Sensitivity Analysis of a State Space Fish Stock Assessment Model Compared to Conventional Approaches

by

© **Prageeth Senadeera**

A practicum submitted to the School of Gradate Studies in partial fulfillment of the requirements for the degree of Master of Applied Statistics.

Department of Mathematics and Statistics

Memorial University

January 2019

St. John's, Newfoundland and Labrador, Canada

Abstract

Surplus production models provide simple analytical methods of assessing fish populations by taking the annual biomass, the growth rate and the carrying capacity into account. However, these simple models may not adequately reflect fish stock dynamics that can be substantially more complex with age and length specific birth, growth, and death processes at play. To account for this, process errors can be included in the production model in a state-space modelling framework, which is used frequently in ecological modelling in recent years. In this study, we compare the sensitivity of estimators of state-space and conventional non-linear production models (without process errors) using both the traditional case deletion diagnostic method and the local influence analysis method introduced by R.D. Cook, 1986 [12]. We apply these diagnostics to different fish stocks to assess how estimated parameters respond to small perturbations of the data.

To my loving wife

Lay summary

Fishing supports billions of people worldwide and it is important to catch fish in a way to promote the long-term survival of fish and the people who rely on them. Overfishing may cause a reduction of the amount of fish available and eventually there will be not much fish left to catch in the future. Therefore, management agencies provide guidelines on how much fish can sustainably be caught. Scientists use customized mathematical and statistical models and fit them to the data available to decide on sustainable harvest strategies. In this research, we examine models commonly used to provide harvest advice. We focus on the traditional way of estimating models and also a more modern state-space approach. We examine how the critical outputs from each approach change when we make minor changes to the inputs of the models. We compare the sensitivity of the models using two methods for five real data sets. Results from our research show that for some data the state-space approach shows less sensitivity than the traditional method.

Acknowledgements

First, I would like to express my sincere gratitude to my supervisor Dr. Noel Cadigan for his expert advice and extraordinary support throughout my study. Without his helpful guidance, valuable suggestions and generous financial support during my program, this practicum would not have been possible. I'm especially grateful for the time he spent reading my work and helping me to improve upon it. I sincerely acknowledge the financial support provided by the School of Graduate Studies, the Department of Mathematics & Statistics, and Center for Fisheries Ecosystems Research (CFER) at the Marine Institute, the Memorial University of Newfoundland in the forms of Graduate Assistantship, Teaching Assistants. I would like to thank all the faculty and staff at the Department of Mathematics & Statistics especially Prof. J C. Loredo-Osti, and Prof. Zhaozhi Fan.

My deepest gratitude goes to my parents and my brother for their love and unconditional support throughout my life and my studies. I must express my very profound gratitude to my wife for providing me with support and encouragement throughout my years of study and through the process of researching and writing this thesis. Finally, a special thank goes to my son who sacrificed a lot of his childhood time giving me space to study. This accomplishment would not have been possible without their help. Thank you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Stock assessments play a vital role in fisheries science and management. A stock is a subpopulation that is reproductively isolated, and in which immigration/emigration only play a minor role in stock productivity. Information obtained from stock assessments helps fisheries management agencies make decisions and regulations to maintain a sustainable and profitable fishing industry [23], [21]. With stock assessments, fisheries scientists try to build the most precise model to fit the data they have. These models range from simple to very complex, depending on the type of data used. In this study, we investigate Surplus Production Models (SPMs). They are a simple and widely used stock assessment model. To fit SPMs and estimate model parameters, we only need a time-series of the total catch each year and an index which reflects the total weight of the fish population (biomass).

Over the last couple of decades, the state space framework has increasingly been used to fit SPMs. These State Space Models (SSMs) are becoming popular among fisheries scientists because they can account for both the measurement error associated with the data and the process error associated with the model of the population dynamic [5]. Since both conventional and state space versions of SPMs are available, it is useful to identify a better model to apply for a stock assessment. One consideration when deciding which model to use is robustness. An aspect of robustness we study is sensitivity to changes made to input data. Therefore, we investigate the sensitivity of some commonly used SPMs by making changes to input data and

examining the effects of these changes on important model outputs. We first apply the traditional case deletion diagnostic method to survey indices, analyze them for influential cases and compare diagnostic results for both contemporary and state-space SPMs. We also use the case weight local influence (CWLI) method (Cook, 1986 [12]) to compare diagnostic results for CSPM's and SSPM's. We also compare these CWLI results with the case deletion results to better understand the correspondence between the methods. Finally, we apply the local influence technique to find influential observations in the annual catch data.

1.1 Fish stock assessment models

Humans have been fishing as a way of life for thousands of years [19]. As an industry, it provides the direct livelihood for over 250 million people worldwide and more than one billion people supply their animal protein needs from fish [2]. However, human activities are increasingly disturbing delicate marine ecosystems, reducing the size of some fish stocks. Overfishing is one of the main reasons for the observed changes and decades of over-exploitation of fish stocks have caused a depletion in the catch over the years [11]. Therefore, sustainable fishing methods were introduced by fisheries management organizations over the past decades, which require reliable and accurate information about fish stocks to make decisions and regulations. Fisheries scientists help them to this end by providing the information through conducting assessments.

Simply, "stock assessments involve understanding the dynamics of fisheries" (Hilborn et al., 1992) [23]. These assessments consist of multiple steps: creating a meaningful database, analyzing these data with proper models, projecting short or long-term fish stock size and fishery catches, determining long-term stock targets and estimating the short and long-term effects of different harvest strategies on stock size and fishery catches [29].

Scientists use multiple methods when gathering information about a particular type of fish stock. They collect information using research ships as well as commercial fishing vessels. These data include biological (age and length of fish) and commercial data (total landing and catch per effort) [29]. Fisheries scientists then use this

information to build a database and analyze the data using customized statistical models. These models provide a simplification of a very complex fishery system to help estimate population changes over time in response to fishing and to predict future growth in response to management actions.

Fitting the most precise model with the data available is extremely challenging because, in most situations, the underlying data are noisy and have substantial measurement errors. We can construct models that fit our data very closely by adding many parameters; however, that does not mean the model will give good predictions of the fishery system. We want parsimonious models that fit the data well in light of the measurement errors in the data, but they should not over-fit the data and capture data measurement errors as part of the fishery system. The models used in stock assessment are relatively simple ones and cannot synthesize all the true processes that drive fish stock dynamics over time.

Production models and structural models are the two primary types of modelling techniques used in fish stock assessments. Which technique to use depends on the availability of data. To fit production models (also known as biomass dynamic models or surplus production models), a time-series of the total catch each year and an index of relative stock abundance are sufficient. We can use structural models if biological data are also available for the fish stock, which typically consist of the age and size composition of the stock. Therefore, structural models generally are more complex than production models. However, in this study, we only focus on production models, further discussed in Chapter 4.

SPMs contain two submodels: the process model which describes how the population biomass changes over time, and the observation model that explains how an abundance of index observations relates to biomass predictions of the model [42]. The basic process model is,

$$B_{t+1} = B_t + f(B_t) - C_t, \tag{1.1}$$

where B_t is the biomass (e.g weight in tonnes) of the fish stock at the starting period t, $f(B_t)$ is the surplus production as a function of biomass (we discuss this in detail

in Section 4.1), and C_t is the fishing catch in period t. Basically, in SPMs the current biomass is the addition of new fish to the previous time periods' population biomass minus the amount of fish removed from the population due to natural mortality and fishing.

Since we do not have a way to estimate the population biomass directly, we use the following observation model to estimate parameters of the model in 1.1,

$$I_t = qB_t, \tag{1.2}$$

where I_t is an index of abundance and q is the catchability coefficient, which is thought of as a measure of availability of a fish stock to the fishing or survey process used to generate the indices [18]. We also assume that C_t/E_t is an index of abundance, such that,

$$I_t = \frac{C_t}{E_t} = qB_t, \tag{1.3}$$

where E_t is the fishing effort producing the catch in period t and C_t and q are the same as described earlier. In this model, we have the assumption that catch rates are linearly related to stock biomass.

A simple difference equation for the process model 1.1 is [23],

$$B_{t+1} = B_t + rB_t(1 - B_t/K) \ - C_t, \tag{1.4}$$

where r is the population growth rate, K is the carrying capacity and $C_t,\ B_t$ are as described earlier. In this model we assume a constant population growth rate (r), and a carrying capacity K (discussed in detail in Chapter 4), and that the population is closed (no immigration and emigration). To estimate model parameters, we consider the errors (ϵ_t) associated with the observation model. These errors are normally distributed with mean 0 and variance σ^2,

$$I_t = qB_t \exp(\epsilon_t), \tag{1.5}$$

where $\epsilon_t \overset{iid}{\sim} N(0, \sigma^2)$. We use maximum likelihood estimation to estimate unknown

parameters $r,\ K,\ q,\ \sigma$ and the initial biomass, B_0, and the estimation method is discussed in detail in Section 4.2.1. Here we assume that the random errors are only associated with observational model 1.3 and that the process model (state equation) 1.1 is deterministic. These random errors are also called observation errors and they are made while collecting data. These types of models are called observation error models and they are commonly used in fisheries population modelling [31], [23], [28].

In reality the process model 1.1 also consists of errors because of the variability in recruitment (the amount of fish added to the exploitable stock each year due to growth [1]), and natural mortality. Therefore, some fisheries scientists (e.g. Breen, 1991 [6]) have considered process-error models and they assumed that the random errors occur only in the state-equation and that observations are deterministic to given states. The process model with errors is,

$$B_{t+1} = \{B_t + rB_t(1 - B_t/K)\ - C_t\} \exp(\gamma_t), \tag{1.6}$$

where γ_t are the error and $\gamma_t \overset{iid}{\sim} N(0, \sigma^2_\gamma)$.

Many scientists prefer the observation error method over process error method if they have to use a single method. Polacheck et al., 1993 [31] noted that observation error estimators are superior to process-error estimators when analyzing real data sets because process-error estimates have higher variance than of observation error estimates. They also suggested to use the process-error approach in simulation studies since process-error estimators gave less variable estimates for the parameters in their simulation studies [31].

However, these assessment models may be unreliable to study the dynamics of fish populations since they only use one error structure (observation error or process error). Hence, ecologists have more recently preferred to use state space models rather than conventional models to address this issue. A better understanding about the dynamics of a fish population can be obtained from these state space models (e.g. [37], [35], and [26]). SSMs can describe changes in the unobservable states of the population biomass and how the observed data relate to the unobservable states [30]. Also, SSMs allow scientists to model both the variations in population dynamics and

observation models. For example, we can use the observation model in Eq. 1.3 and the process model in Eq. 1.1 to fit the state space surplus production model. We discuss state space models in detail in Chapter 3 and the estimation of state space surplus production models in Chapter 4.

With the introduction of software **ADMB** by Fournier et al., 2012 [17] and R package **TMB** by Kristensen et al., 2015 [27], the use of SSMs has increased substantially in fisheries stock assessments [3]. These packages use the Laplace approximation to obtain the marginal likelihood of fixed effects parameters when random process errors and other random effects are integrated out of the model joint likelihood (see Section 3.2).

1.2 Outline of the thesis

In Chapter 2, we discuss the most commonly used influence measures in statistics. We give illustrative examples for most of these measures and compare some of their results. We also discuss the local influence method introduced by Cook, 1986 [12] and an extension of this approach (Cadigan and Farrell)[8].

In Chapter 3, we give a brief introduction to the state space framework and we discuss the open source R package, the Template Model Builder (TMB), which has been designed to estimate nonlinear models containing random effects. Using a simple linear regression example, we illustrate the steps of using this package. In Chapter 4, we discuss both contemporary and state space SPMs and their estimation in detail, using the TMB package.

In Chapter 5, we present the diagnostics and results of our analysis. For this study, we use real data obtained from five different fish stock assessments for redfish, Greenland halibut, megrim, yellowtail flounder, and anglerfish. We first compare traditional case deletion diagnostics results with case weight local influence (CWLI) diagnostics results for selected parameters. We then compare CWLI diagnostics results with SSMs and contemporary SPMs. Finally, we compare local influence diagnostic results for catch observations. In Chapter 6, we give the summary of the study.

Chapter 2

Sensitivity

2.1 Analyzing the sensitivity of a model

There are many models available for stock assessments, and we should always examine their accuracy as well as reliability using sensitivity analysis. A sensible approach in fisheries studies is to investigate how important model outputs react when we modify the inputs of the model. Usually, there are critical parameter estimates or functions of parameter estimates that are considered important. If these critical results change significantly when inputs are changed, then we need to be cautious when interpreting those models results. Better model formulations could be considered that provide more robust and hopefully, reliable estimates.

We first illustrate influence concepts using the familiar multiple linear regression model,

$$Y = X\beta + \epsilon, \tag{2.1}$$

where Y is the $n \times 1$ response vector, X is the $n \times p$ covariate matrix, β is the $p \times 1$ parameter vector and ϵ is the $n \times 1$ error vector with elements $\epsilon_i : i = 1, ..., n$ and the ϵ_i's are assumed to be independent and normally distributed with mean zero and known variance σ^2. There are numerous ways of measuring the influence on both estimated parameters ($\hat{\beta}$) as well as predicted values ($\hat{y}$) for a linear model

like this. Chatterjee and Hadi, 1986 [10] classified these measures into five groups: residuals, prediction matrix, the volume of confidence ellipsoids, influence functions, and partial influence. Before considering these measures in detail, we need first to discuss the concepts of outliers and leverage points. For illustration purposes, we created a hypothetical data set using the linear model $Y = 1 + 10X + e$, where $e \sim N(0, 5)$.

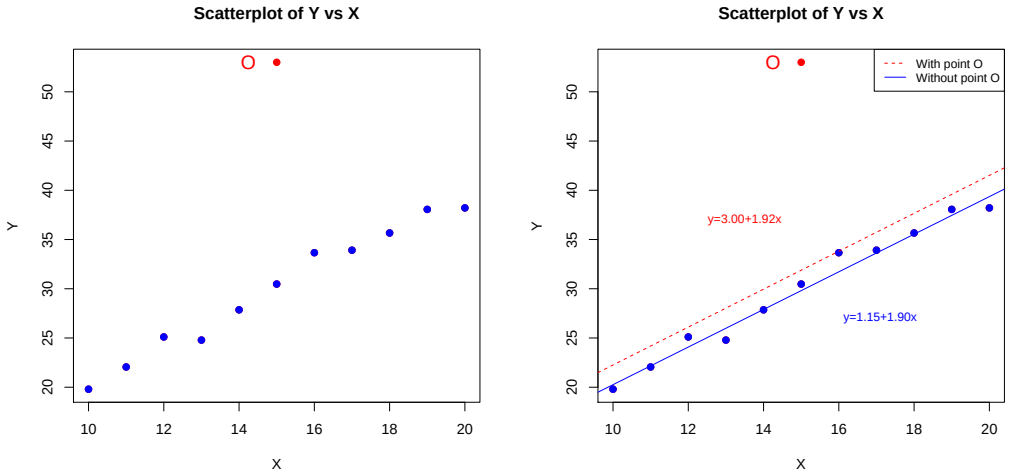
2.1.1 Distinction Between Outliers and High Leverage Observations

a. Outliers

If an observation does not follow the general trend in the data, we call it an outlier. In some situations, we can identify such observations easily by plotting data. To illustrate this, we added the point "O" to the hypothetical data set and plotted the data in Figure 2.1 The point "O" is an outlier because it lies significantly away from the rest of data. It is a data point that deviates from the general trend of the data. We can also use numerical methods to identify outliers in our data, and we will discuss a couple of these methods in the next section.

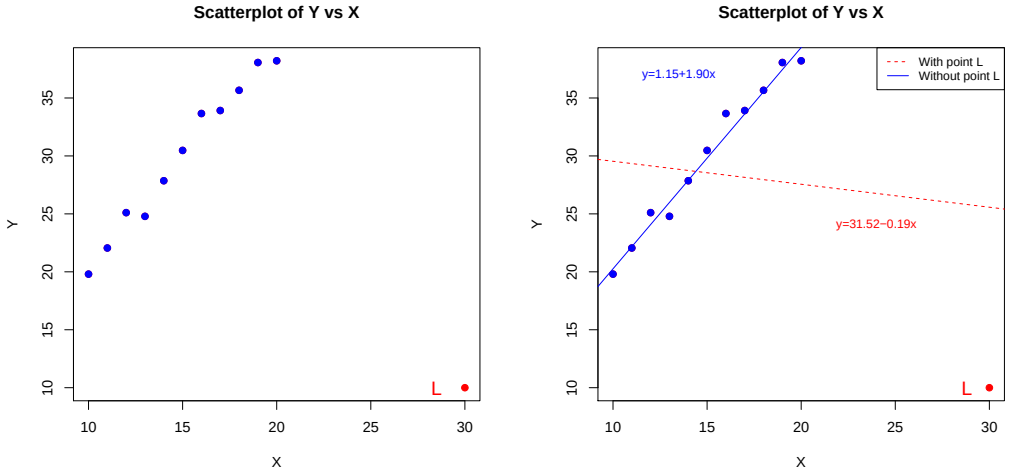
b. Leverage points

If a data point lies far away from the rest of the data along the X-axis (or has an extreme x value), it can be considered a leverage point. These points can affect the slope of the regression line by dragging it towards them from the mass of the data. In Figure 2.2, we added the point "L" to the data set; it is a leverage point because it forces the regression line to tilt towards itself. The plot (b) in Figure 2.2 shows the change in the regression line with and without "L".



(a) (b)

Figure 2.1: Illustration of an outlier. Left: Scatter plot of the values of Y versus the corresponding values of X. Right: Best fitted regression lines with (red-dotted line) and without (blue line) the point "O".



(a) (b)

Figure 2.2: Illustration for a leverage point. Left: Scatter plot of the values of Y versus the corresponding values of X. Right: Best fitted regression lines with (red-dotted line) and without (blue line) the point "L".

2.1.2 Traditional measures of identifying influential points in linear regression

Next, we discuss the influence measures described in Chatterjee and Hadi, 1986 [10].

1. **Residuals**

   This is one of the earliest methods for detecting a data anomaly in a model. If an observation has a much larger residual value than the rest of data, we can think of that observation as an outlier. The least squares residual for the i^{th} observation is,

$$e_i = y_i - x_i\hat{\beta}, \tag{2.2}$$

   where $\hat{\beta}$ is the least squares estimate for β and e_i is the difference between the i^{th} response and its model prediction. Since e_i values highly depend on the unit of measurement, in practice we use "studentized residuals" (t_i's),

$$t_i = \frac{e_i}{\sqrt{MSE(1-p_i)}},\ and \tag{2.3}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{2.4}$$

   where MSE is the mean squared error, and p_i is the i^{th} diagonal element of the prediction matrix (P)

$$P = X(X^TX)^{-1}X^T. \tag{2.5}$$

   In addition to Eq. 2.3, we also use a scaled version of e_i in Eq. 2.2, the

"standardized residuals" (e^*),

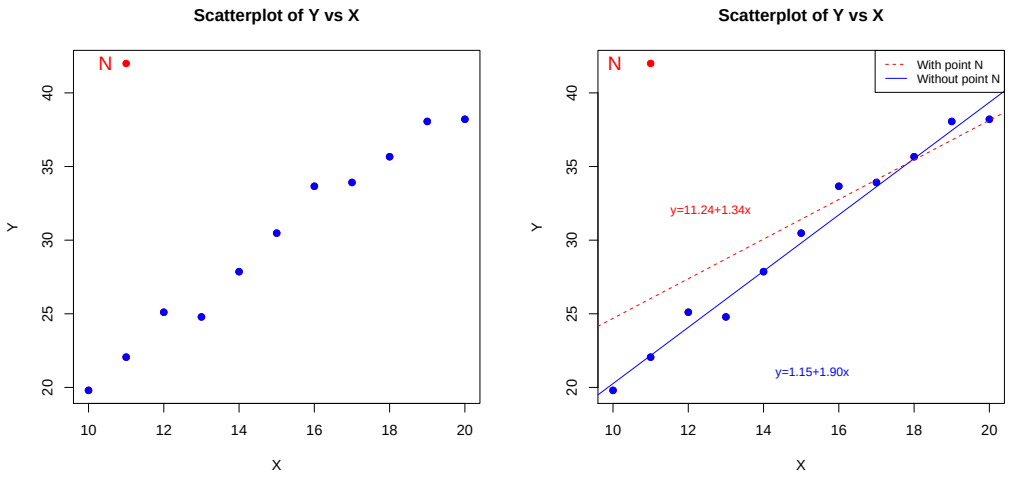
$$e^* = \frac{e_i}{\sqrt{\hat{\sigma}^2(1-p_i)}} \tag{2.6}$$

where $\hat{\sigma}^2$ is the mean squared residual estimate of $var(\epsilon_i) = \sigma^2$ and e_i, p_i are as described earlier. We can find $\hat{\sigma}^2$ as,

$$\hat{\sigma}^2 = \frac{e^T e}{n-p,}$$

where n and p are dimensions of Y and β in Eq. 2.1, respectively.

We can determine whether an observation is an outlier or not using its t_i value. As a rule of thumb, $|t_i|'s > 2$ are identified as outliers. As an example, the t_i value of point "O" in Figure 2.1 is 3.13. Once we identify a data point as an outlier, it is important to check if the point has an undue influence on the model. This is because some outliers may influence the regression parameter estimates while others may not. For example, the point "O" in Figure 2 does not seem to affect the slope of the regression greatly. Plot (b) in Figure 2.1 shows the



(a) (b)

Figure 2.3: Left: Scatter plot of the values of Y versus the corresponding values of X. Right: Best fitted regression lines with (red-dotted line) and without (blue line) the point "N".

fitted regression lines with and without the outlying point "O", and the slopes are almost identical. However, for some outliers, this will not be the same. For example, we added the point "N" to our hypothetical data set shown in Figure 2.3 and we can see it is an outlier ($|t_i| = 3.21$) but, unlike "O", it influences the estimated regression line. We can visually identify this by observing at the plot (b) in Figure 2.3. The slope changes by a noticeable amount when we refit the regression line without "N". Therefore, the assessment of residuals may help to identify outliers, but this alone is not enough to identify an influential observation in data. This is because an observation we identify as an outlier may not always be influential, as we discussed earlier. To distinguish these two situations, we need to understand more about the concept of leverage, which is discussed in the next section.

2. **Prediction matrix**

The diagonal elements (p_i's) of the prediction or projection matrix (P) given in Eq. 2.5 are important when identifying leverage points. This is because p_i's give the amount of leverage of y_i on the corresponding value $\hat{y}_i$. As Chatterjee and Hadi, 1986 [10] described, if an observation has a larger p_i, then it has higher leverage on the fitted regression line. The leverage effect increases when an observation is remote from the rest of the data in X space. Recall the outlying point "N" in Figure 2.3. It has more influence on the regression parameters than the outlier "O" in Figure 2.1 because "N" also has a leverage effect since it is remotely placed along the X axis. In Figure 2.2, point "L" is an example of a high leverage point, since it is an isolated point in the covariate space, and it has a high influence on the regression parameters, as shown in plot (b). However, every leverage point is not always influential. Point "B" in Figure 2.4 is far away from the rest of the covariate observations, but it does not have high leverage because it follows the general trend of the rest of the data.

A summary of what we have discussed so far can be given as follows: a data point can be influential on model parameter estimates if it has an outlying

response value, a high leverage point in the covariate space, or both of these qualities.

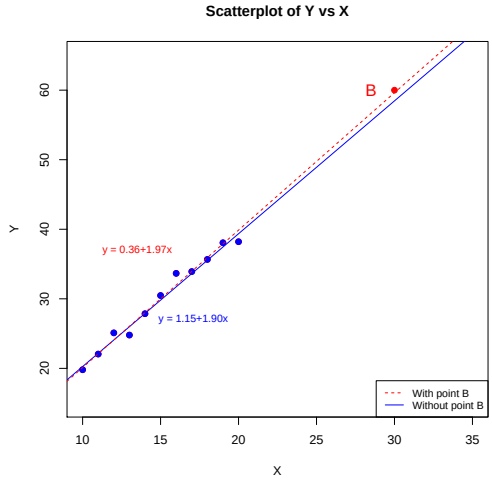


Figure 2.4: Best fitted regression lines with (red-dotted line) and without (blue line) the point "B".

3. The volume of confidence ellipsoids

Confidence intervals for regression parameters change significantly when the data have outliers or leverage points. As an example, in Figure 2.5, we constructed the joint 95% confidence ellipse for the parameters in the simple linear regression models shown in Figure 2.1 and 2.2 using the R package "Ellipse" [15]. The red and green ellipses in Figure 2.5 represent the joint confidence ellipses for the slope and intercept parameters in fitted regression models with and without the outlier "O" and leverage point "L". The areas of these ellipses are significantly different and these figures demonstrate that there is some association with confidence intervals and influential points. Chatterjee and Hadi, 1986 [10] described a few statistics we can use to measure the influence of an observation on model parameter estimates by studying the volume of the confidence ellipsoids. A confidence ellipsoid is the generalization of a confidence interval to more than one dimension. Confidence ellipsoid diagnostics compare the volume of the confidence ellipsoids with and without the i^{th} observation

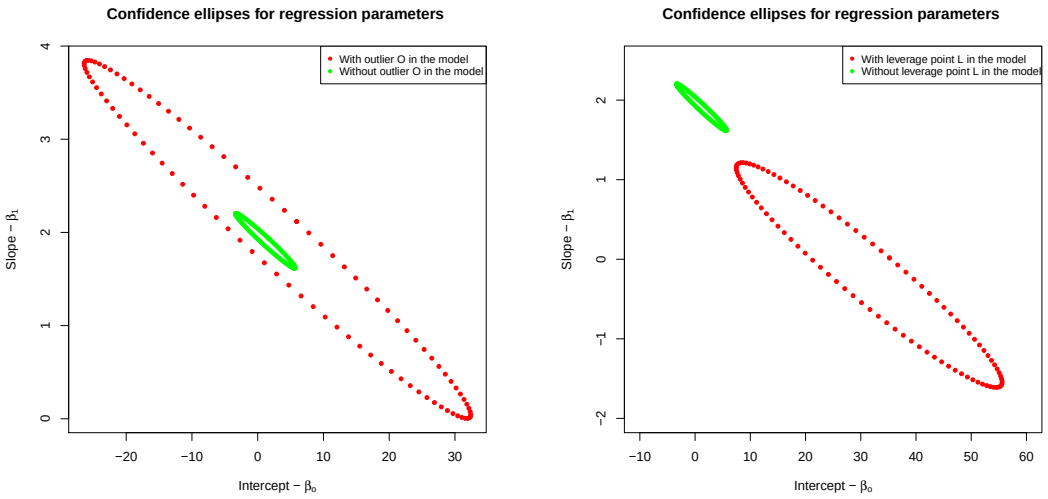


Figure 2.5: Joint confidence ellipses for slope and intercept parameters with and without outlying observation "O" and leverage point "L". Left: confidence ellipses for data in Figure 2.1. Right: confidence ellipses for data in Figure 2.2.

from the model. Some other diagnostics that have been proposed are: the Likelihood Distance (LD_i), Andrews-Pregibon Statistic (AP_i), Covariance Ratio (CVR_i), and Cook-Weisberg Statistic (CW_i). Among these, the Likelihood Distance is important for us in this study because we use an extension of LD_i, the Likelihood displacement ($LD_{(\omega)}$) by Cook and Weisberg, 1982 [13] as the main analytical method. Usually, the Likelihood distance is defined as

$$LD_i = 2[L(\hat{\beta}) - L(\hat{\beta}_{(i)})], \tag{2.7}$$

where $L(\hat{\beta})$ and $L(\hat{\beta}_{(i)})$ are the log likelihoods evaluated at $\hat{\beta}$ (with all the observations) and $\hat{\beta}_{(i)}$ (without the i^{th} observation). Although all the influence measures discussed earlier are strictly numerical, this LD_i is based on the probability model used, which is an important characteristic of LD_i. There is also a relationship between likelihood distance and the asymptotic confidence region, which is given as $\{\beta : L(\hat{\beta}) - L(\beta) \leq \chi^2_{\alpha,p+1}\}$, where $\chi^2_{\alpha,p+1}$ is the upper α point of the chi-squared distribution with $(p+1)$ degrees of freedom [10]. Since the likelihood displacement may be more computationally expensive, Cook and

Table 2.1: Likelihood distance measures for the data in Figure 2.1. LDi: likelihood distance when i^{th} observation is omitted from the model.

Obs. No.	X	Y	LDi
1	10	19.80	0.11
2	11	22.06	0.07
3	12	25.11	0.05
4	13	24.79	0.06
5	14	27.86	0.05
6	15	30.48	0.04
7	16	33.66	0.04
8	17	33.92	0.05
9	18	35.66	0.05
10	19	38.06	0.06
11	20	38.21	0.17
12	15	53.00	574.77

Weisberg, 1982 [13] introduced the following formula to calculate the LD_i for linear regression models.

$$LD_i = N \log \left[\left(\frac{N}{N-1} \right) \frac{N-p-1}{t_i^* + N - p - 1} \right] + \frac{t_i^*(N-1)}{(1-p_i)(N-p-1)} - 1, \quad (2.8)$$

where N is the total number of observations, p and p_i are the rank and the i^{th} diagonal element of the prediction matrix P, and t_i^* is the studentized residual of the i^{th} observation. The t_i^* is calculated using the formula,

$$t_i^* = \frac{e_i}{\sqrt{MSE_{(i)}(1-p_i)}}, \quad (2.9)$$

where $MSE_{(i)}$ is the mean squared error calculated when the i^{th} observation is deleted from the model. As an illustration, we calculated LD_i values for the data in Figure 2.1, and they are summarized in Table 2.1. The 12^{th} case clearly has the highest LD_i, ($LD_{12} = 574.77$), and the second largest is case 11, ($LD_{11} = 0.17$). This indicates that case 12 is an influential case. Also, it is

the same outlying data point we identified in Figure 2.1 as "O".

4. **Influence functions**

Hampel, 1974 [22] proposed influence functions as a more structured way to assess the influence of an observation. This has led to an alternative class of influence measures because, essentially, the influence function is a functional derivative taken with respect to the input probability distribution. The basic form of the influence function (IF_i) is

$$IF_i = (x_i; y_i; F; T) = \lim_{\varepsilon \to 0} \frac{T[(1-\varepsilon)F + \varepsilon \delta_{x_i, y_i}] - T[F]}{\varepsilon}, \tag{2.10}$$

where $T(.)$ is a statistic obtained from a random sample of the CDF F and $\delta_{x_i,y_i} = 1$ at (x_i, y_i) and 0 otherwise. The influence function (IF_i) measures the influence on T of adding one observation (x_i, y_i) to a very large sample. However, for a finite sample, approximations of the influence functions in Eq. 2.10 are used. The three most commonly used methods are: the empirical influence curve, sample influence curve, and sensitivity curve [10].

Empirical Influence Curve (EIC)
The EIC is obtained by replacing F in Eq. 2.10 by the empirical distribution function, $\hat{F}$, based on the full sample. Let $\hat{F_{(i)}}$ be the empirical distribution function when the i^{th} observation is omitted. We can substitute $\hat{F_{(i)}}$ for F and $\hat{\beta}_{(i)}$ (estimate of β when the i^{th} observation is omitted) in Eq. 2.10 and get the EIC,

$$\begin{aligned} EIC_i &= (N-1)(X_{(i)}^T X_{(i)})^{-1} x_i^T (y_i - x_i \hat{\beta}_{(i)}) \\ &= (N-1)(X^T X)^{-1} x_i^T \frac{e_i}{(1-p_i)^2}. \end{aligned} \tag{2.11}$$

Sample Influence Curve (SIC)

If we omit the limit in Eq. 2.10 and take $F = \hat{F}$, $T(\hat{F}) = \beta$, $\epsilon = -1/(N-1)$, we get the SIC,

$$\begin{aligned} SIC_i &= (N-1)(X^TX)^{-1}x_i^T(y_i - x_i\hat{\beta}_{(i)}) \\ &= (N-1)(X^TX)^{-1}x_i^T\frac{e_i}{(1-p_i)}. \end{aligned} \tag{2.12}$$

Sensitivity Curve (SC)
The SC is based on substituting $F = \hat{F}_{(i)}$ and $T(\hat{F}_{(i)}) = \beta_{(i)}$, $\epsilon = -1/N$,

$$SC_i = N(X^TX)^{-1}x_i^T\frac{e_i}{(1-p_i)}. \tag{2.13}$$

Influence curves are usually vectors because there are usually multiple parameters of interest. Therefore, it is useful to consider the norm of the influence function so that observations can be arranged in a meaningful way. The class of norms which are location/scale invariant is,

$$D_i(M;c) = \frac{(IF_i)^TM(IF_i)}{c} \tag{2.14}$$

where M is a symmetric, positive (semi-)definite matrix and c is a positive scale factor. A large value of $D_i(M;c)$ indicates that the observation i has a strong influence on the estimated coefficients relative to M and c.

We commonly use Cook's distance (C_i), Welsch-Kuh distance (WK_i), Welsch's distance (W_i), and Modified Cook's distance (C_i^*) as distance measures to identify influential observations in data sets. The basic idea behind all these methods is to delete a particular case (i^{th}), refit the regression model for the remaining $(n-1)$ cases, and then compare the new regression model results with the original regression model results. We can also obtain these distance measures by modifying M and c values in the influence curves described above. A summary of the distance measures is given below.

Cook's Distance (C_i)

In Eq. 2.14, if we replace IF_i by SIC_i, M by X^TX, and c by $(n-1)^2pS^2$, we obtain

$$\begin{aligned} C_i &= D_i(X^TX; (N-1)^2p\sigma^2) \\ &= \frac{(e_i^*)^2}{p}\frac{p_i}{(1-p_i)}, \end{aligned} \tag{2.15}$$

where e_i^*'s are standardized residuals, and p and p_i's are the trace and the diagonal elements of the prediction matrix P, respectively. C_i can also be written as,

$$C_i = \frac{(\hat{Y} - \hat{Y}_{(i)})^T(\hat{Y} - \hat{Y}_{(i)})}{p\ MSE}, \tag{2.16}$$

where $\hat{Y}_{(i)} = X\hat{\beta}_{(i)}$ is the vector of predicted values when $Y_{(i)}$ is regressed on $X_{(i)}$ ($Y_{(i)}$ and $X_{(i)}$ are vectors of Y and X where the i^{th} observation is deleted). The C_i directly summarizes how much all the fitted values change when the i^{th} case is omitted. If a data point has a large C_i value then it is an indication that the point influences the fitted values. In practice, observations with C_i values greater than the 10^{th} percentile of the F-distribution ($F_{p,n-p}$) are taken as potentially influential cases.

Welsch-Kuh Distance (WK_i)

If we replace $IF = SIC_i$, $M = X^TX$, and $c = (N-1)S^2_{(i)}$ in 2.14 we get

$$WK_i = |t_i^*|\sqrt{\frac{p_i}{1-p_i}}, \tag{2.17}$$

where t_i^*'s are studentized residuals and p_i's are diagonal elements of the prediction matrix. Vellerman and Welsch, 1980 [38] suggested that if an observation has a WK_i value greater than one or two, then that observation has the potential to be an influential observation.

Welsch's Distance (W_i)

Table 2.2: Influence measures based on the Influence function for the data in Figure 2.3. ci: Cook's distance, mci: modified Cook's distance, wki: Welsch-Kuh distance, wi: Welsch's distance

Obs. No.	X	Y	ci	mci	wki	wi
1	10	20	0.17	-1.32	-0.59	-2.27
2	11	22	0.07	-0.83	-0.37	-1.37
3	12	25	0.02	-0.37	-0.17	-0.60
4	13	25	0.03	-0.55	-0.25	-0.86
5	14	28	0.01	-0.27	-0.12	-0.42
6	15	30	0.00	-0.11	-0.05	-0.17
7	16	34	0.00	0.12	0.05	0.18
8	17	34	0.00	-0.03	-0.01	-0.05
9	18	36	0.00	0.04	0.02	0.07
10	19	38	0.01	0.30	0.13	0.51
11	20	38	0.00	0.02	0.01	0.04
12	11	42	1.15	19.93	8.91	32.86

This statistic is based on replacing $IF = EIC_i$, $M = X_{(i)}^T X_{(i)}$, and $c = (N-1)S_{(i)}^2$ in Eq. 2.14,

$$W_i = \frac{t_i^*}{1-p_i}\sqrt{(N-1)\ p_i}, \tag{2.18}$$

where N is the number of observations, t_i^* and p_i are defined above. The W_i is related to WK_i, $W_i = WK_i\sqrt{\frac{N-1}{1-p_i}}$, and W_i has higher sensitivity to p_i than WK_i. Therefore, W_i is a better tool than WK_i to capture any influential observations caused by the effect of leverage.

Modified Cook's Distance (C_i^*)

As a modification to C_i in Eq. 2.15, we can use $c = \sqrt{\frac{p}{n-p}}S_{(i)}^2$ and obtain

$$C_i^* = |t_i^*|\sqrt{\frac{(N-p)}{p}\frac{p_i}{(1-p_i)}}, \tag{2.19}$$

where t_i^*, p, p_i and N are described above. There is a relationship between C_i^* and WK_i, because $C_i^* = WK_i\sqrt{\frac{N-p}{p}}$. The modified Cook's distance was

introduced by Welsch and Kuh, 1977 [41], and it can highlight potential influential cases better than C_i. These influence measures are illustrated in Table 2.3 using the data shown in Figure 2.3. Cook's distance does not identify case 12 as influential, whereas the other measures do.

5. **Partial influence**

The methods discussed in the previous section were based on the assumption that we have an equal interest in all the regression parameters (β) in the model. However, there are some situations where we may want to find how an observation can affect one or several model parameters separately. For example, in a model with nuisance parameters, we will often not be primarily interested in the influence of the nuisance parameter estimates.

An observation might have a moderate influence on one or several regression parameters while having a large amount of influence on other parameters. In this case, there is a possibility of losing some important information using influence measures based on all parameters. Therefore, it is important to check partial influence in multiple linear regression models. Chatterjee and Hadi, 1986 [10] discussed a few of the most commonly used partial influence measures.

A modified version of the influence measure suggested by Cook in Eq. 2.15 can be used to measure the influence of the i^{th} observation on the j^{th} parameter (D_{ij}) as,

$$D_{ij} = \frac{t^2}{(1-p_i)} \frac{w_{ij}}{W_j^T W_j}, \tag{2.20}$$

where $W_j = (I - P_{[j]})X_j$, $P_{[j]} = X_{[j]}(X'_{[j]}X_{[j]})^{-1}X'_{[j]}$ (prediction matrix without the j^{th} independent variable), $X_{[j]}$ denotes $n \times (p-1)$ matrix from X with X_j removed, and w_{ij} the i^{th} element of W_j.

Added variable plots also help to identify partial influential points in multiple regression models (Velleman and Welsch, 1981) [38]. Suppose we want to fit

the model,

$$Y = X_{[j]}\beta + X_j\theta_j + \epsilon, \tag{2.21}$$

where β is a $(p-1) \times 1$ vector. If we multiply this model by $(I - P_{[j]})$ we get,

$$(I - P_{[j]})Y = (I - P_{[j]})X_j\theta_j + (I - P_{[j]})\epsilon, \tag{2.22}$$

and note that $(I - P_{[j]})X_{[j]} = 0$. Let R_j and W_j are the residual vectors when Y and X_j are regressed on $X_{[j]}$, respectively [10]. They are given as,

$$R_j = W_j\theta_j + \epsilon^*, \tag{2.23}$$

and,

$$W_j = (I - P_{[j]})X_j. \tag{2.24}$$

We take the expectation of Eq. 2.23, and obtain $E(R_j) = W_j\theta_j$, which suggests that a plot of R_j vs W_j will be linear through the origin. Also, the residuals from the multiple regression model in Eq. 2.21 and the residuals from the simple regression model in Eq. 2.23 are identical. Therefore, these plots can be used to identify potential data points which affect individual coefficients because, in general, the scatter of the points will give an overall idea of the strength of the relationship. Therefore, points which lie well away from the rest of the data may be influential in determining the magnitude of parameter estimates [10].

We have described many influence measures that are available based on case deletions. However, it is not advisable to use case deletion diagnostic methods all the time because in some situations deleting a case may lose valuable information provided by the observation. Cook, 1986 [12] proposed another combined approach of assessing the local influence of minor perturbations of

a statistical model based on likelihood and elementary ideas in differential geometry. This local influence method is described in the next section.

2.1.3 Local Influence analysis

Cook's local influence analysis

This method is partially motivated by the form of Cook's statistic in Eq. 2.16,

$$D_i = ||\hat{Y} - \hat{Y}_{(i)}||/p\sigma^2, \tag{2.25}$$

where $\hat{Y}$ and $\hat{Y}_{(i)}$ are $n \times 1$ vectors of fitted values based on the full data and data without the i^{th} observation, respectively, and p is the dimension of the vector of unknown parameters, θ. We refer to the i^{th} observation of the response variable y_i and the associated explanatory variables x_i as the *case* we want to examine. To obtain $\hat{Y}_{(i)}$, we need the re-estimated parameters without the i^{th} case, named $\hat{\beta}_{(i)}$. We can minimize the "weighted mean squared error" (WMSE) of the linear model to obtain the $\hat{\beta}_{(i)}$. The WMSE of the linear model given in Eq. 2.1 is,

$$WMSE(\beta, w) = \frac{1}{n}\sum_{i=1}^{n} w_i(y_i - x_i\beta)^2, \tag{2.26}$$

where w is a $n \times 1$ vector of weights, given by, $w = (w_1, \ldots, w_n)'$. We then set the i^{th} element of the weight vector to zero ($w_i = 0$) and estimate model parameters by minimizing the WMSE above.

Although the case deletion method is widely used in sensitivity analysis, the diagnostics only allow us two possibilities. Cook, 1986 [12] noted them as 1) the case specifies the model as it is or 2) the case does not follow the model (or is totally unreliable). It is also interesting to examine the impact of a change in a case weight (other than zero) to parameter estimates of a model. Cook, 1986 [12] suggested the following slightly modified version of Eq. 2.25,

$$D_i(w) = ||\hat{Y} - \hat{Y}_w||/p\sigma^2, \tag{2.27}$$

where $\hat{Y}_w$ is the vector of fitted values obtained when the i^{th} case has weight w_i where $i = 1, \ldots, n$. Although we can assign any value to the weight (w), we need to choose it carefully so that the application is sensible.

Let $L(\theta|\omega)$ denote the log-likelihood corresponding to the perturbed model for a given ω in an open subset Ω of $\mathbb{R}^n$. Assume that there is also an ω_0 in Ω such that $L(\theta) = L(\theta|\omega_0)$ for all θ. Finally, let $\hat{\theta}$ and $\hat{\theta}_\omega$, denote the maximum likelihood estimators under $L(\theta)$ and $L(\theta|\omega)$, respectively, and assume that $L(\theta|\omega)$ is twice continuously differentiable in (θ^T, ω^T), where ω is a $k \times 1$ vector. It is interesting to examine the influence of changing ω throughout its domain Ω. Cook, 1986 [12] suggested the use of the "likelihood displacement" given in the following form to assess the influence,

$$LD(\omega) = 2[L(\hat{\theta}) - L(\hat{\theta}_\omega)]. \tag{2.28}$$

We can use $LD(\omega)$ as a measure of influence and also as a measure of checking the model's adequacy. Using this likelihood displacement $LD(\omega)$ and the perturbation scheme ω, we can construct an influence graph. When we have only one perturbation scheme $(k = 1)$, the graph of $LD(\omega)$ vs. ω is a plane curve. When $k = 2$ the influence graph is a 3-dimensional surface. However, when $k > 2$ visualization of the influence graph is complicated.

Therefore, Cook, 1986 [12] proposed the normalized curvature of the influence graph to measure the influence, which is the geometric surface formed by the values of the vector,

$$\alpha(\omega) = \begin{bmatrix} \omega \\ LD(\omega) \end{bmatrix}_{(k+1)\times 1} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}_{(k+1)\times 1}, \tag{2.29}$$

where ω can reflect any well-defined perturbation scheme in Ω of $\mathbb{R}^n$. $\alpha(\omega)$ is a

$(k+1) \times 1$ vector and α_1 is a $k \times 1$ vector. When $k = 1$, $\alpha(\omega)$ reduces to a plane curve and the curvature of such a plane curve is,

$$C = |\dot{\alpha}_1 \ddot{\alpha}_2 - \dot{\alpha}_2 \ddot{\alpha}_1| / (\dot{\alpha}_1^2 + \dot{\alpha}_2^2)^{3/2}, \tag{2.30}$$

where $\dot{\alpha}_i$ and $\ddot{\alpha}_i$ are the first and second derivatives evaluated at ω_0. Since

$$\dot{\alpha}_1 = \frac{\partial \omega}{\partial \omega}|_{\omega=\omega_0} = 1,\ \ddot{\alpha}_1 = 0 \ and\ \dot{\alpha}_2 = \frac{\partial LD(\omega)}{\partial \omega}|_{\omega=\omega_0} = 0,$$

from Eq. 2.30, the curvature is,

$$C = |\ddot{\alpha}_2| = |\ddot{LD}(\omega_0)|. \tag{2.31}$$

Although it is not easy to characterize the influence graph over the full range of Ω, it is easier to characterize the behavior in the neighbourhood of a specific value for ω. Therefore, Cook, 1986 [12] focused on the behaviour of the influence graph around the null perturbation (ω_0) using the geometric normal curvature, where $L(\theta) = L(\theta|\omega_0)$. He referred to this as the study of "local influence". Let ω be defined as a function of h ($\in \mathbb{R}^1$) and a straight line in Ω passing through ω_0; $\omega(h) = \omega_0 + hd$, where d is the direction vector of length one, and h is the scalar which determines the magnitude of the perturbation scheme. From Eq. 2.31, we can derive the normal curvature in the direction d as,

$$C_d = |\ddot{LD}\{\omega(h)\}|, \tag{2.32}$$

where $\ddot{LD}\{\omega(h)\}$ is the second order derivative of the likelihood displacement function in Eq. 2.28 with respect to h, which is,

$$\frac{\partial^2 LD\{\omega(h)\}}{\partial h^2}. \tag{2.33}$$

We can further evaluate the curvature along the direction d using the chain rule in differentiation as,

$$C_d = 2|d^T \ddot{F} d|, \tag{2.34}$$

where $||d|| = 1$, $\ddot{F}$ is a $k \times k$ matrix with elements $\partial^2 L(\hat{\theta}_\omega)/\partial\omega_l \partial\omega_j, l = 1, 2, ..., k$ (see Appendix A.1 for the derivation). Since we do not have a direct method to evaluate $\ddot{F}$, we simplify it using the chain rule in differentiation as,

$$\ddot{F} = J^T \ddot{L} J, \tag{2.35}$$

where $-\ddot{L}$ is the observed information matrix for the original model (at $\omega = \omega_0$) and J is the $p \times k$ matrix with elements $\partial\hat{\theta}_{i\omega}/\partial\omega_j, i = 1, 2, ..., k$.

$$\ddot{L} = \begin{bmatrix} \partial^2 L(\theta)/\partial\theta_1\partial\theta_1 & ... & \partial^2 L(\theta)/\partial\theta_1\partial\theta_p \\ \vdots & \ddots & \vdots \\ \partial^2 L(\theta)/\partial\theta_p\partial\theta_1 & ... & \partial^2 L(\theta)/\partial\theta_p\partial\theta_p \end{bmatrix}_{\theta=\hat{\theta}}$$

$$J = \begin{bmatrix} \partial\hat{\theta}_{1\omega}/\partial\omega_1 & \partial\hat{\theta}_{1\omega}/\partial\omega_2 & \dots & \partial\hat{\theta}_{1\omega}/\partial\omega_k \\ \vdots & \vdots & \ddots & \vdots \\ \partial\hat{\theta}_{p\omega}/\partial\omega_1 & \partial\hat{\theta}_{p\omega}/\partial\omega_2 & \dots & \partial\hat{\theta}_{p\omega}/\partial\omega_k \end{bmatrix}$$

To evaluate J, we use the fact that

$$\left.\frac{\partial L(\theta|\omega)}{\partial\theta_j}\right|_{\theta=\hat{\theta}_\omega} = 0.$$

Differentiating both sides with respect to ω and evaluating at ω_0, $\ddot{L}J = \Delta$, where

$$\Delta_{ij} = \frac{\partial^2 L(\theta|\omega)}{\partial\theta_i \partial\omega_j}\,. \tag{2.36}$$

Hence,

$$\ddot{F} = \Delta^T (\ddot{L})^{-1} \Delta. \tag{2.37}$$

Therefore, the curvature (C_d) is

$$C_d = 2|d^T \ddot{F} d|. \tag{2.38}$$

Let C_{max} be the maximum eigenvalue of $\ddot{F}$ and let e_{max} be the eigenvector for C_{max}. Cook, 1986 [12] suggested that a large value of C_{max} is an indication of a serious local problem, and if the i^{th} element in e_{max} is relatively large, special attention should be paid to the element being perturbed by ω_i [32].

Local influence first order approach

The likelihood displacement influence measure is focused directly on estimated parameter values ($\hat{\theta}$), although in many situations it is not only the parameters themselves that are of interest, but also some function of the parameter estimates or a forecast of the data may be the prime interest. Cadigan and Farrell, 2002 [8] suggested a more general approach to local influence analysis which can be evaluated directly using numerical methods rather than deriving analytic expressions, which may be quite complex in any event and difficult to understand without computing the analytic results. Cadigan and Farrell, 2002 [8] assumed that the problem involved the estimation of a $p \times 1$ parameter vector θ by maximizing a fit function $F(\theta)$ that is twice differentiable in θ and yields unique interior parameter estimates. The estimate of θ, denoted as $\hat{\theta}$, is the solution to

$$\dot{F}(\hat{\theta}) = \left.\frac{\partial F(\theta)}{\partial \theta}\right|_{\theta=\hat{\theta}} = 0.$$

They considered a perturbation vector ω with dimension $k \times 1$ in the form $\omega = \omega_0 + hd$, where ω_0 is the null perturbation, d is a fixed direction vector of length 1 and h is a scalar that determines the magnitude of the perturbation. The main focus of their estimation was an arbitrary function of parameter estimates ($g(\hat{\theta})$). Also, their interest was to assess the influence for the perturbed result $g_\omega(\hat{\theta}_\omega)$ which they assumed to be a first-order differentiable function in h and θ. Here g_ω depends on ω not only through $\hat{\theta}_\omega$, which is another difference from Cook's, 1986 [12] Likelihood Displacement method. Cadigan and Farrell, 2002 [8] referred to their method as a First order approach. They measured the influence of a perturbation using the slope

in the direction d, denoted as $S(d)$, of the influence graph of g_ω versus $\omega(h)$,

$$S(d) = \frac{\partial g_\omega(\hat{\theta}_\omega)}{\partial h}\bigg|_{h=0} = d' \frac{\partial g_\omega(\hat{\theta}_\omega)}{\partial \omega}\bigg|_{h=0}. \tag{2.39}$$

Using the chain rule, $S(d)$ can be decomposed into more simple derivatives,

$$S(d) = d' \left\{ \frac{\partial g_\omega(\hat{\theta})}{\partial \omega}\bigg|_{\omega=\omega_0} + \frac{\partial \hat{\theta}'_\omega}{\partial \omega}\bigg|_{\omega=\omega_0} \frac{\partial g(\theta)}{\partial \theta}\bigg|_{\theta=\hat{\theta}} \right\},$$

where

$$\frac{\partial \hat{\theta}_\omega}{\partial \omega}\bigg| = -\ddot{F}^{-1}\Delta, \tag{2.40}$$

$$\ddot{F} = \frac{\partial^2 F(\theta)}{\partial \theta \partial \theta'}\bigg|_{\theta=\hat{\theta}},$$

$$\Delta = \frac{\partial^2 F_\omega(\theta)}{\partial \theta \partial \omega'}\bigg|_{\theta=\hat{\theta},\omega=\omega_0}, \tag{2.41}$$

These results can be used to provide a relatively simple formula for $S(d)$,

$$S(d) = d'\dot{g}_0, \tag{2.42}$$

where $\dot{g}_0$ is

$$\dot{g}_0 = \frac{\partial g_\omega(\hat{\theta})}{\partial \omega}\bigg|_{\omega=\omega_0} - \Delta \ddot{F}^{-1} \frac{\partial g(\theta)}{\partial \theta}\bigg|_{\theta=\hat{\theta}}. \tag{2.43}$$

Cadigan and Farrell, 2002 [8] noted a couple of advantages of using their influence diagnostics method. For some models with high dimensional parameters (θ) and perturbation schemes (ω), the evaluation of perturbed parameter estimates ($\hat{\theta}_\omega$) is difficult. We can avoid this problem with the preceding method, since we can obtain $\dot{g}_0$ without evaluating $\hat{\theta}_\omega$. The evaluation of Δ in Eq. 2.41 is also difficult, since it involves computing $k \times p$ number of differentiations. However, we only need to compute $\ddot{F}$ once, because all the influence measures share a common $\ddot{F}$.

$S(d)$ can be used to compute the slope of the influence surface in a variety of directions. The direction that corresponds to the maximum slope of the influence surface is of particular interest. Perturbations with large absolute elements in s_{max} are relatively influential. The maximum slope ($S(s_{max})$) can be computed as,

$$S(s_{max}) = \sqrt{(\dot{g}_0' \dot{g}_0)}, \tag{2.44}$$

and then $s_{max} = \dot{g}_0 / S(s_{max})$. In our study, we also find the local slope as a percent of full sample estimates ($g(\hat{\theta})$) which we denote as pS_{max}. We can use pS_{max} as a scale-free measurement to compare the influence for each case.

Let the least squares of the linear model in Eq. 2.1 be the fit function $F(\beta)$,

$$F(\beta) = \sum_{i=1}^{n} (y_i - x_i \beta)^2.$$

We can perturb this least squares function using the weights $w_1, \ldots, w_n$ and write the perturbed fit function $F(\beta, w)$,

$$F(\beta, w) = \sum_{i=1}^{n} w_i (y_i - x_i \beta)^2.$$

We write the likelihood displacement of the fit function caused by the perturbed parameter estimates,

$$LD = 2\{F(\hat{\beta}) - F(\hat{\beta}_w)\},$$

where $\hat{\beta}$ and $\hat{\beta}_w$ are parameter estimates for full model and perturbed model. LD is conceptually the same as the likelihood displacement in Eq. 2.28. We can use Cook's method to assess the influence of the function LD. The slope of the LD-influence curve in the direction d is,

$$S(d) = -d' \Delta' \ddot{F}^{-1} x. \tag{2.45}$$

Chapter 3

State Space Models (SSMs) and Template Model Builder (TMB)

3.1 State space models

The state space model also called the dynamic linear model, was introduced by Kalman, 1960 [24] and Kalman and Bucy, 1961 [25]. In the early years, the method was primarily used for aerospace-related research by engineers. Later, the method was applied to modelling data from engineering, economics, medicine, ecology, and social sciences by statisticians.

Since SSMs can incorporate both the measurement error associated with sampling methods and biological (or process) variation of an ecological system, scientists increasingly use SSMs to model ecological systems [5], [30]. Buckland et al. [7] described how they used a SSM to model the dynamics of wild animal populations. They claimed that the flexibility of the method allowed them to incorporate the stochastic variation of the processes. Wang [40] noted that most of the field measurements of ecological variables suffer from human errors and inefficiency of equipments. Therefore traditional statistical inferences may not give accurate results. Addressing this, Wang [40] described the importance of using both measurement errors and process errors when modelling the dynamics of a population.

The state space model (SSM) framework includes most of the linear models including the classical and Box-Jenkins models. Therefore, an SSM is an omnibus model classification for most time series models. It can also represent a latent variable model, since the underlying structure of an observed series may be modeled through unobservable latent variables.

For example, consider x_t, which is the observed time series where $t = 1, ..., n$. Suppose we have a random variable w_t which is a vector of d number of terms, $w_t = (w_{1t}, ..., w_{dt})'$ and x_t is a function of w_t. We call w_t the vector of state variables. Like x_t, this vector (w_t) also varies with t, but unlike x_t, we do not observe w_t. Let $\alpha = (\alpha_1, ..., \alpha_d)'$, a vector of parameters and the observation equation is

$$x_t = \alpha' w_t + \epsilon_t, \tag{3.1}$$

where ϵ_t is an independent and identical white noise sequence with zero mean and variance σ_w^2. We have defined the observed variable in terms of the latent or unobserved variables in this model. The second part of SSM is the state equation. To define this, let $\eta = (\eta_t^{(1)}, ..., \eta_t^{(m)})$ be the iid random vector with mean zero and covariance matrix $\Sigma = diag(\sigma_1^2, ..., \sigma_n^2)$. Let Φ, K be $d \times d$ and $d \times m$ matrices of parameters. The state equation is,

$$w_t = \Phi w_{t-1} + K\eta_t. \tag{3.2}$$

This looks like an AR(1) type model in which w_t is modelled by w_{t-1} and η_t are errors. Also, we have an error coefficient (K) in this case and Φ is basically the autoregressive parameter. We can also assume that the ϵ_t (error in the observation equation) and η_t (error in the state equation) are mutually independent and both are independent of w_1 (first state variable). It is commonly assumed that ϵ_t, and η_t are normally distributed.

A common feature in fishery models is that current stock biomass is related to last year's biomass. The use of SSMs in fish stock assessment has increased in recent years because of their ability to handle process errors in population dynamics models and

observation errors in fishery catch and survey time series data. State space models for fisheries can be fitted through a combination of two stochastic processes. Usually, the state equation contains a stochastic process of population dynamics. As described earlier, this consists of the biomass in the year t as a function of biomass in the year $t-1$ and the new addition to the biomass because of the new recruitment of fish as well as the body growth of fish in the population in the last year. However, the catches or different survey indices (or CPUE-catch per unit effort series) are taken as the observed variables for the state space model (Miller and Meyer, 2000) [28]. In Chapter 4, we will describe a state space surplus production model in detail.

3.2 Template Model Builder (TMB)

One of the most critical and challenging tasks in computational statistics is to calculate derivatives of high-dimensional functional matrices or fit functions, including the log-likelihood. With the efforts of scientists and with the advancement of computers, many computer algorithms and packages are available to overcome this challenge. Automatic Differentiation (AD) (Griewank, 2000 [20]) is a technique that computes derivatives of a function given as a computer algorithm. This technique was later adapted for statistical software through packages like ADMB by Fournier et al., 2012 [17] and Ceres Solver by Agarwal and Mierle, 2013 [4].

To estimate surplus production models we use R package TMB (Template Model Builder) (https://github.com/kaskr/adcomp) by Kristensen et al., 2015 [27]. This package is capable of evaluating first, second, and possibly third-order derivatives. Kristensen et al. [27] described a few of the advantages of using TMB over ADMB: faster run times, capability to handle very high-dimensional problems (up to 10^6 random effects), automatic calculation of the gradient vector and hessian matrix for parameters, the use of external libraries, and there is no use of temporary files on the disk. A notable feature of the TMB package is that it automatically integrates out random effects in mixed-effects models (see below for example) using the Laplace approximation when it evaluates the marginal likelihood [27].

Let us consider how to implement TMB with an example of fitting a linear regression model for the response variable y and predictor variable x. The regression model is,

$$y_i = a + bx_i + \epsilon_i, \tag{3.3}$$

where $a,\ b$ are intercept and slope coefficients, respectively, ϵ is the error variable and $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$. In the C++ template we need to provide the function to evaluate the negative log likelihood for y_i. The distribution of y_i is $y_i \sim N(a + bx_i, \sigma^2)$. The joint probability density function for $y_1, y_2, \ldots, y_n$ is,

$$\prod_{i=1}^{n} p(y_i|x_i; a, b, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i-(a+bx_i))^2}{2\sigma^2}}, \tag{3.4}$$

and the negative log-likelihood is,

$$\begin{aligned} L(a, b, \sigma^2) &= log\left[\prod_{i=1}^{n} p(y_i|x_i; a, b, \sigma^2)\right] \\ &= -\frac{n}{2} log(2\pi) - nlog(\sigma) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \Big(y_i - (a + bx_i)\Big)^2. \end{aligned} \tag{3.5}$$

The first step to find the MLE's of a, b, and sigma using TMB is to write the C++ template,

```
#include <TMB.hpp>
template<class Type>
Type objective_function<Type>::operator() ()
{
  DATA_VECTOR(Y);
  DATA_VECTOR(x);
  PARAMETER(a);
  PARAMETER(b);
  PARAMETER(logSigma);
```

```
  ADREPORT(exp(2*logSigma));
  Type nll = -sum(dnorm(Y, a+b*x, exp(logSigma), true));
  return nll;
}
```

For most models, the C++ syntax used in the first four lines and the last line are standard. Many TMB functions have been designed to mimic R functions and syntax. The data used in the model are declared by the line **DATA_VECTOR()**, the parameters are declared by the line **PARAMETER()**, and the log density for a normal distribution is provided by the function **dnorm**, similar to *R*. The **ADREPORT()** macro reports an expression (scalar, vector, matrix or array valued) back to R with derivative information and typically used to obtain point estimate and standard deviation of the expression. After finishing the user template, we can use *R* to compile, link, evaluate, and optimize this model using the TMB package. The *R* code for the corresponding model is,

```
library(TMB)
compile("linreg.cpp")
dyn.load(dynlib("linreg"))
set.seed(123)
data <- list(Y = rnorm(10) + 1:10, x=1:10)
parameters <- list(a=0, b=0, logSigma=0)
obj <- MakeADFun(data, parameters, DLL="linreg")
obj$hessian <- TRUE
opt <- do.call("optim", obj)
opt$par
opt$hessian ## <-- FD hessian from optim
obj$he()    ## <-- Analytical hessian
sdreport(obj)
```

The first line loads the TMB package. The next two lines compile and link the user template. The line **data** $\leftarrow$ **list(y = ...)** creates a data list for passing to

MakeADFun. We must assign the same names for the data components in this list as the **DATA_VECTOR** names in the C++ user template. The values assigned to the components of parameters are used as initial values during optimization. The line that begins **obj** $\leftarrow$ **MakeADFun** defines the object **obj** containing the data, parameters and methods that access the objective function and its derivatives.

In the ninth line we use the standard R optimizer **optim** to minimize the **obj$fn** aided by the gradient **obj$gr** and starting at the point **obj$par**. The last line is used to calculate the standard deviations of all model parameters. We can obtain estimated parameters using the code **opt$par** (See Appendix A.2 for outputs of the model). In the following section, we discuss how to estimate parameters when we have random effects in our model.

We commonly denoted random effects in a state space models by a vector Γ,

$$\Gamma = (\epsilon_1, \ldots, \epsilon_n, \eta_1, \ldots, \eta_n)'$$

where ϵ_i's and η_i's are random errors from observations and state equations in Eq. 3.1 and Eq. 3.2, respectively. Also, fixed effects parameters are denoted by the vector θ. Since we are using the **TMB** package to estimate parameters of our SSMs, we need to provide the joint negative log-likelihood of the data and random and fixed effects in the C++ source code. We cannot directly estimate fixed effects parameters, particularly variance parameters, when random effects are included in the joint likelihood function. It is better to estimate fixed effects by maximizing the marginal likelihood by integrating out random effects from the likelihood function.

This method is usually known as the marginal likelihood estimation method. Let S denote the set of all data, i.e., CPUE indices and catches, used in the model. The marginal likelihood is,

$$L(\theta) = \iiint_{\Gamma} f_\theta(S|\Gamma) g_\theta(\Gamma) \partial\Gamma \tag{3.6}$$

where $f_\theta(S|\Gamma)$ is the probability density function (pdf) of the data conditional on the random effects Γ and $g_\theta(\Gamma)$ is the joint pdf for the random effects Γ. Note that

$f_\theta(S|\Gamma)g_\theta(\Gamma)$ is the joint pdf of S and Γ. There are two main steps to use TMB for maximum marginal likelihood estimation. First, the user needs to provide the C++ computer code to calculate $f_\theta(S|\Gamma)$ and $g_\theta(\Gamma)$. The calculation of the integration in Eq. 3.6 and the calculation of Γ required for the Laplace approximation are provided by TMB in *R*. The random effects Γ can be predicted by maximizing the joint likelihood, $f_\theta(S|\Gamma)g_\theta(\Gamma)$ [9]. TMB uses automatic differentiation to evaluate the gradient function of Eq. 3.6 and in the Laplace approximation. The gradient function is produced automatically from $f_\theta(S|\Gamma)$ and $g_\theta(\Gamma)$. This greatly improves parameter estimation using a derivative-based optimizer. We also use the nlminb() function within R (R Core Team, 2014 [33]) to find the MLE for θ.

3.3 Laplace Approximation

The Laplace approximation is a technique used to numerically approximate integrals and is very accurate for certain types of integrals. It is the approach used in TMB to find the solution for marginal likelihoods. Let $Y_n = (y_1, y_2, ..., y_n)'$ be the vector of n number of observations, $\tau = (\lambda_1, \lambda_2, ..., \lambda_q)'$ be the vector of latent random effects, and let $\theta = (\theta_1, \theta_2, ..., \theta_m)'$ be the vector of parameters (fixed effects). We write the joint negative log likelihood for data, and random and fixed effects as,

$$l(\theta; Y_n, \tau_q) = \log\{f_{Y_n|\tau_q=x}(y_1, \ldots, y_n) f_\tau(x)\}.$$

The joint density function is,

$$f_{Y_n|\tau_q=x}(y_1, \ldots, y_n) f_\tau(x) = f_{Y_n,\tau_q}(Y_n = y, \tau_q = x).$$

We can write the marginal density of Y_n as,

$$\begin{aligned} f_{Y_n}(y) &= \iint \cdots \int f_{Y_n,\tau_q}(Y_n = y, \tau_q = x) dx_1, ..., dx_q \\ &= \iint \cdots \int f_{Y_n|\tau_q} f_{\tau_q} dx_1, ..., dx_q, \end{aligned} \tag{3.7}$$

where $f_{Y_n|\tau_q} f_{\tau_q} = \exp(l(\theta; Y_n, \tau_q))$. The Laplace approximation is based on a second order Taylor series expansion of $l(\theta; Y_n, \tau_q)$ around the mode of τ. Let $\hat{\tau}_\theta$ be the value of τ that maximizes the joint likelihood evaluated at the observations when θ fixed such that,

$$\hat{\tau}_\theta = max_\tau l(\theta; Y, \tau).$$

Note that $\frac{\partial l(\theta; Y_n = y, \tau)}{\partial \tau}|_{\tau=\hat{\tau}} = 0$. Therefore,

$$l(\theta; Y, \tau) \approx l(\theta; Y, \hat{\tau}) + (\tau - \hat{\tau})' H(\theta)(\tau - \hat{\tau}),$$

where $H(\theta) = \frac{\partial^2 l(\theta; Y, \tau)}{\partial \tau \partial \tau'}|_{\tau=\hat{\tau}}$, is a $q \times q$ matrix.
The marginal likelihood of θ is,

$$\begin{aligned} L(\theta; Y) &= \iint \cdots \int exp\{l(\theta; Y, \tau)\} d\tau \\ &\approx \iint \cdots \int exp\{l(\theta; Y, \hat{\tau}_\theta) + (\tau - \hat{\tau})' H(\theta)(\tau - \hat{\tau})\} d\tau \\ &= L(\theta; Y, \hat{\tau}_\theta) \iint \cdots \int exp\{(\tau - \hat{\tau})' H(\theta)(\tau - \hat{\tau})\} d\tau \\ &= L(\theta; Y, \hat{\tau}_\theta)(2\pi)^{n/2} Det\{H(\theta)\}^{-1/2} \end{aligned} \tag{3.8}$$

Finally, the Laplace approximation of $l(\theta; Y)$ is

$$l(\theta; Y) \approx l(\theta; Y, \hat{\tau}_\theta) - \frac{1}{2} \log[Det\{H(\theta)\}] + \frac{n}{2} \log(2\pi).$$

The hessian matrix, H, is evaluated by **CppAD** using TMB. Using the AD and Laplace approximation greatly simplifies the parameter estimation of hierarchical models. The TMB user only needs to specify the joint log-likelihood function. TMB uses the Cholesky decomposition of $H(\theta)$; therefore, the Laplace approximation is well defined only if $H(\theta)$ is positive definite.

In a R session, we read the data, dynamically link the C++ function template, set up the initial values for θ, specify the random effects, and optimize the objective

function. TMB automatically provides a standard error report for $\hat{\theta}$, and also any differentiable function of θ, $\phi(\theta)$ that the user specifies, by using the delta method [27].

Chapter 4

Surplus Production Models (SPMs)

Two main types of models are available for fish stock assessments. The first is surplus production models, which are a less complex type of model because of the simplicity of the data used. Such models only use information on the total (i.e., aggregated over all sizes and ages) catch each year and an aggregated index of the stock size. The second type are structural models, which are more complex because they use more structured data on the fish stock, such as data on the age or length of the fish. In this chapter, we investigate the sensitivity of some important outputs from surplus production models. We measure sensitivity by quantifying the impact of changes in data inputs on model outputs. We investigate the sensitivity of the contemporary surplus production model compared to its state space version to examine if there are differences among these modelling approaches. We are also interested in comparing the traditional case deletion diagnostics with the local influence diagnostics introduced by Cook, 1986 [12]. We compare diagnostics for real data sets obtained from five different fish stocks assessments.

It is first useful to present some important terminology from fishery science, and we start with "carrying capacity". Figure 4.1 shows the behavior of a fish population over time. We can identify three main phases in the figure. Phase 1 is the initial stage

when there are not much fish in the sea; hence, the population size increases slowly. At the beginning of the second phase, the population grows at an increasing rate and at the middle stage the population has the fastest growth rate. In the later part of the second phase, the population is growing but at a decreasing rate. In the last phase, the population growth slows down and eventually reaches an equilibrium. This is mainly because food and habitat space become scarce and then the death rate equals the birth rate. The population size in this stage is called the "carrying capacity". We can simply say that the carrying capacity (K) is the highest population size this environment can fit.

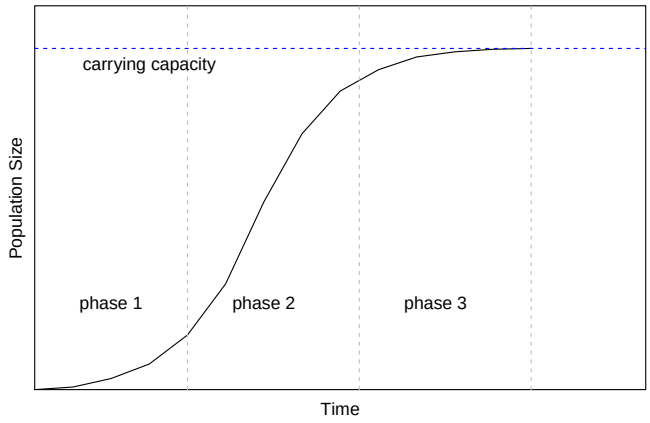


Figure 4.1: Fish population growth over time

The maximum sustainable yield (MSY) is another important concept in fish stock assessment. MSY refers to the largest average catch that can be sustained over a long period by keeping the stock at a level which produces the maximum annual population growth. Often, the MSY is about half of the carrying capacity ($MSY = K/2$). Fisheries management agencies use MSY as a valuable tool to guide fishing regulations to maintain a healthy fish population by avoiding overharvesting. This also helps to sustain a profitable fishing industry. Scientists use MSY concepts to study harvested populations and to determine biological reference points. In fisheries

stock assessments, "biological reference points" are reference values used to measure the status of a stock from a biological perspective [14].

Let B denote the stock biomass, which is the total weight of fish in the stock. Let B_{MSY} be the stock size which would produce the maximum sustainable yield (MSY) of the stock. Similarly, let H denote the harvest rate and H_{MSY} be the harvest rate that produces the MSY. It can be shown for the Schafer's surplus production model that $B_{MSY} = K/2$, and $H_{MSY} = r/2$. The harvest rate is the ratio between the catch and the biomass ($H = C/B$). Here, the catch (C) is the biomass of the stock taken by fishing.

An important focus of fish stock assessments is to estimate the current size of the stock ($B_{current}$) and the current harvest rate ($H_{current}$). Using these values, we can estimate the status of the stock, which is often defined in terms of the ratios of the current value and the respective MSY values. The current status of the stock biomass is given by,

$$B_{status} = B_{current}/B_{MSY}.$$

Similarly, the current status of the harvest rate is $H_{status} = H_{current}/H_{MSY}$. These status values are important because they can be used to guide management actions. For example, if $B_{current}/B_{MSY} < 1$, this means the stock biomass in the last assessment year is less than B_{MSY}, and in this case, the fishery should not harvest the MSY catch in the short-term.

Estimation of most stock assessment models requires some kind of index of abundance (I) to provide information about the trend in stock abundance over time. Catch per unit effort (CPUE) data is commonly used as an index of abundance for some species. CPUE is the number of fish caught per unit of effort from an area over some time. CPUE can be obtained from both commercial catches and scientific surveys. Generally, commercial CPUE is the number or weight (biomass) of fish caught by an amount of effort. The effort is a combination of gear type, gear size, and length of time the gear is used. However, in a scientific trawl survey, CPUE is taken as the average catch per tow. A common assumption for these indices of abundance is that

they are only influenced by changes in abundance. That is, changes in the index of abundance are proportional to changes in the actual stock abundance, and vice versa. The relationship between an index of abundance and true abundance can be written as,

$$I_t = qB_t \tag{4.1}$$

where I_t is the index of relative abundance at time t, B_t is the population biomass at time t, and q is the catchability coefficient which is the portion of a stock caught by a single unit of fishing effort. The time t is usually measured in years.

4.1 Surplus production models

Surplus production models are simple and widely used fish stock assessment models. They are commonly called production models or biomass dynamic models because the behavior or the dynamics of the stock is described in terms of age-aggregated total biomass rather than the numbers at age. The following conceptual equation is a simple way of expressing the change of biomass from one time period to the next if we ignore immigration and emigration.

$$next\ biomass = last\ biomass + recruitment + growth - catch - natural\ mortality, \tag{4.2}$$

where recruitment is the addition of newborn fish to the population, growth is the increase in biomass due to the body size growth of fish from last year, catch is the total biomass of fish taken due to fishing in the period, and the natural mortality is the number of fish that die from causes other than fishing. We can combine recruitment and growth into a single term called production, in the absence of fishing.

$$next\ biomass = last\ biomass + production - catch - natural\ mortality. \tag{4.3}$$

The difference between production and natural mortality is referred to as surplus

production,

$$next\ biomass = last\ biomass + surplus\ production - catch. \tag{4.4}$$

This model formulation is known as the surplus production model [23].

4.2 Schaefer's production model

Schaefer's production model (Schaefer, 1954 [34]) is the most widely used surplus production model in fish stock assessments. The model is based on the following differential equation,

$$\frac{dB(t)}{dt} = rB(t)\left(1 - \frac{B(t)}{K}\right) - C(t), \tag{4.5}$$

where $B(t)$ is the biomass of the stock at time t, r is an intrinsic rate of population growth, K is a parameter that corresponds to the unfished equilibrium stock size (carrying capacity), and $C(t)$ is the catch measured as a rate (e.g., tons per time) [23].

A simple differenced equation approximate solution to Schaefer's original model (Eq. 4.5) is often used (e.g. Walters and Hilborn, 1976 [39]),

$$B_{t+1} = B_t + rB_t\left(1 - \frac{B_t}{K}\right) - C_t, \tag{4.6}$$

where B_t is the biomass at time t, r, and K have the same meanings as in Schaefer's original model and C_t is the catch during the time t.

4.2.1 Schaefer's contemporary production model

We make a modification to the production model given above (Eq. 4.6) by dividing both sides by K. Let $P_t = B_t/K$, and an alternative version of the production model is,

$$P_{t+1} = P_t + rP_t(1 - P_t) - C_t/K. \tag{4.7}$$

We refer to this model as the "Contemporary surplus production model" and all the analyses conducted in this chapter are based on this model and its state space version. We are mainly interested in estimating the following parameters: initial biomass B_0 (or initial production P_0), intrinsic growth rate r, and carrying capacity K.

Parameter estimation : Schaefer's contemporary production model

In most situations, we do not get direct estimates for population biomass (B_t) to estimate surplus production model parameters r and K. However, surveys often provide an index (Eq. 4.1) of stock biomass. Usually, we assume that the catches are measured without errors, and they are considered as fixed covariates. Further, we assume that the model given in Eq. 4.7 has no process error and all of the error assumed to be occurring in the relationship between stock biomass (B_t) and the index of relative abundance (I_t). This is referred to as the "observational error approach". The common statistical observation equation is,

$$I_t = qB_t \exp(\epsilon_t), \tag{4.8}$$

where $\exp(\epsilon_t)$ are the residual errors which are assumed to be log-normally distributed where $\epsilon_t \sim N(0, \sigma^2)$. In this situation σ^2 reflects survey variance and other variations related to how well the survey covers the stock range. Many surveys occur approximately at mid year and in this case we use the observation equation,

$$I_t = qB_{(t+\frac{1}{2})} \exp(\epsilon_t).$$

We get the mid year biomass $B_{(t+\frac{1}{2})}$ by simply averaging two adjacent year's biomass,

$$B_{(t+\frac{1}{2})} = \frac{B_{t+1} + B_t}{2}.$$

The residual of the log index is $e_t = \log(I_t) - \log(E(I_t))$, where $E(I_t)$ is the expected value of the index I_t. Since we are using the production function in the form of

Eq. 4.7 (where, $P_t = B_t/K$), we can express the expected value of the index as $E(I_t) = E(qKP_{(t+\frac{1}{2})}\exp(\epsilon_t)) \approx qKP_{(t+\frac{1}{2})}$, where $P_{(t+\frac{1}{2})}$ is the mid year production $P_{(t+\frac{1}{2})} = \frac{P_{t+1}+P_t}{2}$ and K is the usual carrying capacity.

Next, we discuss how to use TMB to estimate parameters for Schaefer's contemporary surplus production model described above. We use annual catch data and CPUE data for northern Namibian hake obtained from Polachek, 1993 [31]. As the first step, we need to enter the production model given in Eq. 4.7 into the C++ user template and invoke the model using the TMB package in R. The full C++ source code is given in Appendix A.3. However, for illustration purposes, we discuss a few important parts of the code below.

We declare data using key words DATA_VECTOR (reals) or DATA_IVECTOR (integers) and parameters are declared using the key word PARAMETER(),

```
DATA_IVECTOR(year);
DATA_VECTOR(C);
DATA_VECTOR(index);
    .
    .
    .
PARAMETER(log_r);
PARAMETER(log_K);
PARAMETER(log_q);
PARAMETER(log_Po);
PARAMETER(log_sd_log_index);
```

It is often difficult to reliably estimate all the production model parameters and q. Long time-series with high levels of contrast in catch relative to MSY and survey indices are necessary to reliably estimate parameters. In practice P_o is assumed to be one, particularly for stocks that were known to be only lightly harvested before the first year of assessment data. Rather than making this strict assumption, we use a 'prior' distribution on P_o in which the user sets E($\log(P_o)$) and var($\log(P_o)$). Hence, subjective uncertainty about P_o can be included in model inferences. This prior

distribution is included as a negative loglikelihood component. Below, we write the negative log likelihood for initial production **Po**. Similar to R, in C++ the function **dnorm** also provides the density for a normal distribution. **E_log_Po** is the mean of the log of initial production and **sd_log_Po** is its standard deviation.

```
nll -= dnorm(log_Po,E_log_Po,sd_log_Po,true);
```

The contemporary surplus production model given in Eq. 4.7 is declared as,

```
    P(0) = exp(log_Po);
    for (i=1;i<n;i++){
      P(i) = (P(i-1) + r*P(i-1)*(one - P(i-1)) - C(i-1)/K);
}
log_P = log(P);
log_B = log_K + log_P;
```

The mid year production **P_midy** and harvest rate **H** can be found as follows,

```
for (i=0;i<n-1;i++){
  P_midy(i) = half*(P(i)+P(i+1));
}
int ln=n-1;
Type Pnp1 = (P(ln) + r*P(ln)*(one - P(ln)) - C(ln)/K);
P_midy(ln) = half*(P(ln)+ Pnp1);
log_P_midy = log(P_midy);

log_H = log_C - log_B;
H = exp(log_H);
```

Next we provide the code to find the expected log index value and the residual of the log index.

```
  log_Eindex = log_q + log_K + log_P_midy(iyear);
  vector<Type> resid = log_index - log_Eindex;
  vector<Type> std_resid = resid/sd_log_index;
```

The negative log likelihood for the residuals of log index is,

```
nll -= (dnorm(resid,zero,sd_log_index,true)).sum();
```

The rest of the program produces **report** output;

```
  REPORT(log_r);
  REPORT(log_K);
  REPORT(log_q);
  REPORT(log_Po);
     .
     .
     .
  ADREPORT(log_B);
  ADREPORT(log_H);

  return nll;
}
```

Next, we discuss the R code for parameter estimation. In the first line we load the TMB package into our R session and in the next two lines we compile and link the C++ user template. The last line imports the data (created using other R code) to the R working environment. The file tmb.RData includes a data frame names tmb.data. Note that all the data components in tmb.data must have the same names as the DATA_VECTOR names in the C++ user template.

```
library(TMB)
compile("fit.cpp")
dyn.load(dynlib("fit"))
load("tmb.RData")
```

Next, we provide initial values for all the parameters which are to be estimated. For **log_q**'s we need to assign multiple values according to the number of indices (from different surveys or commercial CPUE from different fleets) available for the study. In this example we have only one index (CPUE); therefore, we only assign one value. The initial values for parameter estimates are given as,

```
parameters <- list(
  log_r = log(0.4),
  log_K = log(2700),
  log_q = log(1/1000),
  log_Po = log(1),
  log_sd_log_index = log(0.3)
)
```

The TMB template only returns the nll. Estimation is performed using a R function minimizer. We prefer to use nlminb(). Minimization is often improved using sensible lower and upper bounds on parameter values that prevents optimizers from straying into infeasible parameter space or extreme regions of the parameter space where the nll surface may be nearly flat and cause the optimizer to diverge. We provide appropriate upper and lower bounds for those parameters which need to be estimated.

```
parameters.L <- list(
  log_r = log(0.3),
  log_K = log(2000),
  log_q = -Inf,
  log_sd_log_index = log(0.01))

parameters.U <- list(
  log_r = log(0.5),
  log_K = log(5000),
  log_q = Inf,
  log_sd_log_index = log(1))
```

The line that begins **obj <- MakeADFun** defines the object **obj** containing the data, parameters and methods that access the objective function and its derivatives.

```
obj <- MakeADFun(tmb.data,parameters,DLL="fit",
inner.control=list(maxit=100,trace=T)).
```

Finally, we use the **nlminb** R optimizer to minimize the objective function **obj$fn** aided by the gradient **obj$gr** and starting at the point **obj$par**.

```
opt<-nlminb(obj$par,obj$fn,obj$gr,lower=lower,upper=upper,
control = list(trace=0,iter.max=500,eval.max=1000)).
```

Below, the first line shows the convergence status of the optimization. The second line contains the gradients at the optimized parameter estimates. The final line gives the parameter estimates for the model.

```
opt$message
obj$gr(opt$par)
opt$par
```

A summary of the estimated parameters is given in the table below. Under the column **OE - Estimate**, results obtained by Polacheck et al., 1993 [31] are given. They used the maximum likelihood estimation method to estimate the parameters and used observation error estimators approach to fit the surplus production model (Eq. 4.6). We can see that most of the parameter estimates are similar in both studies.

Table 4.1: Summary of parameters estimated using Schaefer's contemporary surplus production model for northern Namibian hake data.

Parameter	TMB - Estimate	OE - Estimate
r	0.3630	0.379
K	2824	2772.6
q	4.48	4.36
index_log_sd	0.1208	0.124

4.2.2 State space formulation for Schaefer's model

In this study we also use a state space version of Schaefer's model based on the model given in Eq. 4.7. The state equation of the model is,

$$P_{t+1} = \{P_t + rP_t(1 - P_t) \ - H_t P_t\} \ exp(\varepsilon_t), \tag{4.9}$$

where $H_t = C_t/B_t$ is the harvest rate at time t and ε_t is the process error. This process error vector ε_t is generated from an AR(1) stochastic process, $\varepsilon_t = \varphi\varepsilon_{t-1} + \gamma_t$, where γ_t is independent and identically distributed, zero-mean normal vectors with covariance matrix Σ [36], and φ is a scalar autocorrelation parameter that is common to all elements of ε_t. The process errors have stationary distribution,

$$\lim_{t\to\infty} var(\varepsilon_t) = \frac{\sigma_\varepsilon^2}{1 - \varphi_\varepsilon^2}, \tag{4.10}$$

and the covariance and correlation between $\varepsilon_t, \varepsilon_{t-1}$ are,

$$Cov(\varepsilon_t, \varepsilon_{t-1}) = \frac{\sigma_\varepsilon^2 \varphi_\varepsilon}{(1 - \varphi_\varepsilon^2)}, \ and \ Corr(\varepsilon_t, \varepsilon_{t-1}) = \varphi_\varepsilon. \tag{4.11}$$

We model the harvest rate H_t as a random walk,

$$log(H_{t+1}) = log(H_t) + \delta_t, \tag{4.12}$$

where δ_t is the log harvest rate deviation at time t and $\delta_1, ..., \delta_t \overset{iid}{\sim} N(0, \sigma_\delta^2)$.

The state space observations equations are,

$$log(I_t)|B_t \overset{iid}{\sim} N(log(qB_t), \sigma_I^2), \tag{4.13}$$

where I_t is the index of relative abundance at time t. Sometimes there may be several survey indices available, or a combination of survey and CPUE indices. Different index catchability (q) parameters are estimated for each survey index, but the measurement error variance (σ_I^2) may or may not be assumed to be the same for all indices.

Let C_{pt} denote the total model predicted catch,

$$C_{pt} = H_t B_t = H_t K P_t.$$

The log of the total model predicted catch is,

$$log(C_{pt}) = log(H_t) + log(B_t). \tag{4.14}$$

The catch observation equation we use is,

$$log(C_t)|B_t; H_t \overset{iid}{\sim} N(log(C_{pt}), \sigma_C^2) \tag{4.15}$$

where $log(C_t)$ is the log of observed catch and σ_C is the standard deviation of the $log(C_t)$.

Parameter Estimation : State space surplus production model

We implement the model using TMB in R. In the C++ user template we formulate the production model and error structures. As input data, we use time-series of survey indices and aggregate catch data. We estimate variance parameters based on the marginal likelihood in which the random effects have been integrated out. We can predict the random effects based on the joint likelihood with fixed effects parameters fixed at their MLE values. We denote the vector of all random effects by Γ and fixed effects parameters by the vector θ. Fixed effects parameters are estimated by integrating out the random effects from the joint density function of the response indices and random process errors. This method is usually known as the marginal maximum likelihood estimation (MMLE). Let S denote the set of all data, i.e., CPUE, indices and catches, used in the model. Therefore the marginal likelihood is

$$L(\theta) = \iiint_{\Gamma} f_\theta(S|\Gamma) g_\theta(\Gamma) \partial\Gamma \tag{4.16}$$

where $f_\theta(S|\Gamma)$ is the probability density function (pdf) of the data, conditional on

the random effects, Γ, and $g_\theta(\Gamma)$ is the joint pdf for the Γ random effects.

We fit the northern Namibian hake data from Polackeck et al., 1993 [31] to the state space model described above. In the state space model there are extra variance parameters to estimate that specify the distribution of these random effects, and then the random effects may be predicted as we described above. Full C++ and R codes can be found in Appendix A.4. However, here we use some sections of the code to explain some important points we should consider when fitting the model and the parameter estimation.

In addition to the parameters declared in the contemporary model, we add the log of: initial harvest rate (Ho), standard deviation of harvest rate deviations (sd_rw), standard deviation of process errors (sd_pe), process error (pe), harvest rate deviation (H_dev), and logit of process error auto-correlation (ar_pe) to the state space C++ source code.

```
PARAMETER(log_Ho);
PARAMETER(log_sd_rw);
PARAMETER(log_sd_pe);
PARAMETER(logit_ar_pe);
PARAMETER_VECTOR(log_pe);
PARAMETER_VECTOR(log_H_dev);

vector<Type> pe = exp(log_pe);
```

The log of production model is,

```
for (i=1;i<n;i++){
  log_H(i) = log_H(i-1) + log_H_dev(i-1);
  H(i) = exp(log_H(i));
  P(i) = (P(i-1)+r*P(i-1)*(one-P(i-1))-H(i-1)*P(i-1))*pe(i-1);
}
```

Above, the second line represents the harvest rate random walk given in Eq. 4.12. The next lines include the state equation of the state space model given in Eq.

4.9. Therefore we have negative log likelihoods for catches, random walk deviations for the log of the harvest rate, and process errors additional to the negative log likelihoods for indices in the contemporary model.

```
// nll for catch;
  vector<Type> resid_C = log_C - log_EC;
  nll -= dnorm(resid_C,zero,sd_logC,true).sum();

// nll for random walk deviation in log_H;
  nll -= dnorm(log_H_dev,zero,sd_rw,true).sum();

//   nll for log_pe process errors;
  i=0;
  nll -= dnorm(log_pe(i),zero,sd_pe/sqrt(one - ar_pe*ar_pe),true);
  for(int i = 1;i < n;++i){
    nll -= dnorm(log_pe(i) - ar_pe*log_pe(i-1),zero,sd_pe,true);
  }
```

In R, we first need to assign starting values for the parameters and their lower, upper bounds.

```
parameters <- list(
  log_r = log(0.36),
  log_K = log(2800),
  log_q = log(1/10),
  log_Po = log(0.5),
  log_Ho = log(0.1),
  log_sd_rw = log(0.2),
  log_sd_log_index = log(0.3),
  log_sd_pe = log(0.1),
  logit_ar_pe = log(0.50/(1-0.50)),
  log_pe = rep(0,length(tmb.data$C)),
```

```
  log_H_dev = rep(0,length(tmb.data$C)-1)
)

parameters.L <- list(
  log_r = log(0.2),
  log_K = log(2000),
  log_q = -Inf,
  #log_Po = log(0.1),
  log_Ho = log(0.0001),
  log_sd_rw = log(0.01),
  log_sd_log_index = log(0.01),
  log_sd_pe = log(0.05),
  logit_ar_pe = log(0.01/(1-0.01)))

parameters.U <- list(
  log_r = log(0.5),
  log_K = log(4271),
  log_q = Inf,
  #log_Po = log(10),
  log_Ho = log(1),
  log_sd_rw = log(2),
  log_sd_log_index = log(1),
  log_sd_pe = log(0.35),
  logit_ar_pe = log(0.950/(1-0.950)))
```

We introduce the random effects to the model as,

```
rname = c("log_pe","log_H_dev")
```

and they are assigned to the **random** argument in **MakeADFun**, the objective function,

```
obj <- MakeADFun(tmb.data,parameters,map=map,random=rname,DLL="fit",
  inner.control=list(maxit=100,trace=T))
```

The rest of the code is the same as in Schaefer's contemporary surplus production model parameter estimation, described in Section 4.2.1. A summary of the results obtained for the parameter estimates is given in the following table. Under columns **OE - Estimate** and **PE - Estimate**, we give the results obtained by Polachek et al. [31] using the observation error approach and process error approach, respectively.

Table 4.2: Summary of parameters estimated using the state space version of Schaefer's surplus production model for northern Namibian hake data.

Parameter	TMB - Estimate	PE - Estimate	OE - Estimate
r	0.346	0.304	0.379
K	2934	3448	2772
q	3.705	2.701	4.360
index_log_sd	0.0932	0.662	0.124

4.3 Case studies

We conduct five case studies in this practicum for data from: 3LN redfish, 3LNO yellowtail flounder, Divisions 8c and 9a anglerfish, Greenland halibut, and Divisions IVa and VIa megrim. A brief introduction to data and a summary of a few important parameter estimates (r, K, Po, and $sd\_\log\_\text{index}$) for each case study are also given.

4.3.1 Introduction to data and parameter estimates

Redfish

Data for this analysis are obtained from an ASPIC Based Assessment of Redfish (S. mentella and S. fasciatus) in NAFO Divisions 3LN by A. M. Avila de Melo et al. (2014) (document number: NAFO SCR Doc. 14/022). Annual catch data are available for 1959 to 2013 and eight indices (CPUE, 3LN_SPRG, 3LN_FALL, 3LN_RSSN, 3L_WNTR, 3L_SUMR, 3L_FALL, and 3N_SPNH) were used for this study. In Figures 4.2 and 4.3, we plotted the catches and indices used for this study.

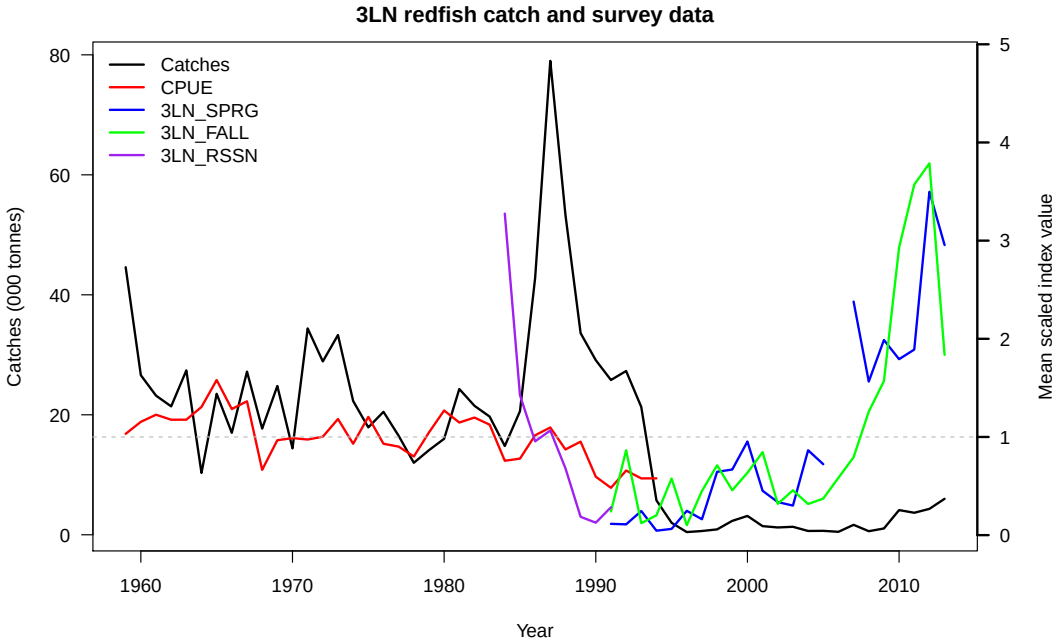


Figure 4.2: Catch data(black line) are in 10^3 tonnes and indices are mean scaled values.

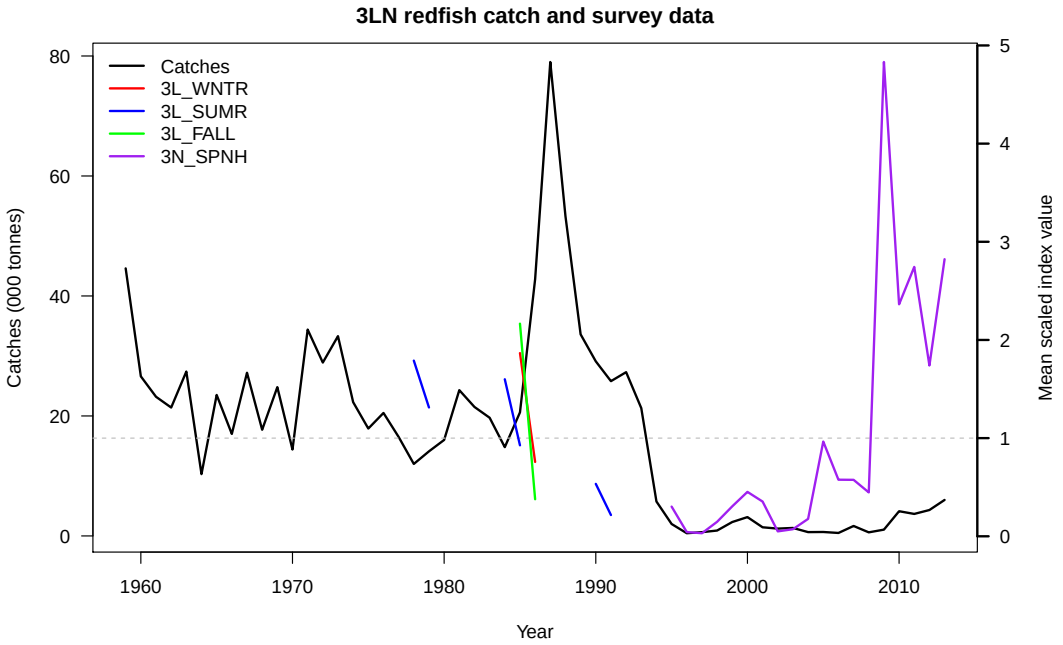


Figure 4.3: Catch data(black line) are in 10^3 tonnes and indices are mean scaled values.

Table 4.3: Critical parameter estimates and their coefficient of variations for redfish data using contemporary and state space production models.

Parameter	Contemporary		SSM	
	Estimate	CV	Estimate	CV
r	0.2518	0.116	0.2303	0.424
K (000s)	415.4323	0.285	437.0644	0.446
Po	0.3689	0.434	0.6480	0.417
index_sd	0.6273	0.231	0.5688	0.071

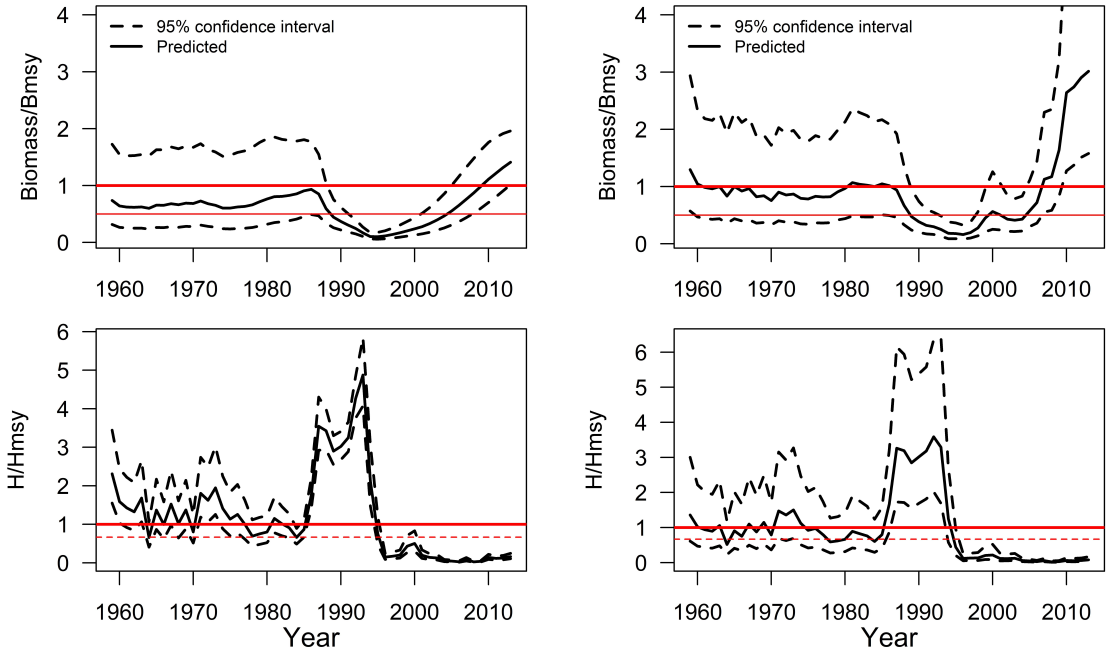


Figure 4.4: Production model for redfish data with a prior on Po: Biomass and exploitation rates (H). Left: contemporary surplus production model, right: state space model.

Yellowtail flounder

Data for this analysis are obtained from the assessment of NAFO Div-3LNO Yellowtail Flounder by Maddock Parsons et al. (document number: NAFO SCR Doc. 15/029). Catch data are available from 1965 to 2015 and four indices (Yankee, Russian, Spring, and Fall) were used for this study.

In Figure 4.5 and 4.6, we plotted the catches and indices used for this study.

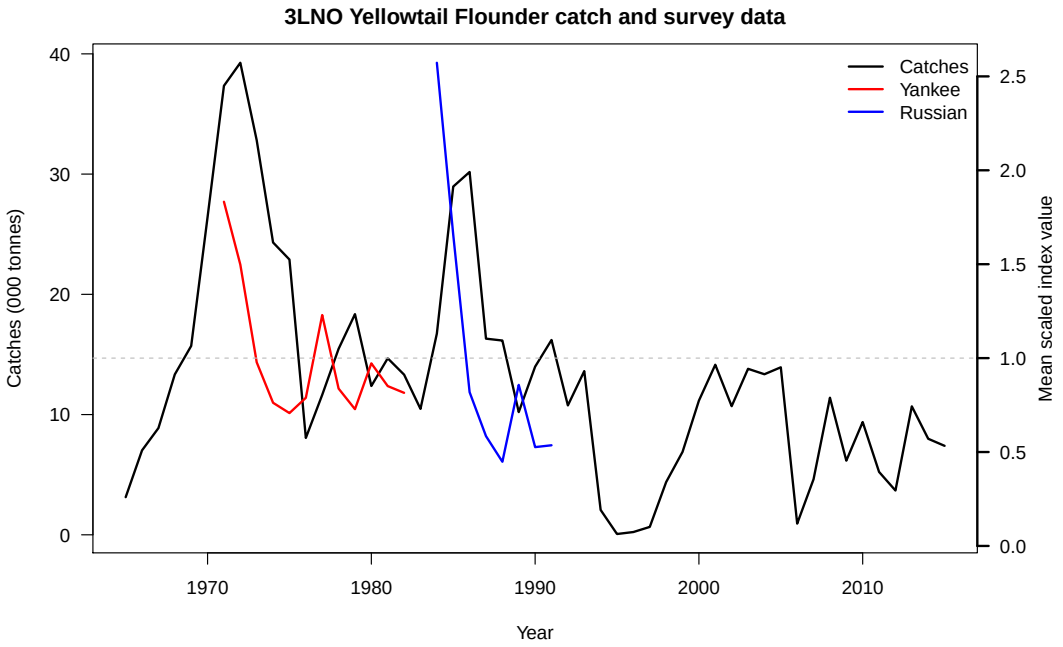


Figure 4.5: Catch data(black line) are in 10^3 tonnes and indices are mean scaled values.

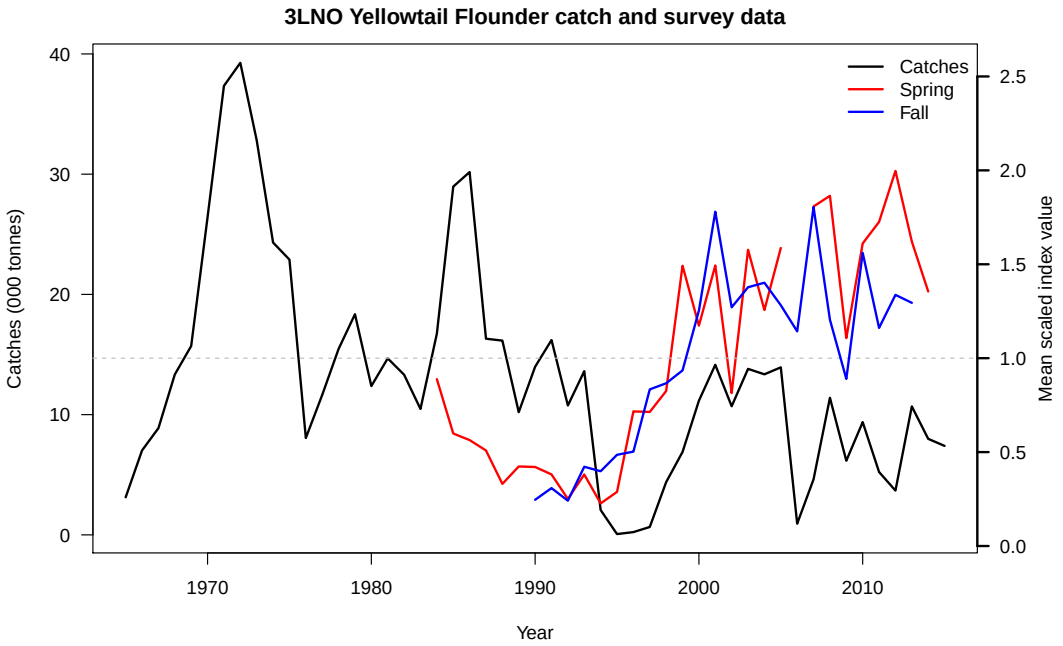


Figure 4.6: Catch data(black line) are in 10^3 tonnes and indices are mean scaled values.

Table 4.4: Critical parameter estimates and their coefficient of variations for yellowtail flounder data using contemporary and state space production models.

Parameter	Contemporary		SSM	
	Estimate	CV	Estimate	CV
r	0.5611	0.093	0.5440	0.101
K (000s)	0.1401	0.057	0.1421	0.084
Po	1.3911	1.372	0.7183	0.705
index_sd	0.073	0.231	0.2761	0.074

Anglerfish

Data for this analysis are obtained from the ICES Working Group for the Bay of Biscay and the Iberian waters Ecoregion (WGBIE) report 2016 in Divisions 8c and 9a. Catch data are available from 1981 to 2015. We also use Spanish and Portuguese C, and F survey results as the indices.

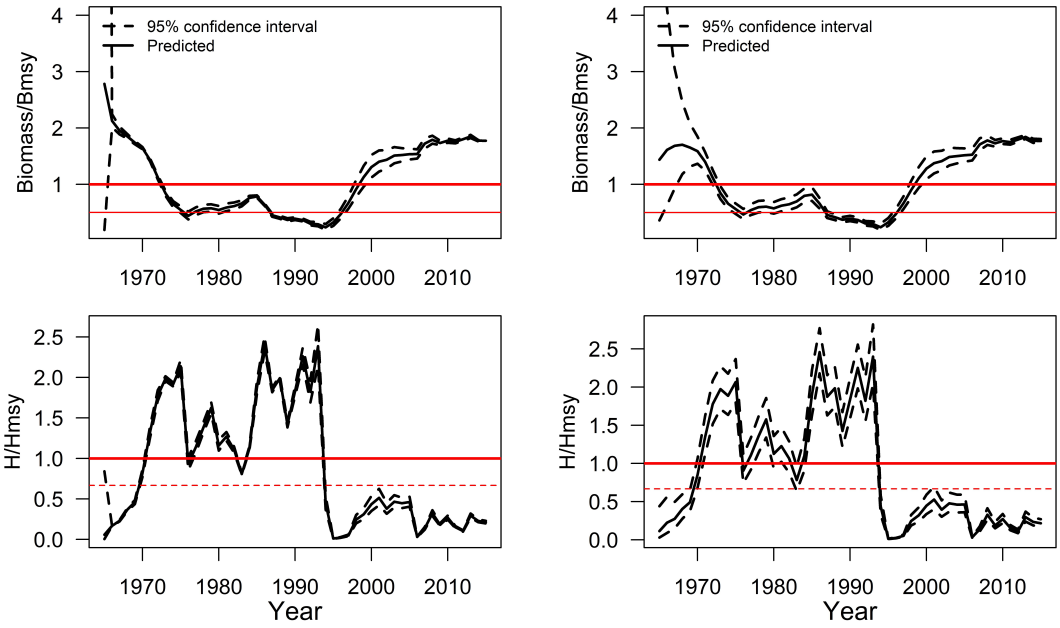


Figure 4.7: Production model for yellowtail flounder data with a prior on Po: Biomass and exploitation rates (H). Left: contemporary surplus production model, right: state space model.

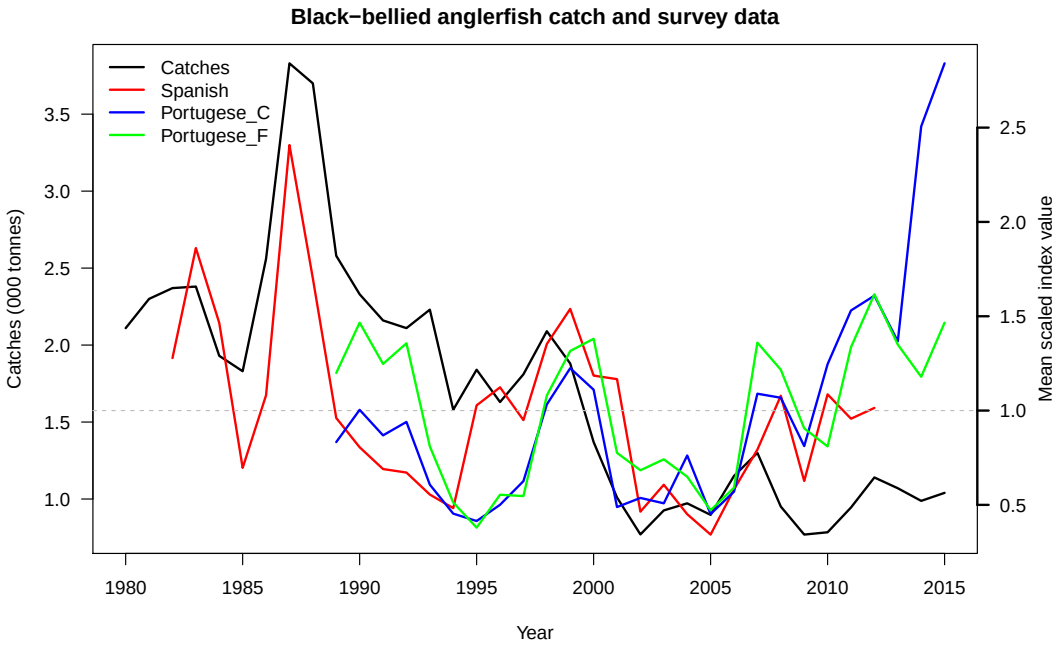


Figure 4.8: Catch data(black line) are in 10^3 tonnes and indices are mean scaled values.

Table 4.5: Critical parameter estimates and their coefficient of variations for anglerfish data using contemporary and state space production models.

Parameter	Contemporary		SSM	
	Estimate	CV	Estimate	CV
r	0.2320	0.204	0.3528	0.711
K (000s)	0.0318	0.160	0.0234	0.726
Po	0.5995	0.020	0.6004	0.020
index_sd	0.3929	0.077	0.2978	0.093

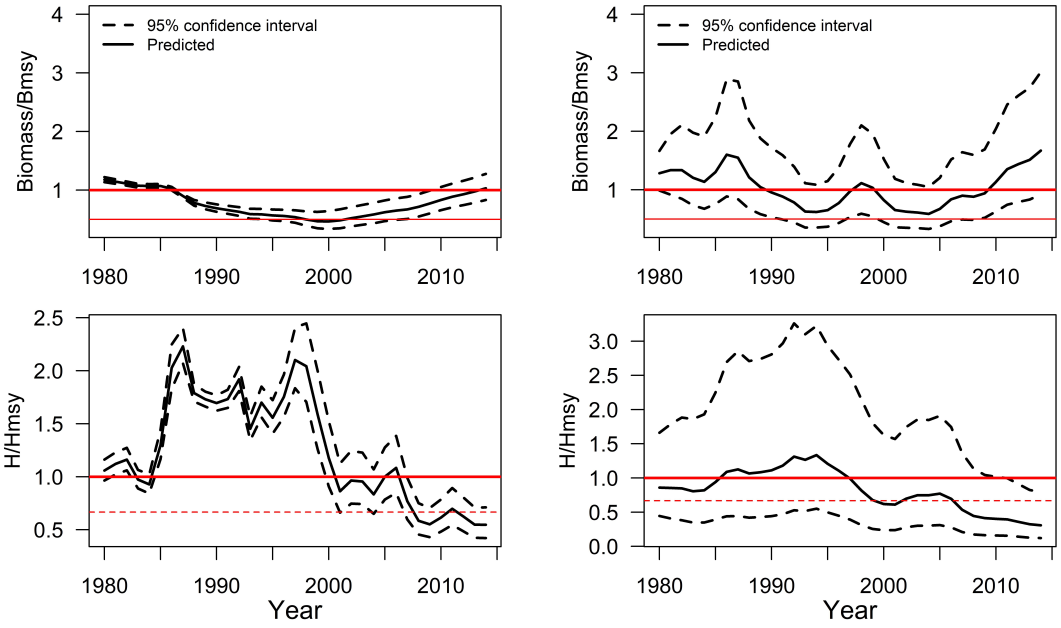


Figure 4.9: Production model for anglerfish data with a prior on Po: Biomass and exploitation rates (H). Left: contemporary surplus production model, right: state space model.

Greenland halibut

Data for this analysis are obtained from the stock assessment carried out by North-Western Working Group Ecoregion (NWWG) in sub areas 5, 6, 12, and 14 (Iceland and Faroes grounds, West of Scotland, North of Azores, East of Greenland). The catch data are available from 1985 to 2015, measured in tonnes and we have two

indices: standardized series of annual commercial-vessel catch rates for 1985-2015, (CPUE), and a combined trawl-survey biomass index for 1996-2015, (Survey).

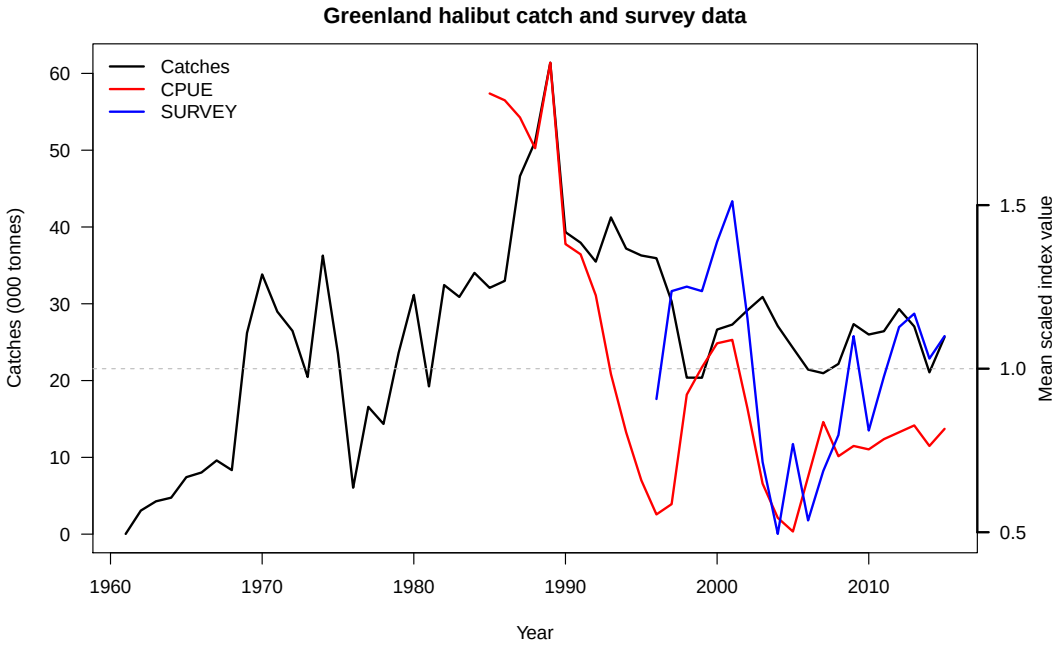


Figure 4.10: Catch data(black line) are in 10^3 tonnes and indices are mean scaled values.

Table 4.6: Critical parameter estimates and their coefficient of variations for Greenland halibut data using contemporary and state space production models.

Parameter	Contemporary		SSM	
	Estimate	CV	Estimate	CV
r	0.4514	0.267	0.2178	0.463
K (000s)	0.2906	0.205	0.6263	0.416
Po	1.0000	0.500	0.6710	0.441
index_sd	0.2216	0.102	0.1344	0.120

Megrim

Data for this analysis are obtained by the stock assessment carried out by Working Group for the Celtic Seas Ecoregion (WGCSE) in the northern North Sea, West of

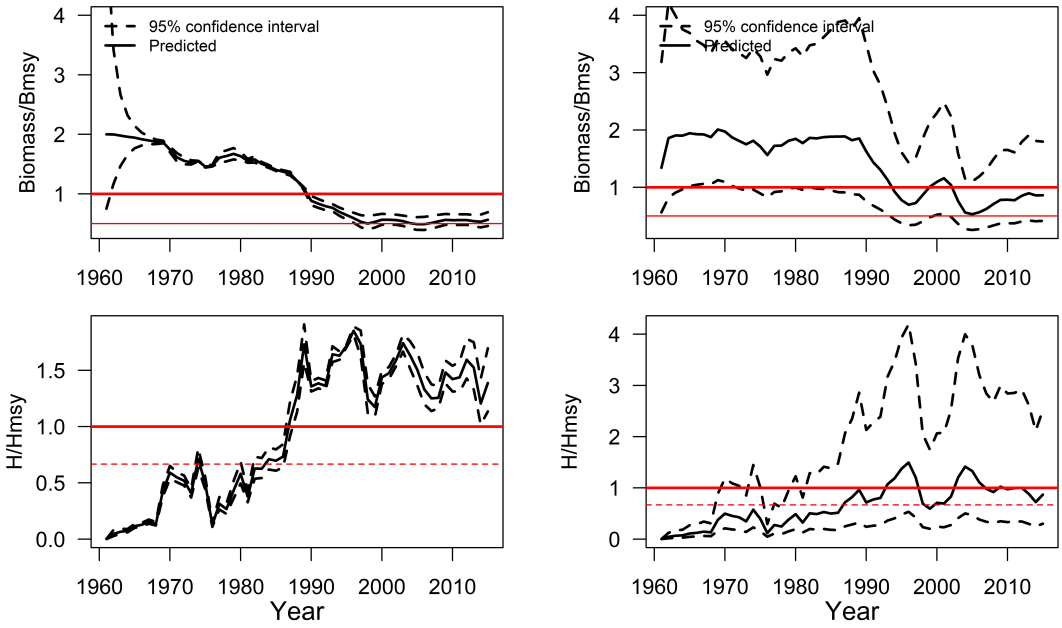


Figure 4.11: Production model for Greenland halibut data with a prior on Po: Biomass and exploitation rates (H). Left: contemporary surplus production model, right: state space model.

Scotland in 2016. The catch data are available from 1985 to 2014, measured in tonnes and we have indices from six independent surveys conducted in divisions 4.a and 6.a (SCOGFS_WIBTS_Q1, SCOGFS_WIBTS_Q4, SCO_IBTS_Q1, SCO_IBTS_Q3, SAMISS_Q2, and IAMISS_Q2).

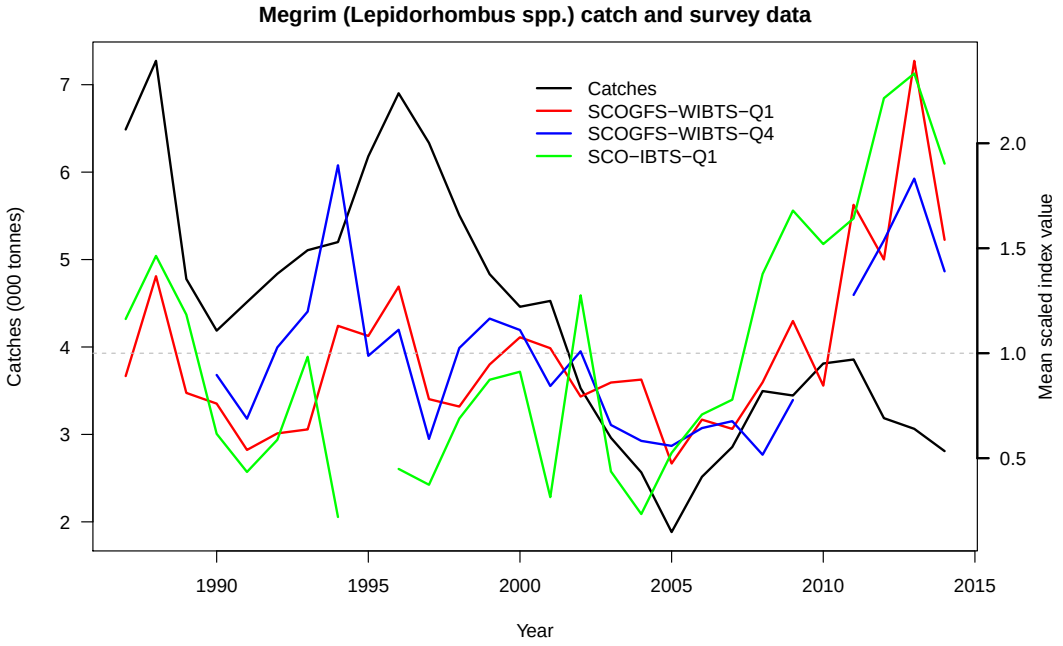


Figure 4.12: Catch data(black line) are in 10^3 tonnes and indices are mean scaled values.

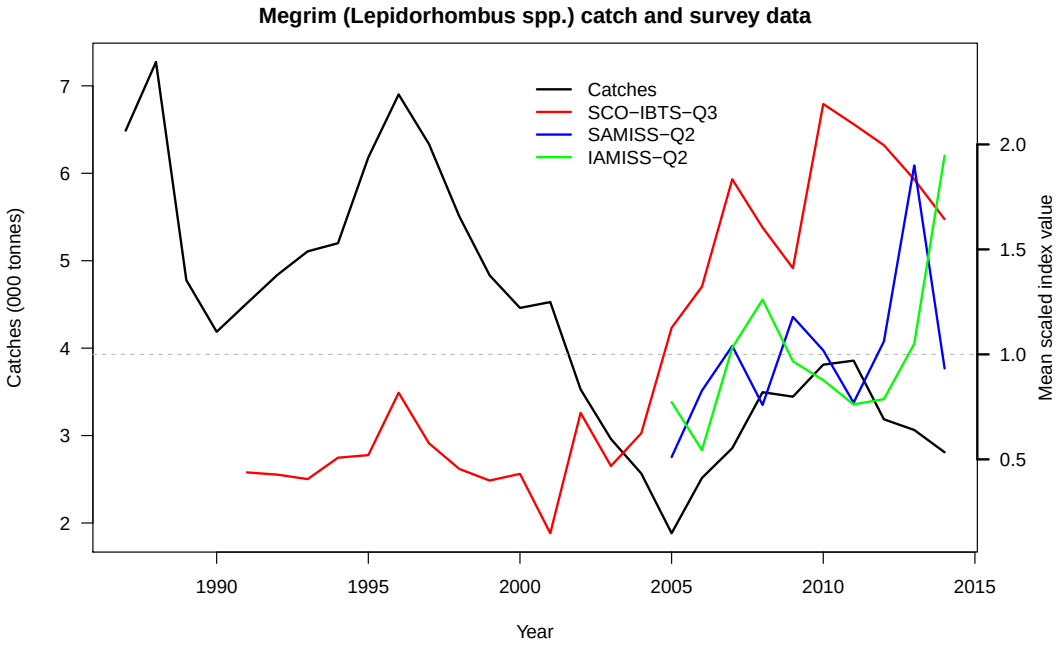


Figure 4.13: Catch data(black line) are in 10^3 tonnes and indices are mean scaled values.

Table 4.7: Critical parameter estimates and their coefficient of variations for megrim data using contemporary and state space production models.

Parameter	Contemporary		SSM	
	Estimate	CV	Estimate	CV
r	0.3994	0.224	0.6452	0.723
K (000s)	58.5453	0.306	30.7072	0.693
Po	0.3546	0.216	0.8918	0.505
index_sd	0.4103	0.063	0.3938	0.065

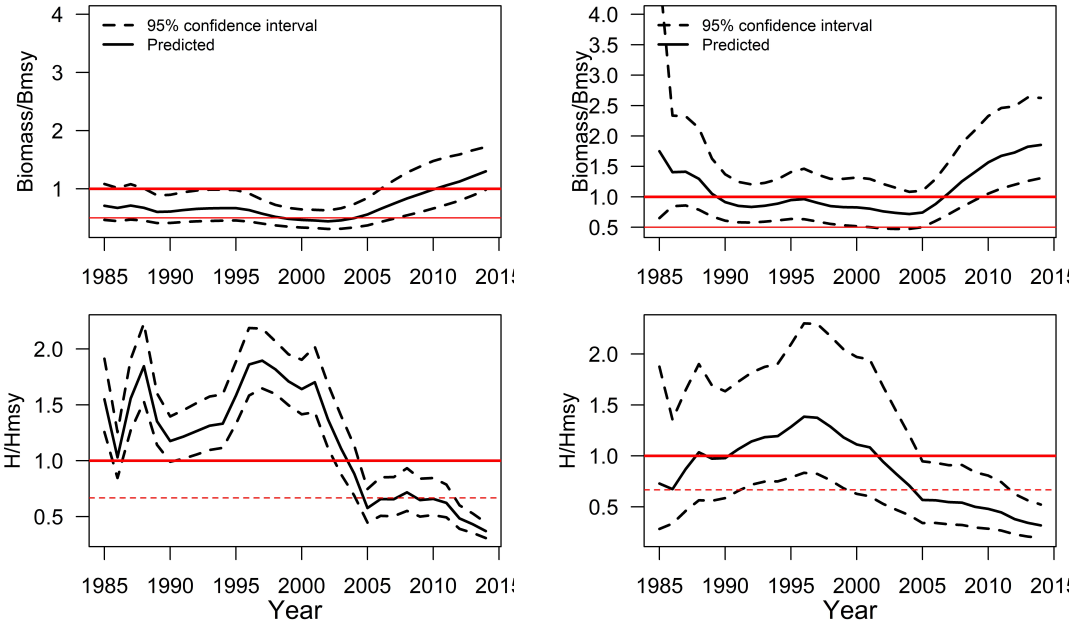


Figure 4.14: Production model for megrim data with a prior on Po: Biomass and exploitation rates (H). Left: contemporary surplus production model, right: state space model.

Table 4.8: Some parameter estimations of the state space model are summarized here. Ho: initial harvest rate, sd_rw: standard deviation of harvest rate deviations, sd_pe: standard deviation of process errors, process error (pe), and logit_ar_pe: logit of process error auto-correlation. (For halibut data logit_ar_pe is fixed to -10)

Stock	Ho	sd_rw	sd_pe	logit_ar_pe
Redfish	0.1562	0.4920	0.2376	-4.5951
Yellowtail flounder	0.0231	0.8494	0.0010	-4.5951
Anglerfish	0.1499	0.1458	0.1506	-0.5414
Halibut	0.0001	0.6851	0.1747	-10.0000
Megrim	0.2324	0.1409	0.0859	0.1419

Chapter 5

Diagnostics and Comparisons

5.1 Influence diagnostics

5.1.1 Case deletion diagnostics for indices

In this chapter, we assess the influence of the surplus production models for small changes made in input data using two methods. The first is the traditional case deletion method and the second is the local influence diagnostic method introduced by Cook, 1986 [12]. We apply both these methods to indices data and fit both the contemporary surplus production model (SPM) and the state space model (SSM). We compare diagnostic results mainly in two directions: a) Correspondence of case deletion and local influence diagnostic methods, and b) comparison of the sensitivity of two production models in use (SPM and SSM). We also apply the local influence method to perturbations of catch data.

Contemporary surplus production model

We are interested in whether the state-space of contemporary production models are more sensitive to changes in the input data. For this investigation, we use the traditional case deletion diagnostic method and the local influence diagnostic method introduced by Cook, 1986 [12]. We use case deletion diagnostics for indices as the

first method.

The following hypothetical example contains annual aggregated catch data (C_t) for n years, and k number of indices ($I_1,\ I_2, \ldots, I_k$) obtained from k individual surveys carried out for a certain fish type. Although we have all the indices for every year in this example, in real life situations, we might not have an index/indices for some years. However, for illustration purposes, we assume that all the indices are available for all the years (from 1 to n).

Table 5.1: Hypothesized catch and indices data available for a certain fish stock assessment.

$Year$	C_t	I_1	I_2	$\ldots$	I_k
Y_1	C_1	I_{11}	I_{21}	$\ldots$	I_{k1}
Y_2	C_2	I_{12}	I_{22}	$\ldots$	I_{k2}
Y_3	C_3	I_{13}	I_{23}	$\ldots$	I_{k3}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Y_n	C_n	I_{1n}	I_{2n}	$\ldots$	I_{kn}

The basic idea of the case deletion technique is to delete each index ($I_{i,j}$) (where $i = 1, \ldots, k$ and $j = 1, \ldots, n$) one at a time and re-estimate the model parameters. This is done iteratively in our R code by setting a weight of zero for each index in each iteration. In this example, we have to estimate the parameters $n \times k$ times.

Weighting the contemporary surplus production model is straightforward. We simply assign weights to the log likelihood function as,

$$l_w(\theta) = \sum_{h=1}^{m} w_h l_h(\theta), \tag{5.1}$$

where $l_h(\theta)$ is the log likelihood for the h^{th} case, w_h is the h^{th} element of the vector of weights (w), and $m = n \times k$. Initially we declare these weights as a vector of $1's$, $w = (1, 1, \ldots, 1)'$. If we want to delete a case (say h^{th} index), we simply set the weight of h^{th} element $w_h = 0$.

Parameter estimation : Contemporary surplus production model

The parameter estimation method is almost the same as discussed in Section 4.2.1 (parameter estimation for the Schaefer's contemporary production model). The only difference is that we need to add the vector of weights to both the C++ source code and R code. We illustrate this using one of the five fish stocks we study in this practicum, **3LN redfish** (S. mentella and S. fasciatus).

We have to make two changes to the C++ source code. First, we add the data vector for weights, **DATA_VECTOR(index_wt)**. We write the negative log likelihood for the indices as follows,

```
nll -= (index_wt*dnorm(resid,zero,sd_log_index,true)).sum().
```

In our R code, we add the vector of weights as,

```
tmb.data$index_wt = rep(1,length(tmb.data$index))
```

where **tmb.data** is our data frame, **length(tmb.data$index)** is the total number of indices available for this study (recall: **m** in Eq. 5.1). We usually do not need to change the first few lines in the R code which loads the **TMB** package and compiles the **C++** user template. However, we need to provide appropriate starting values for each and every parameter to be estimated. Since we have multiple indices, we need to provide starting values for all of the catchability coefficients (q's). For example,

```
log_q = rep(1,length(unique(tmb.data$iq))),
```

where **unique(tmb.data$iq)** is how many survey indices we are using for this example. The case deletion analysis is conducted iteratively using a **for-loop**. In each iteration we assign a zero-weight to the corresponding element in the weight vector as,

```
for(i in 1:n.index){
  tmb.data$index_wt[i]=0
```

```
obj <- MakeADFun(tmb.data,parameters,DLL="fit",
       inner.control=list(maxit=10,trace=T))

opt <- nlminb(opt$par,obj$fn,obj$gr,lower=lower,upper=upper,
       control = list(trace=0,iter.max=5000,eval.max=10000))

rep.index_del[[i]] = obj$report()
}
```

where **n.index** is the total number of indices available for this study. The objective function and the optimization are implemented within the for loop and **rep.index_del**[[**i**]] stores all the output produces by **report**. In this study, we are interested in finding influential indices for the parameters B_{MSY} and H_{MSY}. With this method, we consider each index as a case. We scaled re-estimated parameter values as a percentage to original parameter estimates and the results are plotted below.

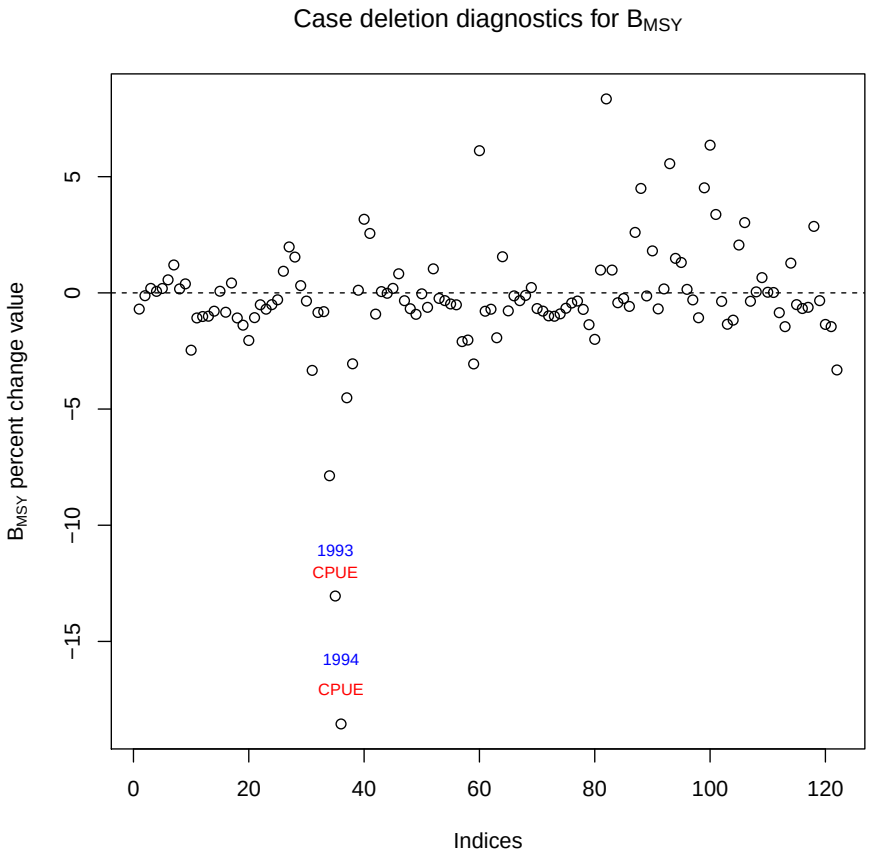


Figure 5.1: Index deletion diagnostics: redfish data for the contemporary surplus production model. The points are B_{MSY} percent difference values of deletion results compared to original results.

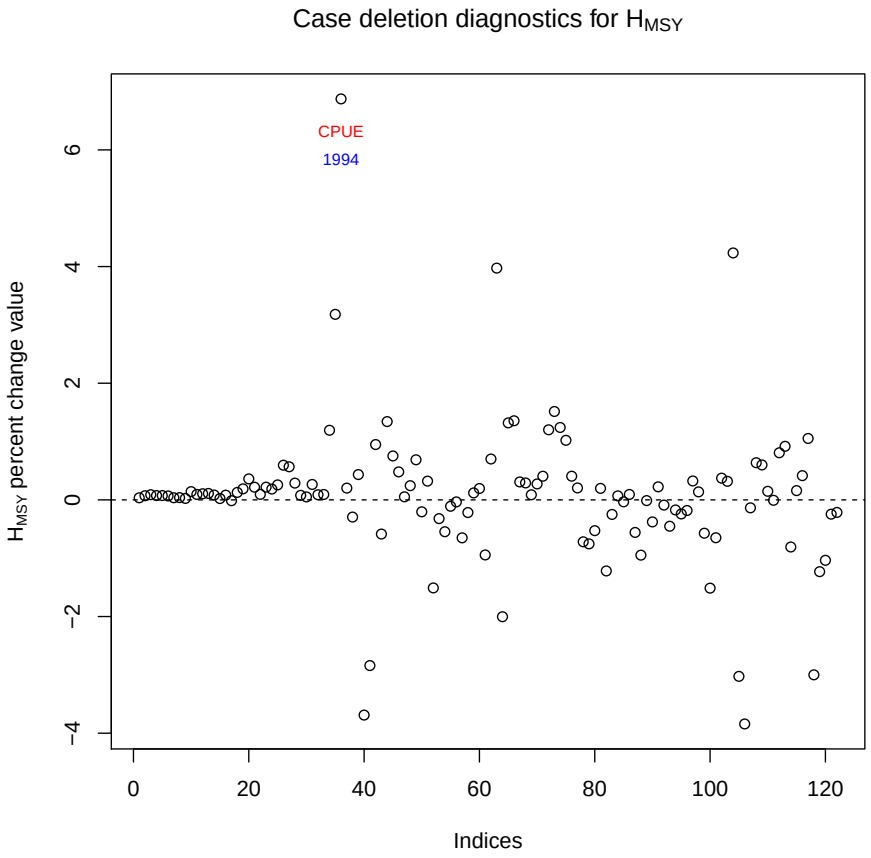


Figure 5.2: Index deletion diagnostics: redfish data for the contemporary surplus production model. The points are H_{MSY} percent difference values of deletion results compared to original results.

Generally, scatters lying far away from the rest of the data can be identified as potential influential observations. In this case, the CPUE index for the year 1994 seems more influential for both H_{MSY} and B_{MSY}.

State space surplus production model

Weighting the likelihood function for the state space model is not as straightforward as the contemporary model. Random errors are integrated out to get the marginal likelihood, which is not the sum of likelihood components for individual survey index responses. Therefore, we need to weight the conditional distribution function

separately. The joint density function for the data, random effects and fixed effects is,

$$p(y_1, ..., y_n, \theta, \Gamma) = p(y_1, ..., y_n|\theta, \Gamma)p(\Gamma|\theta), \tag{5.2}$$

where θ, Γ are fixed and random effects for the model, respectively, $p(y_1, ..., y_n|\theta, \Gamma)$ is the conditional density for the data and $p(\Gamma|\theta)$ is the density for random effects. The random effects vector Γ contains both process errors $\varepsilon's$ and random walk deviations $\delta's$ given in Eq. 4.9 and Eq. 4.12, respectively ($\Gamma = [\varepsilon_1, \ldots, \varepsilon_n, \delta_1, \ldots, \delta_n]'$). The joint log likelihood for the density given in Eq. 5.2 is,

$$l(\Gamma, \theta) = \sum_{i=1}^{n} \log[p(y_i|\theta, \Gamma)] + \log[p(\Gamma|\theta)]. \tag{5.3}$$

Therefore, we can weight the conditional density function separately and then the weighted log likelihood function $l_w(\Gamma, \theta)$ is,

$$l_w(\Gamma, \theta) = \sum_{i=1}^{n} w_i \log[p(y_i|\theta, \Gamma)] + \log[p(\Gamma|\theta)]. \tag{5.4}$$

The marginal weighted loglikelihood is based on integrating out the random effects from the joint weighted loglikelihood using the Laplace method.

Parameter estimation : State-space surplus production model

The parameter estimation method is similar to that discussed in the Section 4.2.2. We need to weight the appropriate log-likelihood function in the C++ source code as discussed above. We fit the redfish data used in Section 5.1.1 to the state space surplus production model discussed in Section 4.2.2. We make only two modifications to the C++ code discussed in Sections 4.2.2 which are, adding the weight vector **DATA_VECTOR(index_wt)**, and weight the negative log-likelihood function for indices as,

```
nll -= (index_wt*dnorm(resid,zero,sd_log_index,true)).sum().
```

Next we make the corresponding changes to the R code. First we declare the weight vector as,

```
tmb.data$index_wt = rep(1,length(tmb.data$index))
```

and then we provide the for-loop to evaluate the objective function and optimize parameter estimates in each iteration as follows,

```
for(i in 1:n.index){
  tmb.data$index_wt[i]=0

  obj <- MakeADFun(tmb.data,parameters,random=rname,DLL="fit",
  inner.control=list(maxit=100,trace=T))

  opt<-nlminb(opt$par,obj$fn,obj$gr,lower=lower,upper=upper,
    control = list(trace=0,iter.max=5000,eval.max=10000))

  rep.index_del[[i]] = obj$report()
  }
```

In Figure 5.3 we plot the index deletion diagnostic results for the state space version of the surplus production model described in Eq. 4.9. In this case, we also plot the scaled re-estimated parameters, as described in the previous section.

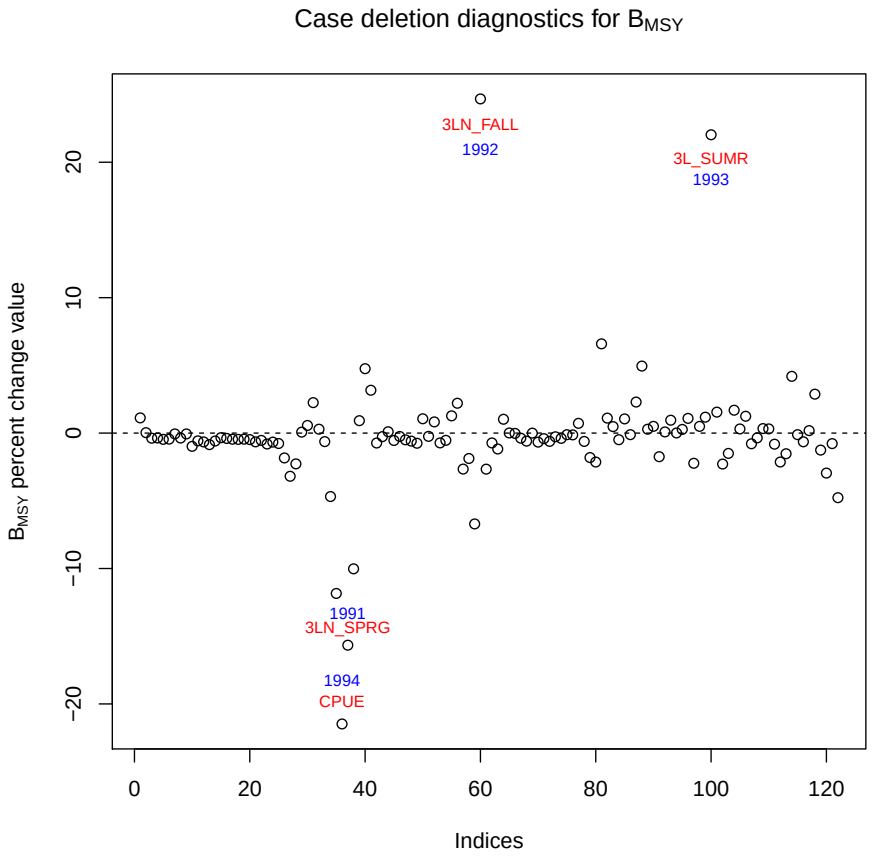


Figure 5.3: Index deletion diagnostics: redfish data for the state space surplus production model. The points are B_{MSY} percent difference values of deletion results compared to original results.

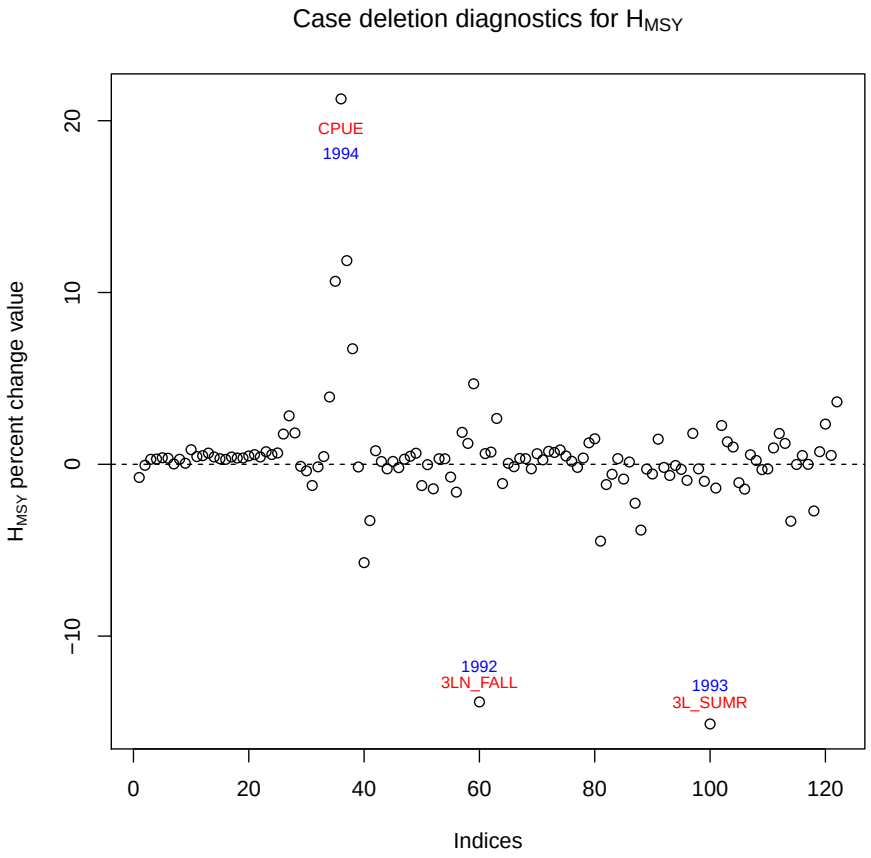


Figure 5.4: Index deletion diagnostics: redfish data for the state space surplus production model. The points are H_{MSY} percent difference values of deletion results compared to original results.

Deletions of the 1991 and 1994 CPUE indices result in relatively large reductions in the estimate of B_{MSY}. On the other hand, deletions of the 1992 3LN fall survey index and 1993 summer index result in relatively large increases in the estimate of B_{MSY}. Similarly, the 1994 CPUE index deletion leads to a relatively large increase in H_{MSY} and 1992 3LN fall survey index and 1993 3L summer index deletions lead to a relatively large decrease in the estimate of H_{MSY}.

Case study: case deletion diagnostics for redfish

We plotted the percent change of the current status for biomass and each harvest rate for eight surveys in Figures 5.5 and 5.6. The red and black dots represent the contemporary model (SPM) and the state space model (SSM), respectively. For the purpose of comparing the values, we also included the average absolute value (A.A.V) for each parameter in each model and each index.

We can see that the overall A.A.V for the biomass is higher for the state space model than the contemporary model. This indicates that the state space model has a higher sensitivity to the case deletion diagnostic than the contemporary model. However, there are a couple of cases (indices: 3L Winter, 3LN Russian) which show higher sensitivity to the contemporary model than the state space model.

For the harvest rate, both the state space and contemporary models show a similar pattern of sensitivity to case deletion. Although the overall A.A.V is slightly higher for the state space model than for the contemporary model, there are a few cases (indices: CPUE, 3LN Russian, and 3L Winter) where the contemporary model is more sensitive to case deletion.

In Figures 5.7 and 5.8, we plotted the percent difference of parameters: intrinsic growth rate (r), carrying capacity (K), and production (Po). We can see that the intrinsic growth rate (r) seems more sensitive to the state space model than the contemporary model. The carrying capacity (K) has an almost similar sensitivity to both models (overall $A.A.V_{SSM} = 1.95 > A.A.V_{SPM} = 1.54$). However, indices: CPUE, 3LN Russian, and 3L Winter show a higher sensitivity to the conventional model than the state space model. However, the production (Po) is more sensitive to the contemporary model than the state space model ($overall\ A.A.V_{SSM} = 1.38 < A.A.V_{SPM} = 2.7$). It is apparent that all of the indices have higher A.A.Vs for the contemporary model than for the state space model.

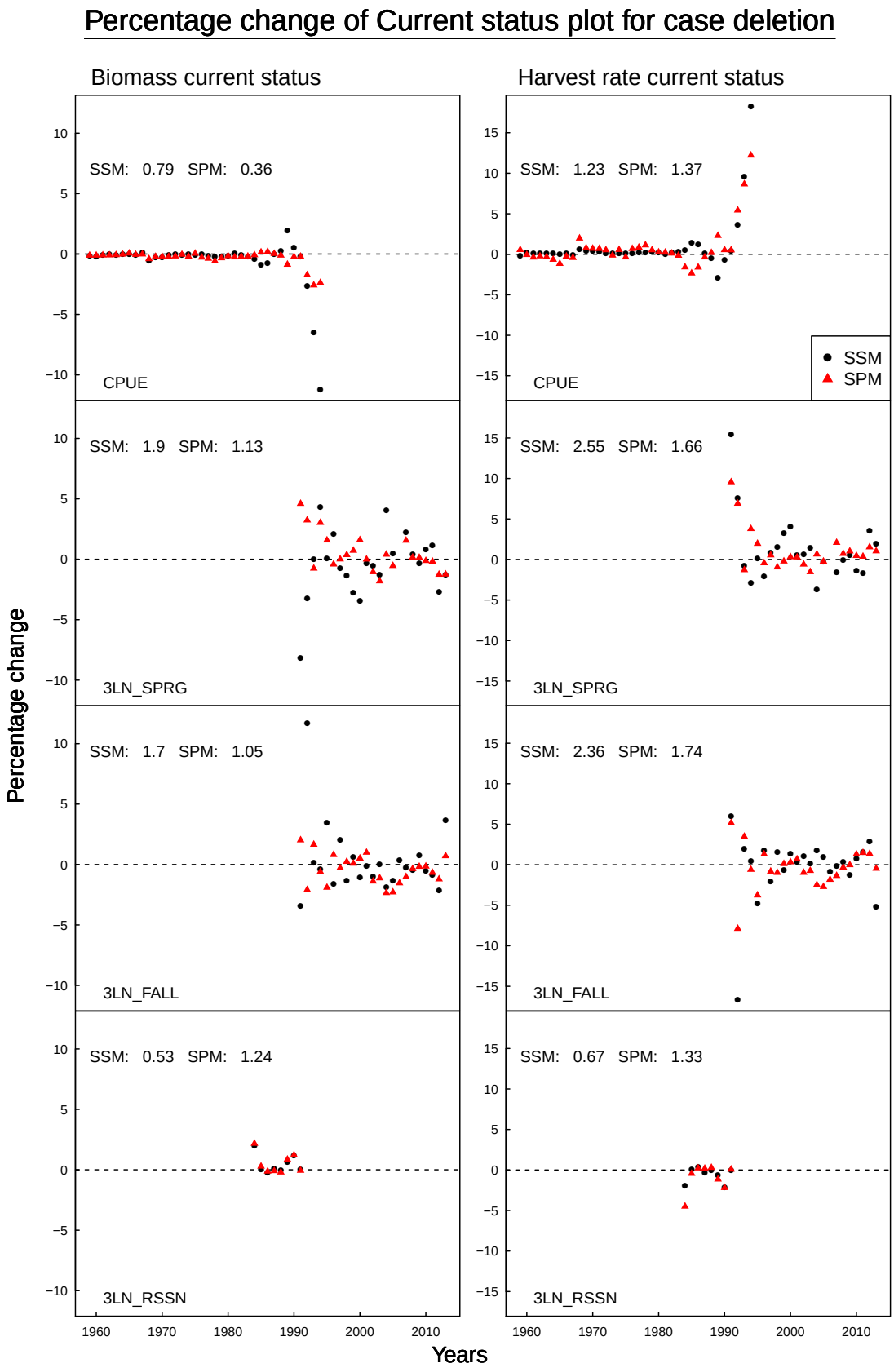


Figure 5.5: Percent change of the biomass and harvest rate for the case deletion diagnostic.

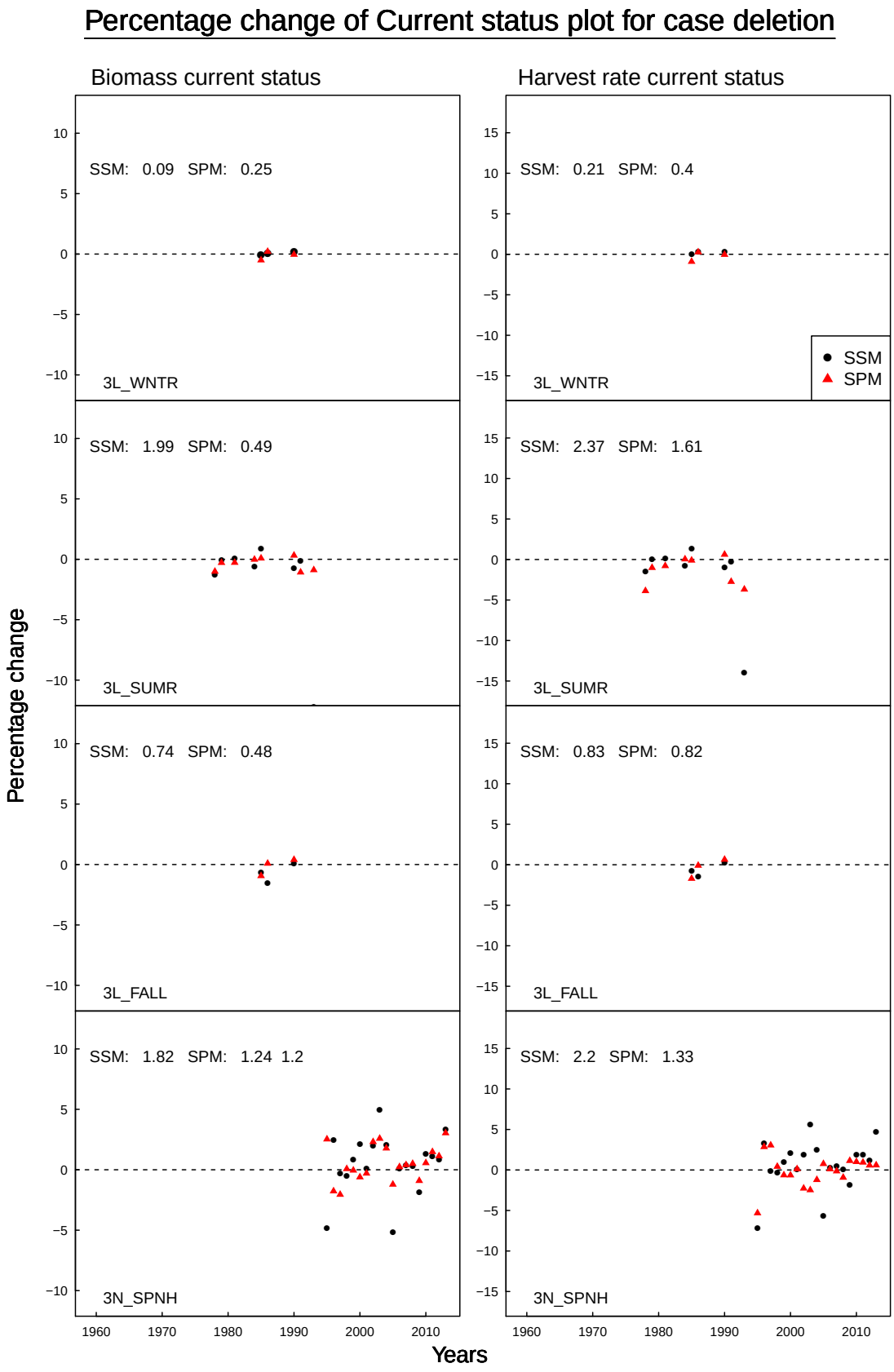


Figure 5.6: Percent change of the biomass and harvest rate for the case deletion diagnostic.

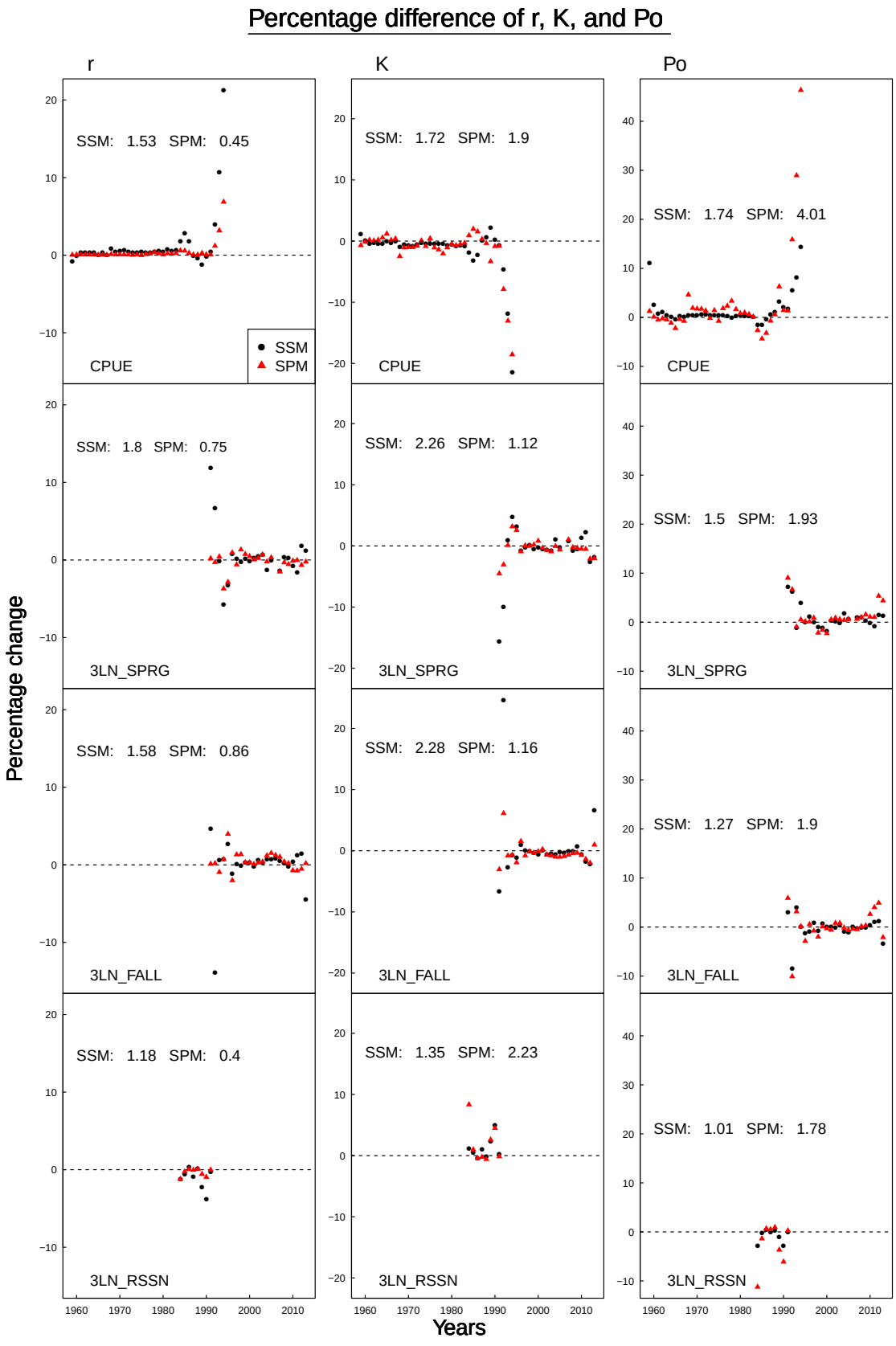


Figure 5.7: Percent change of the growth rate, carrying capacity, and production for the case deletion diagnostic.

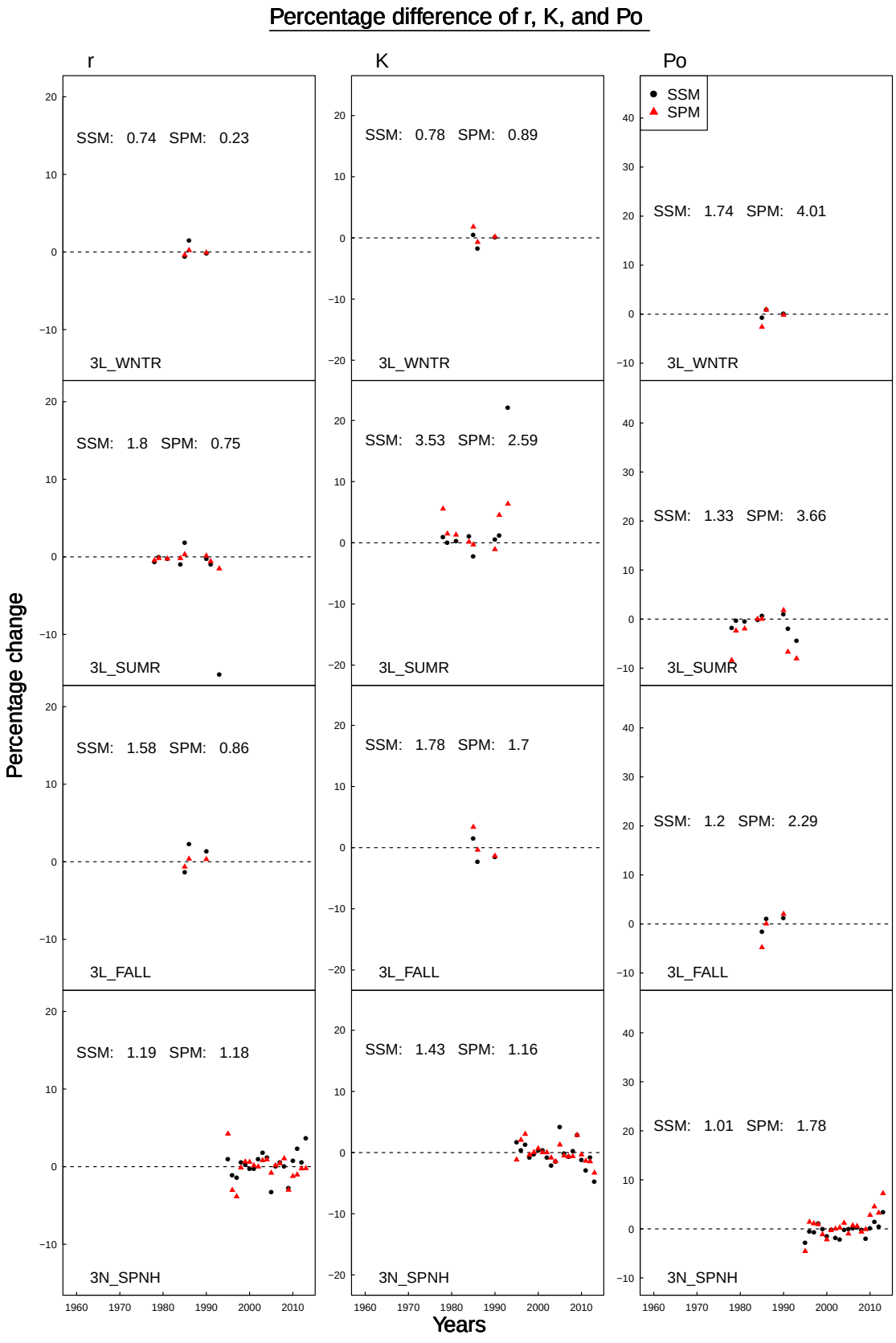


Figure 5.8: Percent change of the growth rate, carrying capacity, and production for the case deletion diagnostic.

In Figures 5.9 and 5.10 we plotted the percent difference of B_{MSY}, H_{MSY}, and MSY with each index deletion. Note that B_{MSY} is sensitive to both models by almost the same amount (overall $A.A.V_{SSM} = 1.78 > A.A.V_{SPM} = 1.70$). H_{MSY} seems noticeably more sensitive to the state space model than the contemporary one (overall $A.A.V_{SSM} = 1.65 > A.A.V_{SPM} = 0.45$). However, MSY seems much more sensitive to the contemporary model than to the state space model. Overall, A.A.Vs ($A.A.V_{SSM} = 0.14 < A.A.V_{SPM} = 1.25$) as well as all of the indices support this fact.

We can see that for some parameters, the state space model shows less sensitivity, and for other parameters, the contemporary model shows less sensitivity. Hence, by studying only one fish stock, we cannot determine which model is more sensitive to the parameters. To address this issue, we conducted the same analysis for other four data sets described in the Chapter 4. Summarized results from the investigation for all the stocks and parameters are given in Tables 5.2, 5.3, and 5.4.

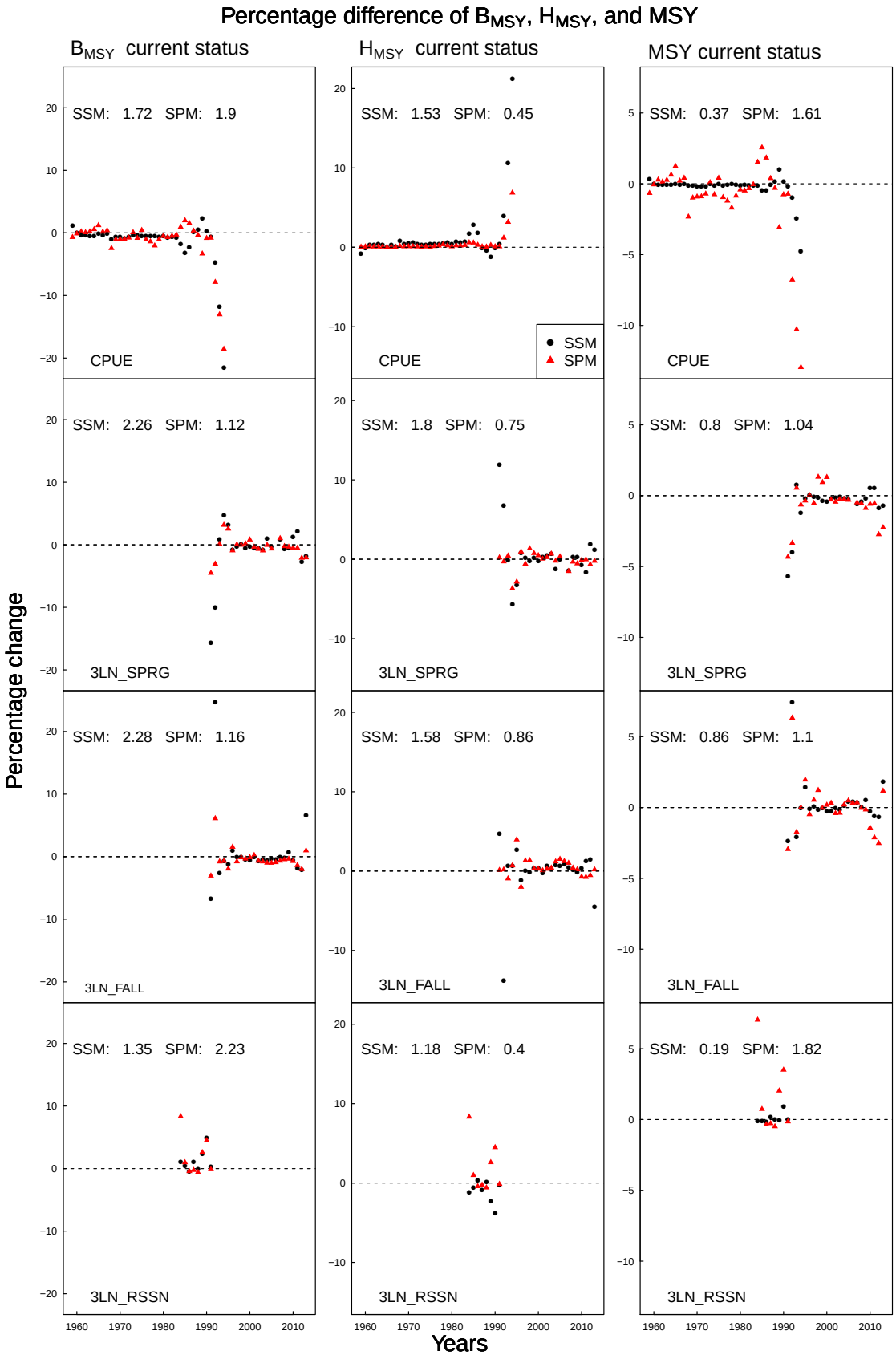


Figure 5.9: Percent change of the B_{MSY}, H_{MSY}, and MSY for the case deletion diagnostic.

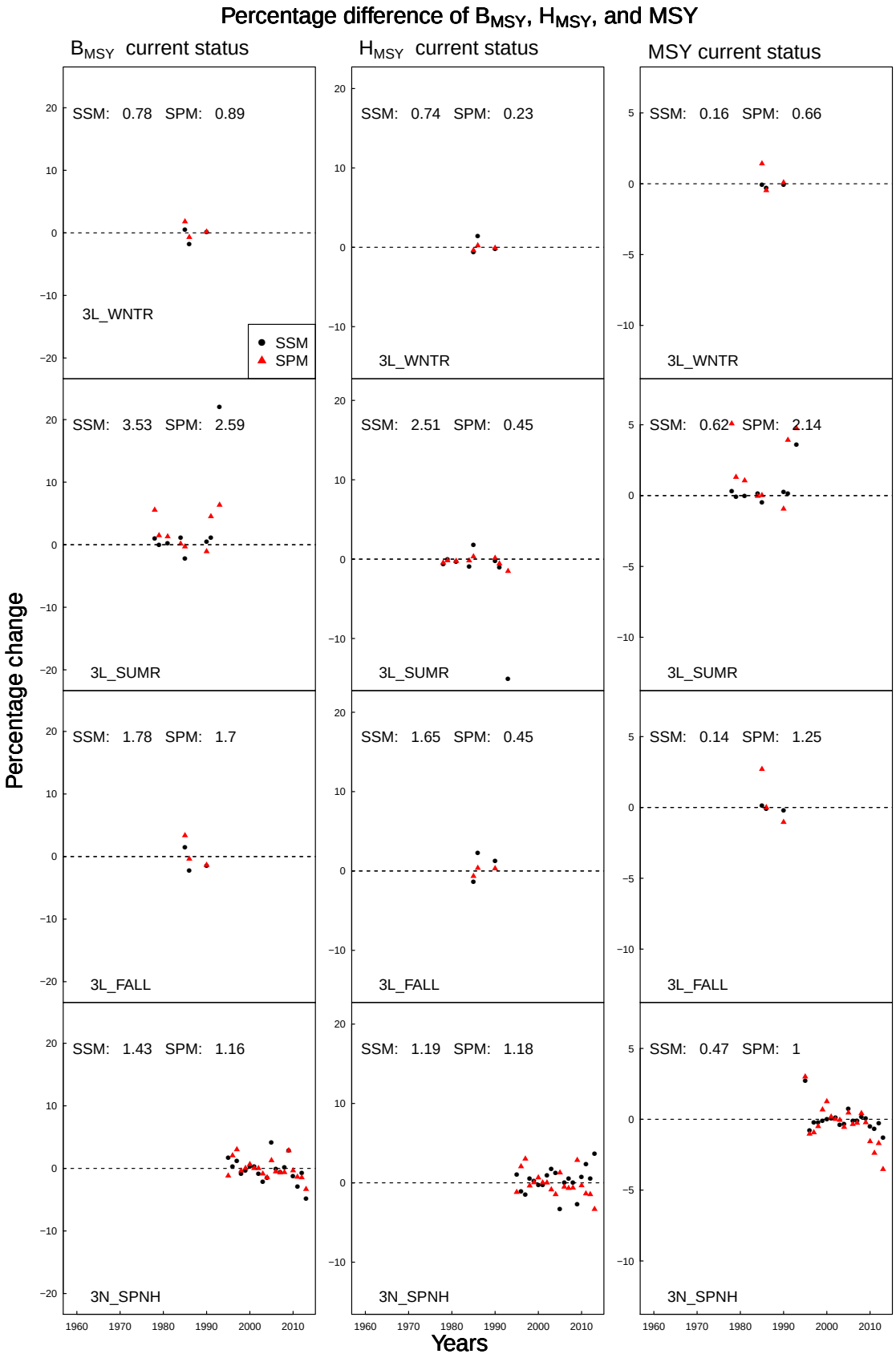


Figure 5.10: Percent change of the B_{MSY}, H_{MSY}, and MSY for the case deletion diagnostic.

Table 5.2: Case deletion analysis of indices: for biomass and the harvest rate.

stock	Average absolute value			
	Biomass		Harvest rate	
	SSM	SPM	SSM	SPM
Redfish	1.3653	0.7925	1.8344	1.4494
Yellowtail_flounder	0.0272	0.0287	0.2980	0.2764
Angler	2.3384	0.8130	2.8717	0.9971
Halibut	10.0273	45.9815	20.9586	34.3925
Megrim	0.4257	0.6307	0.5476	0.6134

Table 5.3: Case deletion analysis of indices: for B_{MSY}, H_{MSY}, and MSY.

stock	Average absolute value					
	B_{MSY}		H_{MSY}		MSY	
	SSM	SPM	SSM	SPM	SS	SPM
Redfish	1.9512	1.5423	1.5609	1.0000	0.5487	1.0000
Yellowtail_flaonder	0.6240	0.4177	0.8997	1.0000	0.2740	0.0000
Angler	2.1365	1.4512	2.1228	2.0000	0.9394	0.0000
Halibut	1.6660	1.4959	1.8202	2.0000	0.7776	0.0000
Megrim	2.0400	1.6428	2.0960	1.0000	0.1671	1.0000

Table 5.4: Case deletion analysis of indices: for growth rate, carrying capacity, and production.

stock	Average absolute value					
	r		K		Po	
	SSM	SPM	SSM	SPM	SSM	SPM
Redfish	1.5609	0.6856	1.9512	2.0000	1.3772	3.0000
Yellowtail_flaonder	0.8997	0.6646	0.6240	0.0001	0.0732	0.0002
Angler	2.1228	1.8502	2.1365	1.0000	0.0124	0.0001
Halibut	1.8202	1.9128	1.6660	1.0000	0.4873	0.0001
Megrim	2.0960	1.2903	2.0400	2.0000	1.4864	1.0000

5.1.2 Local influence diagnostics for indices

Case weight local influence (CWLI)

Fisheries scientists often use the case deletion method for influence diagnostics. However, Cadigan and Farrell (2000) [16] described that local influence diagnostics can provide a more computationally efficient means for obtaining analogous information. Therefore, in this section, we try to diagnose influential indices and catch observations for the redfish case study using the first order approach of the local influence method described in [8].

Recall the first order local influence approach by Cadigan and Farrell, 2002 [8]. Using this method we can measure how much a certain perturbation scheme, w, can influence multiple components of a model. Suppose we have a vector of parameters (θ) to estimate using the fit function $F(\theta)$. Assume this fit function can be written as a sum of each individual case such as $F(\theta) = \sum_{i=1}^{n} f_i$ and we can write the weighted form of the fit function as $F_w(\theta) = \sum_{i=1}^{n} w_i f_i$. w is the case-weight perturbation vector and it is written as $w = (1 + hd)'$, where h is the scalar that determines the size of the perturbation and d is known as the direction vector with $\sum_{i=1}^{n} d_i = 1$. If we want to perturb the j^{th} case, we can set the j^{th} element of d equal to 1 and all the other elements to zero, and then $w_j = 1 + h$. Suppose $F_w(\theta)$ is a first order

differentiable function in h and θ. The influence of a perturbation is measured by the slope in the direction d, and we denote that by $S(d)$. Further details are given in Section 2.1.3. In what follows we provide some of the R code we use to compute the local influence diagnostics. As described in earlier sections, first we need to estimate the parameters for each model using TMB. Additionally, we need the following R code to evaluate the local influence diagnostics.

First we declare a list of full data parameter estimates as follows. The **pnames** contains names of all the parameters originally estimated.

```
pnames = names(opt$par)
parameters.est <- list(
  log_r = opt$par[pnames=='log_r'],
  log_K = opt$par[pnames=='log_K'],
  log_q = opt$par[pnames=='log_q'],
  log_Po = opt$par[pnames=='log_Po'],
  log_sd_log_index = opt$par[pnames=='log_sd_log_index']
)
```

The following function will compute Δ in Eq. 2.41. The object **tmb.data.orig** contains the original data we are using for this analysis and within this function we evaluate the objective function using **MakeADFun** in **TMB**. As the output, we get the gradients of the parameters from objective function.

```
dFgrad_dtheta = function(w,i){
  tmb.data = tmb.data.orig
  tmb.data$index_wt[i]=w

  obj <- MakeADFun(tmb.data,parameters.est,DLL="fit",
           inner.control=list(maxit=10,trace=FALSE))
  return(obj$gr(obj$par))
}
```

The following function provides the g function described in Section 2.1.3. In this case,

we are interested in obtaining parameter estimates for H_{MSY} and B_{MSY}. However, we can add any number of parameters from **report** as the output of this function.

```
gfunc = function(w,theta){

  parameters.est <- list(
    log_r = theta[pnames=='log_r'],
    log_K = theta[pnames=='log_K'],
    log_q = theta[pnames=='log_q'],
    log_Po = theta[pnames=='log_Po'],
    log_sd_log_index = theta[pnames=='log_sd_log_index']
  )
  tmb.data = tmb.data.orig
  tmb.data$index_wt=w

  obj <- MakeADFun(tmb.data,parameters.est,DLL="fit",inner.control
  =list(maxit=10,
  trace=FALSE))
  rep = obj$report()

  ret = c(rep$Hmsy,rep$Bmsy)
return(ret)
}
```

We use the following function as a device to get the derivative with respect to θ,

```
gfunc1 = function(theta,w){
  return(gfunc(w,theta))}

Del = matrix(NA,n.index,length(opt$par))
for(i in 1:n.index){
  Del[i,] = t(jacobian(dFgrad_dtheta,1,,,,i))
```

```
}
w = tmb.data$index_wt
theta = opt$par
# first term in equation below (2) in CF2002
dg_dw = t(jacobian(gfunc,w,,,,theta))
# need this for 2nd term in equation below (2) in CF2002
dg_dtheta = t(jacobian(gfunc1,theta,,,,w))
```

We obtain the the local influence slopes for individual case-weight perturbations,

```
Si = dg_dw - Del%*%solve(hess)%*%dg_dtheta
```

This is the numerical solution to the analytical equations given in Eq. 2.42 and Eq. 2.43.

Parameter estimation

We also refer to this method as the local influence case weight method because even though we do not delete the case entirely, we use the weight to apply perturbations to the case-weights. Since we are using the case weights, we do not make any change to the C++ code used in the case deletion diagnostics.

To estimate perturbed parameters we need to follow these steps. First, in the R code we need to estimate the parameters for the original data as we discussed in Section 4.2.1 or 4.2.2, depending on the model we are using. We add the local influence parameter estimation part described in Section 5.1.2. We apply this method to redfish indices data and obtained the following results.

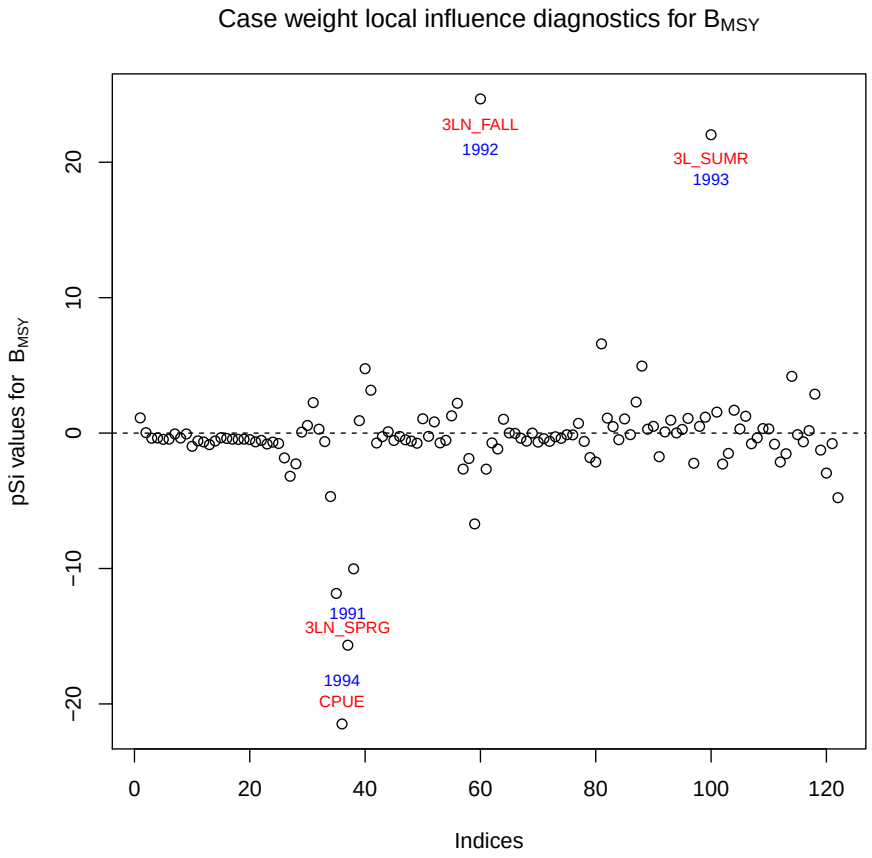


Figure 5.11: Local influence diagnostics: redfish data for the state space production model (SSM). The points are B_{MSY} local slope as a percent of full sample estimates (pSi).

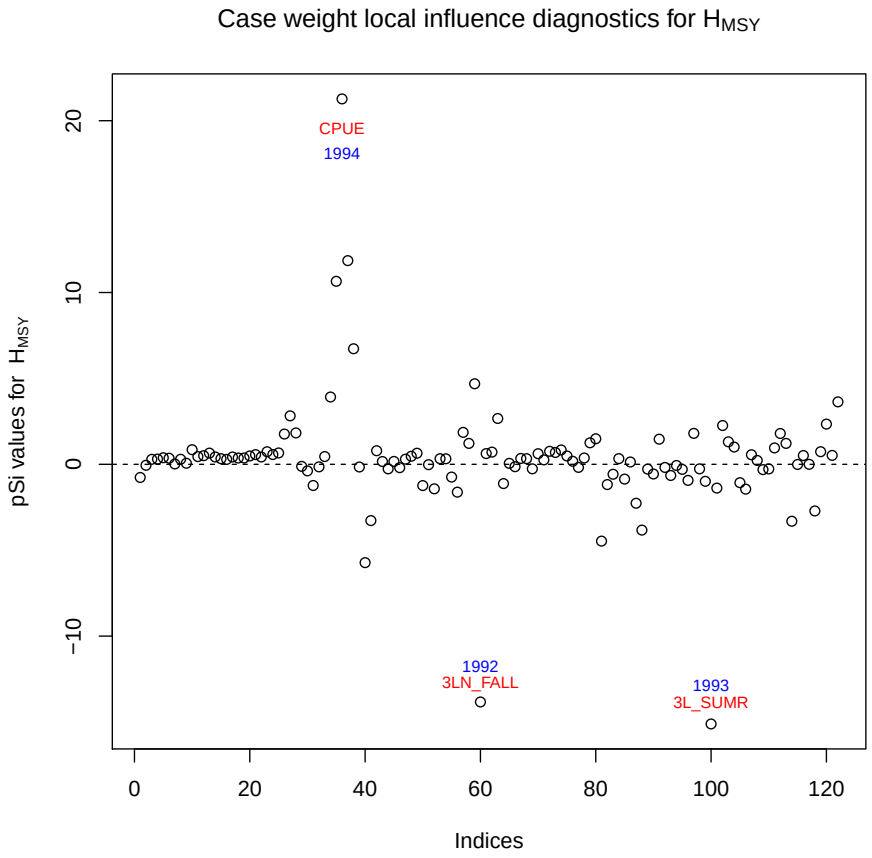


Figure 5.12: Local influence diagnostics: redfish data for the state space production model (SSM). The points are H_{MSY} local slope as a percent of full sample estimates (pSi).

We can observe from Figure 5.11 that the deletion of the 1994 CPUE index results in relatively large reductions in the estimate of B_{MSY}. On the other hand, deletions of the 1992 3LN fall survey index and 1993 summer index result in relatively large increases in the estimate of B_{MSY}. Similarly, from Figure 5.12 we can see that the 1994 CPUE index deletion leads to a relatively large increase in H_{MSY} and 1992 3LN fall survey index and 1993 3L summer index deletions lead to a relatively large decrease in the estimate of H_{MSY}.

5.1.3 Local influence diagnostics for catch data

In this section, we investigate how important fishery parameters are sensitive to catch data. Since we use the local influence diagnostic method, we can make small changes to catch data via a well-defined perturbation scheme. We perturb the catch in the form, $C_w = C * w$, where C_w is the perturbed catch and w is the perturbation for the catch. We can define the perturbation scheme as $w = 1 + hd$, where h determines the magnitude of the perturbation and d determines the direction, as described earlier.

Parameter estimation

In the C++ source code, we declare a new data vector for catch perturbations as **DATA_VECTOR(catch_p)**. Before modelling the production, we change the catch into the perturbed catch as,

```
C = C*catch_p;
vector<Type> log_catch_p= log(catch_p);
log_C = log_C + log_catch_p;
```

In this analysis, we no longer need the weight applied to indices (**index_wt**) because we only change catch data. The rest of the C++ source code remains the same as discussed in Section 5.1.1 for the contemporary model and state space model, respectively. In the R code we need to provide the data input for the catch perturbations as **tmb.data$catch_p = rep(1,length(tmb.data$year))**,
where **length(tmb.data$year)** is the number of catch data available. In the local influence diagnostic section we should pass this catch perturbation values as the weight (w). As an example, see the following function which evaluates the Δ in Eq. 2.40.

```
dFgrad_dtheta1 = function(w){
  tmb.data = tmb.data.orig
  tmb.data$catch_p=w
```

```
  obj <- MakeADFun(tmb.data,parameters.est,DLL="fit",
         inner.control=list(maxit=1000,trace=FALSE))
  return(obj$gr(obj$par))
}
```

In the third line we have assigned the catch perturbation as the weight used in the function. We applied this technique to find influential observations in redfish catch data. We measured the sensitivity of H_{MSY} and B_{MSY} using the local slope as a percent of full sample estimates. In the plots below we have marked the years which show the highest sensitivity to the change in the catch.

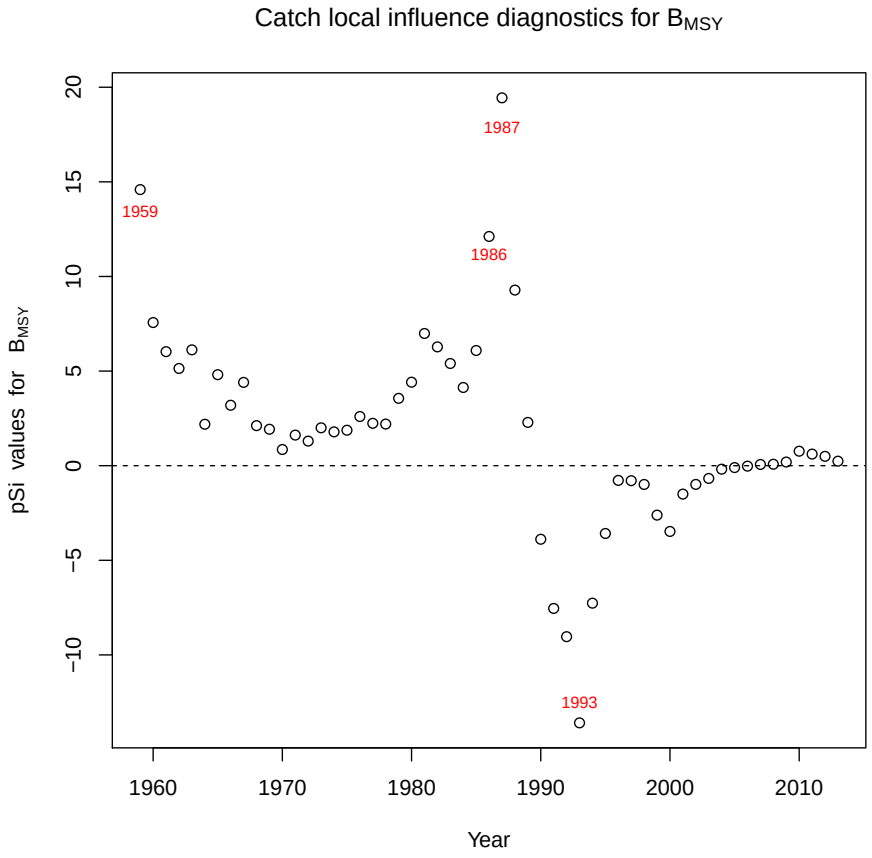


Figure 5.13: Local influence catch diagnostics: redfish data for the contemporary production model (SPM). The points are B_{MSY} local slope as a percent of full sample estimates pSi.

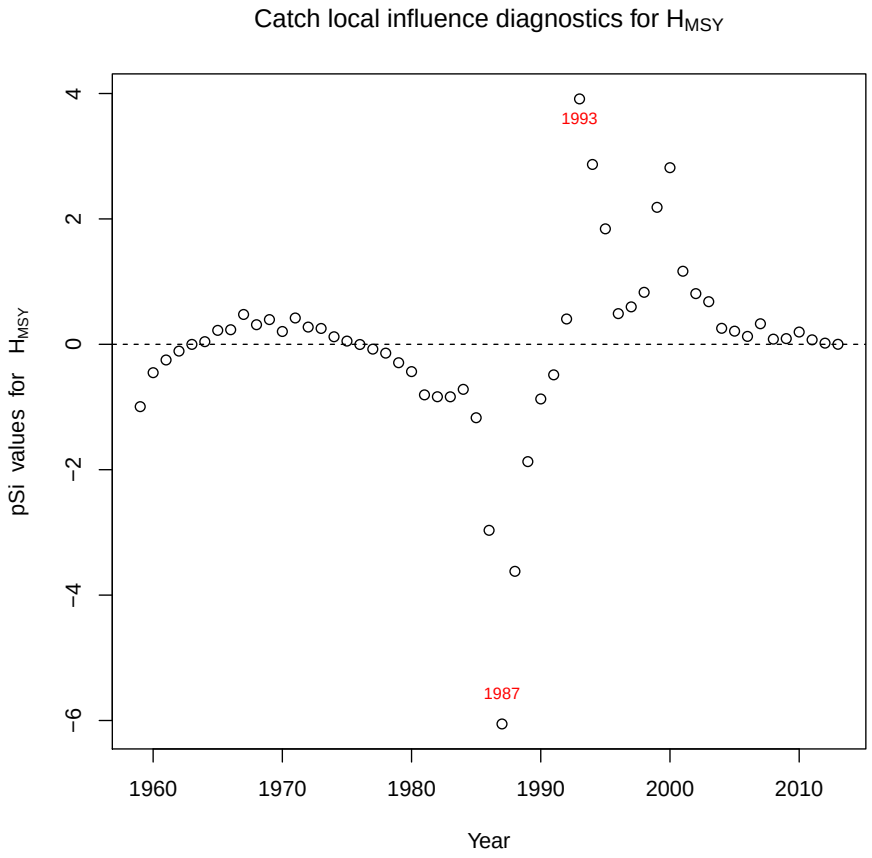


Figure 5.14: Local influence catch diagnostics: redfish data for the contemporary surplus production model (SPM). The points are H_{MSY} local slope as a percent of full sample estimates pSi.

From the local influence diagnostics in Figure 5.13, we can observe that the changes made to catches in 1959, 1985, and 1986 result in relatively large increases in the estimate of B_{MSY} and the changes made to catch in 1993 result in relatively large reductions in the estimate of B_{MSY}. Similarly, in Figure 5.14, the changes in catch in 1993 leads to a relatively large increase in H_{MSY} and changes in catch in 1987 lead to a relatively large decrease in the estimate of H_{MSY}.

5.2 Comparisons

5.2.1 Case Deletion Vs. Case Weight Local Influence

Comparison of case weight local influence diagnostics with case deletion diagnostics

In Figures 5.15, 5.16, 5.17, and 5.18 below we compare sensitivity of both B_{MSY} and H_{MSY} parameter estimates with two influence diagnostic techniques we described earlier: case deletion and local influence. In case deletion method the sensitivity of re-estimated parameter are measured as the percent difference to original parameter estimates. For the local influence method, the sensitivity is given as the percent local slope. Form these figures we can observe that the highly influential points can be identified identically using both case deletion and local influence methods for B_{MSY} and H_{MSY}. Therefore, we can consider the local influence method as a better alternative method of influence analysis for the traditional case deletion method.

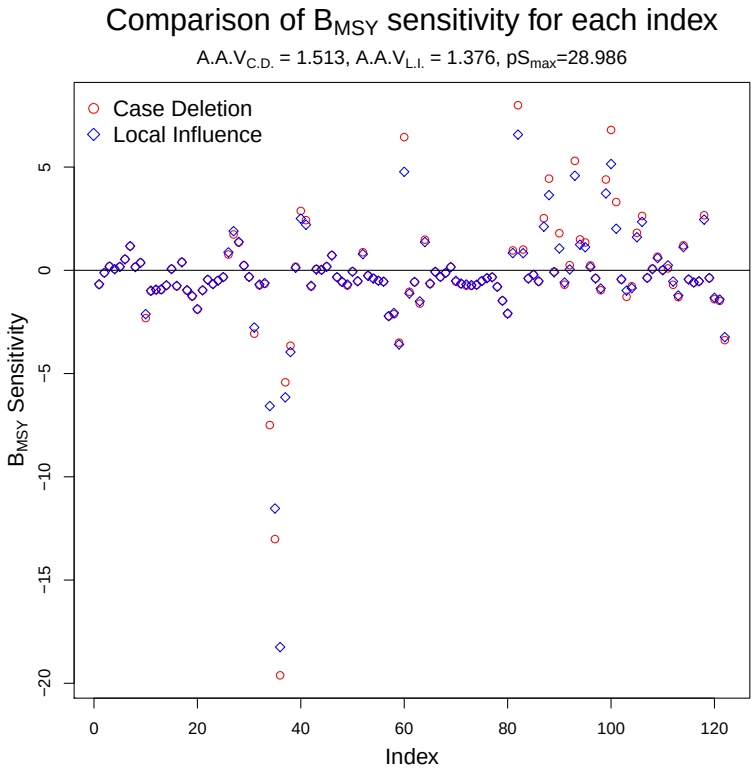


Figure 5.15: Comparative sensitivity of B_{MSY} to each index for contemporary production model (SPM). A.A.V. stands for average absolute value. For case deletion, percent change of re estimated parameters to original estimates are plotted. For local influence, local slope as a percent of full sample estimates (pS_i's) are plotted. pS_{max} is the maximum local slope.

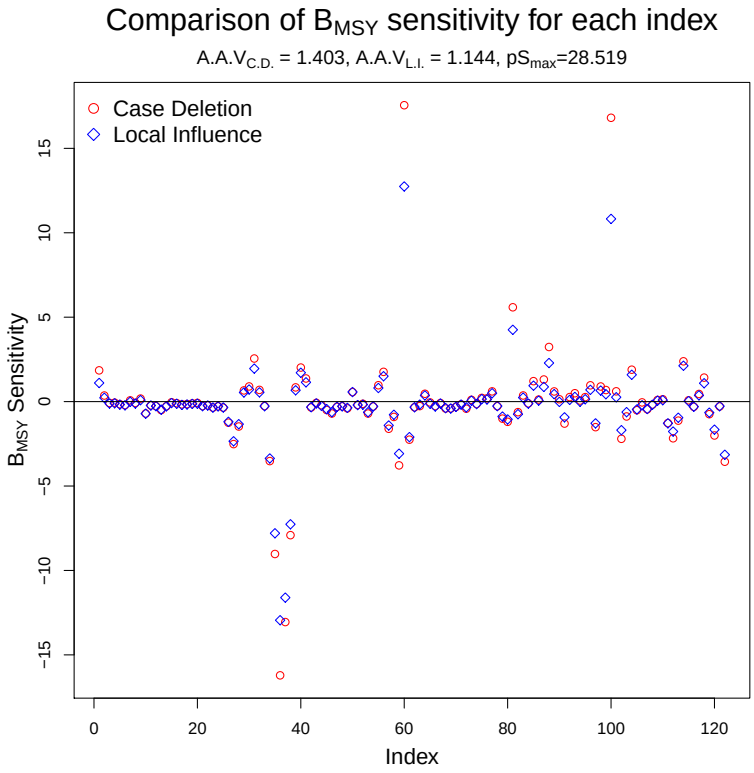


Figure 5.16: Comparative sensitivity of B_{MSY} to each index for state space model (SSM). A.A.V. stands for average absolute value. For case deletion, percent change of re-estimated parameters to original estimates are plotted. For local influence, local slope as a percent of full sample estimates (pS_i's) are plotted. pS_{max} is the maximum local slope.

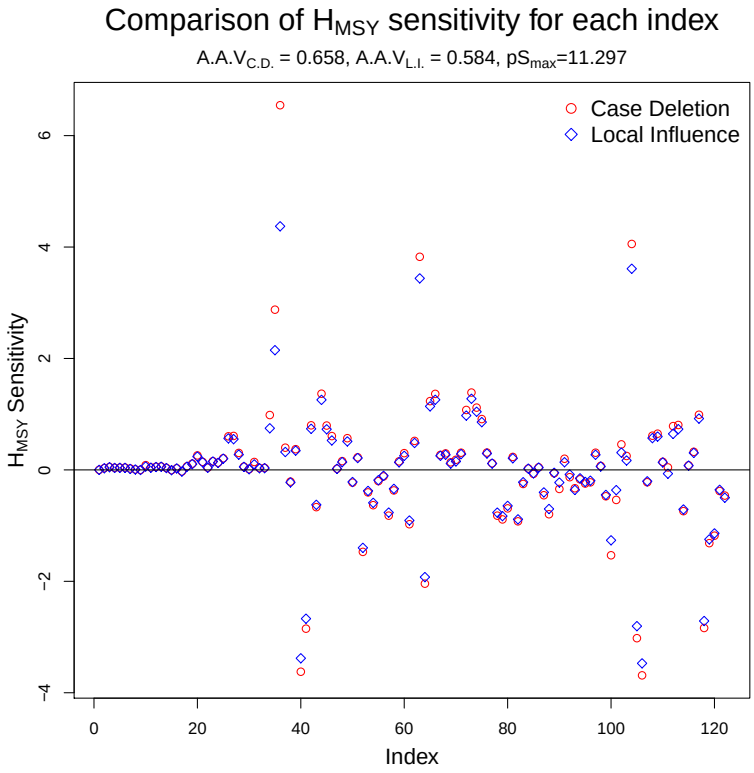


Figure 5.17: Comparative sensitivity of H_{MSY} to each index for contemporary production model (SPM). A.A.V. stands for average absolute value. For case deletion, percent change of re-estimated parameters to original estimates are plotted. For local influence, local slope as a percent of full sample estimates (pS_i's) are plotted. pS_{max} is the maximum local slope.

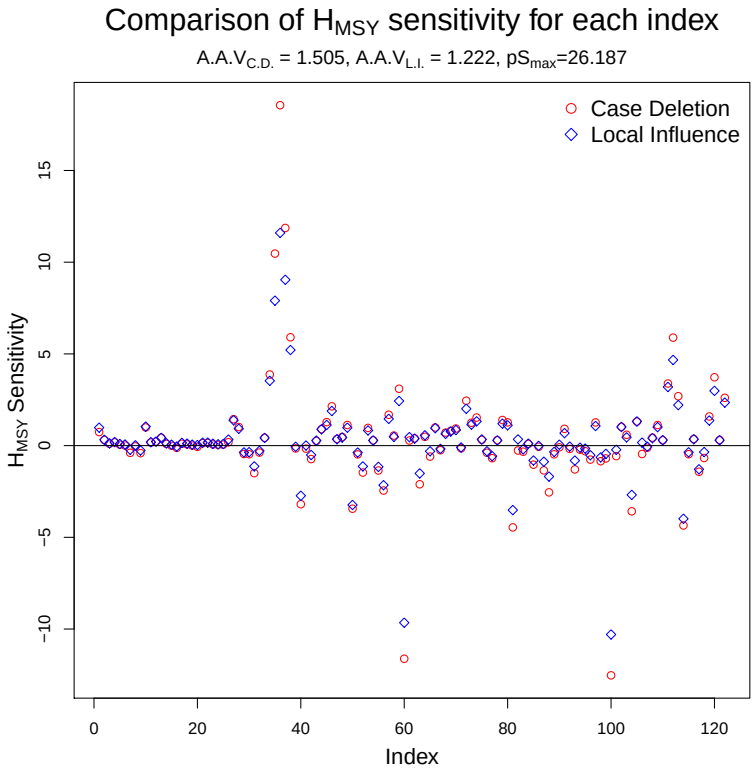


Figure 5.18: Comparative sensitivity of B_{MSY} to each index for state space model (SSM). A.A.V. stands for average absolute value. For case deletion, percent change of re-estimated parameters to original estimates are plotted. For local influence, local slope as a percent of full sample estimates (pS_i's) are plotted. pS_{max} is the maximum local slope.

Summary table

As shown in the figures above, we can see that we can obtain similar influence diagnostics using the case deletion and case weight local influence method. We investigated that relationship by conducting the same analysis for four more data obtained from surveys: Div-3LNO yellowtail flounder by NAFO, anglerfish by ICES, Greenland halibut by ICES, and megrim by ICES. As we expected, we can see that there is a very high correlation between the results obtained from the case deletion and the case weight local influence method. We summarized correlations between both case deletion and case weight local influence for parameter estimates (B_{MSY} and H_{MSY}) for both state space and contemporary surplus production models as

follows.

Table 5.5: Summary of correlations between case deletion and case weight local influence diagnostics. B_{MSY} and H_{MSY} parameters for state space model (SSM) and contemporary production model (SPM).

Stock	Case deletion vs. case weight local influence			
	B_{MSY_SSM}	H_{MSY_SSM}	B_{MSY_SPM}	H_{MSY_SPM}
Redfish	0.9929	0.9872	0.9962	0.9913
Yellowtail flounder	0.9877	0.9829	0.9992	0.9973
Anglerfish	0.9909	0.9872	0.9983	0.9994
Greenland halibut	0.4964	0.7822	0.9989	0.9992
Megrim	0.8251	0.7492	0.9991	0.9992

5.2.2 Case Weight Local Influence: SSM Vs. Contemporary SPM

In the previous analysis, we found that there is a high correlation between case deletion results and case weight local influence (CWLI) results for both parameters B_{MSY} and H_{MSY}. Therefore, we use CWLI method to analyse both the contemporary SPM and the SSM in this section. We compare influence diagnostics of these two models using both parameters estimates B_{MSY} and H_{MSY}.

In Figure 5.19, we plotted pS_i values for B_{MSY} for SSM (in black) and contemporary SPM (in red) for 3LN redfish data. We can observe that the most sensitive cases are the same for both the models. However, the SSM results show higher sensitivity than the contemporary SPM in this analysis. To support this we use the average absolute values (A.A.Vs). Similar results can be observed in Figure 5.20 where we plotted pS_i values for H_{MSY} for SSM (in black) and contemporary SPM (in red) for 3LN redfish data. The SSM results show higher sensitivity than the contemporary SPM in this analysis.

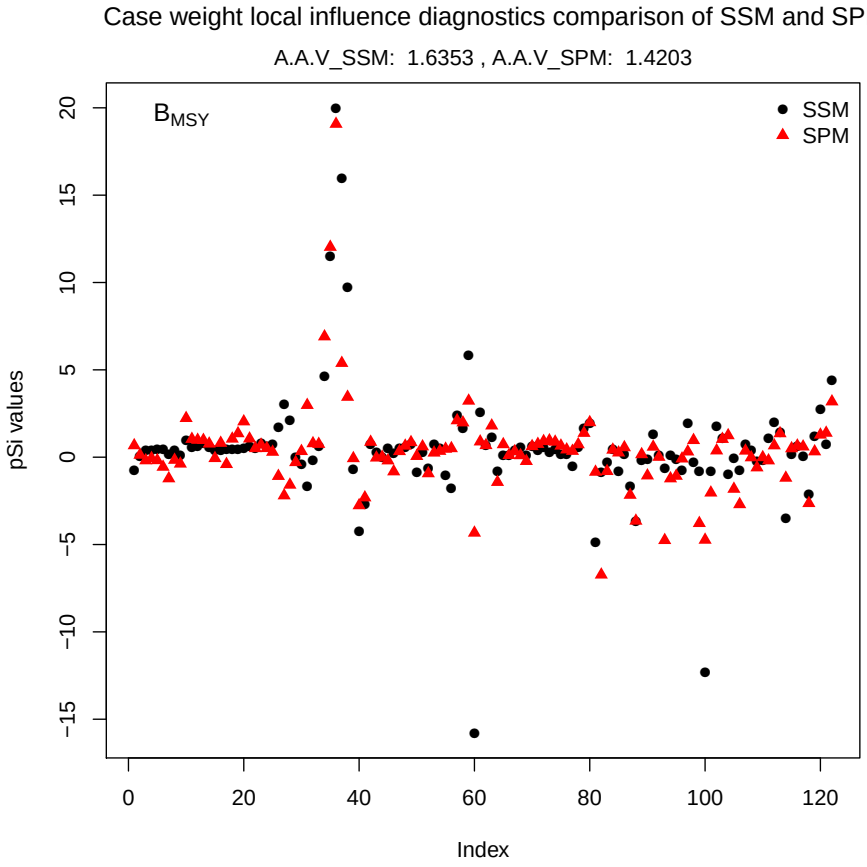


Figure 5.19: Local influence diagnostics for redfish indices: The points are B_{MSY} local slope as a percent of full sample estimates for state space model (SSM) and contemporary model (SPM).

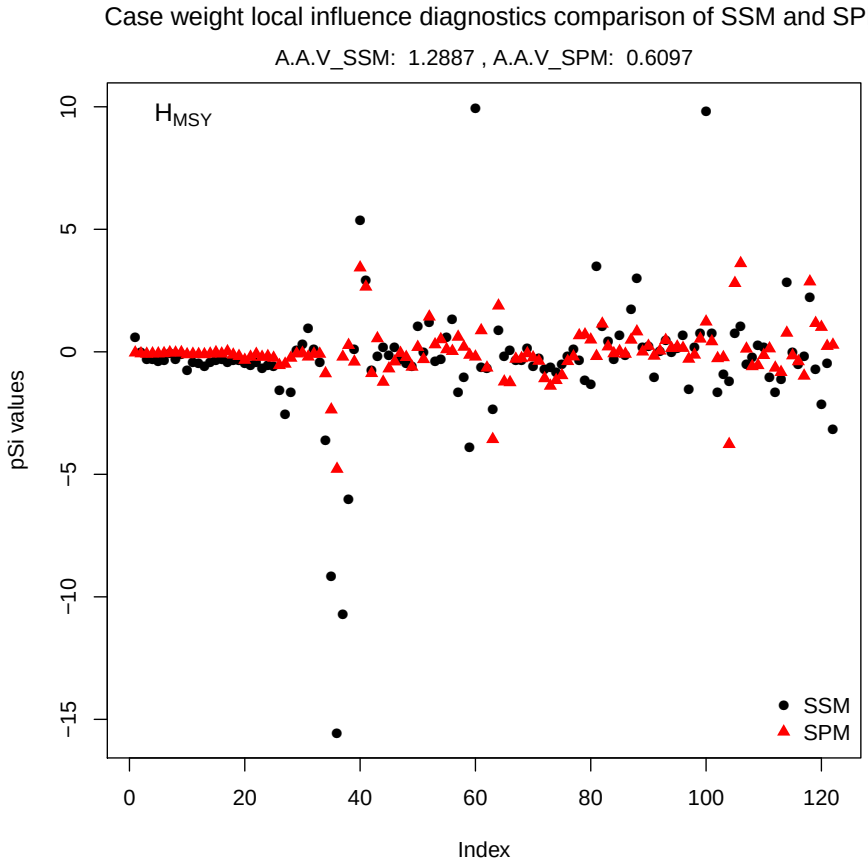


Figure 5.20: Local influence diagnostics for redfish indices: The points are H_{MSY} local slope as a percent of full sample estimates for state space model (SSM) and contemporary model (SPM).

We conducted the same analysis for the other four case studies: yellowtail flounder, anglerfish, halibut, and megrim. A.A.V's for the SSM and the contemporary SPM are summarized for both the parameters B_{MSY} and H_{MSY} in Table 5.6. For anglerfish and halibut data, both B_{MSY} and H_{MSY} estimates are less sensitive to SSM than to the contemporary SPM. However, for yellowtail flounder and megrim data, both parameters estimates are less sensitive to contemporary model than to SSM.

Table 5.6: Case weight local influence (CWLI) analysis of indices: average of absolute pSi values for B_{MSY} and the H_{MSY} are given in the table.

	B_{MSY}		H_{MSY}	
	SSM	SPM	SSM	SPM
Redfish	1.6352	1.4202	1.2886	0.6097
Yellowtail_flounder	0.4420	0.3736	0.6262	0.6066
Anglerfish	1.0754	1.3720	1.4519	1.7436
Halibut	0.9863	1.6403	1.5342	2.1400
Megrim	1.7587	1.5364	1.5476	1.2106

5.2.3 Compare Catch Local Influence

Catch local influence diagnostics comparison for state space model and contemporary model

In Figure 5.21 we plotted local influence results for both B_{MSY} and H_{MSY} to compare state space production model (SSM) and the contemporary production model (SPM). To make comparison easy, we give the average of the absolute values (A.A.V's) of pS_i's for both parameter estimates. For H_{MSY}, SSM is more sensitive than contemporary model and for B_{MSY}, the contemporary model shows more sensitiveness than the SSM.

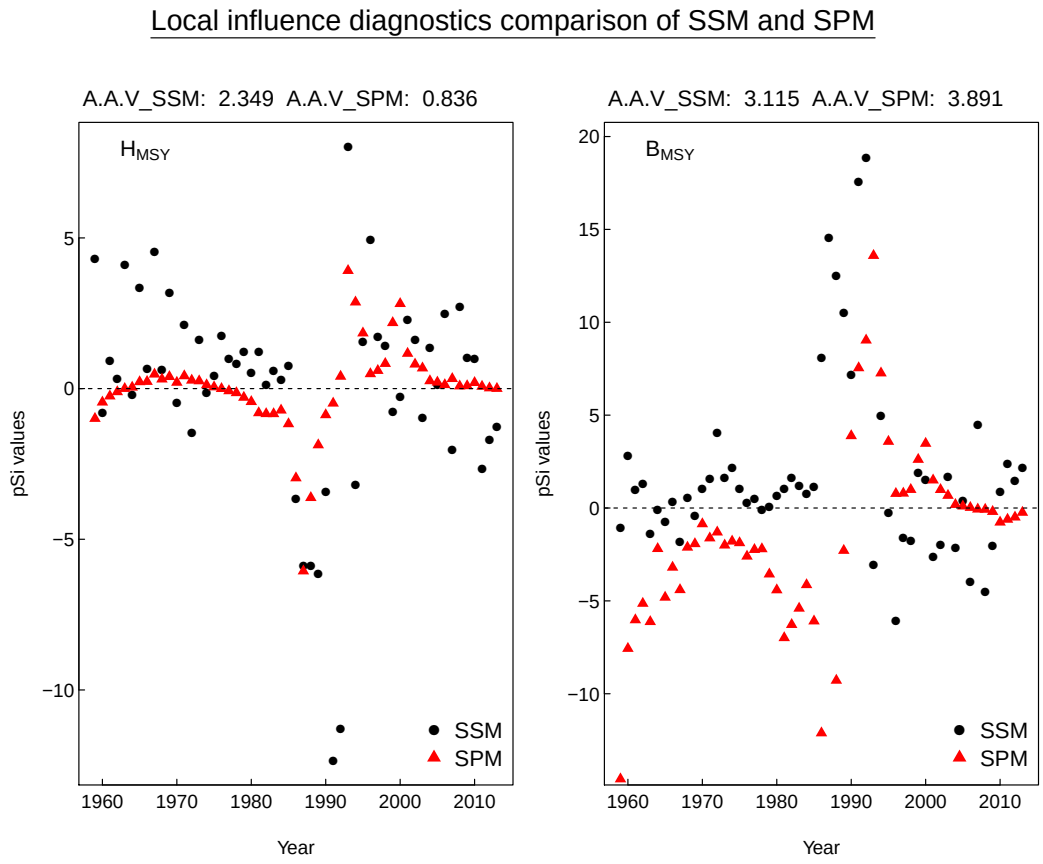


Figure 5.21: Local influence results comparison for contemporary surplus production model (SPM) and state space model (SSM) for catch data. B_{MSY} and H_{MSY} local slopes are plotted as a percent of full sample estimates (pS_i).

We extended this analysis to other four data sets studied in the previous section. A.A.V's for the SSM and the contemporary SPM are summarized for both the parameters B_{MSY} and H_{MSY} in Table 5.7. For yellowtail flounder and halibut data, both B_{MSY} and H_{MSY} estimates are less sensitive to SSM than to the contemporary SPM. However, for anglerfish and megrim data, both parameters estimates are less sensitive to contemporary model than to SSM.

Table 5.7: Local influence analysis of catch data: average of absolute pSi values for B_{MSY} and H_{MSY} are given in the table.

	B_{MSY}		H_{MSY}	
	SSM	SPM	SSM	SPM
Redfish	3.11	3.89	2.34	0.83
Yellowtail_flounder	3.32	3.37	2.76	3.33
Anglerfish	16.86	9.58	16.83	9.37
Halibut	4.44	9.66	3.29	11.48
Megrim	57.35	10.61	58.83	8.74

Chapter 6

Summary

Fishing industries can increase their production by increasing the effort. Unfortunately, this leads to overharvesting or even collapse of fish stocks. Therefore, fishery management agencies need information about the status of fish stocks that they are harvesting. Fisheries scientists try to provide this information by conducting stock assessments. Scientists use mathematical and statistical models to estimate abundance or biomass of the fish stocks. The complexity of these models differs upon the availability of data. We use the simple and the most widely used Schaefer's SPM for this study (we refer to it as the contemporary SPM). In recent years the state-space modelling framework was also widely used to fit the SPMs. In this study, we compare the sensitivity of estimators of state-space SPMs and contemporary SPMs (without process errors) using the traditional case deletion method and local influence analysis method introduced by R.D. Cook, 1986 [12]. We applied these methods to five different data sets and examined how important parameter estimates respond to small changes made in the input data. We used **R** package **TMB** for each parameter estimation. TMB uses the Laplace approximation to find the solution for marginal likelihoods by integrating out the random effects in the SSM. In the first analysis, we compared the two diagnostic methods. For the comparison, we used the B_{MSY} and H_{MSY} as the two model outputs. The Comparison shows a high positive correlation of influential observations between the two diagnostic methods (see Table 5.5). This finding is beneficial in other studies where the case deletion method cannot apply.

For example, we cannot delete catches to find influential observations in catch data.

In the second analysis, we compared the sensitivity of state space and the contemporary SPMs using the case weight local influence method for indices data. Anglerfish and halibut data showed less sensitivity for both B_{MSY} and H_{MSY} parameter estimates of the SSM. However, 3LN redfish, yellowtail flounder and megrim data showed less sensitivity for both B_{MSY} and H_{MSY} parameter estimates of the contemporary SPM.

As the last analysis, we compared the sensitivity of two models using the local influence diagnostic method with catch data. Here also we obtained mixed results. For redfish data, B_{MSY} estimates showed less sensitivity to the contemporary SPM while H_{MSY} estimates showed less sensitivity to the SSM. Both the parameters B_{MSY} and H_{MSY} showed less sensitivity to SSM for yellowtail flounder and halibut data than contemporary SPM. However, contemporary SPM showed less sensitivity for anglerfish and megrim data.

Bibliography

[1] Acronyms and terminology. `http://www.ices.dk/community/Documents/Advice/Acronyms_and_terminology.pdf`. Accessed: 2018-11-20.

[2] Worldfish why fish. `https://www.worldfishcenter.org/why-fish`. Accessed: 2018-11-22.

[3] W. H. Aeberhard, J. Mills Flemming, and A. Nielsen. Review of state-space models for fisheries science. *Annual Review of Statistics and Its Application*, 5:215–235, 2018.

[4] S. Agarwal and K. Mierle. TMB: Automatic Differentiation and Laplace Approximation. *Journal of Statistical Software*, 70:1–21, 2013.

[5] M. Auger-Méthé, C. Field, C. M. Albertsen, A. E. Derocher, M. A. Lewis, I. D. Jonsen, and J. M. Flemming. State-space models dirty little secrets: even simple linear gaussian models can have estimation problems. *Scientific reports*, 6:26677, 2016.

[6] P. A. Breen. *Assessment of the red rock lobster (Jasus edwardsii) North and South Island stock, November 1991.* MAF Fisheries, 1991.

[7] S. Buckland, K. Newman, L. Thomas, and N. Koesters. State-space models for the dynamics of wild animal populations. *Ecological modelling*, 171(1-2):157–175, 2004.

[8] N. G. Cadigan and P. J. Farrel. Generalized Local Influence With Applications To Fish Stock Cohort Analysis. *Journal of the Royal Statistical Society*, 51:469–483, 2002.

[9] N. G. Cadigan, E. Wade, and A. Nielsen. A spatiotemporal model for snow crab (chionoecetes opilio) stock size in the southern gulf of st. lawrence. *Canadian Journal of Fisheries and Aquatic Sciences*, 74(11):1808–1820, 2017.

[10] S. Chatterjee and A. S. Hadi. Influential Observations, High Leverage Points, and Outliers in Linear Regression. *Statistical Science*, 1:379–393, 1986.

[11] C. W. Clark. *The worldwide crisis in fisheries: economic models and human behavior*. Cambridge University Press, 2006.

[12] R. D. Cook. Assessment of Local Influence. *Journal of the Royal Statistical Society*, 48:133–169, 1986.

[13] R. D. Cook and S. Weisberg. *Residuals and Influence in Regression*. Chapman & Hall, 1982.

[14] A. B. Cooper. *A guide to fisheries stock assessment: from data to recommendations*. University of New Hampshire, Sea Grant College Program, 2006.

[15] Duncan Murdoch and E. D. Chow . *Ellipse: Functions for Drawing Ellipses and Ellipse-Like Confidence Regions*. R Foundation for Statistical Computing, 2018.

[16] P. Farrell and N. Cadigan. Local influence in binary regression models, and its correspondence with global influence. *Communications in Statistics-Theory and Methods*, 29(2):349–368, 2000.

[17] D. Fournier, H. Skaug, J. Ancheta, J. Ianelli, A. Magnusson, M. Maunder, A. Nielsen, and J. Sibert. AD Model Builder: using Automatic Differentiation for Statistical Inference of Highly Parameterized Complex Nonlinear Models. *Optimization Methods and Software*, 27:233249, 2012.

[18] R. Francis, R. J. Hurst, and J. A. Renwick. *An evaluation of catchability assumptions in New Zealand stock assessments*. Ministry of Fisheries, 2001.

[19] D. Gartside and I. Kirkegaard. A history of fishing. *Interactions: Food, Agriculture And Environment*, 2:70–80, 2010.

[20] A. Griewank and A. Walther. Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. *Society for Industrial and Applied Mathematics*, 70:1–21, 2008.

[21] M. Haddon. *Modelling and quantitative methods in fisheries*. CRC press, 2010.

[22] F. R. Hampel. The Influence Curve and Its Role in Robust Estimation. *Journal of The American Statistical Association*, 69:383–393, 1974.

[23] R. Hilborn and C. J. Walters. Quantitative Fisheries Stock Assessment : Choice, Dynamics and Uncertainty. 1992.

[24] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[25] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83:95–108, 1961.

[26] D. Kimura, J. Balsiger, and D. Ito. Kalman filtering the delay-difference equation: practical approaches and simulations. *Fishery Bulletin*, 94(4):678–691, 1996.

[27] K. Kristensen, A. Nielsen, C. W. Berg, H. Skaug, and B. Bell. Tmb: automatic differentiation and laplace approximation. *arXiv preprint arXiv:1509.00660*, 2015.

[28] R. B. Millar and R. Meyer. Nonlinear state space modelling of fisheries biomass dynamics by using MetropolisHastings withinGibbs sampling. *Journal of the Royal Statistical Society*, 49:327–342, 2000.

[29] J. C. Patrick Kilduff and R. Latour. Guide to Fisheries Science and Stock Assessments. 2009.

[30] M. Pedersen, C. Berg, U. Thygesen, A. Nielsen, and H. Madsen. Estimation Methods For Nonlinear State-Space Models In Ecology. *Ecological Modelling*, 222:13941400, 2011.

[31] T. Polacheck, R. Hilborn, and A. E. Punt. Fitting surplus production models: comparing methods and measuring uncertainty. *Canadian Journal of Fisheries and Aquatic Sciences*, 50(12):2597–2607, 1993.

[32] W.-Y. Poon and Y. S. Poon. Conformal Normal Curvature and Assessment of Local Influence. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61:51–61, 1999.

[33] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.

[34] M. B. Schaefer. Some aspects of the dynamics of populations important to the management of commercial marine fisheries. *IATTC Bull.*, 1:25–56, 1954.

[35] J. T. Schnute. A general framework for developing sequential fisheries models. *Canadian Journal of Fisheries and Aquatic Sciences*, 51(8):1676–1688, 1994.

[36] R. H. Shumway and D. S. Stoffer. Time series analysis and its applications. volume 9, pages 287–375. Springer, 2000.

[37] P. J. Sullivan. A Kalman Filter Approach to Catch-at-Length Analysis. *Biometrics*, 48:237–257, 1992.

[38] P. F. Velleman. and R. E. Welsch. Efficient Computing of Regression Diagnostics. *The American Statistician*, 35:234–242, 1981.

[39] C. J. Walters and R. Hilborn. Adaptive control of fishing systems. *Journal of the Fisheries Board of Canada*, 33(1):145–159, 1976.

[40] G. Wang. On The Latent State Estimation Of Nonlinear Population Dynamics Using Bayesian And Non-Bayesian State-Space Models. *Ecological Modelling*, 200:521–528, 2007.

[41] R. E. Welsch and E. Kuh. Linear regression diagnostics. Working Paper 173, National Bureau of Economic Research, March 1977.

[42] S. Zhou, A. E. Punt, R. Deng, and J. Bishop. Estimating multifleet catchability coefficients and natural mortality from fishery catch and effort data: comparison of bayesian state–space and observation error models. *Canadian Journal of Fisheries and Aquatic Sciences*, 68(7):1171–1181, 2011.

Appendix A

Appendices

A.1 Derivative

$$\begin{aligned} \frac{\partial LD\{\omega(h)\}}{\partial h} &= \frac{\partial \omega}{\partial h}\frac{\partial LD\{\omega(h)\}}{\partial \omega} = d'\frac{\partial LD\{\omega(h)\}}{\partial \omega} \\ \frac{\partial^2 LD\{\omega(h)\}}{\partial h^2} &= d'\frac{\partial LD\{\omega(h)\}}{\partial \omega \partial \omega'}\frac{\partial \omega'}{\partial d} = d'\frac{\partial LD\{\omega(h)\}}{\partial \omega \partial \omega'}d \end{aligned} \tag{A.1}$$

A.2 TMB linear regression results

```
> opt
$par
         a          b   logSigma
 0.5254673  0.9180288 -0.1350524

$value
[1] 12.83886

$counts
function gradient
      87       34
```

```
$convergence
[1] 0

$message
NULL

$hessian
                      a             b      logSigma
a         1.310102e+01  7.205559e+01 -1.976378e-07
b         7.205559e+01  5.043891e+02 -2.022709e-06
logSigma -1.976378e-07 -2.022709e-06  2.000001e+01

> opt$hessian ## <-- FD hessian from optim
                      a             b      logSigma
a         1.310102e+01  7.205559e+01 -1.976378e-07
b         7.205559e+01  5.043891e+02 -2.022709e-06
logSigma -1.976378e-07 -2.022709e-06  2.000001e+01
> obj$he()    ## <-- Analytical hessian
              [,1]          [,2]          [,3]
[1,]  1.312724e+01  7.219985e+01 -1.980334e-07
[2,]  7.219985e+01  5.053989e+02 -2.026757e-06
[3,] -1.980334e-07 -2.026757e-06  2.004004e+01
> sdreport(obj)
outer mgc:  1.011354e-06
outer mgc:  0.0720566
outer mgc:  0.07205458
outer mgc:  0.5043902
outer mgc:  0.5043881
outer mgc:  0.01997994
outer mgc:  0.02002009
outer mgc:  1.526599
```

```
sdreport(.) result
           Estimate Std. Error
a         0.5254673 0.59683034
b         0.9180288 0.09618792
logSigma -0.1350524 0.22360672
Maximum gradient component: 1.011354e-06
```

A.3 TMB C++ code for the contemporary model: Namibian hake data

```
#include <TMB.hpp>
#include <iostream>

template<class Type>
Type objective_function<Type>::operator() ()
{
  DATA_IVECTOR(year);
  DATA_VECTOR(C);
  DATA_VECTOR(index);
  DATA_IVECTOR(iyear);
  DATA_IVECTOR(iq);
  //DATA_VECTOR(index_wt);
  DATA_VECTOR(log_C);
  DATA_VECTOR(log_index);
  DATA_SCALAR(E_log_r);
  DATA_SCALAR(sd_log_r);
  DATA_SCALAR(E_log_Po);
  DATA_SCALAR(sd_log_Po);
  DATA_SCALAR(sd_logC);
```

```
  PARAMETER(log_r);
  PARAMETER(log_K);
  PARAMETER_VECTOR(log_q);
  PARAMETER(log_Po);
  PARAMETER(log_sd_log_index);

  int n = year.size();
  int ni = index.size();
  int i;

  Type r = exp(log_r);
  Type K = exp(log_K);
  Type sd_log_index = exp(log_sd_log_index);

  vector<Type> log_P(n); //log population biomass divided by K at
  start of the year;
  vector<Type> log_P_midy(n); // log P at middle of year;
  vector<Type> P(n);
  vector<Type> P_midy(n);
  vector<Type> log_B(n);
  vector<Type> log_H(n);
  vector<Type> H(n);
  vector<Type> log_Eindex(ni);

  Type one = 1.0;
  Type half = 0.5;
  Type zero = 0.0;

  Type nll=0;

// prior nll for log_r;
```

```
//  nll -= dnorm(log_r,E_log_r,sd_log_r,true);

// prior nll for log_Po;
  //nll -= dnorm(log_Po,E_log_Po,sd_log_Po,true);

// log of production model;

  P(0) = exp(log_Po);
  for (i=1;i<n;i++){
    P(i) = (P(i-1) + r*P(i-1)*(one - P(i-1)) - C(i-1)/K);
  }
  log_P = log(P);
  log_B = log_K + log_P;
  for (i=0;i<n-1;i++){
    P_midy(i) = half*(P(i)+P(i+1));
  }
  int ln=n-1;
  Type Pnp1 = (P(ln) + r*P(ln)*(one - P(ln)) - C(ln)/K);
  P_midy(ln) = half*(P(ln)+ Pnp1);
  log_P_midy = log(P_midy);

  log_H = log_C - log_B;
  H = exp(log_H);


// nll for index;

  log_Eindex = log_q(iq) + log_K + log_P_midy(iyear);
  vector<Type> resid = log_index - log_Eindex;
  vector<Type> std_resid = resid/sd_log_index;
```

```
  nll -= (dnorm(resid,zero,sd_log_index,true)).sum();


// the rest of the program produces report output;

  Type Hmsy = half*r;
  Type Bmsy = half*K;
  Type MSY = Hmsy*Bmsy;
  vector<Type> log_rB = log_B - log(Bmsy);
  vector<Type> log_rH = log_H - log(Hmsy);
  vector<Type> B = exp(log_B);
  vector<Type> Eindex = exp(log_Eindex);

  REPORT(log_r);
  REPORT(log_K);
  REPORT(log_q);
  REPORT(log_Po);
  REPORT(Hmsy);
  REPORT(Bmsy);
  REPORT(MSY);
  REPORT(log_B);
  REPORT(log_H);
  REPORT(B);
  REPORT(P);
  REPORT(H);
  REPORT(log_rB);
  REPORT(log_rH);
  REPORT(log_Eindex);
  REPORT(Eindex);
  REPORT(resid);
  REPORT(std_resid);
```

```
  ADREPORT(Hmsy);
  ADREPORT(Bmsy);
  ADREPORT(MSY);
  ADREPORT(log_rB);
  ADREPORT(log_rH);
  ADREPORT(log_B);
  ADREPORT(log_H);

  return nll;
}
```

A.4 TMB C++ and R codes for the state space model: Namibian hake data

A.4.1 C++ code

```
#include <TMB.hpp>
#include <iostream>

template<class Type>
Type objective_function<Type>::operator() ()
{
  DATA_IVECTOR(year);
  DATA_VECTOR(C);
  DATA_VECTOR(index);
  DATA_IVECTOR(iyear);
  DATA_IVECTOR(iq);
  DATA_VECTOR(log_C);
  DATA_VECTOR(log_index);
```

```
DATA_SCALAR(E_log_r);
DATA_SCALAR(sd_log_r);
DATA_SCALAR(E_log_Po);
DATA_SCALAR(sd_log_Po);
DATA_SCALAR(sd_logC);

PARAMETER(log_r);
PARAMETER(log_K);
PARAMETER_VECTOR(log_q);
PARAMETER(log_Po);
PARAMETER(log_Ho);
PARAMETER(log_sd_rw);
PARAMETER(log_sd_log_index);
PARAMETER(log_sd_pe);
PARAMETER(logit_ar_pe);

PARAMETER_VECTOR(log_pe);
PARAMETER_VECTOR(log_H_dev);

int n = year.size();
int ni = index.size();
int i;

Type r = exp(log_r);
Type K = exp(log_K);
Type sd_rw = exp(log_sd_rw);
Type sd_log_index = exp(log_sd_log_index);
Type sd_pe = exp(log_sd_pe);

vector<Type> log_P(n); //log population biomass divided by K at
start of the year;
```

```
  vector<Type> log_P_midy(n); // log P at middle of year;
  vector<Type> P(n);
  vector<Type> P_midy(n);
  vector<Type> log_B(n);
  vector<Type> log_H(n);
  vector<Type> H(n);
  vector<Type> log_Eindex(ni);
  vector<Type> log_EC(n); // model log catch;
  vector<Type> pe = exp(log_pe);

  Type one = 1.0;
  Type half = 0.5;
  Type zero = 0.0;
  Type ar_pe = exp(logit_ar_pe)/(one + exp(logit_ar_pe));

  Type nll=0;

// prior nll for log_r;
//  nll -= dnorm(log_r,E_log_r,sd_log_r,true);

// prior nll for log_Po;
 // nll -= dnorm(log_Po,E_log_Po,sd_log_Po,true);

// log of production model;

  P(0) = exp(log_Po);
  log_H(0) = log_Ho;
  H(0) = exp(log_H(0));
  for (i=1;i<n;i++){
    log_H(i) = log_H(i-1) + log_H_dev(i-1);
    H(i) = exp(log_H(i));
```

```
    P(i) = (P(i-1)+ r*P(i-1)*(one-P(i-1))-H(i-1)*P(i-1))*pe(i-1);
  }
  log_P = log(P);
  log_B = log_K + log_P;
  log_EC = log_B + log_H;
  for (i=0;i<n-1;i++){
    P_midy(i) = half*(P(i)+P(i+1));
  }
  int ln=n-1;
  Type Pnp1 = (P(ln) + r*P(ln)*(one - P(ln)) - H(ln)*P(ln))*pe(ln);
  P_midy(ln) = half*(P(ln)+ Pnp1);
  log_P_midy = log(P_midy);


// nll for index;

  log_Eindex = log_q(iq) + log_K + log_P_midy(iyear);
  vector<Type> resid = log_index - log_Eindex;
  vector<Type> std_resid = resid/sd_log_index;

  nll -= (dnorm(resid,zero,sd_log_index,true)).sum();

// nll for catch;

  vector<Type> resid_C = log_C - log_EC;
  nll -= dnorm(resid_C,zero,sd_logC,true).sum();

// nll for random walk deviation in log_H;

  nll -= dnorm(log_H_dev,zero,sd_rw,true).sum();
```

```
//   nll for log_pe process errors;
  i=0;
  nll -= dnorm(log_pe(i),zero,sd_pe/sqrt(one - ar_pe*ar_pe),true);
  for(int i = 1;i < n;++i){
    nll -= dnorm(log_pe(i) - ar_pe*log_pe(i-1),zero,sd_pe,true);
  }

// the rest of the program produces report output;

  Type Hmsy = half*r;
  Type Bmsy = half*K;
  Type MSY = Hmsy*Bmsy;
  vector<Type> log_rB = log_B - log(Bmsy);
  vector<Type> log_rH = log_H - log(Hmsy);
  vector<Type> B = exp(log_B);
  vector<Type> EC = exp(log_EC);
  vector<Type> Eindex = exp(log_Eindex);

  REPORT(log_r);
  REPORT(log_K);
  REPORT(log_q);
  REPORT(log_Po);
  REPORT(log_Ho);
  REPORT(Hmsy);
  REPORT(Bmsy);
  REPORT(MSY);
  REPORT(log_B);
  REPORT(log_H);
  REPORT(B);
  REPORT(H);
  REPORT(log_rB);
```

```
  REPORT(log_rH);
  REPORT(log_EC);
  REPORT(resid_C);
  REPORT(log_Eindex);
  REPORT(Eindex);
  REPORT(resid);
  REPORT(std_resid);
  REPORT(log_pe);
  REPORT(log_H_dev);

  ADREPORT(Hmsy);
  ADREPORT(Bmsy);
  ADREPORT(MSY);
  ADREPORT(ar_pe);
  ADREPORT(log_rB);
  ADREPORT(log_rH);
  ADREPORT(log_B);
  ADREPORT(log_H);

  return nll;
}
```

A.4.2 R code

```
load("tmb.RData")

library(TMB)
library(numDeriv)

compile("fit.cpp")
```

```
dyn.load("fit")


parameters <- list(
  log_r = log(0.36),
  log_K = log(2800),
  log_q = log(1/10),
  log_Po = log(1),
  log_Ho = log(0.1),
  log_sd_rw = log(0.2),
  log_sd_log_index = log(0.3),
  log_sd_pe = log(0.1),
  logit_ar_pe = log(0.50/(1-0.50)),
  log_pe = rep(0,length(tmb.data$C)),
  log_H_dev = rep(0,length(tmb.data$C)-1)
)

parameters.L <- list(
  log_r = log(0.2),
  log_K = log(2000),
  log_q = -Inf,
  #log_Po = log(0.1),
  log_Ho = log(0.0001),
  log_sd_rw = log(0.01),
  log_sd_log_index = log(0.01),
  log_sd_pe = -Inf,
  logit_ar_pe = -Inf)

parameters.U <- list(
  log_r = log(0.5),
  log_K = log(14271),
```

```
  log_q = Inf,
  #log_Po = log(10),
  log_Ho = log(1),
  log_sd_rw = log(2),
  log_sd_log_index = log(1),
  log_sd_pe = log(0.35),
  logit_ar_pe = log(0.950/(1-0.950)))


lower = unlist(parameters.L);
upper = unlist(parameters.U);


## random effects;
rname = c("log_pe","log_H_dev")

map = list(
##log_sd_pe = factor(NA),
log_Po = factor(NA))

obj <- MakeADFun(tmb.data,parameters,,map=map,random=rname,DLL="fit",
  inner.control=list(maxit=100,trace=T))

obj$gr(obj$par)

opt<-nlminb(obj$par,obj$fn,obj$gr,lower=lower,upper=upper,
control = list(trace=0,iter.max=5000,eval.max=10000))

opt$message
opt$convergence # will gives "0"
```

```
obj$gr(opt$par)

opt$par
exp(opt$par)
```

A.4.3 Classical linear models

The following assumptions define the classical linear model (CLM).

CLM: y = Xβ +ε, where y is a $n \times 1$ vector of observations on a dependent variable, X is a $n \times p$ matrix of observations on explanatory variables, β is a $p \times 1$ vector of fixed parameters, and ε is a $n \times 1$ vector of random disturbances.

Assumptions: ε_i's are assumed to be independent and identically distributed normal random variables with mean zero and known variance σ^2.