a,y} = q_a N_{a,y}$
- Conceptual estimator: $\hat{q}_a = \bar{R}_a^* / \bar{N}_a^*$; * indicates averages for the converged (i.e early) years
- survivors could then be estimated as $\hat{N}_{aY} = R_{aY} / \hat{q}_a$

Recall the model

						Age					
Year	0	1	2	3	4	5	6	7	8	9	10
			Converged years								
⋮											
1992											
1993											
1994											
1995											
1996											
1997											
1998											
1999											
2000											
2001											

Parameter Estimation

- The ratio method is not used. *It is just a useful analogy*.
- Estimation is based on regression methods in which the SPA cohort model is used for estimation over the entire time-series of the stock indices, and not just the converged portion.
- We may have multiple surveys – subscript s
- A common method is to estimate survivors as those that, jointly with q_a parameters, minimize
- $Fit(\theta) = \sum_{s,a,y}\{\log(R_{say}) - \log(q_{sa}) - \log(N_{say})\}^2$

q's and survivors

$$N_{say} = \exp(-t_{fsy} Z_{say})\, N_{ay};$$

$$N_{ay} = N_{ay}(\theta_1)$$

Fraction of year that survey *s* occurred in year *y*

Parameter Estimation

- When there are several surveys then most of the parameters will be for q_{sa}'s
- Possible to get a closed form estimates of q_{sa} given survivors (θ_1)
- $\hat{q}_{sa}(\theta_1) = n_s^{-1} \sum_y \log\left(R_{say}/N_{say}(\theta_1)\right)$
  - The number of years for survey s is n_s^{-1}
- Concentrated fit function for θ_1 :

$$Fit(\theta_1) = \sum_{say} \left\{\log\left(R_{say}\right) - \log\left[\hat{q}_{sa}(\theta_1) N_{say}(\theta_1)\right]\right\}^2$$

- More generally, we can form concentrated or profile likelihood functions this way

Parameter Estimation

- When there are several surveys then most of the parameters will
- Pos … vivors (θ_1)
- $\hat{q}_{sa}$(
  - T
- Con
- Mor … lihood functions this way

Only makes sense if estimation formula for *q*'s is closed form

A change in the model means changing the estimate of *q*

Can't generally do this in state-space or integrated models

Concentrated fit function not usually worth the effort these days

We won't pursue this

Example

NOT TO BE CITED WITHOUT PRIOR
REFERENCE TO THE AUTHOR(S)

Northwest Atlantic Fisheries Organization

NAFO SCR Doc. No. 17-042

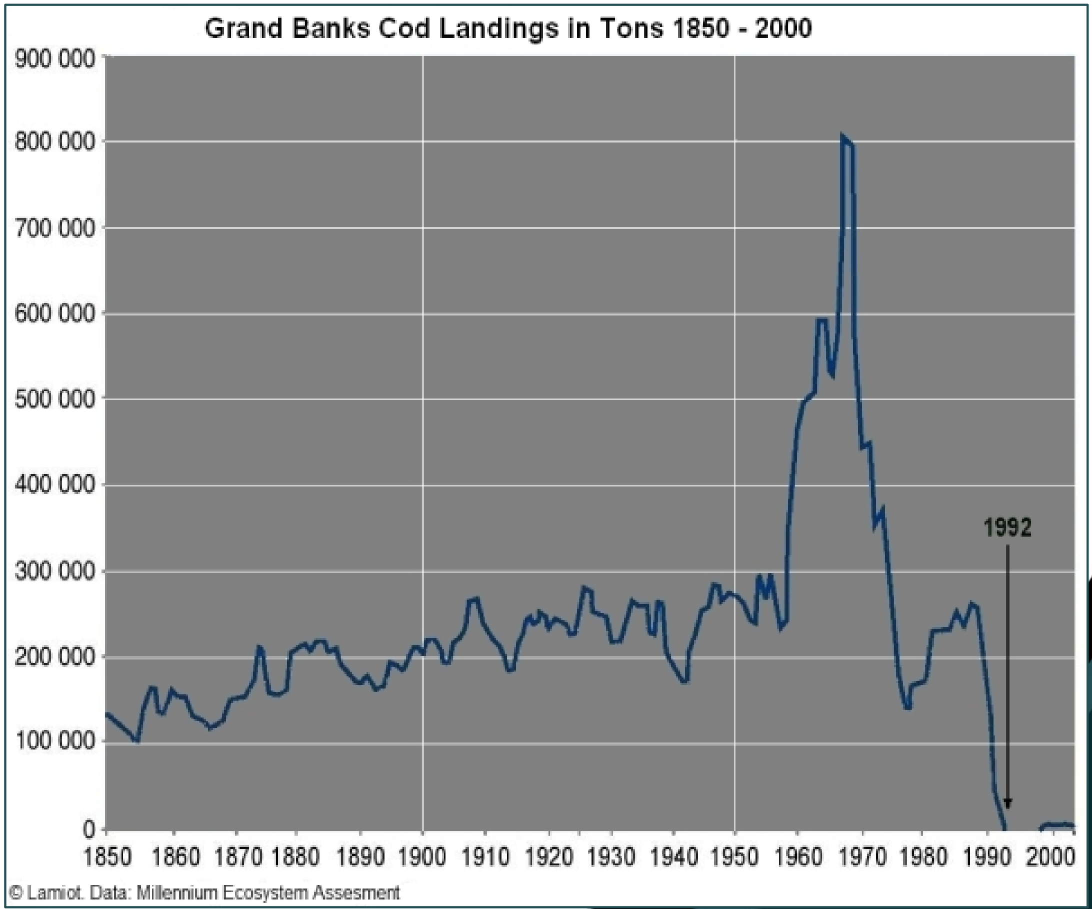
SCIENTIFIC COUNCIL MEETING – JUNE 2017

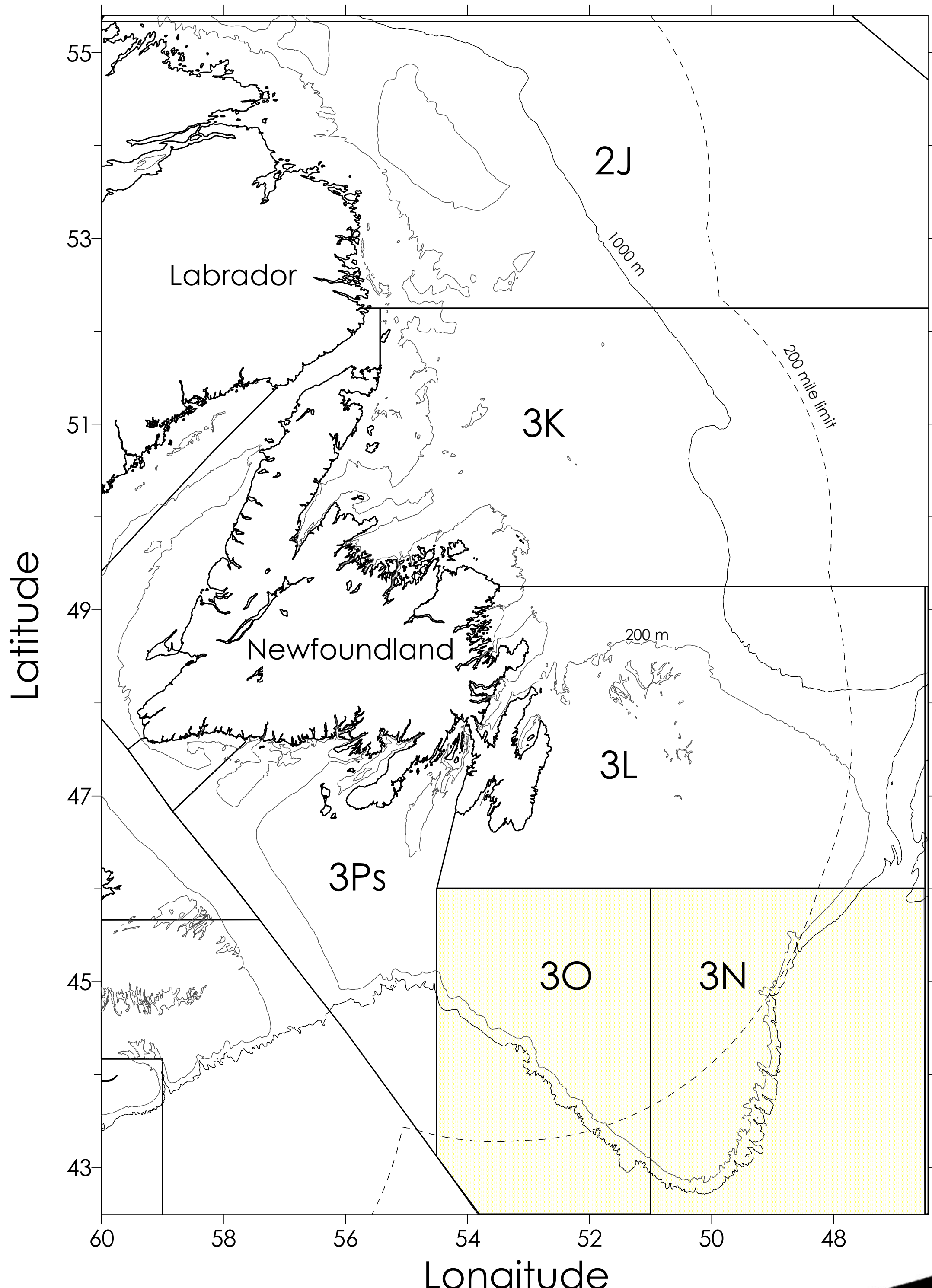
An Assessment of the Cod Stock in NAFO Divisions 3NO

by

R.M. Rideout, D.W. Ings, J. Brattey

Science Branch, Fisheries and Oceans Canada,
P. O. Box 5667, St. John's, Newfoundland, Canada A1C-5X1




2J
1000 m
Labrador
3K
200 mile limit
200 m
Newfoundland
3L
3Ps
3O
3N
55
53
51
49
47
45
43
60
58
56
54
52
50
48
Latitude
Longitude

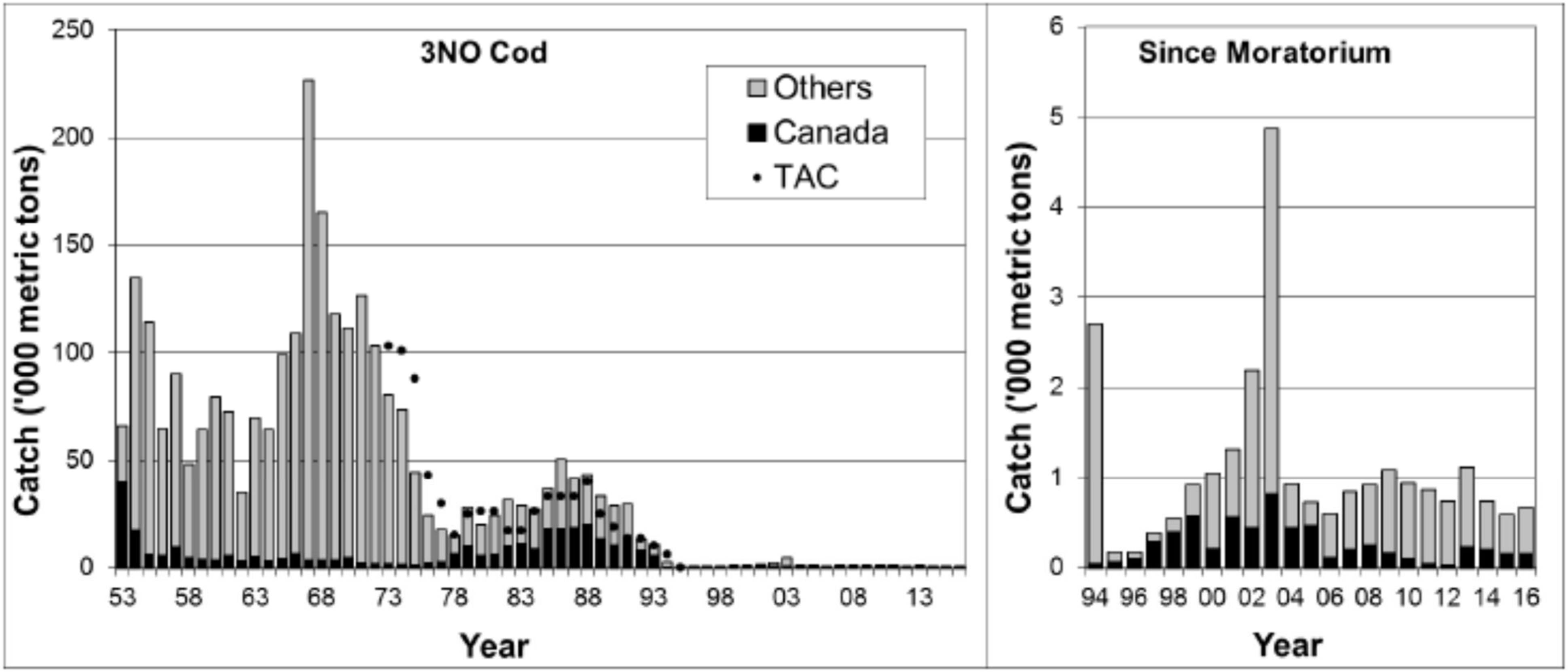
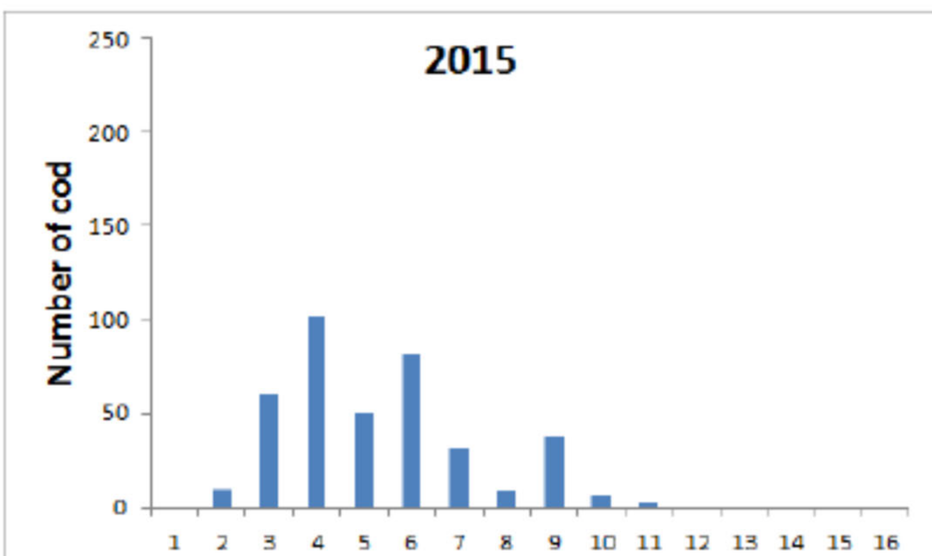
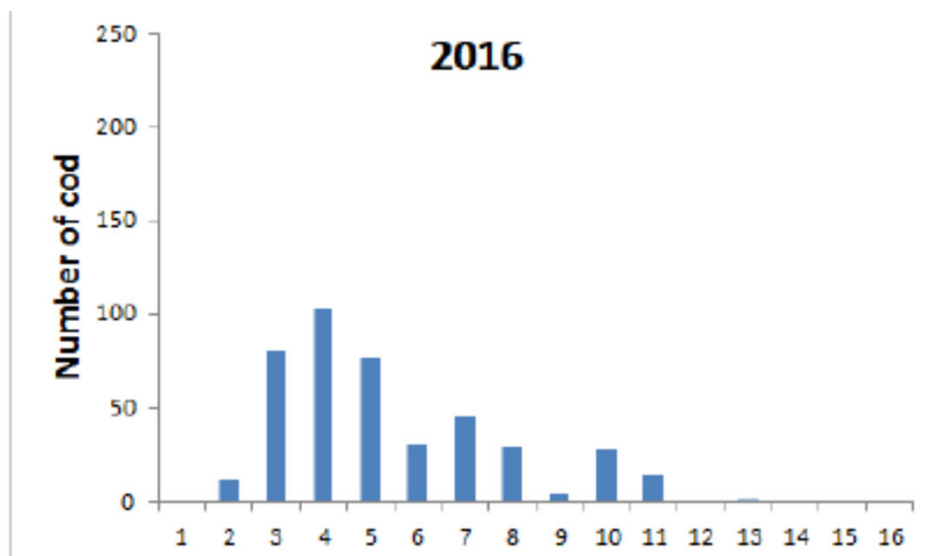


Fig. 1. Catches of cod in NAFO Divs. 3NO. The panel on the right represents catches since the moratorium in Feb. 1994.





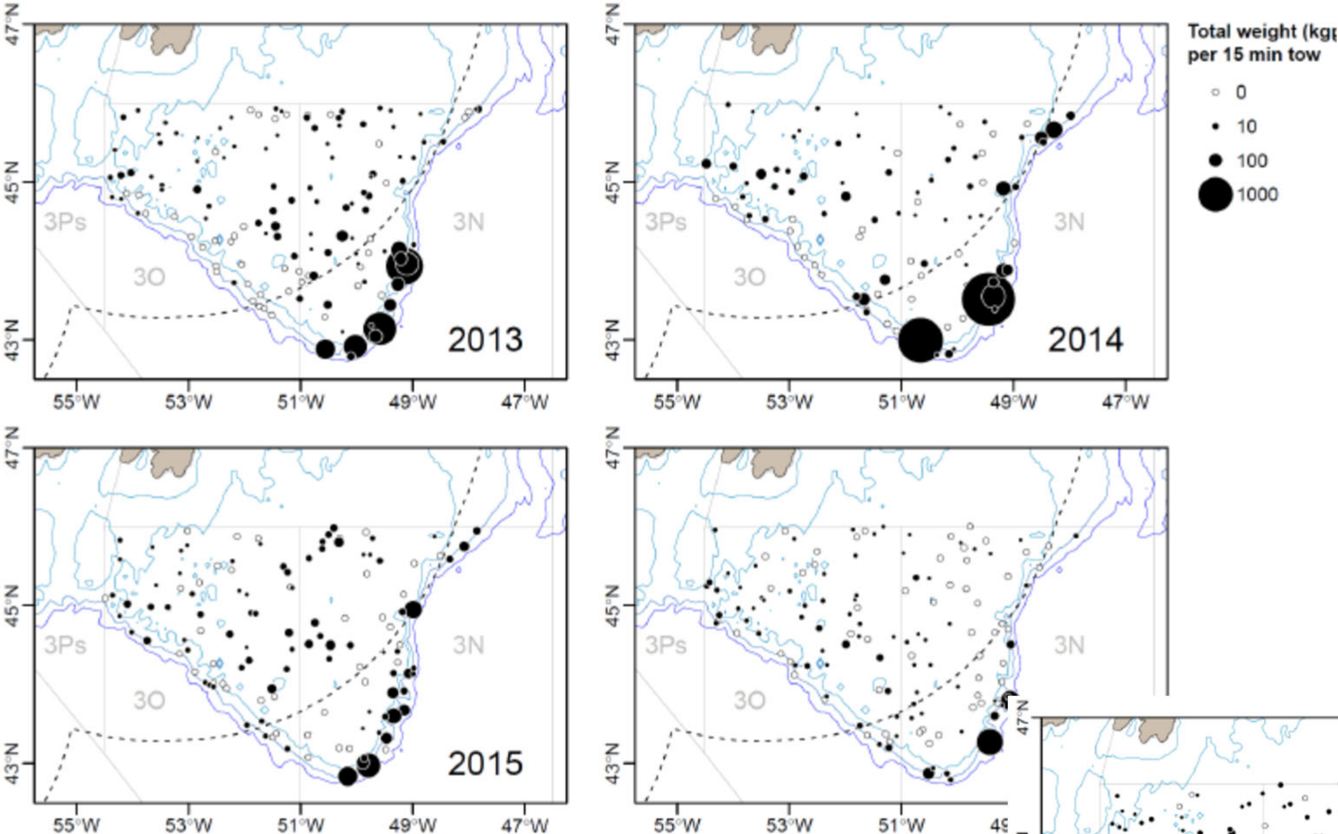


Fig. 4. Distribution plots demonstrating Spring survey set locations and total weight of location. Symbol area is proportional to catch weight.

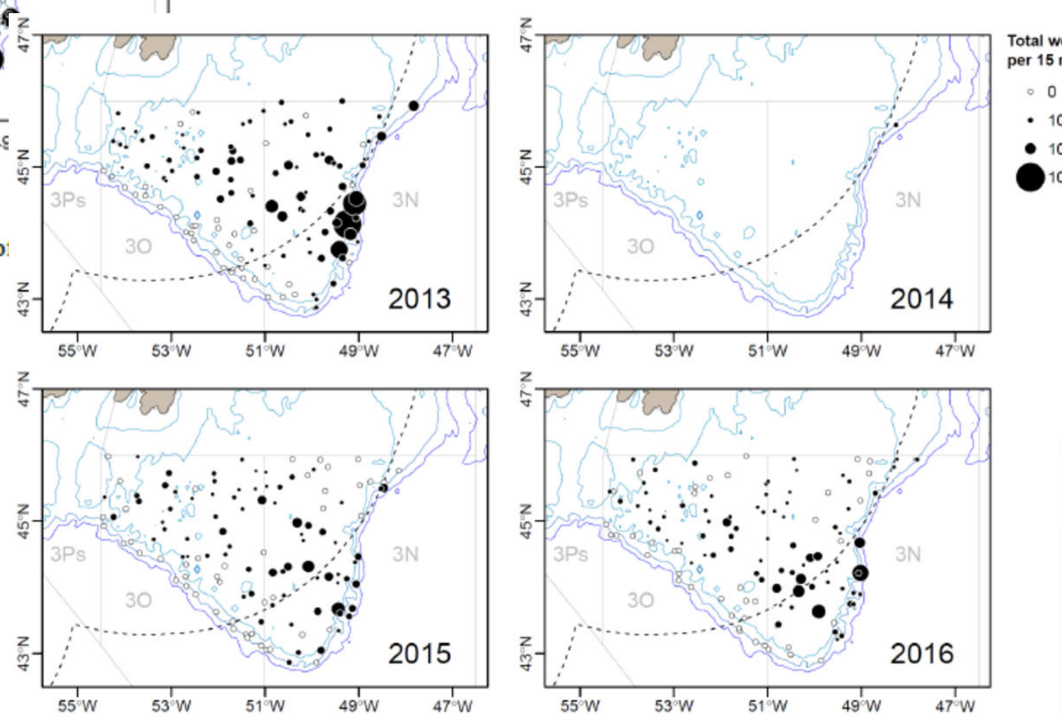
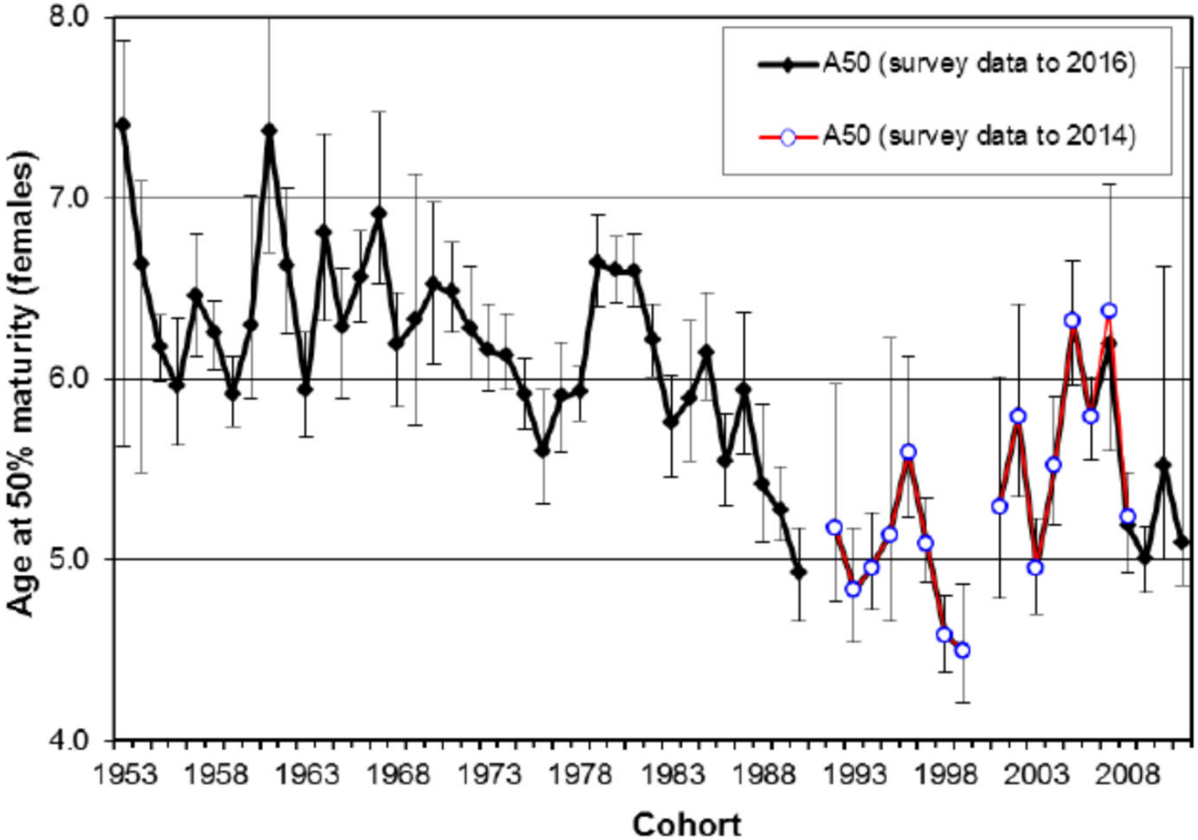
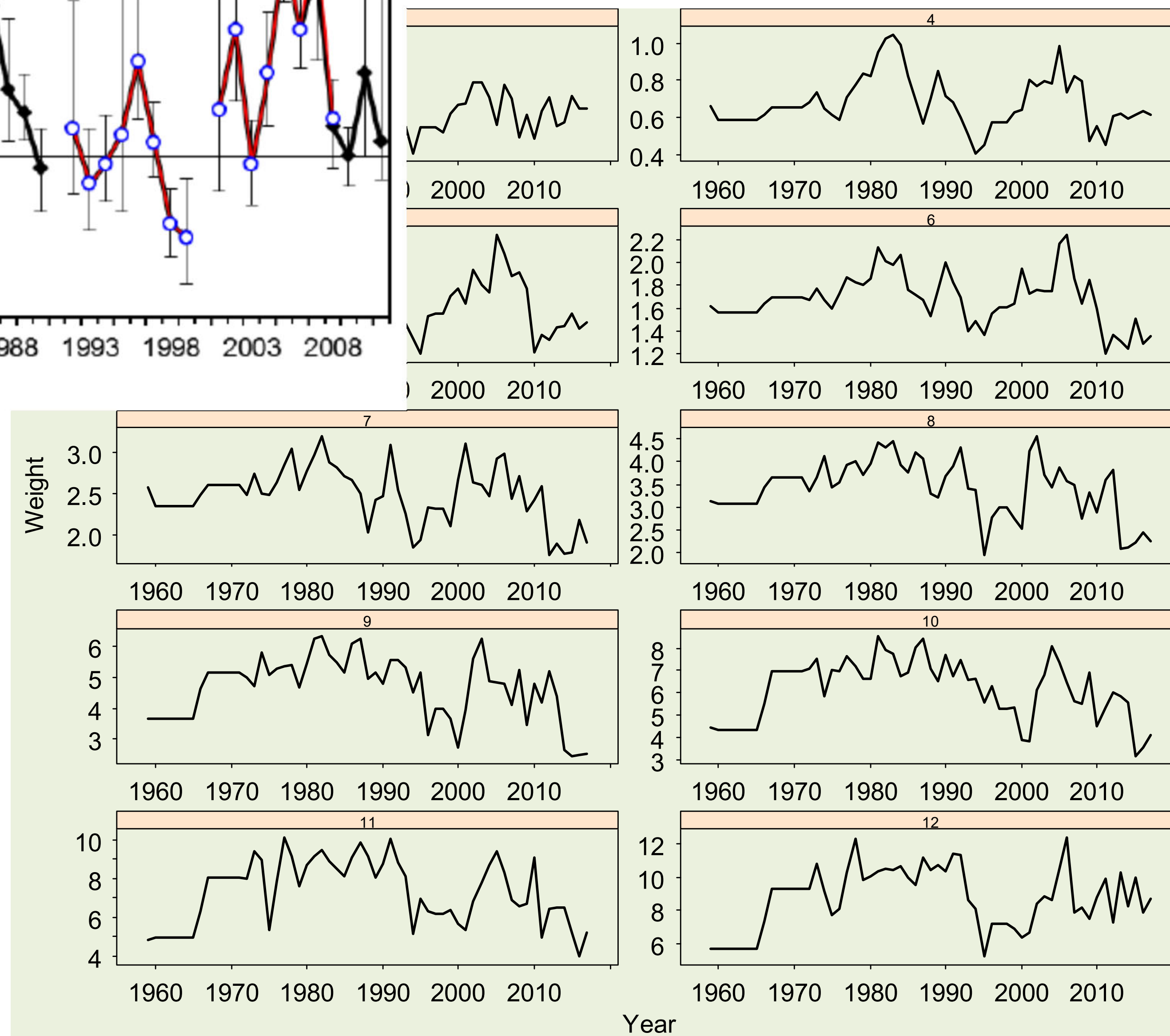


Fig. 5. Distribution plots demonstrating Autumn survey set locations and total weight of fish caught at each location. Symbol area is proportional to catch weight.


A50 (survey data to 2016)
A50 (survey data to 2014)
Age at 50% maturity (females)
Cohort


Weight
Year

3NO cod VPA

ADAPTive Framework

Input data were:

Catch numbers at age,
$C_{i,t}$ where i = 2 to 12 and t = 1959 to 2016 ,
Canadian Research Vessel survey estimates of mean numbers per tow-at-age (Campelen or Campelen equivalent values),
$RV1_{i,t}$ where i = 2 to 10 and t = 1984 to 2005 and 2007-2016, spring
$RV2_{i,t}$ where i = 2 to 10 and t = 1990 to 2013 and 2015-2016, autumn
and Canadian juvenile Research Vessel survey estimates of mean numbers per tow-at-age (Yankee 41.5 shrimp trawl in August – September)
$RV3_{i,t}$ where i = 2 to 10 and t = 1989 to 1994 .

The objective function minimized is

$$SS = \sum_{s,i,t} \{\ln(RV_{s,i,t}) - \ln(q_{s,i} N_{i,t})\}^2$$

Around the corner

where s= Survey 1 to 3 , i =age 2 to 10, t= year of survey.

The following structure was imposed:

natural mortality was assumed to be 0.2,
fishing mortality on the oldest age (12) set equal to the average F for ages 6 to 9 for years 1959-1993,
no "plus" age class,
equal weighting of all indices,
no error in the catch numbers-at-age.

Around the corner VPA

F-constraint: $F_A = F_{ave(a1:a2)} \Longrightarrow N_A = C_A e^{M_A/2} / (1 - e^{-F_{ave}})$

						Age					
	0	1	2	3	4	5	6	7	8	9	10
Year											
⋮											
1995											
1996											
1997											
1998											
1999											
2000											
2001											

Estimate around the corner because there are some 0's in catch at age 12 => $N_{Ay} = 0$

Around the corner VPA

F-constraint: $F_A = F_{ave(a1:a2)} \Longrightarrow N_A = C_A e^{M_A/2} / (1 - e^{-F_{ave}})$

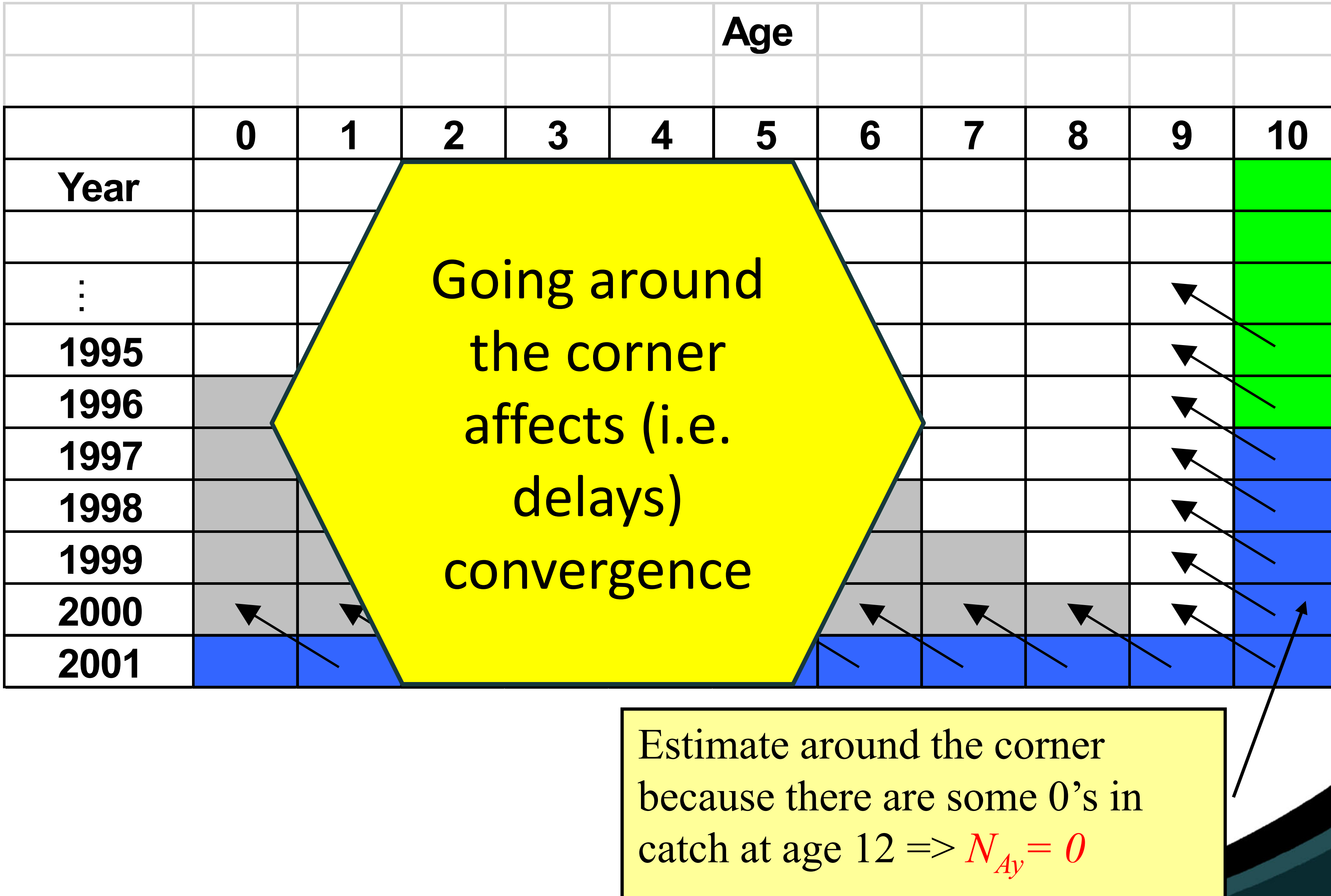
						Age					
	0	1	2	3	4	5	6	7	8	9	10
Year											
⋮											
1995											
1996											
1997											
1998											
1999											
2000											
2001											

Indicates a problem with the F-constraint approach

Estimate around the corner because there are some 0's in catch at age 12 => $N_{Ay} = 0$

Around the corner VPA

F-constraint: $F_A = F_{ave(a1:a2)} \Longrightarrow N_A = C_A e^{M_A/2} / (1 - e^{-F_{ave}})$



3NO cod VPA

The ADAPT was calibrated with Canadian RV survey indices at age from spring 1984-2005 and 2007-2016 (Table 16), autumn 1990-2013 and 2015-2016 (Table 17) and a Canadian juvenile survey 1989-94 (Table 18) to estimate population numbers $N_{i,t}$,

where i = 3 to 12, for t = 2017 (10 parameters) and i =12, for t=1994 to 2016 (23 parameters),

and Catchabilities

- $q1_i$ where i = 2 to 10 for the Canadian Research Vessel survey spring (RV1) (9 parameters)
- $q2_i$ where i = 2 to 10 for the Canadian Research Vessel survey autumn (RV2) (9 parameters)
- $q3_i$ where i = 2 to 10 for the Juvenile Research Vessel survey (RV3) (9 parameters)

27 parameters

60 parameters in total

3NO cod uses ADAPT software???

ADAPT framework: Gavaris, S. MS 1988. An adaptive framework for the estimation of population size. CAFSAC Res. Doc. 88/29, 12p.

www2.mar.dfo-mpo.gc.ca/science/adapt/adapt-e.html

Apps | Outlook Web App | Google Scholar | St. John's, Newfound | The Telegram - St. Jo | CBC Newfoundland & | Imported | Google

Fisheries and Oceans Canada | Pêches et Océans Canada | Canada

Français	Contact Us	Help	Search	Canada Site
Home	What's New	DFO National	Site Map	Media

FISHERIES AND OCEANS CANADA

Last Updated : 2011-03-28

Maritimes Region

- Fishing Industry
- General Public
- Marine and Oceans Industry
- Media
- Students and Teachers
- Scientists and Researchers

ADAPT

no longer maintained to my knowledge

The ADAPT software is used to conduct fish stock assessments using the ADAPTive framework. It implements non-linear least squares to calibrate virtual population analysis using indices of stock abundance.

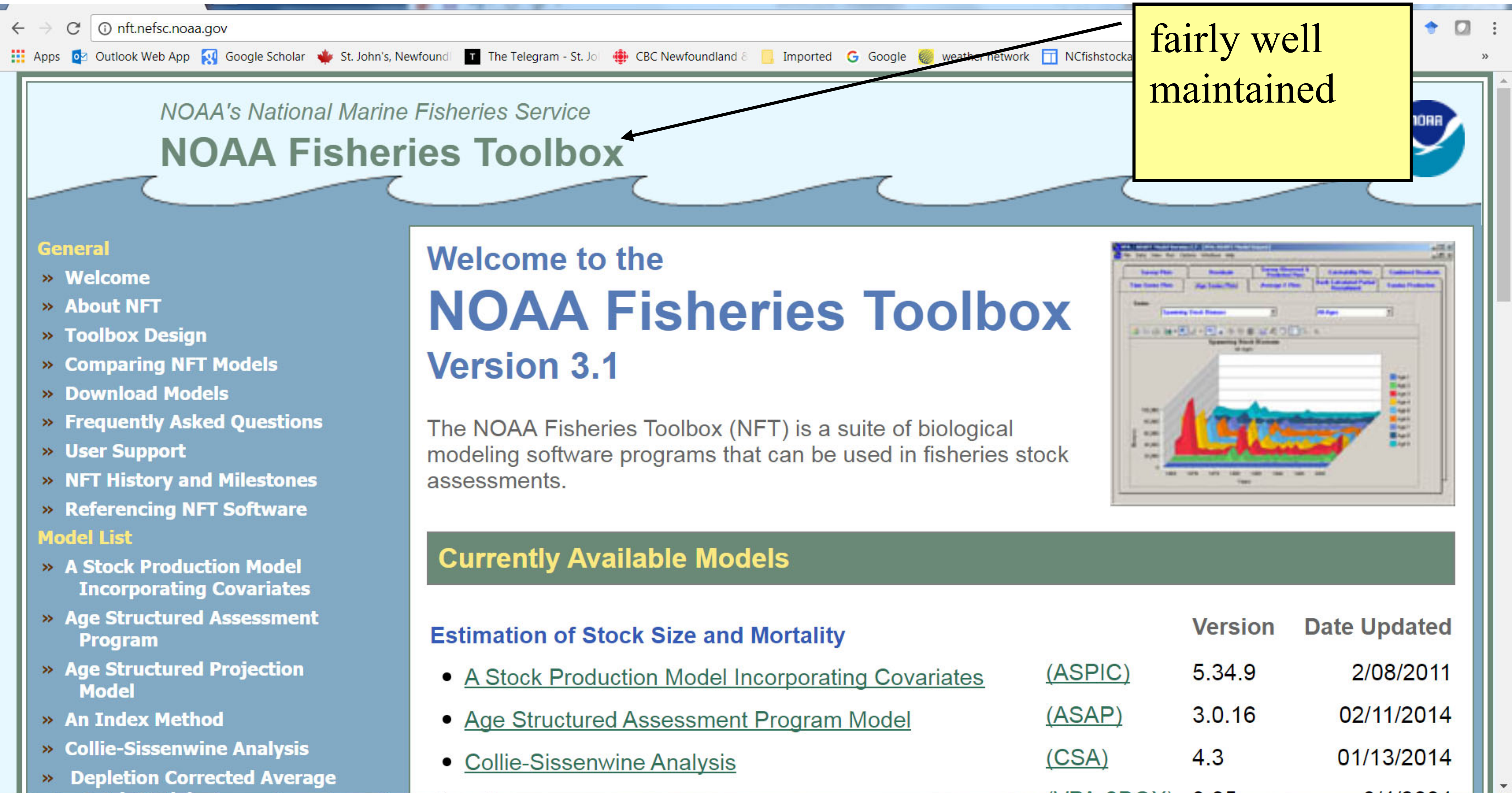
A) APL Run Time

ADAPT is written in the APL programming language. The APL Run Time System is distributed free of charge here and permits you to use all the features of ADAPT but you will not be able to modify the program. If you have the APL Development System and wish to receive the source code please e-mail Stratis Gavaris at « GavarisS@mar.dfo-mpo.gc.ca ».

Skip this step if you already have Version 6.0 of the APL run time interpreter installed.

US Stock assessment software

fairly well maintained



- » A Stock Production Model Incorporating Covariates
- » Age Structured Assessment Program
- » Age Structured Projection Model
- » An Index Method
- » Collie-Sissenwine Analysis
- » Depletion Corrected Average Catch Model
- » Dual Zone VPA
- » Instanteous Rates
- » Kalman Filter
- » Length Based Yield Per Recruit
- » Management Strategy Evaluation
- » Model Compare
- » Population Simulator - Age Based
- » Population Simulator - Length Based
- » Productivity and Susceptibility Analysis
- » Rivard Weights
- » Statistical Catch at Age Model
- » Statistical Catch at Length Model
- » Stock Recruitment Fitting Model
- » Stock Synthesis Version 3
- » Survival Estimation in Non-Equilibrium situations
- » Virtual Population Analysis
- » Visual Report Designer
- » Yield Per Recruit

Currently Available Models

Model		Version	Date Updated
Estimation of Stock Size and Mortality			
• A Stock Production Model Incorporating Covariates	(ASPIC)	5.34.9	2/08/2011
• Age Structured Assessment Program Model	(ASAP)	3.0.16	02/11/2014
• Collie-Sissenwine Analysis	(CSA)	4.3	01/13/2014
• Dual Zone Virtual Population Analysis	(VPA-2BOX)	3.05	8/4/2004
• Statistical Catch at Age Model	(STATCAM)	1.4.1	5/2/2008
• Statistical Catch at Length Model	(SCALE)	1.0.11	9/13/2013
• Stock Synthesis Version 3	(SS3)	3.45f	10/18/2012
• Virtual Population Analysis	(VPA)	3.4.4	3/3/2014
Management Scenario Projections			
• Age Structured Projection Model	(AGEPRO)	4.2.2	9/17/2013
Biological Reference Points			
• Age Based Yield Per Recruit	(YPR)	3.3	9/17/2013
• An Index Method	(AIM)	2.5.0	1/31/2014
• Length Based Yield Per Recruit	(YPRLEN)	2.1	4/20/2012
• Stock Recruitment Fitting Model	(SRFIT)	7.0.1	3/18/2010
Model Performance Evaluation			
• Population Simulator - Age Based	(POPSIM-A)	8.2	12/12/2013
• Population Simulator - Length Based	(POPSIM-L)	8.0	12/12/2013
• Management Strategy Evaluation	(MSE)	4.0	12/23/2013
• Visual Report Designer	(VisRpt)	1.6.1	4/2/2008
Models for Data Limited Situations			
• Depletion Corrected Average Catch Model	(DCAC)	2.1.1	10/4/2012
• Survival Estimation in Non-Equilibrium situations	(SEINE)	1.3	9/15/2008

US Stock assessment software

Model for Analyzing Tagging Data

• Instantaneous Rates	(IRATE)	2.0	4/19/2013

Additional Tools

• Kalman Filter	(KALMAN)	2.3	7/24/2009
• Model Compare	(MCOMP)	4.3	2/10/2014
• Productivity and Susceptibility Analysis	(PSA)	1.4	3/4/2010
• Rivard Weights Calculator	(RIVARD)	2.0	10/24/2008

Notes

This suite of programs supersedes the former toolboxes FACT (Fisheries Assessment and Computation Toolbox) and WHAT (Woods Hole Assessment Toolbox).

This information is distributed solely for the purpose of pre-dissemination peer review under applicable information quality guidelines. It has not been formally disseminated by NMFS. It does not represent and should not be construed to represent any agency determination or policy.

NOAA Fisheries Toolbox Version 3.1
website last modified May 16, 2016

Printer friendly view

www.nefsc.noaa.gov NMFS Search Link Disclaimer webMASTER Privacy Policy (File Modified Jan. 10 2012)

European Stock assessment software

and Code ×

ices.dk/marine-data/tools/Pages/Software.aspx

Outlook Web App | Google Scholar | St. John's, Newfound | The Telegram - St. Jo | CBC Newfoundland & | Imported | Google | weather network | NCfishstockassessme | LISTSERV 15.5 - FIS

ICES CIEM

Contact | Sitemap | FAQ | Glossary | SharePoint Login | Admin | Search this site

EXPLORE US | NEWS AND EVENTS | MARINE DATA | PUBLICATIONS | COMMUNITY

Dataset Collections | Data Portals | Tools | Maps | Vocabularies | Guidelines and Policy

TOOLS

DATA BASICS
- Submission status
- Data calls
- Data formats
- Code lookup
- Code request

DATA SCREENING
- Environment
- Fisheries

Software and Code

Print it | Send to | Share it

Tools provided by ICES, other institutes and projects

ICES GitHub

GitHub is a web-based repository hosting service to share and develop source code on scientific methods. It allows anyone to view and download data tools and analytical methods, as well as suggesting changes. ICES has three GitHub sites for different purposes: products, tools in development and a site for Expert Groups. Links to these areas, and guidelines are provided in the menu to the right.

ICES GITHUB

- ICES tools (production)
- ICES tools (development)
- ICES Expert Groups GitHub area
- ICES GitHub guidelines

SOFTWARE LINKS

European Stock assessment software

- Quality control checks

DATA TOOLS

- Oceanographic calculator
- **Software and Code**
- Station dictionary
- Web Services

ASSESSMENT TOOLS

- Stock assessment graphs
- Transparent Assessment Framework

Software disclaimer

The downloading and use of any software from this website grants you only a limited, nonexclusive license for use solely by you for your own personal use and not for republication, distribution, assignment, sublicense, sale or other use. No part of any content, coding or documentation may be reproduced in any form or incorporated into any information retrieval system without the express permission of ICES or the third party that provided the software.

The downloading and installation of software from this website is at your own risk. ICES accepts no responsibility for the correct functioning and operation of software downloaded from this website, nor any incidental, special or consequential damages associated with downloading and installing software provided by ICES.

Environmental Data Tool

- DATSU
  Data Screening Utility
- Download DATSU user instructions

Fisheries Assessment Tools

- AMCI (2005)
  A flexible assessment model combining information from various sources.
- aspic382 (2011)
  A Stock-Production Model Incorporating Covariates
- HCS (2010)
  A family of general purpose programs stochastic simulation
  simulation of management decision rules
- PROST (2014)
  A software to estimate Fmsy and SSBmsy including density dependent growth and maturity, and cannibalism in a stochastic framework
- SAM (2011)
  The SAM model developed by Anders Nielsen, is a state space model used by ICES working groups
- TASACS (2009)
  A toolbox of programs to perform an analytic assessment of a fish stock

Old Stuff!

www.fao.org/docrep/003/x9026e/x9026e00.htm

Outlook Web App | Google Scholar | St. John's, Newfoundl | The Telegram - St. Jo | CBC Newfoundland & | Imported | Google | weather network | NCfishstockassessme | LISTSERV 15.5 -

FAO CORPORATE DOCUMENT REPOSITORY

Title: Virtual Population Analysis A Practical Manual for Stock Assessment...

Produced by: Fisheries and Aquaculture Department

PDF version

More details

FAO Fisheries Technical Paper 400

Virtual Population Analysis - A Practical Manual for Stock Assessment

Hans Lassen
and
Paul Medley

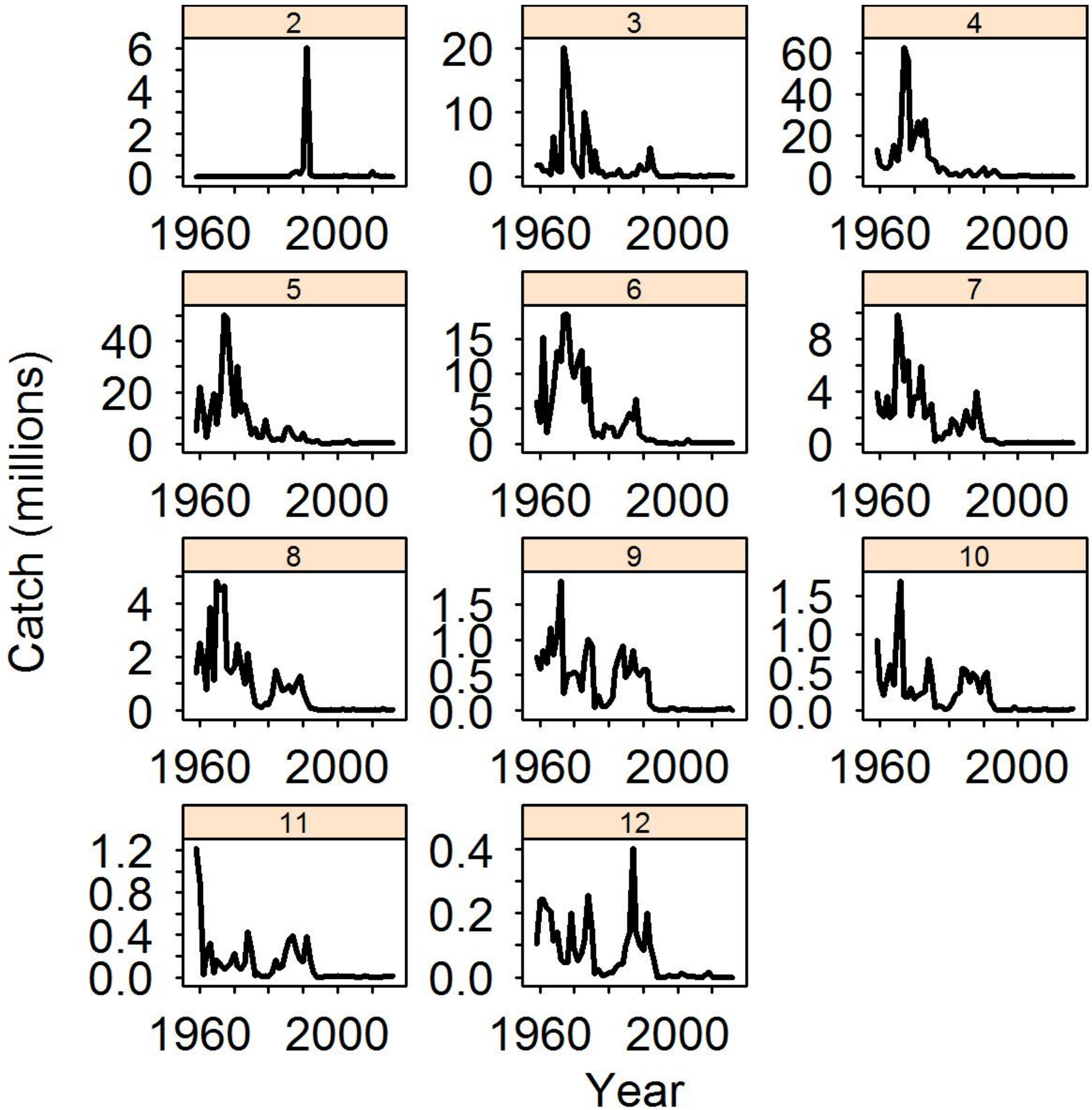
FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS

Rome, 2001

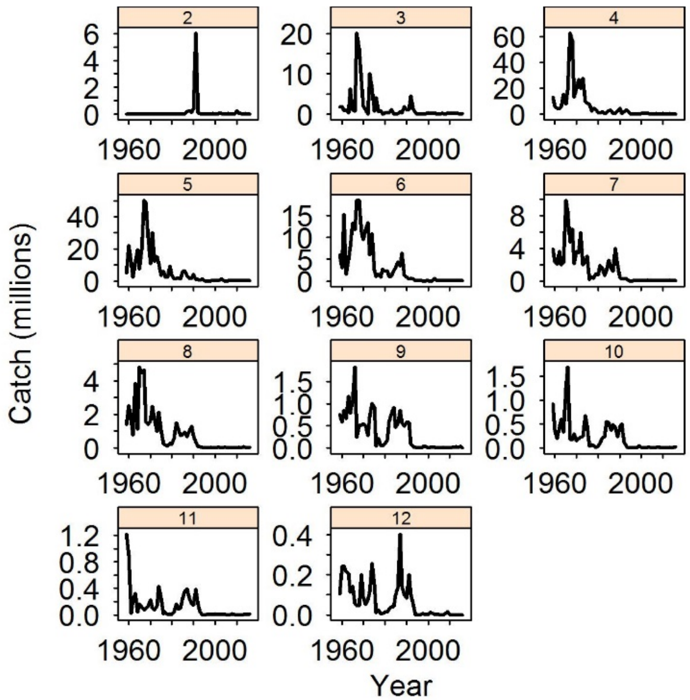
SPA in R

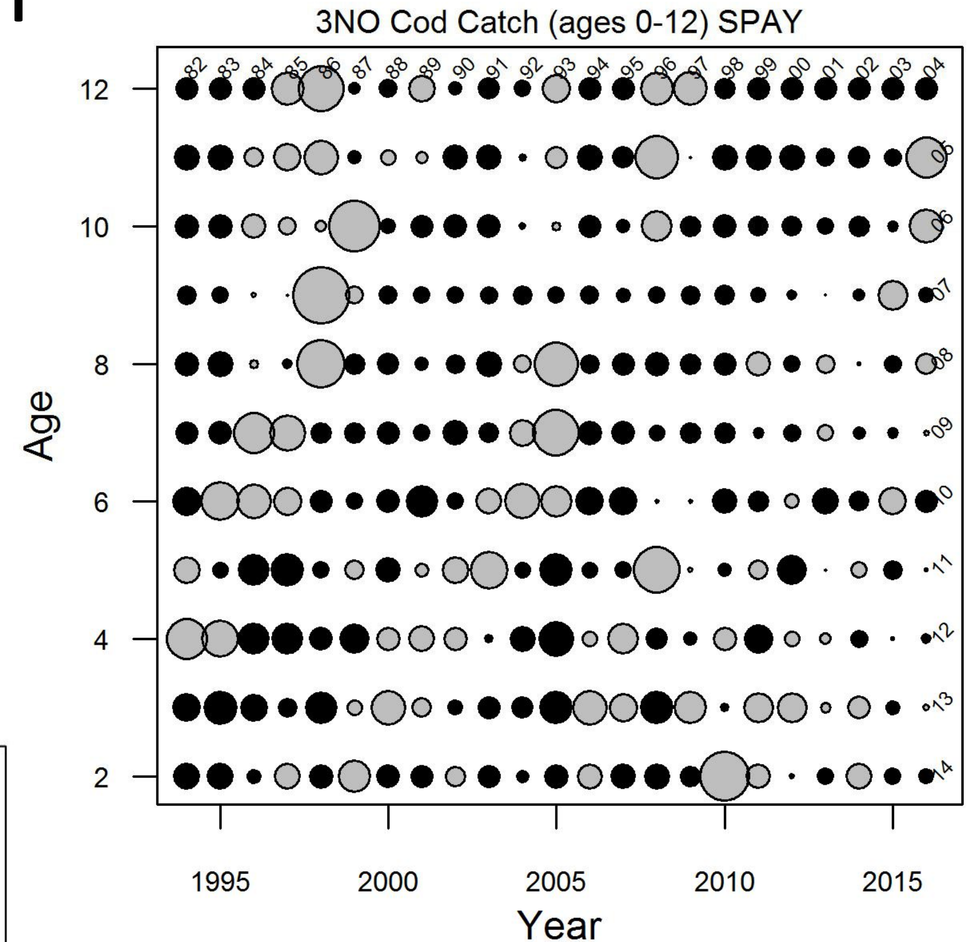
- Lets do 3NO cod ADAPT in R!
- And compare with results in NAFO SCR Doc
- First we plot the data (ALWAYS!)

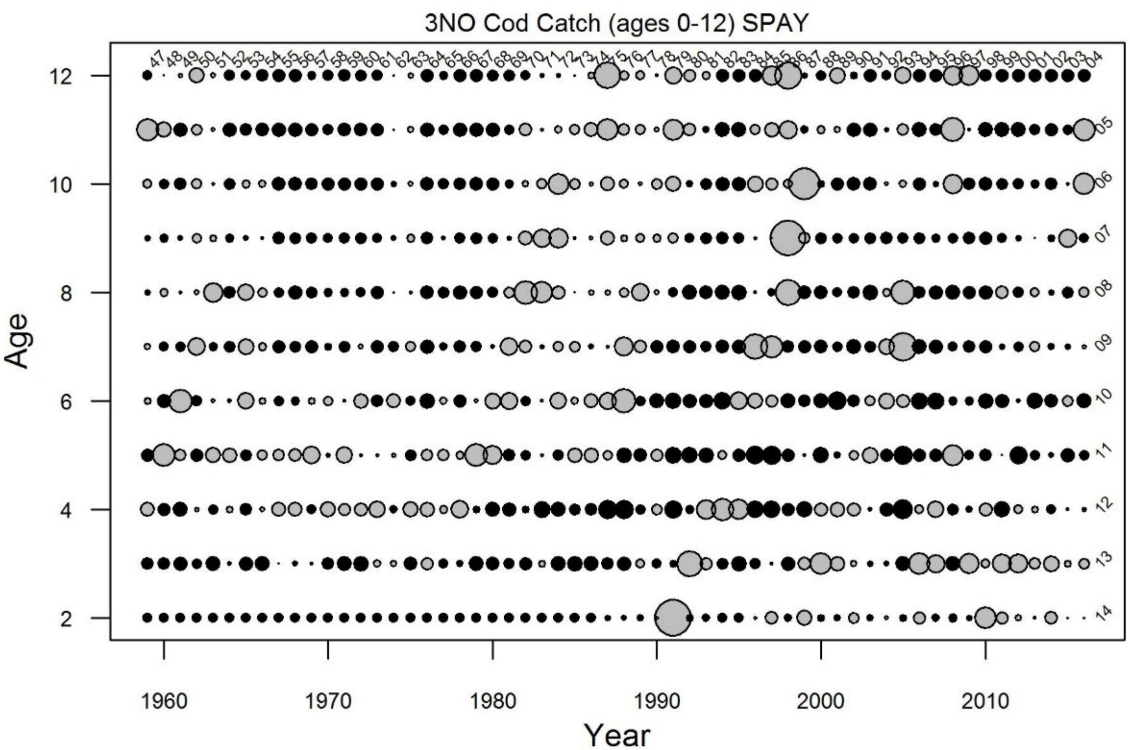
Catch



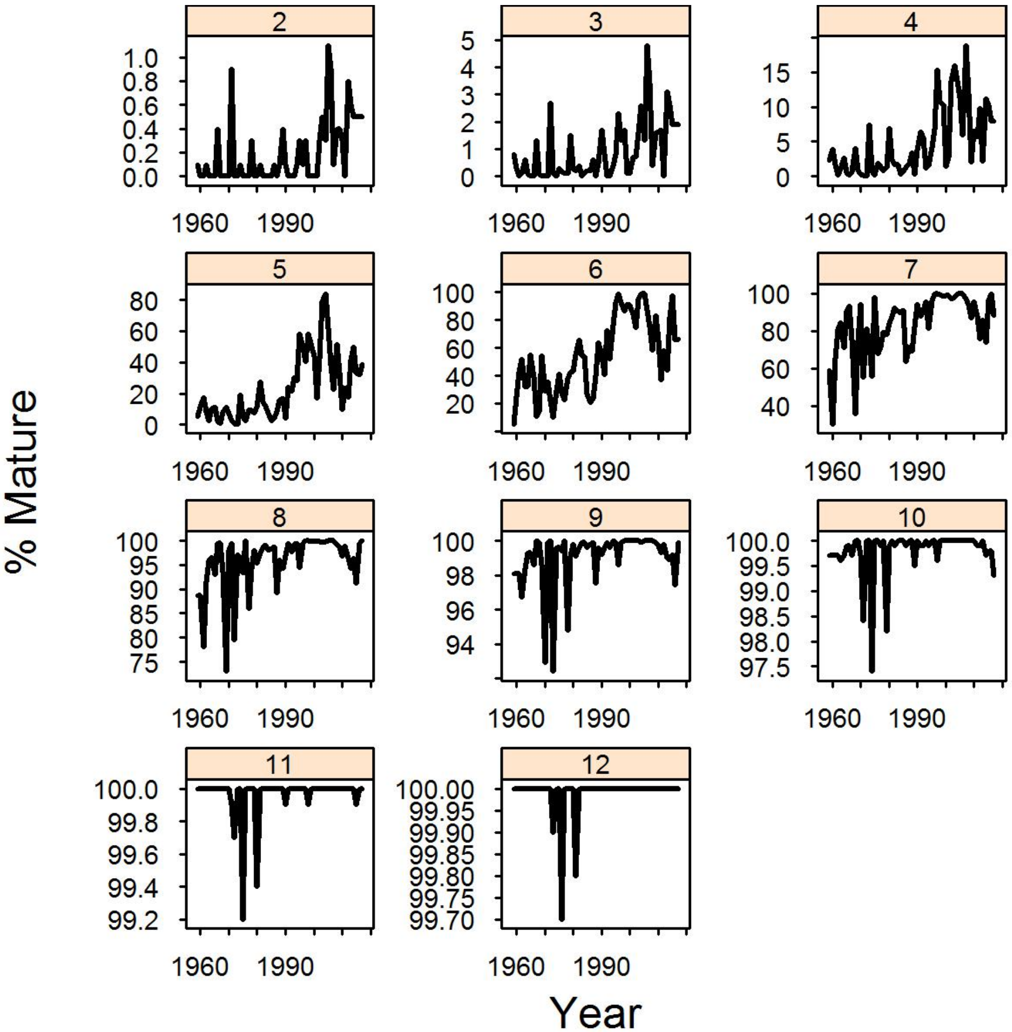
Catch

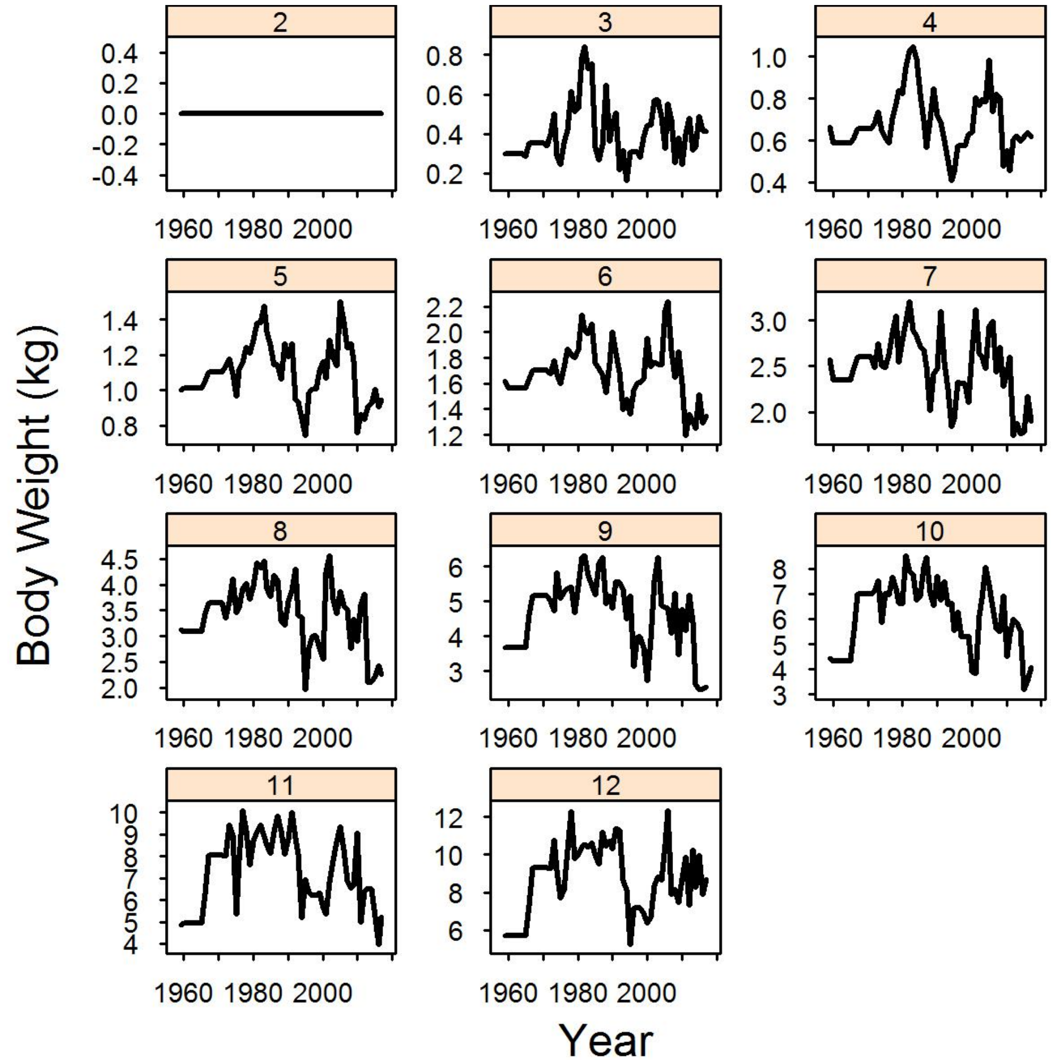

Catch (millions)
Year
2
3
4
5
6
7
8
9
10
11
12


3NO Cod Catch (ages 0-12) SPAY
Age
Year
1995
2000
2005
2010
2015

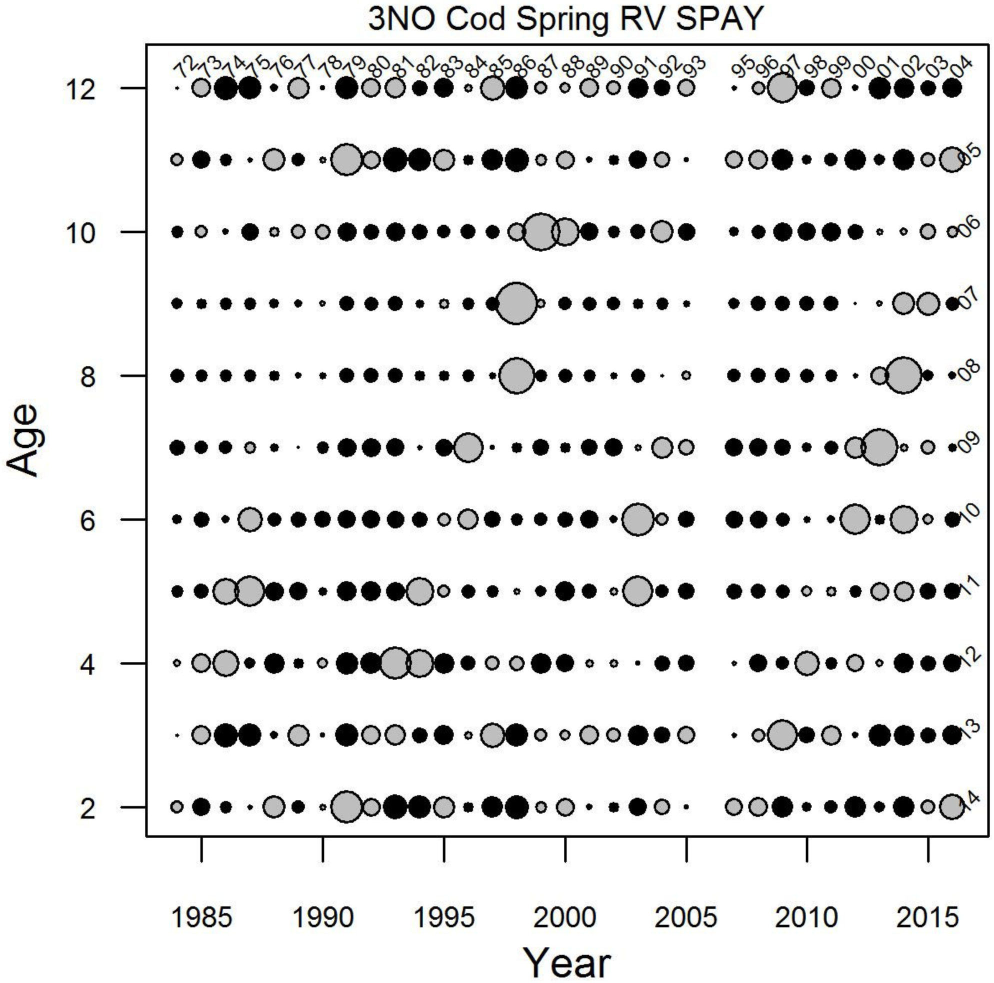

3NO Cod Catch (ages 0-12) SPAY
Age
Year
1960
1970
1980
1990
2000
2010

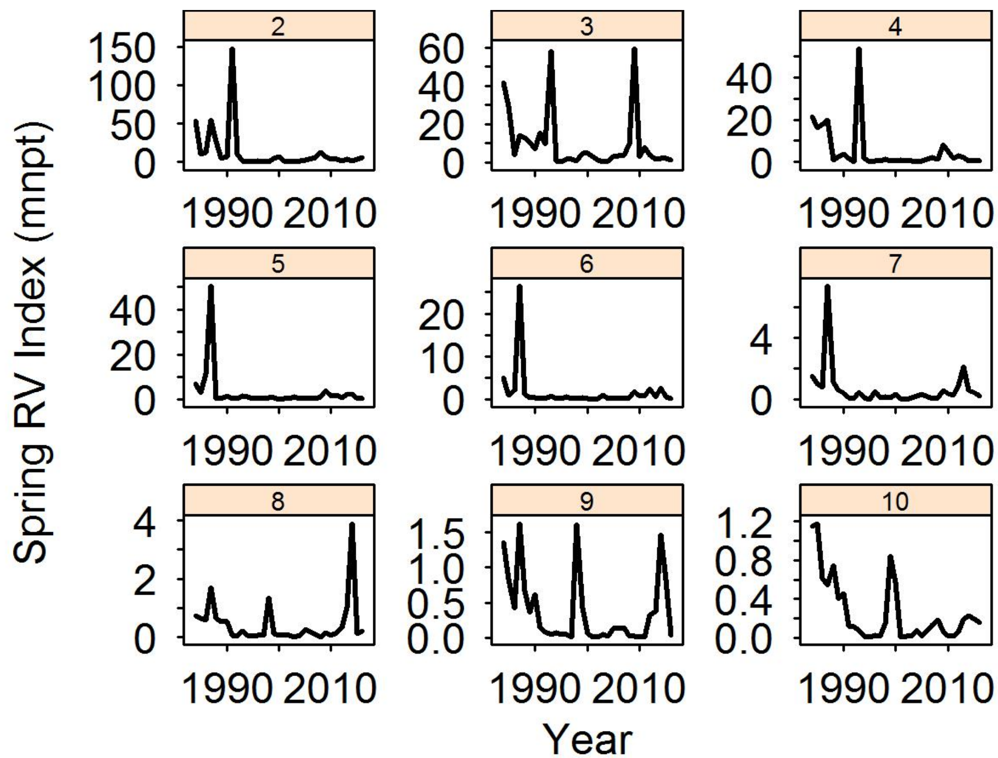
Biological inputs: mats and weights



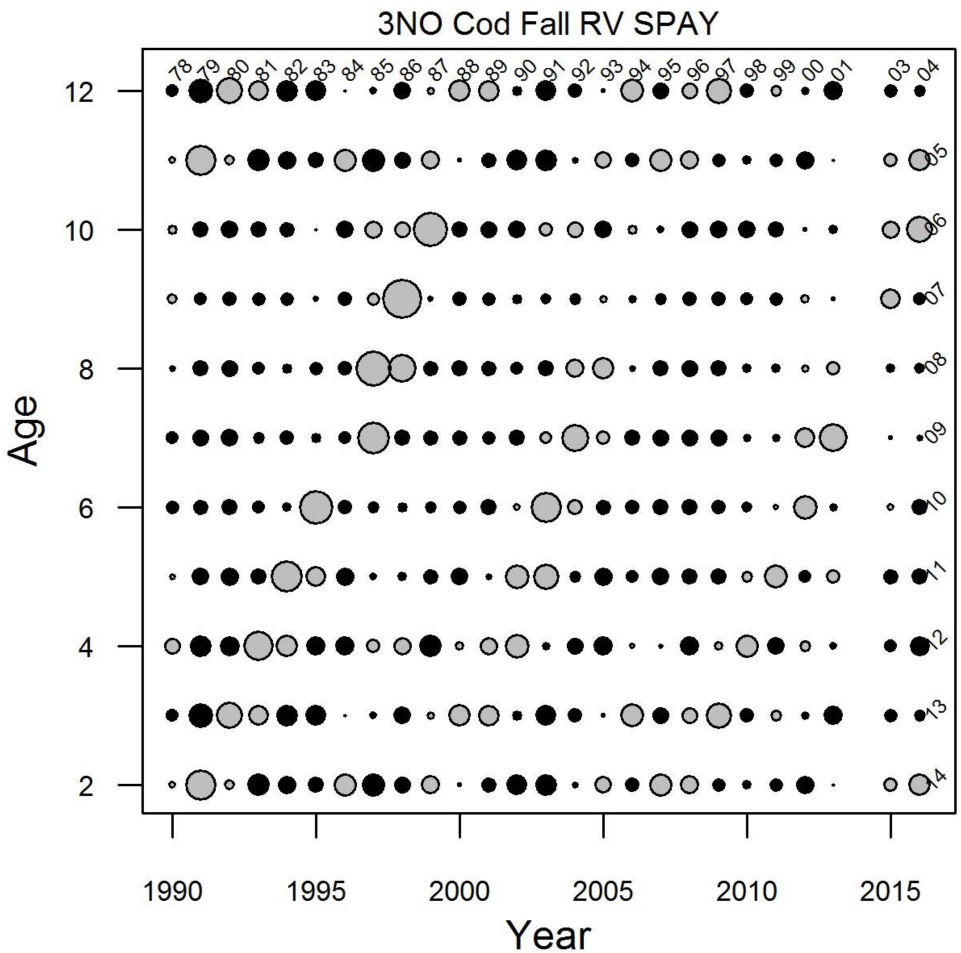


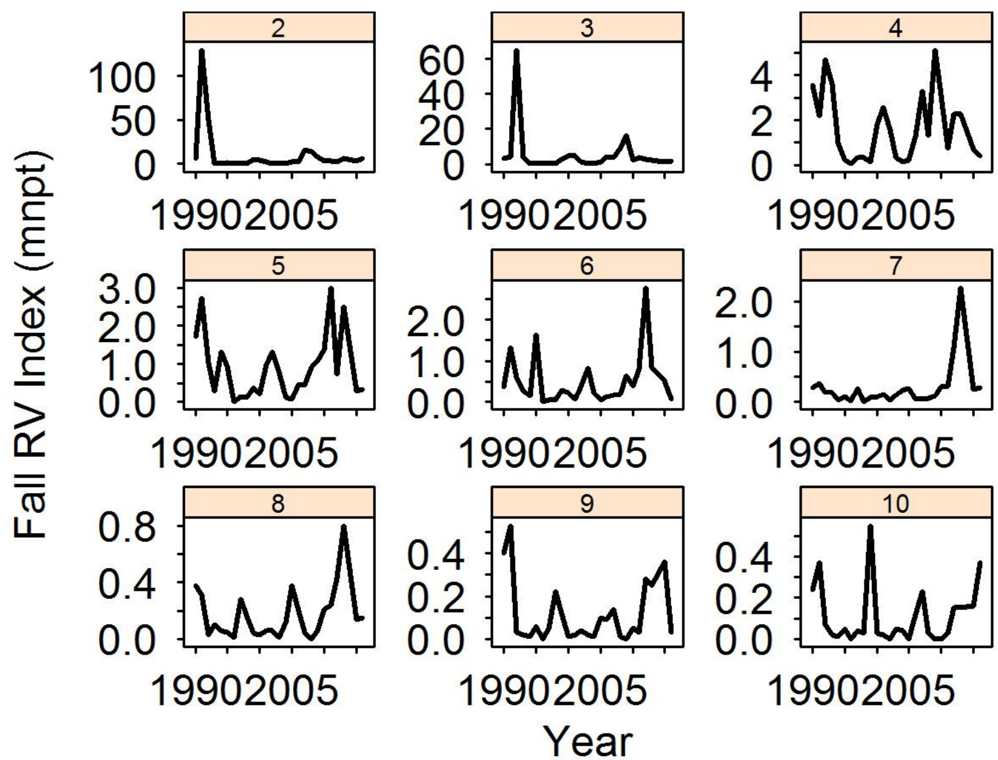
Survey Indices



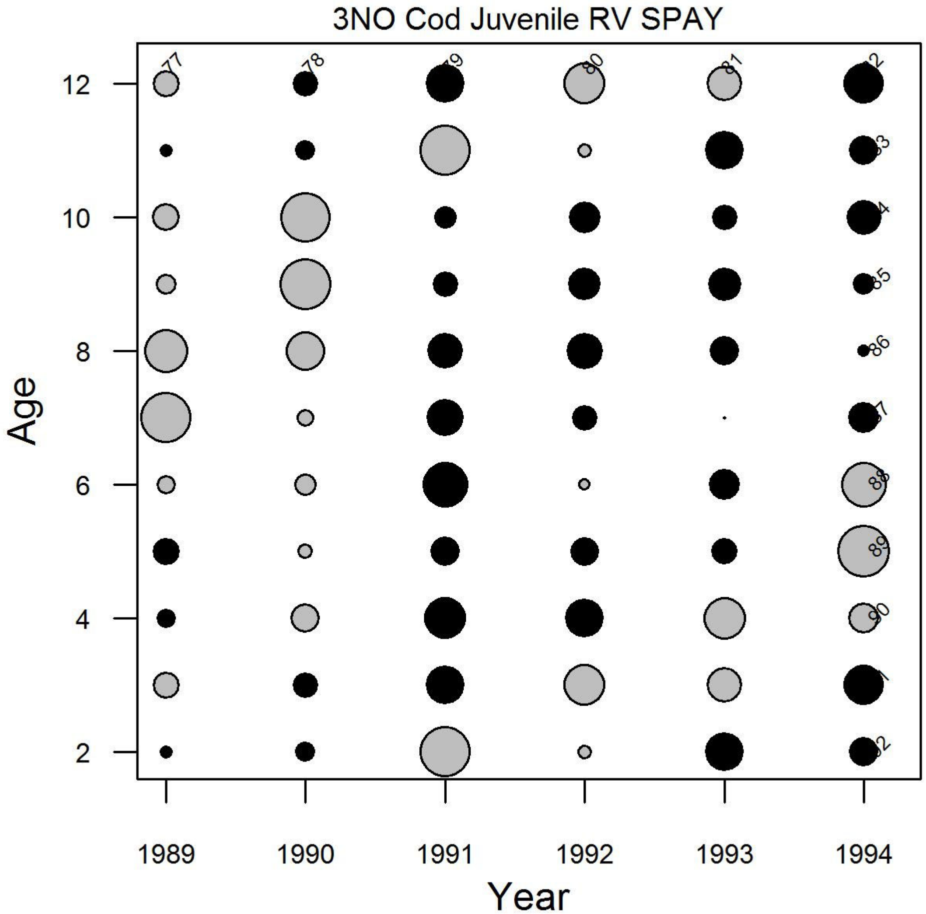


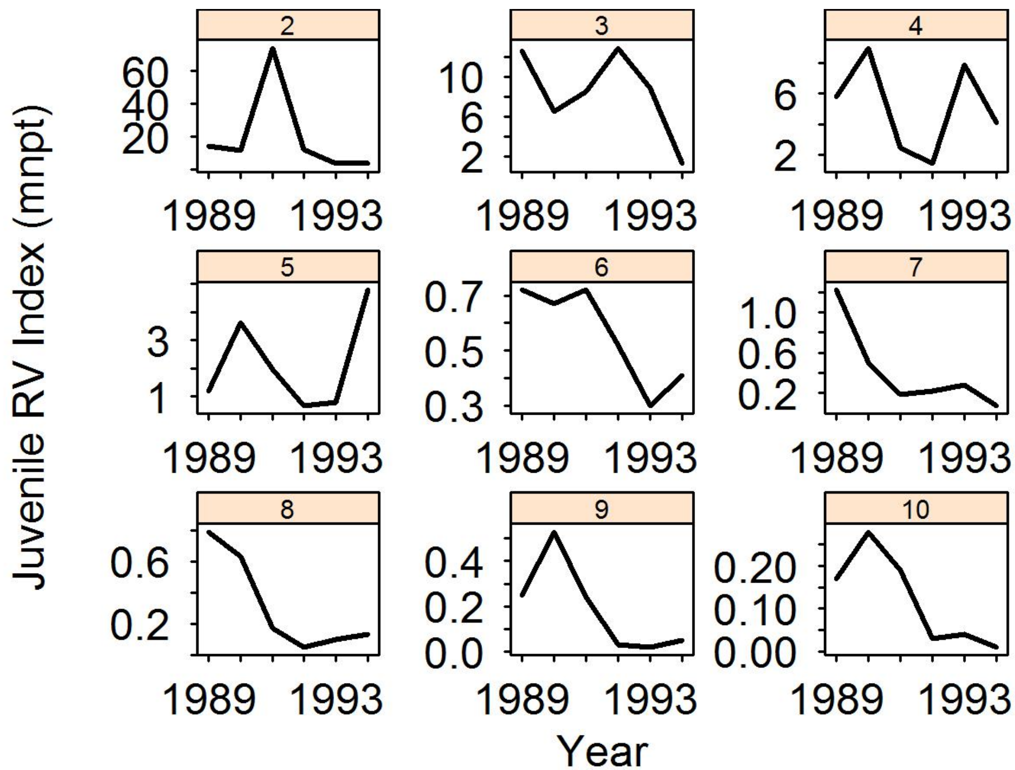
Survey Indices





Survey Indices





R ADAPT

Substantial R code to read and plot data inputs

```
> load('data\\3LNO.RData')
> ls()
[1] "catch"     "catch.vec" "indices"   "mat"       "mat.vec"   "wt"        "wt.vec"
```

```
> str(catch)
'data.frame':   58 obs. of  11 variables:
 $ Age2 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Age3 : num  1.711 1.846 0.812 1.026 0.313 ...
 $ Age4 : num  13.04 6.5 4.4 3.88 5.76 ...
 $ Age5 : num  5.07 22.05 11.7 2.21 11.21 ...
 $ Age6 : num  6.03 3.1 15.26 1.58 4.85 ...
 $ Age7 : num  3.94 2.38 2.01 3.59 1.94 ...
 $ Age8 : num  1.392 2.504 1.672 0.773 3.84 ...
 $ Age9 : num  0.757 0.583 0.847 0.668 1.165 ...
 $ Age10: num  0.926 0.387 0.196 0.433 0.608 ...
 $ Age11: num  1.22 0.898 0.025 0.226 0.322 0.037 0.163 0.122 0.071 0.09 ...
 $ Age12: num  0.103 0.242 0.245 0.216 0.208 0.112 0.141 0.057 0.045 0.045 ...
```

R ADAPT Inputs

```
> str(catch.vec)
'data.frame':   638 obs. of  3 variables:
 $ Year : int  1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 ...
 $ Age  : num  2 2 2 2 2 2 2 2 2 2 ...
 $ index: num  0 0 0 0 0 0 0 0 0 0 ...
```

```
> str(indices)
'data.frame':   576 obs. of  4 variables:
 $ Year  : int  1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 ...
 $ Age   : num  2 2 2 2 2 2 2 2 2 2 ...
 $ index : num  6.15 129.66 49.65 0.72 0.62 ...
 $ survey: chr  "Fall" "Fall" "Fall" "Fall" ...
```

```
> str(mat.vec)
'data.frame':   649 obs. of  3 variables:
 $ Year : num  1959 1960 1961 1962 1963 ...
 $ Age  : num  2 2 2 2 2 2 2 2 2 2 ...
 $ index: num  0.0006 0 0.0001 0.0012 0.0002 0 0 0.0043 0.0001 0 ...
```

R ADAPT Inputs

```
> str(mat)
'data.frame':   59 obs. of  11 variables:
 $ Age2 : num  0.0006 0 0.0001 0.0012 0.0002 0 0 0.0043 0.0001 0 ...
 $ Age3 : num  0.0079 0.0033 0.0001 0.0009 0.0056 0.001 0 0.0005 0.0132
0.0012 ...
 $ Age4 : num  0.0226 0.0393 0.0168 0.0012 0.0099 0.0266 0.0049 0.0006
0.0067 0.0398 ...
 $ Age5 : num  0.0555 0.1157 0.1731 0.0825 0.0229 ...
 $ Age6 : num  0.0509 0.2512 0.4251 0.5172 0.3206 ...
 $ Age7 : num  0.59 0.304 0.657 0.807 0.846 ...
 $ Age8 : num  0.886 0.886 0.781 0.916 0.959 ...
 $ Age9 : num  0.981 0.981 0.981 0.967 0.984 ...
 $ Age10: num  0.997 0.997 0.997 0.997 0.996 ...
 $ Age11: num  1 1 1 1 1 ...
 $ Age12: num  1 1 1 1 1 ...
```

```
> str(wt.vec)
'data.frame':   649 obs. of  3 variables:
 $ Year : num  1959 1960 1961 1962 1963 ...
 $ Age  : num  2 2 2 2 2 2 2 2 2 2 ...
 $ index: num  0 0 0 0 0 0 0 0 0 0 ...
```

R ADAPT Inputs

```
> str(wt)
'data.frame':   59 obs. of  11 variables:
 $ Age2 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Age3 : num  0.301 0.301 0.301 0.301 0.301 0.301 0.287 0.351 0.351 0.351 ...
 $ Age4 : num  0.664 0.587 0.587 0.587 0.587 0.587 0.587 0.615 0.657 0.657 ...
 $ Age5 : num  1 1.01 1.01 1.01 1.01 ...
 $ Age6 : num  1.62 1.56 1.56 1.56 1.56 ...
 $ Age7 : num  2.57 2.35 2.35 2.35 2.35 ...
 $ Age8 : num  3.13 3.09 3.09 3.09 3.09 ...
 $ Age9 : num  3.67 3.67 3.67 3.67 3.67 ...
 $ Age10: num  4.42 4.32 4.32 4.32 4.32 ...
 $ Age11: num  4.84 4.96 4.96 4.96 4.96 ...
 $ Age12: num  5.69 5.69 5.69 5.69 5.69 ...
```

R ADAPT: Step 1

```
# get log indices and wt=0 for index=0
indices$YC = indices$Year-indices$Age
indices$log.index = NA
indices$wt = 0
ind = indices$index==0
indices$wt[!ind]=1
indices$log.index[!ind] = log(indices$index[!ind])

#set fraction of year survey takes palce
indices$fs=NA
indices$fs[indices$survey=="Juvenile"] = 8.5/12
indices$fs[indices$survey=="Spring"] = 5.5/12
indices$fs[indices$survey=="Fall"] = 10.5/12
```

R ADAPT: Step 2

```
## create some useful objects

## all the inputs for cohort model in the list pop.dat
pop.dat = list(
  catch=catch,
  mat=mat,
  wt=wt,
  Year=sort(unique(wt.vec$Year)),
  Age=sort(unique(wt.vec$Age))
)
pop.dat$M = 0.2
pop.dat$A = length(pop.dat$Age)
pop.dat$Y = length(pop.dat$Year)
pop.dat$n = length(wt.vec$Age)
pop.dat$imap = 1:pop.dat$n
```

R ADAPT: Step 3

```
## setup the survey observation model
indices.nz = subset(indices,wt==1); #use data with estimation wt=1

temp1 = paste(wt.vec$Year,"_",wt.vec$Age,sep='')
temp2 = paste(indices.nz$Year,"_",indices.nz$Age,sep='')
indices.nz$imap = pop.dat$imap[match(temp2,temp1)]

#define q parmeters and map
indices.nz$qparm = paste(indices.nz$survey,"_",indices.nz$Age,sep='')
q.name = unique(indices.nz$qparm)
nq = length(q.name)
indices.nz$qmap = (1:nq)[match(indices.nz$qparm,q.name)]

#define index std parmseters and map
##indices.nz$stdparm = indices.nz$survey; ##self-weight by survey
indices.nz$stdparm = 'all'
std.name = unique(indices.nz$stdparm)
nstd = length(std.name)
indices.nz$stdmap = (1:nstd)[match(indices.nz$stdparm,std.name)]
```

R ADAPT: Step 4

```
#For now sort of cheat and take starting values from assessment

qfall = c(11,11,8,7,5,4,4,3,3)/10
qspring = c(10,14,7,5,3,3,3,3,4)/10
qjuvenile = c(35,18,13,11,8,6,5,3,3)/10
q.start = c(qfall,qspring,qjuvenile)
names(q.start)=q.name

## equal weighting all indices => common log error variance parameter
std.start=0.5

##set survivors
N_Y = c(5.4,1.7,0.7,0.7,0.3,1,1,0.4,1.9,1)
names(N_Y) = paste('N_2017,',3:12,sep='')
N_Aest = c(12,6,4,8,11,5,6,52,24,3,4,7,8,2,12,27,19,5,3,6,13,41,45)/100
Years_NAest = 1994:2016
names(N_Aest) = paste('N_',Years_NAest,',12',sep='')
pop.dat$NA.map = match(Years_NAest,pop.dat$Year)
pop.dat$Fave.map = match(6:9,pop.dat$Age)
```

R ADAPT: Step 5 – the model!

```
Npop = function(N_Y,N_Aest,pop.dat){
  A=pop.dat$A;  Y=pop.dat$Y;  C=pop.dat$catch;  M=pop.dat$M
  N = matrix(NA,nrow=Y,ncol=A);    F = matrix(NA,nrow=Y-1,ncol=A)
  last.age = 1:(A-1) ;  all.age = 1:A
  N[Y,2:A]=N_Y
  N[pop.dat$NA.map,A] = N_Aest

  for (y in seq(Y-1,1,-1)){
    for (i in seq_along(last.age)){
      N[y,i] <- N[y+1,i+1]*exp(M) + C[y,i]*exp(M/2)
      F[y,i] = -log(1 - exp(M/2)*C[y,i]/N[y,i]);
     }
    test = is.na(N[y,A])
    F[y,A] = ifelse(test,mean(F[y,pop.dat$Fave.map]),-log(1 - exp(M/2)*C[y,A]/N[y,A]))
    if(test){N[y,A] = C[y,A]*exp(M/2)/(1-exp(-F[y,A]))}
  }
  N[Y,1] = exp(mean(log(N[(Y-3):(Y-1),1])))
  B = pop.dat$wt*N
  SSB = pop.dat$mat*B
  pop = list(N=N,B=B,SSB=SSB,F=F)
  return(pop)
}
```

ADAPT computes N 1 year more than last year for catch/surveys

ADAPT has no information for this cohort – it is a projection

R ADAPT: Step 6 – some useful functions


```
create.parm = function(z){
  pname = names(z)
  ret = list(
      N_Y = exp(z[substr(pname,1,6)=="logN_Y"]),
      N_Aest=exp(z[substr(pname,1,6)=="logN_A"]),
      q=exp(z[substr(pname,1,4)=="logq"]),
      std_index=exp(z[substr(pname,1,9)=="std_index"]))
  return(ret)
}
```

```
resid = function(parms,pop.dat,indices.nz){
  x = create.parm(parms)
  N = Npop(x$N_Y,x$N_Aest,pop.dat)$N
  vecN = as.vector(N);
  Elog_index = log(x$q[indices.nz$qmap]) + log(vecN[indices.nz$imap])
  std_log_index = x$std_index[indices.nz$stdmap]
  resid = indices.nz$log.index - Elog_index
  s.resid = resid/std_log_index
  ret = data.frame(resid=resid,resid_std = s.resid)
  return(ret)
}
```

R ADAPT: Step 6 – some useful functions


```
pred = function(parms,pop.dat,indices.nz){
  x = create.parm(parms)
  N = Npop(x$N_Y,x$N_Aest,pop.dat)$N
  vecN = as.vector(N);
  Elog_index = log(x$q[indices.nz$qmap]) + log(vecN[indices.nz$imap])
  return(Elog_index)
}
```

The Fit Function

```
fit = function(parms,pop.dat,indices.nz){
  x = create.parm(parms)
  N = Npop(x$N_Y,x$N_Aest,pop.dat)$N
  vecN = as.vector(N);
  ## make sure years and ages assoc. with vecN are the same as wt.vec
  EI = log(x$q[indices.nz$qmap]) + log(vecN[indices.nz$imap])
  stdI = x$std_index[indices.nz$stdmap]
  nll = -sum(dnorm(indices.nz$log.index,EI,stdI,log=TRUE))
  return(nll)
}
```

R ADAPT: Step 7 – starting values


```
start.parms.list = list(
  logN_Y = log(N_Y),
  logN_Aest = log(N_Aest),
  logq = log(q.start),
  std_index = log(0.5)
)
start.parms = unlist(start.parms.list)
```

```
lower = list(
  logN_Y = rep(-10,length(start.parms.list$logN_Y)),
  logN_Aest = rep(-10,length(start.parms.list$logN_Aest)),
  logq = rep(-10,length(start.parms.list$logq)),
  std_index = log(0.1)
)
```

```
upper = list(
  logN_Y = rep(100,length(start.parms.list$logN_Y)),
  logN_Aest = rep(100,length(start.parms.list$logN_Aest)),
  logq = rep(100,length(start.parms.list$logq)),
  std_index = log(10)
)
```

```
lower=unlist(lower)
upper=unlist(upper)
```

R ADAPT: Step 7 – fitting


```
## check initial fit
fit(start.parms,pop.dat,indices.nz)
## check gradient
grad(fit,start.parms,,,,pop.dat,indices.nz)

## Do the estimatiom
adapt.fit <-
nlminb(start.parms,fit,,,pop.dat,indices.nz,lower=lower,upper=u
pper)
##get hessian
Hfit = hessian(fit,adapt.fit$par,,,pop.dat,indices.nz)
```

Not much to this?

$$H(\hat{\theta}) = \left.\frac{\partial^2 nll(\theta)}{\partial\theta\partial\theta^t}\right|_{\theta=\hat{\theta}}$$

Asymptotic (i.e. large n) approximation:

$$\hat{\theta} \sim N\{\theta, Cov = H^{-1}(\hat{\theta})\}$$

Standard errors are sqrt of diagonal

R ADAPT: Step 8 – output estimates and CI's

```
parms = data.frame(log_est=adapt.fit$par,
log_se = sqrt(diag(solve(Hfit))))
parms$est = exp(parms$log_est)
parms$se = parms$log_se*parms$est
parms$L = exp(parms$log_est - qnorm(0.975)*parms$log_se)
parms$U = exp(parms$log_est + qnorm(0.975)*parms$log_se)
```

```
stargazer(parms[,3:6], type = "html",  out="out.doc",align = TRUE,
 title="3LNO cod ADAPT", single.row=TRUE, summary=FALSE)
```

```
x = create.parm(adapt.fit$par)
pop = Npop(x$N_Y,x$N_Aest,pop.dat)
```

```
ind = match(4:6,pop.dat$Age)
aveF = apply(pop$F[,ind],1,mean)
ssb = apply(pop$SSB,1,sum)
```

R ADAPT: Step 9 – model fit object


```
fit = resid(adapt.fit$par,pop.dat,indices.nz)
fit$Age = indices.nz$Age
fit$Year = indices.nz$Year
fit$survey = indices.nz$survey
fit$pred = pred(adapt.fit$par,pop.dat,indices.nz)
fit$index = indices.nz$index
fit$Eindex = exp(fit$pred)
fit$log_index = indices.nz$log.index
fit$Elog_index = log(fit$Eindex)
```

My Help: an object to plot q's

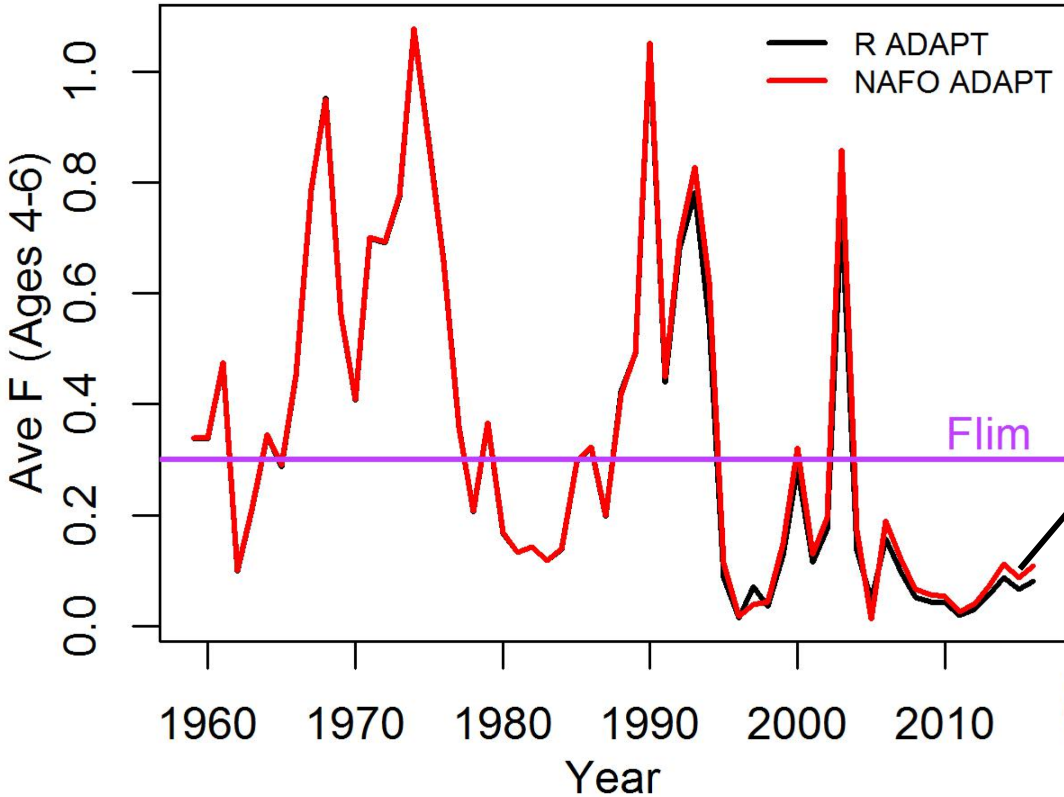
```
qdat = subset(parms,substr(rownames(parms),1,1)=='q')
temp = strsplit(rownames(qdat),"_")
qdat$Age=as.numeric(unlist(lapply(temp,function(x){x[2]})))
qdat$survey=gsub('q.','',unlist(lapply(temp,function(x){x[1]})))
```

R ADAPT: Output

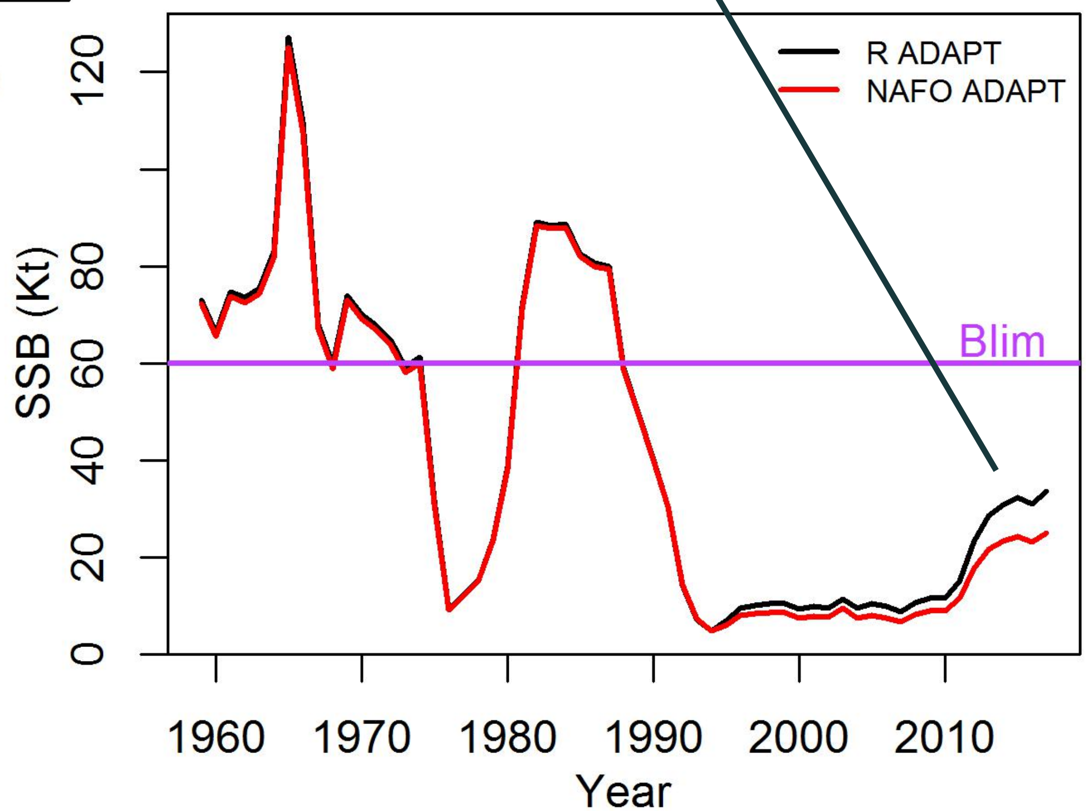
3LNO cod ADAPT

	est	se	L	U
N_Y,3	6.200	3.235	2.230	17.241
N_Y,4	1.987	0.764	0.935	4.221
N_Y,5	0.806	0.309	0.380	1.711
N_Y,6	0.977	0.327	0.507	1.884
N_Y,7	0.415	0.137	0.218	0.791
N_Y,8	1.224	0.333	0.719	2.085
N_Y,9	1.322	0.319	0.825	2.121
N_Y,10	0.525	0.123	0.332	0.831
N_Y,11	2.423	0.502	1.614	3.637
N_Y,12	1.238	0.256	0.825	1.856
N_A_1994	0.054	0.084	0.003	1.130
N_A_1995	0.058	0.032	0.020	0.169
N_A_1996	0.038	0.013	0.019	0.076
N_A_1997	0.083	0.027	0.045	0.156
N_A_1998	0.119	0.039	0.063	0.226
N_A_1999	0.060	0.021	0.030	0.117
N_A_2000	0.067	0.019	0.038	0.118
N_A_2001	0.618	0.169	0.362	1.056
N_A_2002	0.307	0.079	0.186	0.508
N_A_2003	0.042	0.011	0.026	0.070
N_A_2004	0.058	0.014	0.035	0.094
N_A_2005	0.099	0.026	0.060	0.166
N_A_2006	0.106	0.030	0.061	0.184
N_A_2007	0.022	0.007	0.012	0.040
N_A_2008	0.162	0.043	0.096	0.272
N_A_2009	0.347	0.095	0.202	0.594
N_A_2010	0.232	0.066	0.133	0.404
N_A_2011	0.064	0.018	0.037	0.111
N_A_2012	0.045	0.013	0.026	0.078
N_A_2013	0.079	0.019	0.049	0.128
N_A_2014	0.167	0.040	0.104	0.268
N_A_2015	0.530	0.119	0.342	0.822
N_A_2016	0.579	0.127	0.377	0.891
q.Fall_2	0.767	0.118	0.567	1.036
q.Fall_3	0.732	0.112	0.543	0.987
q.Fall_4	0.457	0.070	0.338	0.617
q.Fall_5	0.377	0.059	0.278	0.512
q.Fall_6	0.317	0.051	0.232	0.433
q.Fall_7	0.229	0.038	0.166	0.315
q.Fall_8	0.234	0.040	0.167	0.328
q.Fall_9	0.171	0.030	0.120	0.242
q.Fall_10	0.235	0.045	0.161	0.343
q.Spr_2	0.836	0.115	0.639	1.094
q.Spr_3	1.102	0.150	0.844	1.438
q.Spr_4	0.521	0.071	0.399	0.681
q.Spr_5	0.347	0.048	0.265	0.456
q.Spr_6	0.247	0.035	0.187	0.326
q.Spr_7	0.247	0.036	0.186	0.328
q.Spr_8	0.244	0.036	0.183	0.327
q.Spr_9	0.242	0.037	0.180	0.326
q.Spr_10	0.308	0.048	0.227	0.418
q.Juv_2	2.713	0.806	1.516	4.856
q.Juv_3	1.242	0.367	0.696	2.216
q.Juv_4	0.735	0.217	0.412	1.312
q.Juv_5	0.564	0.167	0.316	1.009
q.Juv_6	0.427	0.127	0.238	0.764
q.Juv_7	0.332	0.100	0.185	0.599
q.Juv_8	0.293	0.090	0.161	0.533
q.Juv_9	0.195	0.061	0.106	0.359
q.Juv_10	0.181	0.059	0.095	0.343
std_index	0.720	0.021	0.679	0.763

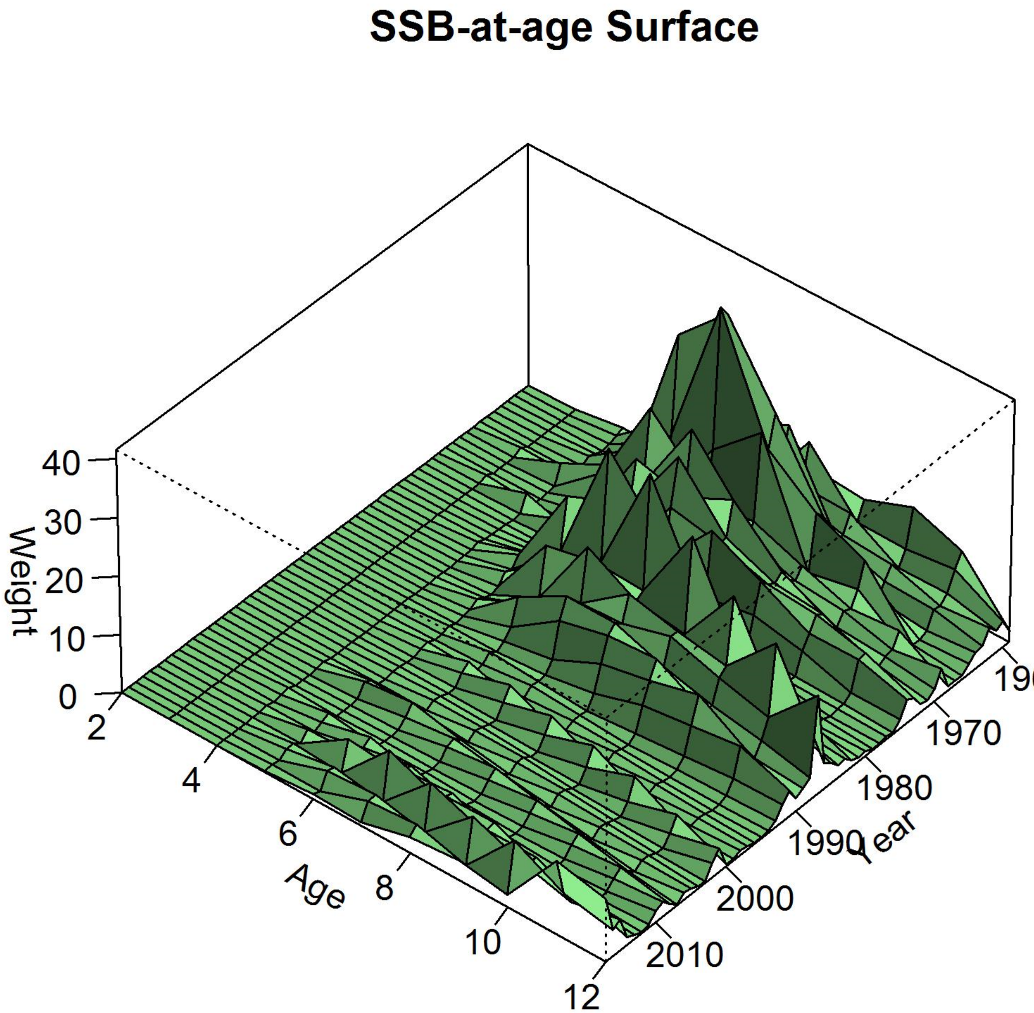
R ADAPT: Output

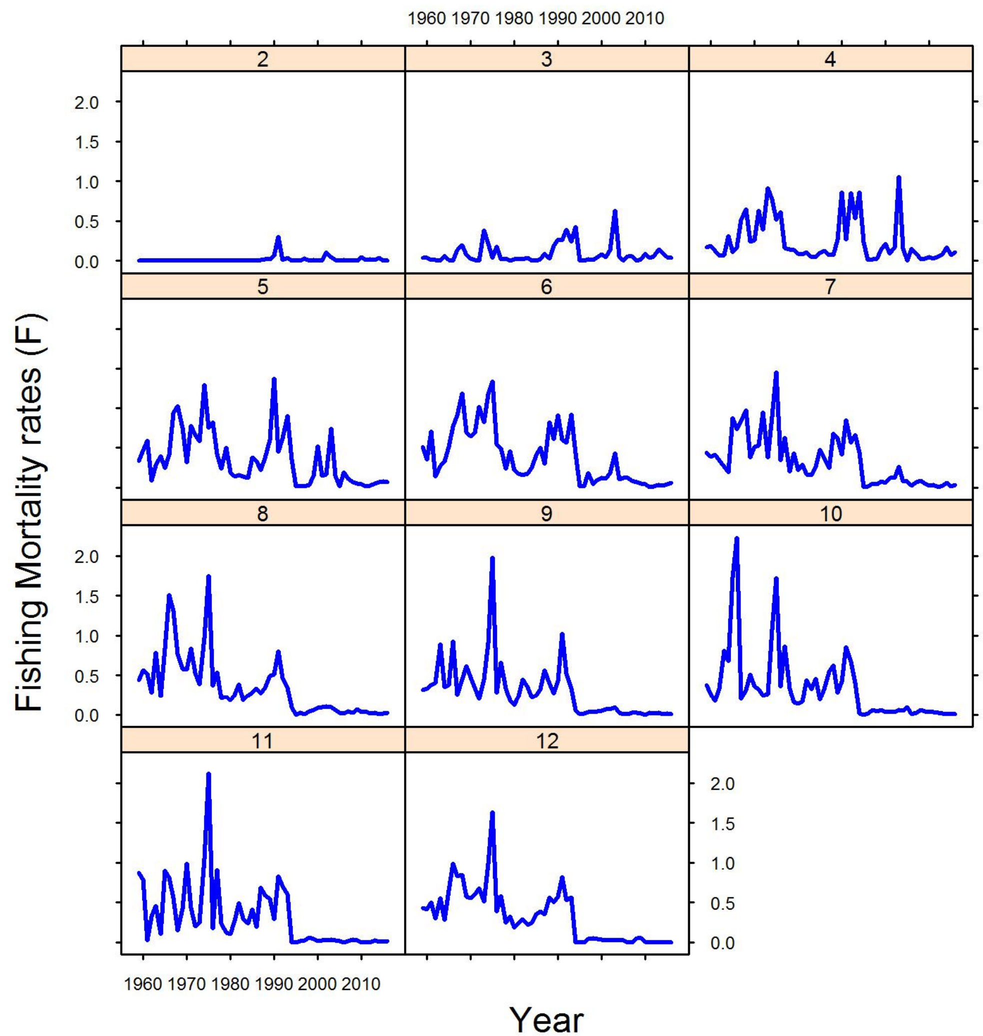


Difference mostly due to DFO ADAPT bias correction

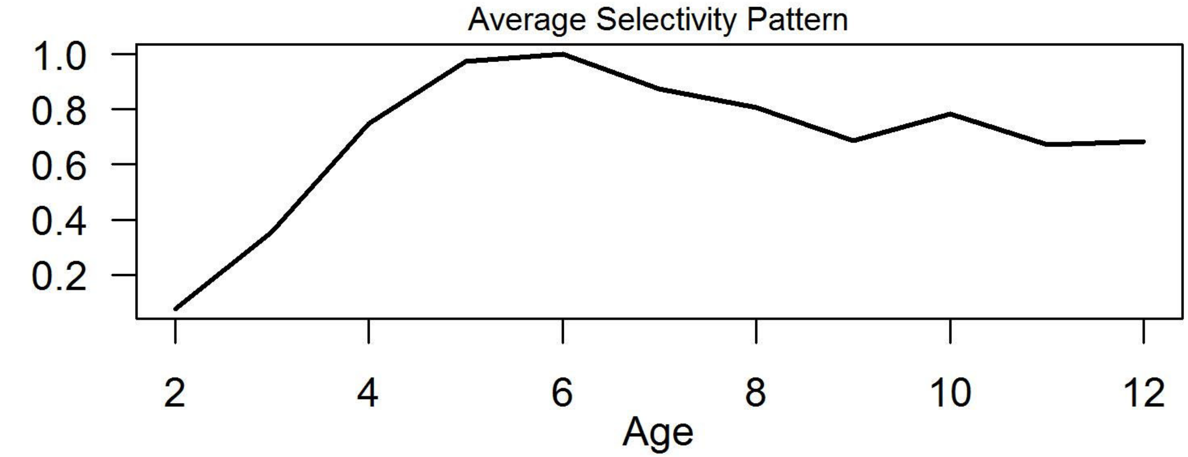


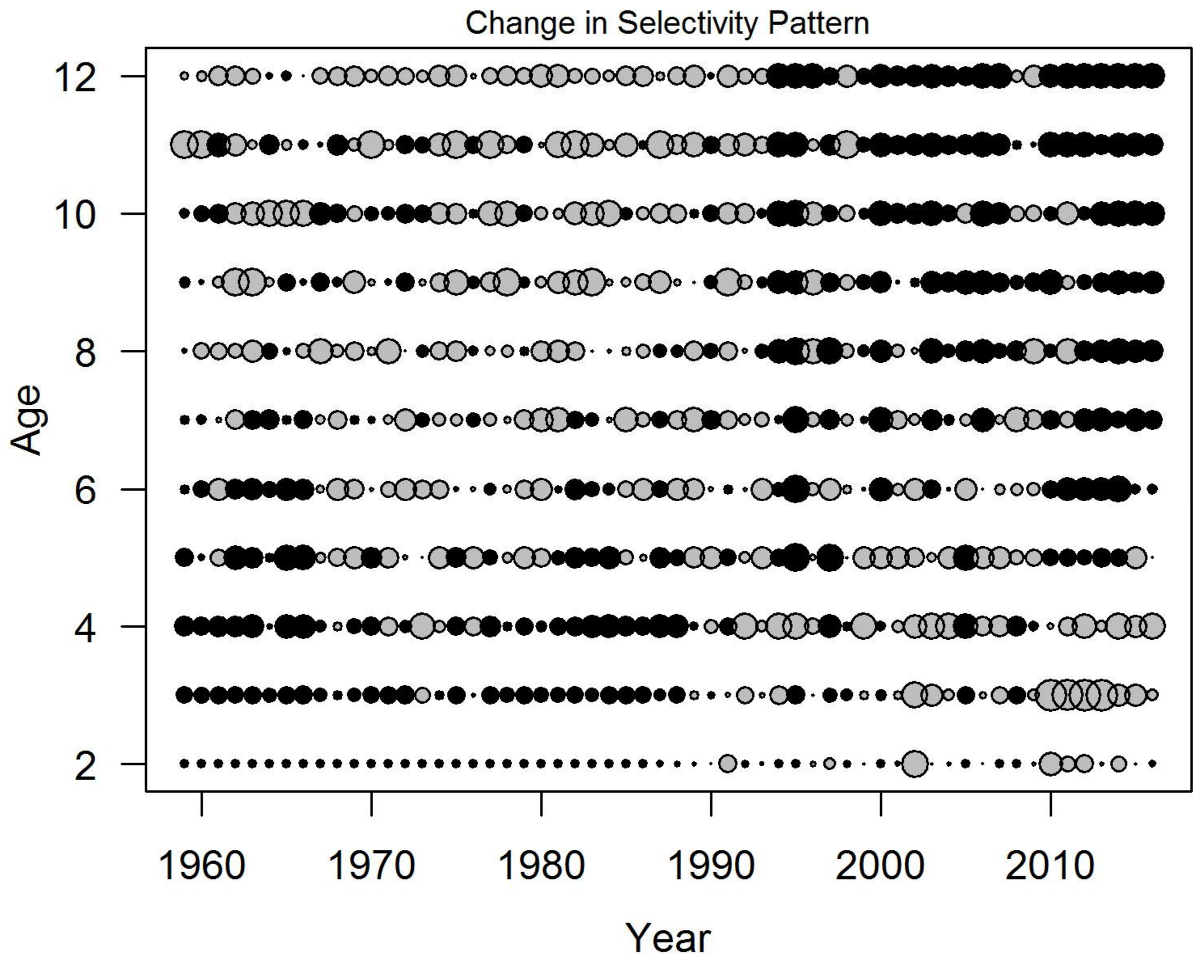
R ADAPT: Output



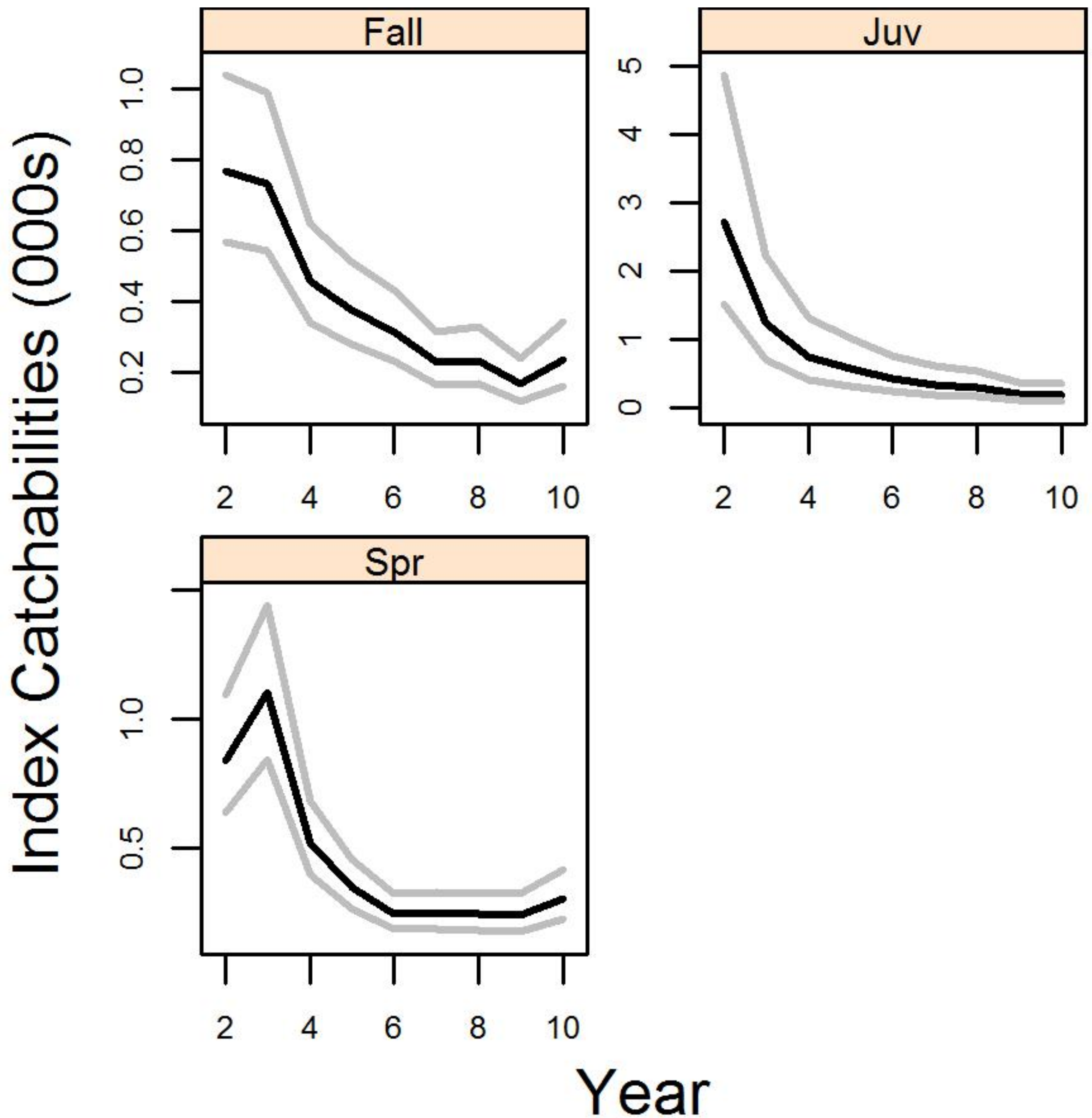


R ADAPT: F Age Pattern (selectivity)

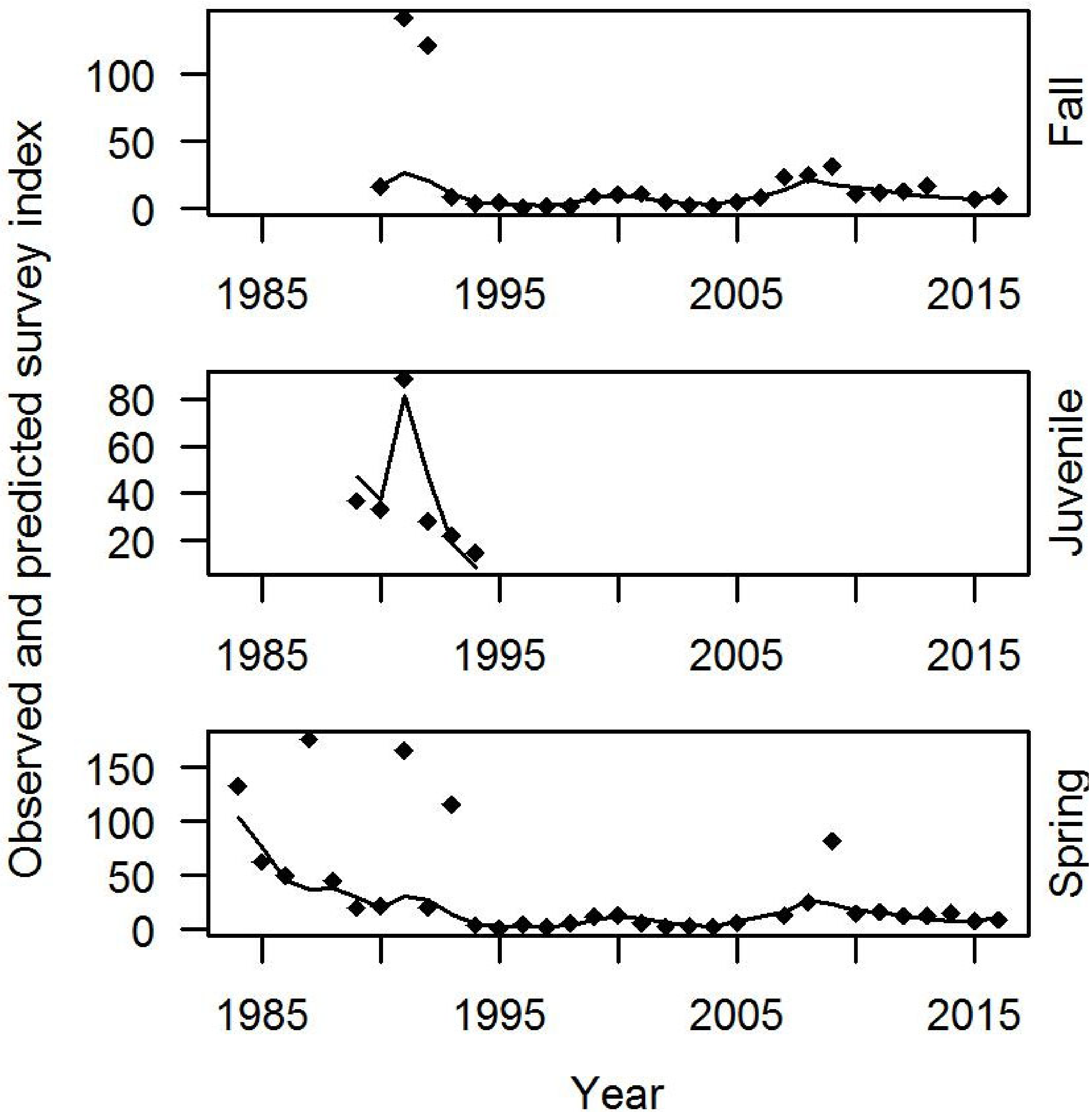




R ADAPT: Survey Index Catchability

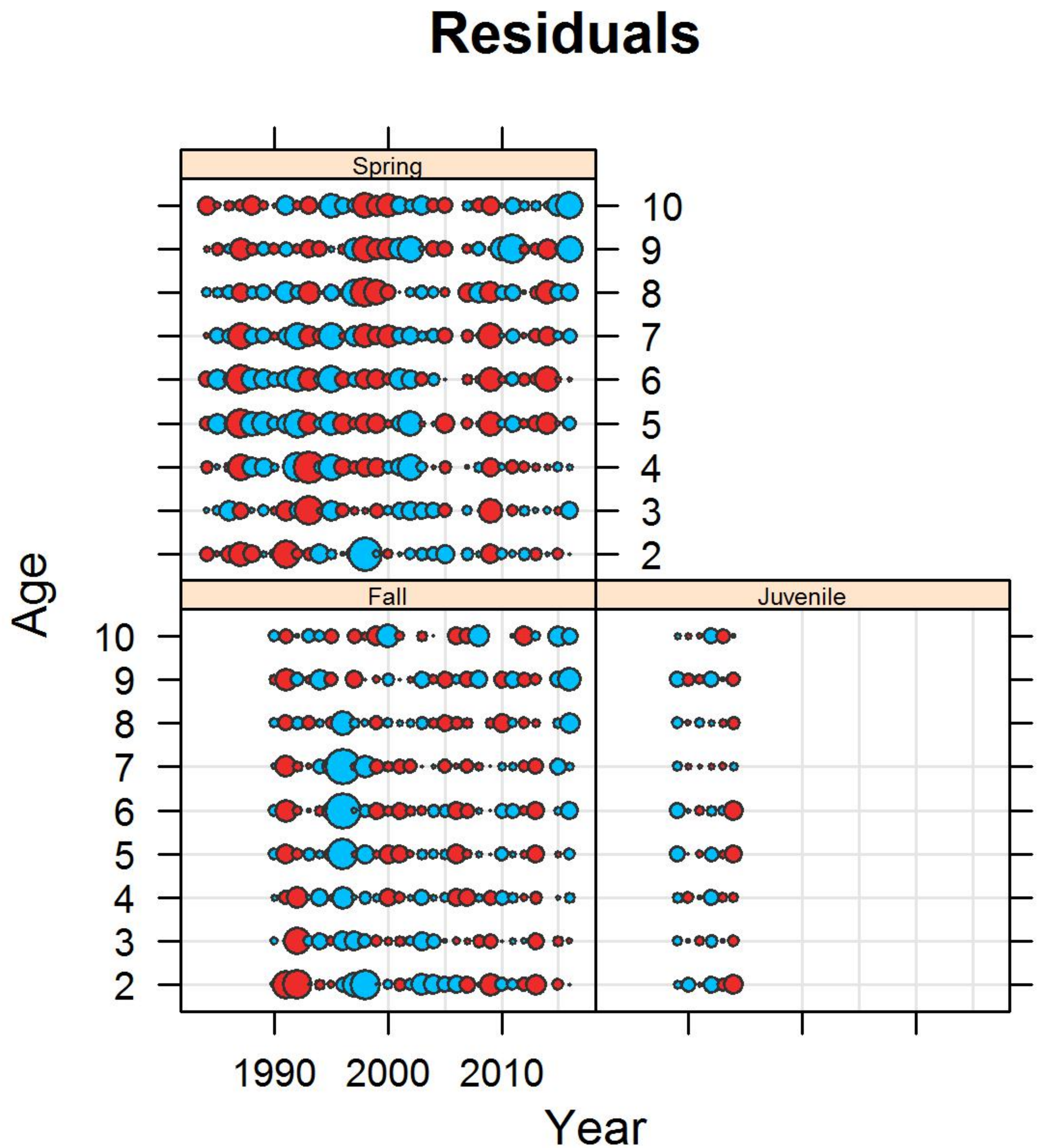


R ADAPT: Survey Index Age-aggregated "Fits"

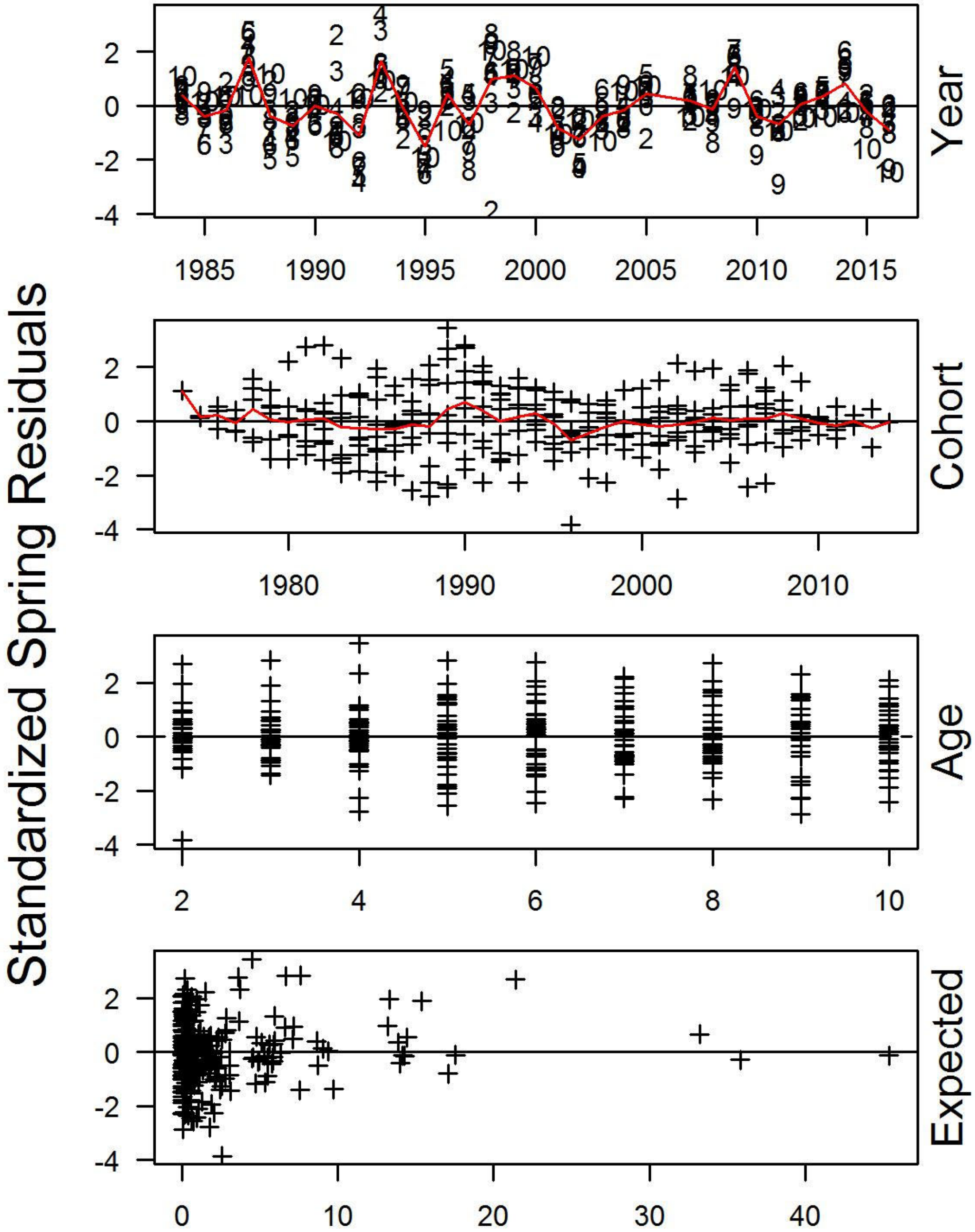


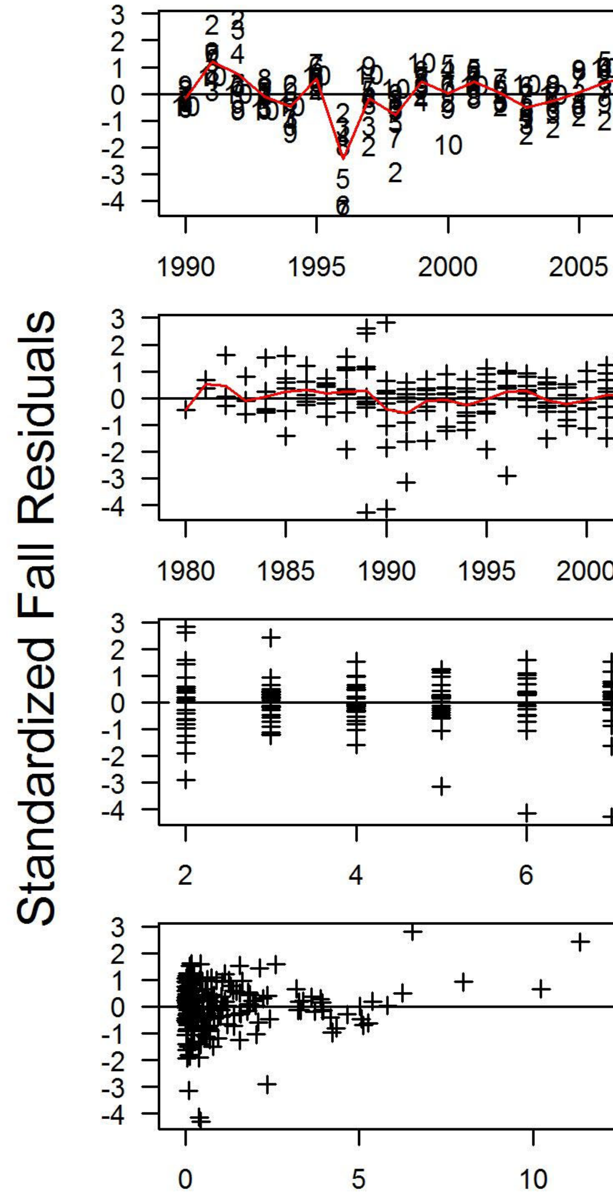
We don't fit age-aggregated indices – this is just a summary of age-specific fits

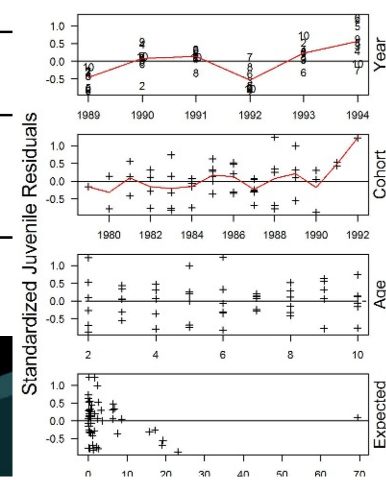
R ADAPT: Survey Index Fit



R ADAPT: Survey Index Fit

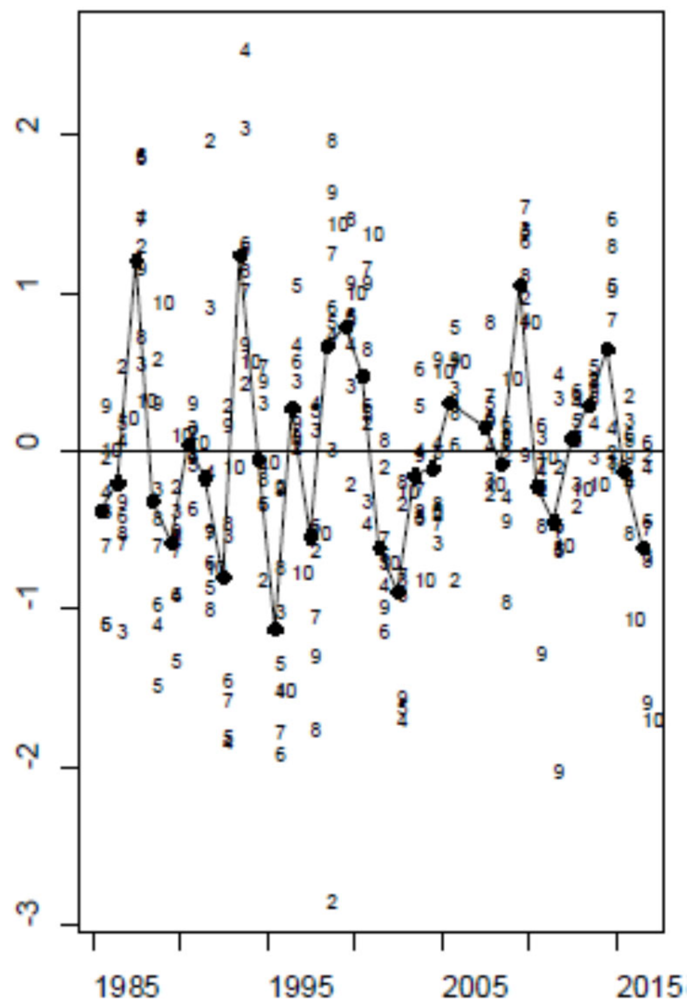




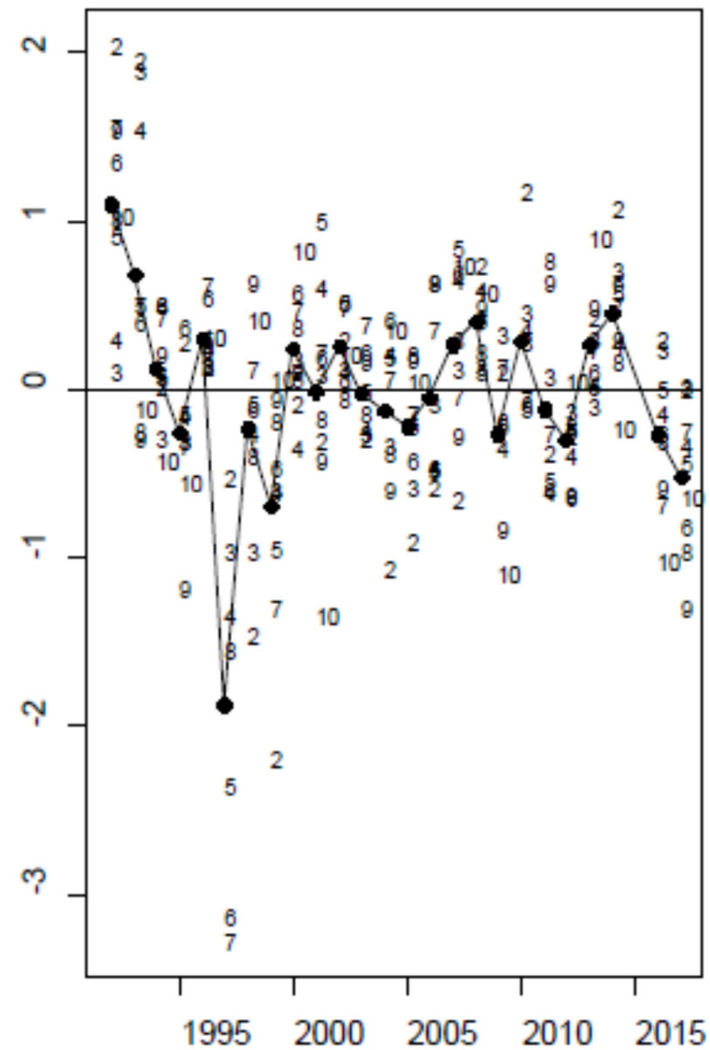


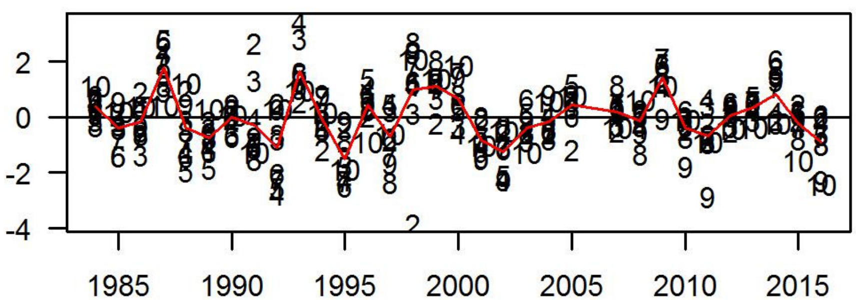
R ADAPT: Survey Index Fit

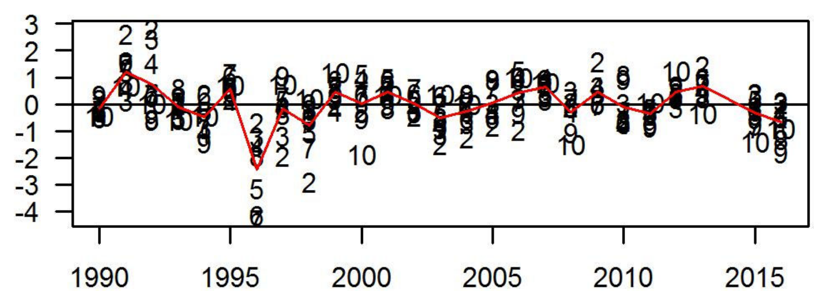
Cdn.Spr



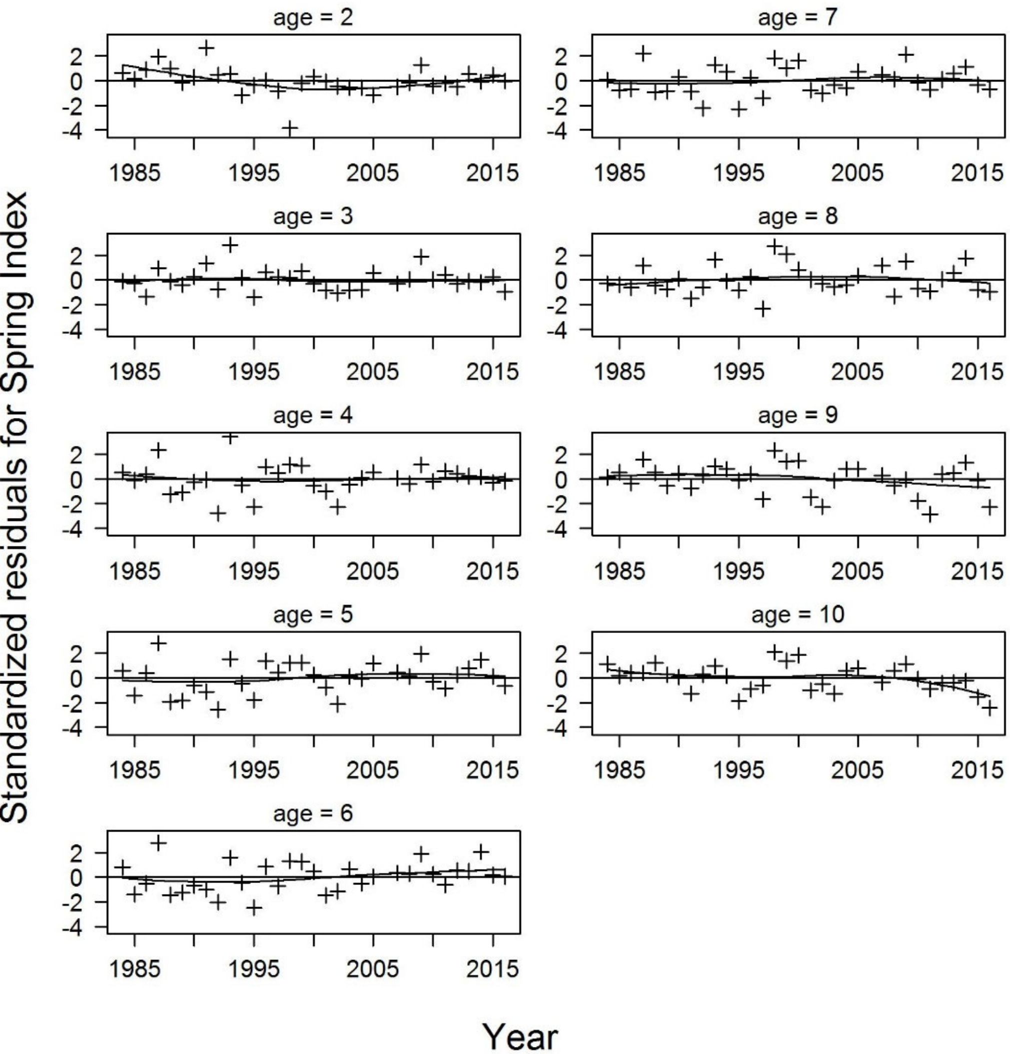
Cdn.Aut

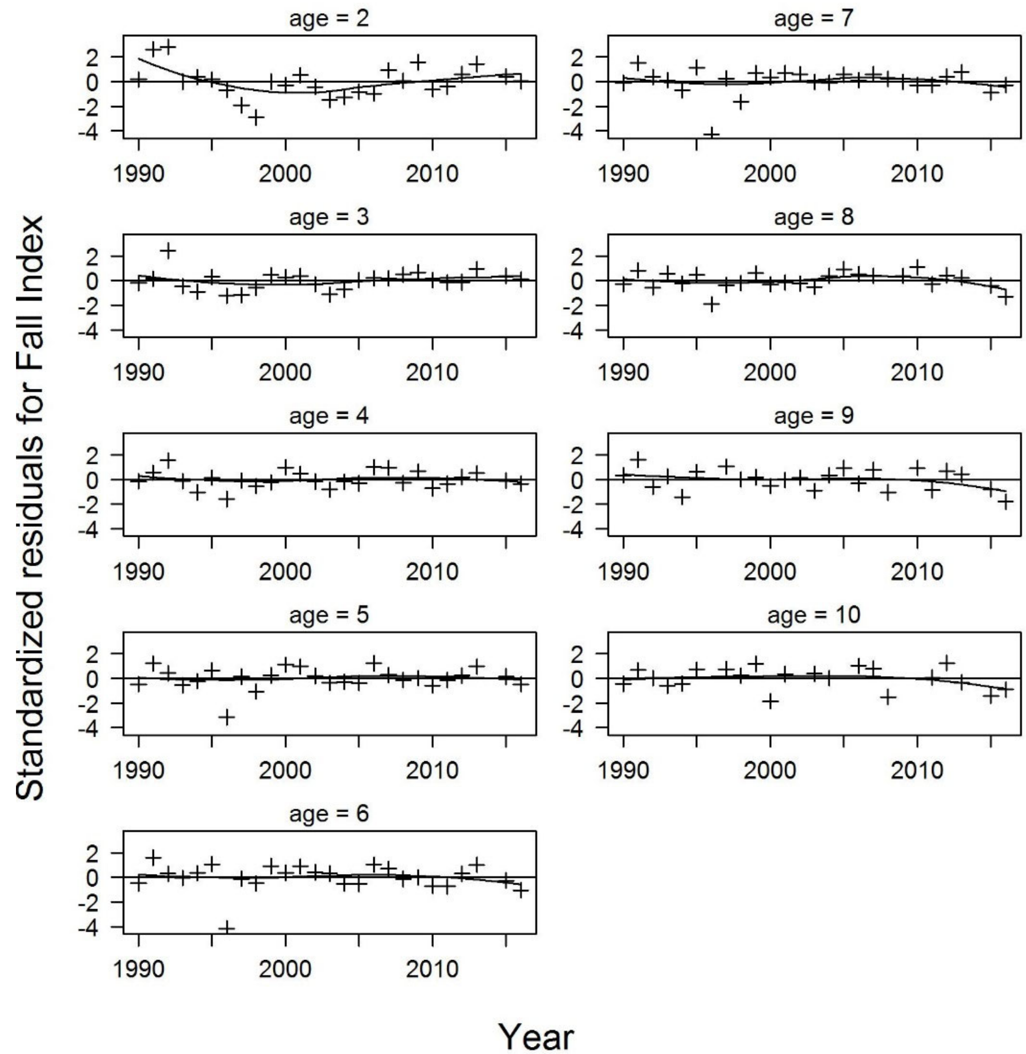






R ADAPT: Survey Index Fit





Statistical Inferences

- Confidence intervals for parameter estimators were traditionally derived from asymptotic normal distribution results based on the hessian of the “lognormal” fit function.

- Confidence intervals for derived results (e.g. SSB) were based on the delta method, or a numerical simulation method based on a normal approximation for parameter estimators.

- More recently the nonparametric bias-corrected percentile bootstrap method has been used for confidence intervals and hypothesis tests.

Statistical Inference Problems

- Large y asymptotic approaches for SPA inferences, such as the delta method, may not be good because the relevant sample size for inferences about survivors never gets very large.

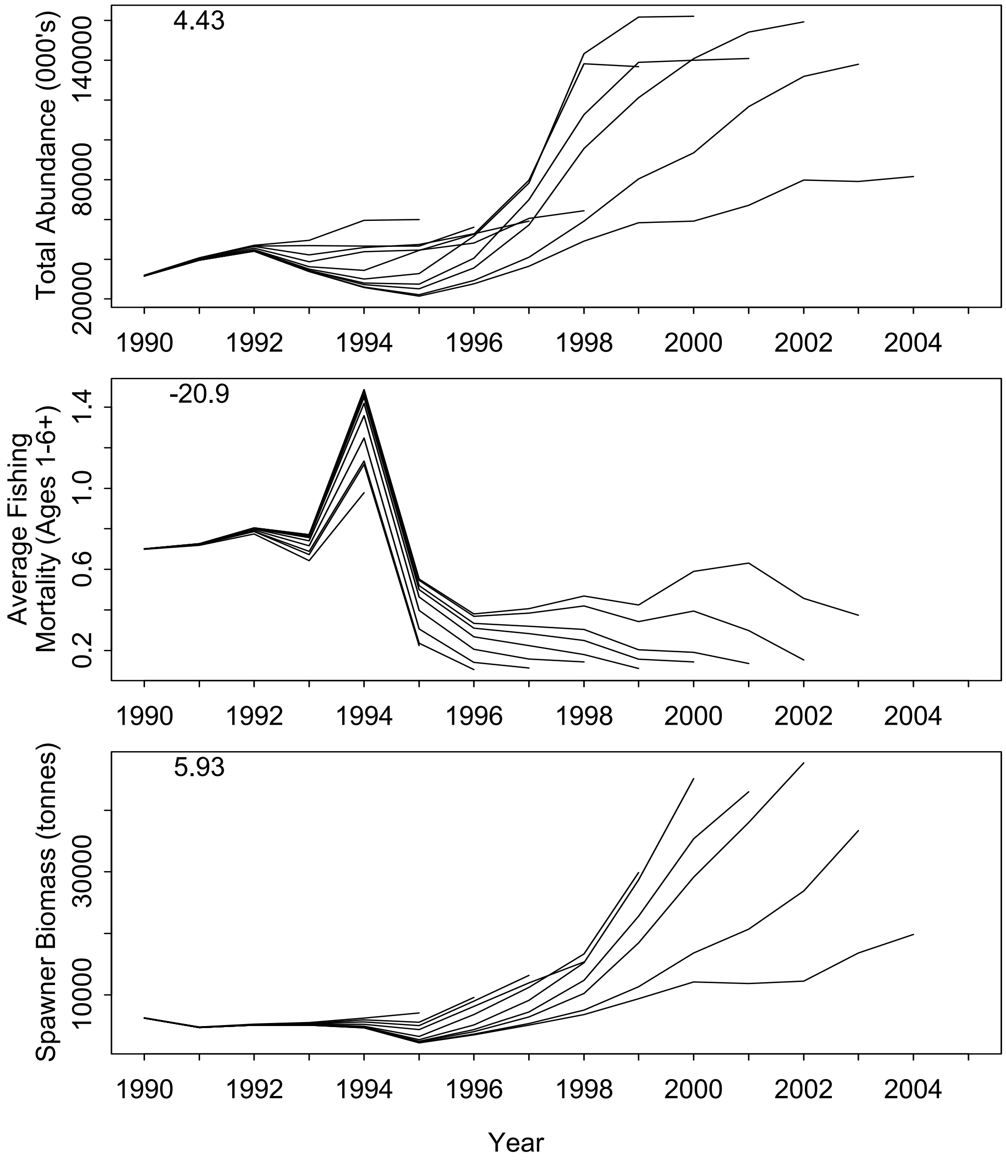
- This problem may also affect the bootstrap?

- State-space models may be better for statistical inferences because they have fewer parameters, but they do have random effects

Other Inference Problems

- Inaccurate reported catches.
- M not 0.2, and not constant over time of age.
- q_a changes with time.
- Maturities and weights are sampled, and have noise.
- SPA does not use all available information.

Retrospective Problems



The retrospective problem involves systematic differences in SPA estimates of stock size.

The differences occur as successively more data are used for estimation, and they appear to be structural biases caused by a mis-specification of the SPA.

In some cases the retrospective problem is so severe that the SPA is considered to be too unreliable for stock assessment purposes.