Canadian Journal of Fisheries and Aquatic Sciences
Journal canadien des sciences halieutiques et aquatiques

Precision and bias of parameter estimates through simulation analysis of a multi-region tag-integrated catch-at-age assessment model

Journal:	*Canadian Journal of Fisheries and Aquatic Sciences*
Manuscript ID	cjfas-2016-0297
Manuscript Type:	Article
Date Submitted by the Author:	01-Jul-2016
Complete List of Authors:	Vincent, Matthew; Michigan State University, Department of Fisheries and Wildlife Brenden, Travis; Michigan State University, Department of Fisheries and Wildlife Bence, James; Michigan State University
Keyword:	TAGGING < General, STOCK ASSESSMENT < General, MARK-RECAPTURE < General, POPULATION DYNAMICS < General, POPULATION STRUCTURE < General

Precision and bias of parameter estimates through simulation analysis of a multi-region tag-integrated catch-at-age assessment model

Matthew T. Vincent

vince151@msu.edu Quantitative Fisheries Center, Department of Fisheries and Wildlife, Michigan State University, 293 Farm Lane Room 153, East Lansing, Michigan, USA, 48824 [1]

Travis O. Brenden

brenden@anr.msu.edu Quantitative Fisheries Center, Department of Fisheries and Wildlife, Michigan State University, 293 Farm Lane Room 153, East Lansing, Michigan, USA, 48824

James R. Bence

bence@msu.edu Quantitative Fisheries Center, Department of Fisheries and Wildlife, Michigan State University, 293 Farm Lane Room 153, East Lansing, Michigan, USA, 48824

[1]Corresponding author: Matthew T Vincent (email: mtvincen@vt.edu, Telephone: 517-355-0003).

Abstract

Integrated Tagging and Catch-at-Age ANalysis (ITCAAN) models frequently involve the estimation of many parameters, but the influence of model complexity on precision and bias of estimated parameters is not well understood. Simulation analysis was used to investigate the accuracy and precision of ITCAAN models. We simulated the dynamics of four fish stocks with natal homing that intermixed during periods of harvest. Scenarios examined included varying levels of movement, whether natural mortality and/or reporting rate were treated as known or estimated, tagging cohort size, assumed spatial complexity in parameters, and degree of similarity in spawning stock productivities. We found that ITCAAN models were robust for estimating movement rates. Accuracy and precision of model estimates generally decreased with greater model complexity, but were more precise and less biased than when natural mortality or reporting rate was misspecified. At high movement rates, recruitments for the least productive stocks were overestimated, whereas the most productive stocks' recruitments were underestimated. ITCAAN model estimates of recruitment were unbiased regardless of movement when spawning stocks had similar productivity levels.

Introduction

Fisheries scientist and managers commonly assume fish harvested in a single management unit originate from an isolated spawning stock (Beverton and Holt 1957). The validity of this assumption has come under increased scrutiny as a result of recent recognition of stock complexity (Cadrin et al. 2004; Haponski and Stepien 2014; La Valley and Feeney 2013), and the migratory behavior exhibited by both freshwater and marine species (Haist et al. 1999; Punt et al. 2000; McGarvey et al. 2010; Goethel et al. 2015a; Hayden et al. 2014; Vandergoot and Brenden 2014). Numerous simulation studies have shown that ignoring spatial complexity of intermixed stocks can result in the overexploitation of less productive stocks (Ying et al. 2011; Guan et al. 2013; Hulson et al. 2013; Molton et al. 2013; Li et al. 2014). The collapse of some high profile fish stocks such as Atlantic cod (*Gadus morhua*) (Fu and Fanning 2004; Hutchinson 2008) and many Pacific salmon stocks (Morishima and Henry 1999) are believed to have in part been caused by inadequate accounting of spatial complexity arising from movement. This in turn led to the development and use of assessment methods that account for the spatial complexity of migratory behavior (Hampton 1991; Goethel et al. 2011; Maunder and Punt 2013) and that can reliably estimate abundances, mortality components, and/or movement rates of multiple stocks in mixed populations.

Integrated tagging and catch-at-age analysis (ITCAAN) assessment models, which incorporate return/recovery information from tagging studies as data components within statistical catch-at-age (i.e., integrated) assessment models, have been used to estimate the abundance and mortality rates of multiple species (Maunder 2001; Fielder and Bence 2014; Goethel et al. 2015a). However, parameter estimability has not been rigorously tested for ITCAAN models. Most simulation evaluations have been for models with subpopulations with reproductive mixing, but few analyses have been conducted for models assuming overlapping populations with natal homing (Goethel et al. 2011; Hulson et al. 2013; Goethel et al. 2015b). The incorporation of tagging data in ITCAAN models ostensibly allows the models to estimate parameters beyond those in traditional assessment models (e.g., fishing mortalities, catchabilities, selectivities, recruitment, and initial abundances); however, the extend of parameterization that can accurately be estimated by incorporating tagging data is uncertain and requires investigation. The mixing or movement rate of the stocks is an essential parameter for a spatially-explicit catch-at-age assessment model, and incorporating tagging data will likely inform the estimation of these rates. Natural mortality may be another parameter that is estimable in an ITCAAN model. Catch-at-age models typically require an a priori estimate of natural mortality (Doubleday 1976; Deriso et al. 1985), though attempts have been made to estimate natural mortalities using information inherent in the age composition with varied success (Wang and Liu 2006; Lee et al. 2011). However, tag-recovery data frequently are used to estimate the natural mortality rates of tagged cohorts of fish (Hoenig et al. 1998;

Latour et al. 2001; 2003; Frusher and Hoenig 2003; Vandergoot and Brenden 2014). ITCAAN models that combines the information in harvest age composition and tag-recovery data may result in reliable estimates of natural mortality. Fishery reporting rates of recovered tags are additional parameters that are seemingly estimable in ITCAAN models (Goethel et al. 2015b). However, simulation studies have not tested to what extent parameters such as movement, tag reporting, and natural mortality rates can be reliably estimated in combination with each other or with parameters such as recruitment, initial abundances and fishing mortality. It is possible, even likely, that some of these parameters may be confounded and the estimability of some parameters may be affected by the degree of movement among or differences in productivity of the fish stocks.

The goal of our research was to evaluate the estimability of model parameters within a release-conditioned ITCAAN model under a range of conditions. Simulations were conducted exploring how accuracy and precision of parameter estimates were affected by the following: 1. movement rate and whether natural mortality and/or reporting rates were estimated or treated as known and fixed quantities; 2. misspecification of reporting rate and natural mortality when treated as known and fixed quantities in the ITCAAN model; 3. varying levels of cohort tagging size; 4. assumptions as to spatial complexity of reporting rate and natural mortality in the ITCAAN model; 5. parity in productivities of the spawning stocks (i.e., equal stock-recruitment relationships).

Methods

The simulation framework used in this research consisted of an operating model that generated and tracked the true dynamics of four fish stocks and tagged cohorts. Given the true dynamics from the operating model, an observed time-series of data (e.g., fishery harvest, fishery harvest age composition, tag-recovery data) were generated and a release-conditioned ITCAAN model used this data to estimate dynamics of the fish stocks. Some aspects of the operating model were based on walleye (*Sander vitreus*) populations in Lakes Erie and Huron from the Laurentian Great Lakes region of North America, but the operating model was intended to be sufficiently generic for the results to be applicable to elsewhere. The spatial framework consisted of four major spawning stocks that overlapped with four regions of harvest. After spawning at the beginning of the year, individuals from each of the spawning stocks could move to any of the harvest regions. Unique fisheries operated in each harvest region with independent fishing dynamics, so that fishing mortality could vary among regions. The four spawning stocks differed considerably with regards to productivity (i.e., stock-recruitment steepness), although as part of sensitivity analyses we explored how results changed when productivity was similar among spawning stocks. The operating model generated a 40-year time period of observations, with the ITCAAN model applied once at the end of the time period. Both the operating model and ITCAAN models followed the dynamics for an age range of 2 to 7 years, with the last age class an aggregate group including age-7 and older fish. The operating model was programmed in R version 2.15.1 (R Development Core Team 2016), whereas the ITCAAN model was programmed in AD Model Builder version 11.5 (Fournier et al. 2012). Symbols and equations used to model the dynamics described below for both the operating model and ITCAAN model are presented in the section A. Parameter values assumed for the operating model regardless of simulation scenario are presented in the Supplementary Materials

Operating Model

Recruitment within spawning regions assumed Ricker stock-recruitment functions with spawners equal to the spawning biomass two years prior to the year of recruitment and an autocorrelated recruitment deviation randomly generated independently for each stock (Equation A.1). Recruitment steepnesses, which were used to represent productivity of the individual stocks, were chosen so there was considerable variation among stocks, consistent with information for walleye stock-recruitment patterns within areas of Lakes Erie and Huron (Figure 1; Supplementary Materials). Spawning site fidelity was assumed to be 100 % with instantaneous return annually at time of spawning. Recruitment deviations on log_e scale for the spawning stocks were generated from a first-order autoregressive process (Equation A.2). A hierarchical Bayesian approach was used by Thorson et al. (2014) to estimate posterior distributions of the mean autocorrelation coefficient and innovations (uncorrelated errors) variance for a variety of exploited taxonomic orders. Values of the autocorrelation coefficient (ρ_s) and innovations variance (σ_s) for each spawning stock and simulation iteration were randomly generated from the posterior distribution estimated for Percidae (Thorson et al. 2014). The mean of the autocorrelation process, δ_y, was configured such that it would have a mean of 1 when exponentiated (Thorson et al. 2016). Spawning was assumed to occur at the beginning of the year.

Abundances at age for the spawning stocks were modeled using an exponential population model that accounted for movement of stocks to each of the harvest regions (Equation A.3 and A.4). Region-specific total mortality was partitioned into natural and fishing mortality (Equation A). The apical fishing mortality rates for the harvest regions were randomly generated from first-order autoregressive processes (Equation A.5), where the means of the processes were based on estimates of fully selected fishing mortality from Lakes Erie and Huron (Supplementary Materials). The autocorrelation coefficients and innovations variances of the processes were based on the fully-selected total fishing mortality estimated for walleye in the western basin of Lake Erie (Wills et al. 2015). Age-specific fishing mortalities for the regions and years were generated by multiplying the corresponding region and year specific apical fishing mortality rates by age-specific selectivities (i.e., vulnerabilities) that were constant over time (Equation A.6). Fishery effort data were generated by dividing apical fishing mortality time-series by assumed region-specific catchability coefficients and multiplying by a lognormal observation error (Equation A.7). Assumed selectivities and catchabilities were based on estimated values for Walleye in Lake Erie (Wills et al. 2015) and Lake Huron (Fielder and Bence 2014). Movement was based on a box-transfer process that assumed instantaneous movement to harvest regions in which fish remained for the remainder of the year (Goethel et al. 2011). The proportion of the stock that moved to each harvest region varied depending on the examined scenario (see Simulation Scenarios), but in all cases were assumed to be spatially, temporally, and age invariant. A fishery independent survey was assumed to be conducted on each harvest region during October when fish were intermixed. Survey indices of abundance at age were generated from true abundances at age multiplied by region-specific catchabilities and region- and age-specific vulnerabilities (Equation A.8).

Cohorts of tagged fish from each spawning stock were assumed to experience the same dynamics as the at-large populations. All tagging was conducted when fish were located in their spawning regions. Tags were allocated to different ages based on an assumed set of proportions (Supplementary Material), but a single batch code was assumed to be applied to all ages (Equation A.9). Tagging-induced mortality and tag shedding were assumed to not occur. Actual numbers of fish from a tagged cohort that moved to each harvest region after spawning were generated from a multinomial distribution with the underlying proportions equal to the assumed movement rates under examination (Equation A.10). The probability of a tagged fish being harvested (Equation A.11), surviving (Equation A.12), or dying (Equation A.13) in a harvest region was then calculated based on the fishing, total, and natural mortalities in a region. These probabilities (Equation A.14) were used in a multinomial random number generator to determine the number of fish in each fate category (Equation A.15). A binomial random variable based on an assumed reporting rate with a sample size equal to the number of tags recovered was then used to determine the number of harvested fish that were reported (Equation A.16). A reporting rate of 50 % was assumed in the operating model for all examined scenarios and fisheries. The number of tagged fish that survived were then progressed to the next age and year (Equation A.17). The number of tags never recovered was calculated as the number of tags released minus the total tags returned summed across year, age and region (Equation A.18).

Total harvest, harvest age composition, fishing effort, survey index of abundance, survey index of abundance age composition, and tag recovery data were assumed to be available for each region and most were subject to observation error. Total annual harvest and fishing effort were generated by multiplying the true harvest and fishing effort by a log-normal observation error with a CV of 10 % (Equation A.19 and A.7, respectively). The observed survey index of abundance was generated by multiplying the true index of abundance summed across ages by a log-normal random variable with a CV of 20 % (Equation A.20).

The age composition samples for the harvest and survey (Equation A.8) were simulated from multinomial distributions with samples sizes of 1000 for every year.

ITCAAN Model

The ITCAAN model was similar to Goethel et al. (2015a), but differed in that ours modeled 100 % natal homing and in the number of spawning stocks and harvest regions. The dynamics assumed in the ITCAAN model were similar to assumptions in the operating model (e.g., box-transfer movement, 100 % spawning site fidelity, tagged cohorts, and at-large populations experiencing the same dynamics) (Equation A.21, A.22 and A.23). Preliminary investigations found that estimating the coefficients of a Ricker stock-recruitment function as part of the ITCAAN model resulted in poor performance and model convergence problems. As a consequence, annual recruitment in the ITCAAN model was estimated through a random walk process in which the multiplicative random walk deviations were assumed to be from a log-normal distribution with a standard deviation equal to 4.0 (Equation A.24 and A.25). The ITCAAN model assumed that recruitment during the last two modeled years were equal to the mean recruitment for the previous three years for each region, which was necessary for the model to converge with a positive definite Hessian matrix. Abundance for ages 3 to 7 in the first modeled year were estimated as the product of a mean abundance and multiplicative age deviation terms that were constrained to sum to 0 (Equation A.26) and that were assumed to be from a log-normal distribution with standard deviation equal to 4.0 (Equation A.27). Region-specific fishing mortalities were assumed to be products of annual fishing effort data, age-specific selectivities and year-specific catchabilities (Equation A.28). Year-specific fishery catchability were modeled using a random-walk process, as advocated by Wilberg and Bence (2006) as a default approach for modeling fishery catchability based on simulation results (Equation A.29 and A.30). Age specific vulnerabilities (selectivities), which were constant through time were estimated for ages 2 through 7 for each fishery. Depending on the examined scenario, natural mortality and reporting rate were either estimated or set equal to assumed (sometimes misspecified) values. Region specific survey catchabilities and survey vulnerabilities-at-age were assumed constant over time and estimated in the ITCAAN model. Movement rates, including the stay rates, were estimated through a multinomial logit transformation that constrained movement rates to be between 0 and 1 and to sum to 1 (Vandergoot and Brenden 2014). The formulation is similar to Goethel et al. (2015a) except the parameter for movement to Region 4 for all stocks was set equal to 0, instead of the residency parameter, to make the model identifiable (Equation A.31). As with the operating model, movement rates were assumed to be spatially, temporally and age invariant. Reporting rates were estimated through a logistic function, which constrained the reporting rate to be between 0 and 1, while allowing the estimated parameter to be a real number.

Highest posterior density estimation, which is also referred to as maximum penalized likelihood, was used to estimate the parameters of the ITCAAN model. Diffuse upper and lower bounds were specified for all parameters to keep the optimization algorithm from flat parts of the likelihood surface. The objective function was the sum of multiple negative log-likelihood and log-penalty components. Log-normal distributions were assumed for the log-likelihoods for region-specific total fishery harvests (Equation A.32) and survey indices(Equation A.33) and log-penalties for the catchability (Equation A.34) and recruitment random walk deviations (Equation A.25) and initial abundance-at-age white-noise deviations (Equation A.27). The log-standard deviation of the harvest data for each fishery was an estimated parameter. The log-standard deviations of the fishing effort and survey indices of abundance were calculated based on assumed ratios of their variances relative to the estimated variance of the harvest data (Equation A.35). The assumed ratios were equal to the actual ratios in variances from the operating model. The log-standard deviations for the recruitment and initial abundance deviations were set equal to 4.0 (Equation A.35). Age-composition data from the harvest and survey were assumed to be multinomially distributed with effective samples sizes equal to 150 (Equation A.36 and A.37). The number of tags returned was predicted based on the known number of tags released by age each year, estimated stock movement rates, survival estimates, and a fishery specific estimated reporting rate (Equation A.38). Yearly proportion of regional tag returns and tags never recovered relative to the total number of tags released were assumed to be multinomially distributed (Equation A.39). The proportion of tags recovered were calculated as the predicted number of recoveries by a fishery during a given year divided by the total number of tags released for a tagging cohort(Equation A.40). The proportion of tags never recovered were calculated as the total number of tags released minus the total tags returned summed over recovery year and region for each individual release event divided by the total number of tags released in the event (Equation A.40).

The maximum gradient convergence criterion in ADMB was set to 0.05. The simulated data was created and the ITCAAN model was applied to give an observed maximum gradient value. To be included in the analysis, the value of the maximum gradient for the simulation must be less than 0.05 and a positive definite Hessian must exist. Simulations were conducted until 1000 iterations had successfully met this convergence criteria. Code for the simulation and ITCAAN models can be found in the supplementary materials.

Simulation Scenarios

Five groups of simulation analyses were conducted to explore ITCAAN model performance (Table 1). The first group of scenarios explored in combination how parameter estimates were affected by variation in movement rates and whether reporting rates and/or natural mortality rates estimated or assumed known. The second group of scenarios explored sensitivity of parameter estimates to misspecification of reporting rates or natural mortalities when these parameters were assumed known. The third group of scenarios were conducted to examine the influence of tagging cohort size on parameter estimates. The fourth group of scenarios examined the ability to estimate spatially varying reporting rates and/or natural mortalities. The fifth scenario examined the influence on parameter estimates of assuming the same productivity for all regions.

Varying Movement and Reporting Rate and Natural Mortality Estimation

A total of 16 scenarios were conducted examining the interconnection between movement rates and the estimation of reporting rate and natural mortality. Four movement rates (the percentage of the stocks that moved to each non-natal region, (1 %, 5 %, 10 %, and 20 %) were examined, each crossed with four estimation scenarios. The four estimation scenarios were defined by whether or not reporting rate was estimated, crossed with whether or not natural mortality rate was estimated. If reporting rate and/or natural mortality was not estimated, it was assumed to be fixed at the correct value. For all cases, a natural mortality of 0.32 and a reporting rate of 50 % were assumed in the operating model. These scenarios were examined assuming a tagging cohort size of 2000 fish in each region and year. The ITCAAN model in these scenarios assumed one natural mortality rate for all regions but estimated a unique reporting rate for each fishery.

Sensitivity to Misspecified Reporting Rate and Natural Mortality

The second group of scenarios explored the consequence on parameter estimates of misspecifying the reporting rate or natural mortality parameters in the ITCAAN model. Three scenarios were explored under this group, all assuming the highest movement rate (20 %) and tagging cohort size of 2000 fish in each region. In the first scenario, a reporting rate of 75 % was assumed in the ITCAAN model when the true reporting rate in the operating model was 50 % for all fisheries. The second and third scenarios consisted of assuming a natural mortality of 0.16 and 0.48, respectively, in the ITCAAN model when the true natural mortality rate was 0.32. When natural mortality was estimated in the ITCAAN models it was assumed to be constant across regions, whereas when reporting rates were estimated they were assumed to be unique for each fishery.

Tag Cohort Size

The third group of scenarios examined sensitivity of parameter estimates to tagging cohort size. In these scenarios, both reporting rate and the natural mortality rate were estimated as described for the first group of scenarios. Four scenarios were considered with fewer tags than in the previous scenarios released in each region (1500, 1000, 500 and 250, per year and region). The simulations were conducted assuming a 20 % movement rate in the operating model. Natural mortality and reporting rate parameters were estimated with the same assumptions as the first scenario.

Spatial Complexity in Reporting Rates and Natural Mortalities

The fourth group of scenarios evaluated the consequences of estimating spatially varying or constant natural mortality and/or reporting rate in the ITCAAN model.The operating model generated data assuming that both rates were spatially constant using the values described above (reporting rates=50 %; M=0.32). We investigated all combinations of cases where natural mortality was either assumed constant spatially or estimated by region, crossed with cases where reporting rates were spatially constant or estimated by region (4 scenarios). These simulations were conducted assuming a 20% movement rate and a tagging cohort size of 2000 fish in each region and year in the operating model. The spatially constant natural mortality and regionally estimated reporting rate scenario was investigated in the first group scenario (B20), which are included in figures for comparison.

Equal Productivity

The final scenario investigated the influence of assuming the same Ricker stock-recruit parameters for all the regions. The Region 2 stock-recruit parameters were used as the basis for this scenario. However, the autocorrelation, standard deviation from the recruitment curve and annual recruitment values were unique for each region. The operating model assumed that the emigration rate was 20 % and 2000 tags were released in each region every year. The ITCAAN model assumed natural mortality was constant across regions and reporting rates were regionally unique, as described in the first group of scenarios.

Performance Metrics

The performance of the ITCAAN model was explored by comparing parameter estimates to the true values assumed in the operating model. For the sake of brevity, we discuss the precision and bias in fishery catchability coefficients, annual recruitment estimates, natural mortality, and reporting rates, results for all other parameters are shown in the Supplementary Material. The fishery catchability coefficients were investigated as a measure of fishing mortality estimation accuracy, whereas annual recruitment estimates were investigated to give a measure of abundance estimation accuracy. Error in natural mortality and reporting rates were investigated to assess parameter estimability and the influence of misspecification in the ITCAAN model. The percent relative error for all estimated parameters were calculated by subtracting the true value from the estimate and then dividing by the true value and multiplying by 100. However, error in movement rate estimates were also assessed (and presented) as actual error, estimate minus true, given that these values were already percentages. The median and interquartile ranges (IQR) of the percent relative and actual error of the 1000 simulated datasets were used to gauge ITCAAN model accuracy and precision for each scenario.

Results

Although we quantified percent relative error in estimates for all ITCAAN parameter estimates, for the sake of brevity we focus on the results for fishery catchabilities, recruitments, movement rates, natural mortalities and reporting rates. The results for the survey catchabilities were overall quite similar to that of the fishery catchabilities, whereas the results for initial abundances were similar to those of recruitment. Results for fishery and survey selectivities were relatively unaffected by the different scenarios. Exceptions to these general results for particular scenarios are noted below. Figures displaying the relative errors for all parameters not touched on below are presented in the Supplementary Materials.

Varying Movement and Reporting Rate and Natural Mortality Estimation

Movement rates had minimal influence on the precision of fishery catchabilities for the regions. Precision of the fishery catchabilities decreased (i.e., IQR of relative error increased) when reporting rate and natural mortality were both estimated in the ITCAAN model (Figure 2). This was most noticeable under the 20 % movement rate. A positive bias in fishery catchability was observed in Region 2 (i.e., the most productive region) under the 10 and 20 % movement rates; the degree of bias was greatest under a 20 % movement rate when reporting rates were estimated, either alone or in conjunction with natural mortality. For the other movement rates and estimation combinations, the degree of bias in fishery catchabilities was low.

Unlike fishery catchabilities, different movement rates had a much larger effect on both bias and precision of recruitment estimates (Figure 3). Under the 1 % movement rate, recruitment estimates were largely unbiased (median relative error between -0.73 and 1.27 %) and precise (IQR of relative error between 6.39 and 15.61 %). As movement rate increased, the level of imprecision increased and the magnitude of bias increased, with the direction of bias depending on the region. For Region 2 (i.e., the most productive region), recruitment estimates were generally negatively biased, whereas recruitment estimates in other regions were positively biased. The degree of bias and imprecision were generally the greatest when reporting rate was estimated, either alone or in conjunction with natural mortality. For example, when movement rates were 20 %, median relative error in recruitment estimates was 87 % for region 3 when neither natural mortality nor reporting rate were estimated, but was 288 % when both were estimated (Figure 3). With a movement rate of 20 %, IQR was 322 % for region 3 when neither natural mortality nor reporting rates were estimated, whereas IQR when both were estimated was 742 % for region 3.

Movement rate estimates were largely unbiased regardless of the assumed movement rate and whether reporting rates and/or natural mortality were estimated or treated as known (Figure 4). Median actual errors were within -0.12 and 0.07 % for all examined scenarios and regions. Precision in the movement rate estimates decreased as the level of assumed movement increased, however the degree of precision was similar regardless of whether reporting rates and/or natural mortality were estimated (Figure 4).

Natural mortality estimates were generally accurate and precise when estimated as a parameter in the ITCAAN model across each of the examined scenarios (Figure 5). When natural mortality was estimated and reporting rate was fixed at its true value, the IQR of the relative error was 0.91 %. However, estimating both reporting rates and natural mortality decreased the precision of natural mortality estimates, and the extent of change was larger when movement rates were higher. Similar results with respect to precision were observed for reporting rate (i.e., precision decreased when both natural mortality and reporting rate were estimated and movement rate increased) (Figure 5). Unlike natural mortality estimates, however, biased reporting rates for some regions did occur for some of the examined scenarios. This bias was most noticeable for the most productive region under a 20 % movement rate; a median relative error of approximately -10 % was observed when reporting rate was estimated alone or in conjunction with natural mortality for Region 2. Conversely, median relative errors in reporting rates for the other regions were generally within ±5 %.

Sensitivity to Misspecified Reporting Rate and Natural Mortality

Misspecification of the reporting rate or natural mortality in the ITCAAN model caused biases in nearly all parameter estimates. A negative bias in fishery catchabilities resulted when assuming a natural mortality value in the ITCAAN model that was 1.5 times that of the true value in the operating model (0.48 versus 0.32). Across the regions, the median relative error in fishery catchability coefficients was approximately -25 % (Figure 6). The precision in fishery catchability estimates were overall similar to the results obtained under the first group of examined scenarios at comparable rates of movement. Similar results were obtained when the reporting rates in the ITCAAN model were 1.5 times greater than in the operating model (75 % versus 50 %). Conversely, when the natural mortality rate in the ITCAAN model was half the true value in the operating model (0.16 versus 0.32), the median relative error in the catchability coefficients was close to 100 %. Precision in the fishery catchabilities was also affected by fixing natural mortality in the ITCAAN model at too low of a value. The IQR of the relative error for fishery catchabilies was between 8.61 and 9.18 %, which was approximately half that obtained under the first group of examined scenarios at comparable rates of movement. Estimates of fishery and survey selectivities were not strongly influenced by misspecification of the reporting rates to half of the true value in the ITCAAN model. However, misspecification of natural mortality in the ITCAAN model to 1.5 times the value in the operating model resulted in a decreased precision (IQR of survey selectivities were approximately three times the value in the B20 scenario) and large biases in median estimates, but the direction of bias varied among region (Supplementary Materials). Similarly, natural mortality misspecified in the ITCAAN model to half the true value in the operating model resulted in larger IQRs in relative errors for selectivity estimates but median relative errors were typically close to zero.

Misspecification of natural mortality or reporting rates in a positive direction (i.e., 1.5 times that of the true value) resulted in a positive bias in annual recruitment estimates for all regions (Figure 7). Conversely, misspecification of natural mortality in a negative direction (i.e., half of the true value) resulted in a positive bias regions 1 and 3, but a negative bias for regions 2 and 4. The precision of recruitment estimates increased for all regions when natural mortality was misspecified in the negative direction in the ITCAAN model (IQR of relative error: region 1 = 243 %; region 2 = 11.4 %; region 3 = 398 %; region 4 = 221 %) compared to the same movement rate scenario in the first group (B20: IQR of relative error: region 1 = 294 %; region 2 = 16 %; region3 = 742 %; region 4 = 520 %). IQRs of recruitment relative error when reporting rate was misspecified in the positive direction in the ITCAAN model were similar to those when natural mortality was misspecified in the negative direction. On the other hand, precision in recruitment estimates decreased when natural mortality was misspecified in the positive direction in the ITCAAN model (IQR of relative error: region 1 = 329 %; region 2 = 31 %; region 3 = 1201 %; region 4 = 969 %).

Estimated movement rates were largely unaffected by misspecification of reporting rates or natural mortality in the ITCAAN model. Precision in the estimates as measured by the IQR of the actual errors in movement rates were comparable to those found in the first group of scenarios under comparable movement rates. Slight biases were observed for regions 1 (median error = 1.05 %) and 2 (median actual error = -0.90 %) when natural mortality in the ITCAAN model was specified at 1.5 times the value assumed in the operating model (Figure 8); however, compared to the biases of other parameter estimates the degree of bias observed in movement estimates for this region were relatively small.

Natural mortality and reporting rate estimates were very sensitive to misspecification. When reporting rate in the ITCAAN model was specified at 1.5 times the value in the operating model, median relative error was around 25 % for the natural mortality estimates, with a very narrow IQR of relative error (Figure 9). When natural mortality in the ITCAAN model was specified at too high a value, the median relative error of reporting rate estimates was near 100 % with very little variability among simulation iterations. Effectively, this translated to reporting rates being estimated close to 100 % (i.e., perfect reporting) for all simulations. When natural mortality was specified at too low a value in the ITCAAN model, the median relative error of reporting rate estimates was near -50 %, also with little variability among the simulation iterations. The IQR in relative error for this scenario was similar to the RR20 scenario from the first group of examined scenarios.

Tag Cohort Size

Reducing tagging cohort size mostly resulted in reduced precision for estimated parameters with relatively little effect on the accuracy of parameters (figs. 6 to 9). The major exceptions to this were the fishery catchability (Figure 6) and reporting rate (Figure 9)for the most productive region and recruitment estimates for all regions (Figure 7). For these parameters, bias in estimates increased as tagged cohort size decreased.

Spatial Complexity in Reporting Rates and Natural Mortalities

Increasing the complexity of the ITCAAN model by allowing for spatially-specific estimates of natural mortality in addition to spatially-specific estimates of reporting rates, resulted in moderately lower precision in fishery catchability estimates for all regions and slightly increased the bias for fishery catchability in region 2 when compared to simulation scenario B20 (Figure 6). With respect to recruitment, greater spatial complexity in the ITCAAN model increased the bias and imprecision of estimates for all regions, with the direction of the bias remaining consistent for each region across the range of examined scenarios (Figure 7). Although movement rate estimates were unaffected by allowing for greater spatial complexity in the ITCAAN model (Figure 8), bias and imprecision of both natural mortalities and reporting rates increased at least for some regions (Figure 9). The most noticeable increase in bias was in the most productive region, although a small bias in reporting rate also occurred in region 3 under the most complex ITCAAN model (i.e., spatially unique estimates for reporting rates and natural mortality). For regions 1 and 4, reporting rate and natural mortality estimates were unbiased regardless of spatial complexity, although precision of the estimates decreased as the ITCAAN model became more complex (Figure 9).

Equal Productivity

When all regions had the same Ricker stock recruitment parameters, but different annual recruitments, the biases observed and reported for the first group of scenarios at comparable movement rates largely disappeared. This included biases in fishery catchability, natural mortality, and reporting rate in region 2 and recruitment in all regions (figs. 6 to 9). As with other investigated scenarios, movement rate estimates were very accurate. With respect to precision of estimates, the most notable consequence of the assumption of equal productivity was that precision of recruitment in region 1, 3 and 4 improved while the precision in recruitment estimates for Region 2 decreased (Figure 7). Precision for natural mortality and reporting rate improved with equal productivity across regions (Figure 9).

Discussion

Simulation analysis is an important tool for fisheries scientist to determine how well models can accurately and precisely estimate parameters. Extensive simulation analysis should be conducted whenever a new assessment model is proposed to evaluate its performance and gauge its robustness/sensitivity to violations in model assumptions, data quality, and/or structural uncertainties. Although ITCAAN models have been in use for a number of years (Maunder 1998; 2001; Goethel et al. 2015a), only models assuming subpopulations with reproductive mixing have been investigated (Hulson et al. 2011; 2013; Goethel et al. 2015b). Simulations to evaluate the performance of ITCAAN models with overlapping populations with natal homing have not been widely conducted and/or reported. Additionally, previous simulation studies have assumed two or three spawning stocks and harvest regions with one underlying recruitment and fishing mortality time series with region-specific observation error (Maunder 2001; Hulson et al. 2011; 2013; Goethel et al. 2015b). In our study, we expanded the number of assumed stocks and fishing regions and allowed for drastically different magnitude of spawning stock size along with autocorrelated recruitment deviations for the stocks. Additionally, for each simulation iteration, a different time-series of fishing mortalities for each region were generated in the operating model. In other words, we explored ITCAAN performance under a broader set of conditions compared to earlier studies.

In most previous simulation studies, the ability of ITCAAN models to estimate movement rates has been of primary interest. Investigation into the estimability of different movement rates were conducted assuming movement varied as a function of environmental variables (Hulson et al. 2013) or as a function regional population density (Goethel et al. 2015b). These studies found that ITCAAN models were unbiased for movement rates and biomass in most scenarios investigated, even when the movement parameters in the ITCAAN model were allowed to vary when the underlying movement rate varied. Our finding that ITCAAN models produced accurate and precise movement rates under conditions of natal homing and across a wide range of scenarios suggests that movement estimation is an overall robust feature of these models and a high degree of confidence can be placed in movement rates estimated in ITCAAN models.

Simulation analysis regarding the estimation of natural mortality in spatially explicit assessment models require additional investigation. The simulation analyses of Maunder (2001); Hulson et al. (2011; 2013);

Goethel et al. (2015b) all assumed a known and fixed value of natural mortality. The results of this study show that natural mortality and reporting rates can both be estimated with high precision and low bias in ITCAAN models and should serve as a starting point for future simulations to evaluate the precision and bias of estimating spatially varying natural mortality and reporting rates when the underlying dynamics also vary. However, careful consideration is required whether to model the rate of natural mortality as due to environmental conditions (i.e., due to the current region of residency), or due to genetics (i.e., attributed to natal region). Simulation analyses of misidentification of these factors could be conducted or model selection techniques (such as AIC or BIC) could be used to identify the most appropriate assumption for a specific situation.

Until recently (Goethel et al. 2015a;b), estimation of reporting rates in ITCAAN models has not generally been attempted. Our study found that reporting rates could be estimated with low bias and moderate precision when the natural mortality rate was correctly specified or estimated, which aligns with the findings of Goethel et al. (2015b). Conversely, when reporting rate is incorrectly specified in the ITCAAN model, the estimates of abundance (or biomass) are biased, but estimates of movement are unbiased (Goethel et al. 2015b, this study,). Our results also show that misspecification in natural mortality can result in biased estimation in reporting rates and other population dynamics parameters, which had previously not been investigated. We hypothesize that the low reporting rates estimated in the ITCAAN model of Goethel et al. (2015a) were lower than the estimated reporting rates from high reward tagging (Cadrin 2006) due to misspecification of natural mortality in the ITCAAN model. We advocate that natural mortality and reporting rate be estimated simultaneously in ITCAAN models and high reward tag data be incorporated into the model as well to inform reporting rate estimates.

One of the key findings from our simulations was that under conditions of high movement and varying stock-recruitment conditions for spawning populations, ITCAAN models are biased and imprecise estimators of recruitment. Higher movement rates affected precision of most other parameters that were considered, but not accuracy. The only exceptions to this were fishery catchability and natural mortality for the most productive regions, which also became biased under the highest movement rate evaluated. These biases in recruitments, natural mortalities, and fishery catchabilities dissipated when parity in stock-recruitment relationships was assumed for the spawning stocks. One of the major motivators for incorporating spatially-explicit dynamics in assessment models is the concern that less productive stocks may be overexploited or even extirpated if spatially-varying dynamics are not incorporated in the management process (Molton et al. 2013; Li et al. 2014). Although in our simulations we assumed fairly large differences in stock-recruitment relationships among the spawning stocks, such differences arguably may be more reflective of actual conditions for many species than an assumption of equality in stock-recruitment relationships.

Consequently, the potential for overestimating recruitment of less productive stocks and underestimating recruitment of more productive stocks under conditions of high movement and large differences in relative size of spawning stocks is an important issue to consider for fishery scientists looking to implement ITCAAN models.

Part of our motivations for this research was that we envisioned there could be confounding among parameters that fishery scientists might be interested in estimating when incorporating tagging data into a statistical catch-at-age model. In particular, we envisioned there could be complications that could arise when estimating movement rates, reporting rates, and/or natural mortality along with other parameters that are routinely estimated in assessment models. Across the range of scenarios considered in this research, movement rates were estimated accurately and precisely and thus can be regarded as very robust to recapture data quality and model assumptions. However, Goethel et al. (2015b) demonstrated that spatially explicit catch-at-age models without tagging data and high variance in catch-at-age data can result in poor movement estimates. We reiterate their assertion that high quality age composition data are imperative for accurate estimation in all age-structured stock assessment models. Our simulations showed some biases in reporting rates and natural mortalities could occur under conditions of high movement, small tagging cohort sizes, and level of assumed spatial complexity in the estimation of reporting rates and natural mortalities. Despite these biases observed in ITCAAN models under certain conditions, the level of bias was much smaller than what resulted when parameters were fixed at incorrect values in the ITCAAN model. For example, assuming a natural mortality 1.5 times greater than the true value resulted in median relative error in estimates of recruitment, reporting rates, and fishery catchabilities that were about 4 times, -80 times, and 16 times larger than when these parameters were estimated together under a 20 % movement rate (i.e., B20 scenario). Likewise, assuming too high of a reporting rate resulted in a median relative error in estimates of recruitment, natural mortalities, and fishery catchabilities that were about 1.25 times, -21 times, and 13 times larger than when parameters were estimated together. This sensitivity of parameter estimates to misspecification suggests that unless scientists have a high degree of confidence in external estimates of reporting rates and natural mortalities they would be better off estimating these parameters as part of the ITCAAN model even though the greater model complexity may lead to greater imprecision.

The decrease in precision in parameter estimates with reduced tagging cohort size and greater model complexity was anticipated. With models of this nature, there inevitably will be a compromise between ITCAAN model complexity and data quality. The application of model-selection approaches such as AIC or DIC, applied to an ITCAAN model may be beneficial for determining how complex of a model can be supported based on available data. For example, it is commonly assumed that different fishery types and/or areas have different reporting rates (Hilborn 1990; Brenden et al. 2010; Vandergoot and Brenden 2014;

Konrad et al. 2016) and model-selection criteria may be useful in determining whether spatially and/or fishery unique reporting rates can be supported by existing data (Wilberg and Bence 2008; Linton and Bence 2011). Other factors, such as tagging-induced mortality, tag shedding and spatial-allocation of tags, can influence how much tagging are available to incorporate in ITCAAN models, which if not accounted for could affect accuracy and precision of parameter estimates. Conversely, ITCAAN models could accommodate other data sources not considered in our research that could lead to improvements in both accuracy and precision of parameter estimates. For example, tagging studies sometimes include the release of both high- and low-reward tags or employ fishery observers or use planted tags for the purpose of estimating fishery reporting rates (Polacheck et al. 2006; Eveson et al. 2007); the inclusion of these additional tagging data would greatly aid in the estimation of reporting rates and likely improve the precision of other parameter estimates. The incorporation of other data sources, such as surveys when stocks were located on spawning grounds or information that could help identify harvested fish to individual spawning locations (e.g., genetic stock identification results) similarly could improve accuracy and precision of parameter estimates (Li et al. 2014; Tsehaye et al. 2016).

Although we attempted to incorporate a range of scenarios in this study, it is important to acknowledge that our results are nevertheless influenced by the assumed conditions, both in the operating and ITCAAN estimation model. Our assumed 40 year time-series of tagging and fishery harvest data is perhaps unlikely scenario of data availability except for highly valued species and a shorter time series of data may encounter different estimation issues. Second, we generated recoveries assuming a multinomial process, but overdispersion relative to a multinomial distribution in tagging data is often observed (Bacheler et al. 2008; Hanselman et al. 2015; Vandergoot and Brenden 2014; Mayakoshi and Kitada 2016), meaning our estimates of precision may be conservative. Third, we assumed tagged cohorts were fully mixed with the at-large population and that movements between spawning areas and harvest regions were instantaneous. If in reality there was delayed mixing of tagged cohorts (i.e., fish moved between harvest regions during other parts of the year), parameter estimation could be affected. Fourth, we assumed a known spatial stock structure, consisting of four distinct reproductive stocks that overlapped in four regions during the harvest season. Stock identification has challenges (Cadrin et al. 2004) in that lack of adequate spatial data could preclude spatially separating both reproductive stocks and fishery areas within an ITCAAN model, which would result in incorrect model specification. For example, an identified reproductive population could in fact consist of several sub-stocks, and a fishing region could have sub-regions with different fishing effort trends, which could be occupied differentially by the reproductive populations. We additionally assumed a single fishery operated in each harvest region, whereas in actuality there can be many fisheries that differ with respect to harvest levels, length of fishing season, reporting rate and other harvest dynamics that can make ITCAAN model estimation more complex. Lastly, we did not consider the weighting of tagging data versus harvest and survey data in the ITCAAN model and there may be situations where down-weighting of tagging data may be beneficial or necessary due to some of the issues mentioned above (non-mixing, more complex movements, etc.) (Fielder and Bence 2014; Goethel et al. 2015a).

In fitting the ITCAAN models as part of this research, one issue encountered was that different parameterizations could have large consequences on model fit. For example, initial configurations of our ITCAAN model estimated recruitment as a white-noise process that resulted in large systematic biases in recruitment estimates. Specifically, the ITCAAN model underestimated recruitment early in the time series and overestimated recruitment later in the time series; this result was consistent across many of the examined scenarios. Conversely, estimating recruitment through a random walk process with a large assumed variance term removed these systematic biases and greatly improved precision of the recruitment estimates. We suspect that the key feature here is that we allowed for the mean recruitment to be non-stationary (Maunder and Deriso 2003; Li et al. 2014). Similarly,issues were encountered in estimating initial abundances and the last few years of recruitment that affected whether the model could reliably produce a positive-definite Hessian matrix for many simulated datasets. Ultimately, the ITCAAN model parameterization that we used in our study was the best approach we could find to correct many of the estimation complications encountered. Other approaches might have worked better than our solution and it is possible that our solution may perform poorly under other conditions. Our purpose in pointing out the estimation issues that we encountered is that such issues can be easily overlooked or ignored in empirical applications of ITCAAN models ; therefore, it may be beneficial in real-world applications to attempt different parameterizations and determine sensitivity of estimates to these parameterizations. Incorporating the best-available information for a specific fishery/species of interest may be beneficial for determining an appropriate parameterization for an ITCAAN model. Therefore, we support the recommendation of Goethel et al. (2015b) that a simulation analysis should precede implementation of an ITCAAN model, based on the estimation issues we encountered.

In conclusion, we found that release-conditioned ITCAAN models yielded accurate and precise parameter estimates under moderate to low movement rates, but biases in some parameters could result under conditions of high movement and large differences in stock-recruitment relationships among spawning stocks. Misspecification of certain parameters, such as natural mortalities and reporting rates, were imparted larger biases, for observed misspecification levels, than when parameters were estimated; thus, we urge caution in fixing parameters at assumed values when utilizing ITCAAN models. We recommend additional investigation of factors such as the inclusion of additional data sources, greater levels of uncertainty in data sources, greater spatial complexity, weighting of tagging data relative to fishery harvest/survey data, temporal complexity of parameters (e.g., natural mortality and reporting rate), and alternative parameterizations (e.g., recruitment estimation as random walk or white noise) to gain additional perspectives on the performance of ITCAAN models.

Acknowledgments

This work was partially funded by grant 1250 of the Great Lakes Fishery Trust and 2013_BRE_44025 from the Great Lakes Fishery Commission. Additional funding support was provided by Supporting Partners of the Michigan State University Quantitative Fisheries Center. The authors acknowledge the support of Michigan State University High Performance Computing Center and the Institute for Cyber-Enable Research. This is publication number 20XX-XX of the Quantitative Fisheries Center, Michigan State University.

References

Bacheler, N.M., Hightower, J.E., Paramore, L.M., Buckel, J.A., and Pollock, J.H. 2008. An age-dependent tag return model for estimating mortality and selectivity of an estuarine-dependent fish with high rates of catch and release. Trans. Am. Fish. Soc **137**(5): 1422–1432. Doi: 10.1577/T07-254.1.

Beverton, R.J.H. and Holt, S.J. 1957. On the dynamics of exploited fish populations. Fishery Investigations Series II Volume XIX. Her Majesty's Stationary Office.

Brenden, T.O., Jones, M.L., and Ebener, M.P. 2010. Sensitivity of tag-recovery mortality estimates to inaccuracies in tag shedding, handling mortality and tag reporting. J. Great Lakes Res. **36**(1): 100–109. Doi: 10.1016/j.jglr.2009.09.002.

Cadrin, S.X. 2006. Yellowtail flounder tagging study. Technical report, Northeast Consortium Interim Final Report.

Cadrin, S.X., Friedland, K.D., and Waldman, J.R., editors 2004. Stock Identification Methods: Applications in Fishery Science. Elsevier Academic Press, Burlington, MA.

Deriso, R.B., Quinn, II, T., and Neal, P.R. 1985. Catch-age analysis with auxiliary information. Can. J. Fish. Aquat. Sci. **42**: 815–824. Doi: 10.1139/f85-104.

Doubleday, W.G. 1976. Least squares approach to analyzing catch-at-age data. Int. Comm. Northwest Atl. Fish. Res. Bull. **12**: 69–81. Available from http://icnaf.nafo.int/res-bull/res-bull-12.pdf#page=69 [accessed 25 May 2016].

Eveson, J.P., Polacheck, T., and Laslett, G.M. 2007. Incorporating fishery observer data into an integrated catch-at-age and multiyear tagging model for estimating mortality rates and abundnace. Fish. Bull. **105**(4): 493–508. Available from http://fishbull.noaa.gov/1054/eveson.pdf [accessed 25 May 2016].

Fielder, D.G. and Bence, J.R. 2014. Integration of auxiliary information in statisticial catch-at-age (sca) analysis of the Saginaw Bay stock of walleye in Lake Huron. N. Am. J. Fish. Manage. **34**(5): 970–987. Doi: 10.1080/02755947.2014.938141.

Fournier, D.A., Skaug, H.J., Ancheta, J., Ianellu, J., Magnusson, A., Maunder, M.N., Nielsen, A., and Sibert, J. 2012. AD Model Builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. Optim. Method. Softw. **27**(2): 233–249. Doi: 10.1080/10556788.2011.597854.

Frusher, S.D. and Hoenig, J.M. 2003. Recent developments in estimating fishing and natural mortality and tag reporting rate of lobsters using multi-year tagging models. Fish. Res. **65**: 379–390. Doi: 10.1016/j.fishres.2003.09.027.

Fu, C. and Fanning, L.P. 2004. Spatial considerations in the management of Atlantic Cod off Nova Scotia, Canada. N. Am. J. Fish. Manage. **24**(3): 775–784. Doi: 10.1577/M03-134.1.

Goethel, D.R., Legault, C.M., and Cadrin, S.X. 2015a. Demonstration of a spatially explicit, tag-integrated stock assessment model with application to three interconnected stocks of yellow tail founder off of New England. ICES J. Mar. Sci. **72**(1): 164–177. Doi: 10.1093/icesjms/fsu014.

Goethel, D.R., Legault, C.M., and Cardin, S.X. 2015b. Testing the performance of a spatially explicit tag-integrated stock assessment model of yellowtail flounder (*Limanda ferruginea* through simulation analysis. Can. J. Fish. Aquat. Sci. **72**(4): 582–601. Doi: 10.1139/cjfas-2014-0244.

Goethel, D.R., Quinn, II, T.J., and Cadrin, S.X. 2011. Incorporating spatial structure in stock assessment: Movement modeling in marine fish population dynamics. Reviews in Fisheries Science **19**(2): 119–136. Doi: 10.1080/10641262.2011.557451.

Guan, W., Cao, J., Chen, Y., and Cieri, M. 2013. Impacts of population and fishery spatial structures on fishery stock assessment. Can. J. Fish. Aquat. Sci. **70**(8): 1178–1189. Doi: 10.1139/cjfas-2012-0364.

Haist, V., Fournier, D., and Saunders, M.W. 1999. Reconstruction of B.C. sablefish stocks, 1966-198, and catch projections for 1999, using an integrated catch-age mark-recapture model with area and depth movement. Technical Report Research Document 99/79, Canadian Stock Assessment Secretariat, Ottawa, Canada.

Hampton, J. 1991. Estimation of southern bluefin tuna *Thunnus maccoyii* natural mortality and movement rates from tagging experiments. Fish. Bull. **89**(4): 591–610. Available from http://fishbull.noaa.gov/894/hampton2.pdf.

Hanselman, D.H., Heifetz, J., Echave, K.B., and Dressel, S.C. 2015. More it or lost it: movement and mortality of sablefish tagged in Alaska. Can. J. Fish. Aquat. Sci. **72**(2): 238–251. Doi: 10.1139/cjfas-2014-0251.

Haponski, A.E. and Stepien, C.A. 2014. Genetic connectivity and diversity of walleye (*Sander vitreus*) spawning groups in the Huron-Erie corridor. J. Great Lakes Res. **40**(2): 89–100. Doi: 10.1016/j.jglr.2012.12.006.

Hayden, T.A., Holbrook, C.M., Fielder, D.G., Vandergoot, C.S., Bergstedt, R.A., Dettmers, J.M., Krueger, C.C., and Cooke, S.J. 2014. Acoustic telemetry reveals large-scale migration patterns of Walleye in Lake Huron. PLoS ONE **9**(12): e114833. Doi: 10.1371/journal.pone.0114833.

Hilborn, R. 1990. Determination of fish movement patterns from tag recoveries using maximum likelihood estimators. Can. J. Fish. Aquat. Sci. **47**(3): 635–643. Doi: 10.1139/f90-071.

Hoenig, J.M., Barrowman, N.J., Hearn, W.S., and Pollock, K.H. 1998. Mutliyear tagging studies incorporating fishing effort data. Can. J. Fish. Aquat. Sci. **55**(6): 1466–1476. Doi: 10.1139/f97-256.

Hulson, P.J.F., Miller, S.E., Ianelli, J.N., and Quinn, II, T.J. 2011. Including mark-recapture data into a spatial age-structured model: Walleye pollock (*Theragra chalcogramma*) in the eastern Bering Sea. Can. J. Fish. Aquat. Sci. **68**(9): 1625–1634. Doi: 10.1139/F2011-060.

Hulson, P.J.F., Quinn II, T.J., Hanselman, D.H., and Ianelli, J.N. 2013. Spatial modeling of Bering Sea walleye pollock with integrated age-structured assessment models in a changing environment. Can. J. Fish. Aquat. Sci. **70**(9): 1402–1416. Doi: 10.1139/cjfas-2013-0020.

Hutchinson, W.F. 2008. The dangers of ignoring stock complexity in fisheries management: The case of the North Sea cod. Biol. Lett. **4**(6): 693–695. Doi: 10.1098/rsbl.2008.0443.

Konrad, C., Brattey, J., and Cadigan, N.G. 2016. Modelling temporal and spatial variability in tag reporting-rate for Newfoundland cod (*Gadus morhus*). Environ. Ecol. Stat. pp. 1–17. Doi: 10.1007/s10651-016-0344-0.

La Valley, K.J. and Feeney, R.G. 2013. Reconciling spatial scales and stock structure for fisheries science and management. Fish. Res. **141**(1). Doi: 10.1016/j.fishres.2013.02.014.

Latour, R.J., Hoenig, J.M., Hepworth, D.A., and Frusher, S.D. 2003. A novel tag-recovery model with two size classes for estimating fishing and natural mortality, with implications for the southern rock lobster (*Jasus edwardsii*) in Tasmania, Australia. ICES J. Mar. Sci. **60**(5): 1075–1085. Doi: 10.1016/S10543139(03)00093-6.

Latour, R.J., Hoenig, J.M., Olney, J.E., and Pollock, K.H. 2001. Diagnostics for multiyear tagging models with application to Atlantic striped bass (*Morone saxatilis*). Ca. J. Fish. Aquat. Sci. .

Lee, H.H., Maunder, M.N., Piner, K.R., and Methot, R.D. 2011. Estimating natural mortality within a fisheries stock assessment model: An evaluation using simulation analysis based on twelve stock assessments. Fish. Res. **109**(1): 89–94. Doi:10.1016/j.fishres.2011.01.021.

Li, Y., Bence, J.R., and Brenden, T.O. 2014. An evaluation of alternative assessment approaches for intermixing fish populations: a case study with Great Lakes white fish. ICES J. Mar. Sci. **72**(1): 70–81. Doi: 10.1093/icesjms/fsu057.

Linton, B.C. and Bence, J.R. 2011. Catch-at-age assessment in the face of time-varying selectivity. ICES J. Mar. Sci. **68**(3): 611–625.

Maunder, M.N. 2001. Integrated tagging and catch-at-age analysis (ITCAAN): Model development and simulation testing. In Spatial Processes and Management of Marine Populations, edited by G.H. Kruse, N. Bez, A. Booth, M.W. Dorn, S. Hills, R.N. Lipcius, D. Pelletier, C. Roy, S.J. Smith, and D. Witherell. University of Alaska Sea Grant, Fairbanks, Alaska, USA, number AK-SG-01-02 in Alaska Sea Grant College Program Report, pp. 123–142.

Maunder, M.N. and Deriso, R.B. 2003. Estimation of recruitment in catch-at-age model. Can. J. Fish. Aquat. Res. **60**(10): 1204–1216. Doi: 10.1139/F03-104.

Maunder, M.N. and Punt, A.E. 2013. A review of integrated analysis in fisheries stock assessment. Fish. Res. **142**(1): 61–74. Doi: 10.1016/j.fishres.2012.07.025.

Maunder, M.N. 1998. Integration of tagging and population dynamics models in fisheries stock assessment. Doctor of philosophy dissertation, University of Washington.

Mayakoshi, Y. and Kitada, S. 2016. Overdispersion in the estiamtion of salmon escapement by mark-recapture surveys and its related factors. Trans. Am. Fish. Soc **145**(2): 269–276. Doi: 10.1080/00028487.2015.1121922.

McGarvey, R., Linnane, A.J., Feenstra, J.E., Punt, A.E., and Matthews, J.M. 2010. Integrated recapture-conditioned movement estimation into spatial stock assessment: A south Australian lobster fishery application. Fish. Res. **105**(2): 80–90.

Molton, K.J., Brenden, T.O., and Bence, J.R. 2013. Harvest levels that conserve spawning biomass can provide larger and more stable and sustainable yields in intermixed fisheries. Fish. Res. **147**: 264–283. Doi: 10.1016/j.fishres.2013.07.004.

Morishima, G.S. and Henry, K.A. 1999. The history and status of Pacific Northwest Chinook and Coho Salmon ocean fisheries and prospects for sustainability. In Sustianable Fisheries Management, edited by E.E. Knudsen, C.R. Steward, D.D. MacDonald, J.E. Williams, and D.W. Reiser, CRC Press. Doi: 10.1201/9781439822678.ch16.

Polacheck, T., Eveson, J.P., Laslett, G.M., Pollock, K.H., and Hearn, W.S. 2006. Integrating catch-at-age and multiyear tagging data: a combined brownie and petersen estimation approach in a fishery context. Can. J. Fish. Aquat. Sci. **63**(3): 534–548.

Punt, A.E., Pribac, F., Walker, T.I., Taylor, B.L., and Prince, J.D. 2000. Stock assessment of school shark, *Galeorhinus galeus*, based on a spatially explicit population dynamics model. Mar. Freshwater Res. **51**(3): 205–220. Doi: 10.1071/MF99124.

R Development Core Team 2016. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Thorson, J.T., Jensen, O.P., and Zipkin, E.F. 2014. How variable is recruitment for exploited marine fishes? A hierarchical model for testing life history theory. Can. J. Fish. Aquat. Sci. **71**(7): 973–983. Doi: 10.1139/cjfas-2013-0645.

Thorson, J.T., Jensen, O.P., and Zipkin, E.F. 2016. Corrigendum: How variable is recruitment for exploited marine fishes? A hierarchical model for testing life history theory. Can. J. Fish. Aquat. Sci. Doi: 10.1139/cjfas-2013-0645.

Tsehaye, I., Brenden, T.O., Bence, J.R., Lui, W., Scribner, K.T., Kanefsky, J., Bott, K., and Elliott, R.F. 2016. Combining genetics with age/length data to estimate temporal changes in year-class strength of source populations contributing to mixtures. Fish. Res. **173**(3): 236–249. Doi: 10.1016/j.fishres.2015.09.004.

Vandergoot, C.S. and Brenden, T.O. 2014. Spatially varying population demographics and fishery characteristics of Lake Erie walleyes inferred from a long-term tag recovery study. Trans. Am. Fish. Soc **143**(1): 188–204. Doi: 10.1080/00028487.2013.837095.

Wang, Y. and Liu, Q. 2006. Estimation of natural mortality using statistical analysis of fisheries catch-at-age data. Fish. Res. **78**(2–3): 342–351. Doi: 10.1016/j.fishres.2005.11.020.

Wilberg, M.J. and Bence, J.R. 2006. Performance of time-varying catchability estimators in statistical catch-at-age analysis. Can. J. Fish. Aquat. Sci. **63**(10): 2275–2285. Doi: 10.1139/f06-111.

Wilberg, M.J. and Bence, J.R. 2008. Performance of deviance information criterion model selection in statistical catch-at-age analysis. Fish. Res **93**(1–2): 212–221. Doi: 10.1016/j.fishres.2008.04.010.

Wills, T., Robinson, J., Kayle, K., Faust, M., Gorman, A.M., Belore, M., Cook, A., Drouin, R., MacDougall, T., Zhao, Y., Murray, C., and Hosack, M. 2015. Report for 2014 by the lake erie walleye task group. Technical report, Lake Erie Committee Great Lakes Fishery Commission.

Ying, Y., Chen, Y., Lin, L., and Gao, T. 2011. Risks of ignoring fish population spatial structure in fisheries management. Can. J. Fish. Aquat. Sci. **68**(12): 2101–2120. Doi: 10.1139/f2011-116.

Appendix A: Parameter List and Model Equations

Table A1: Symbols and descriptions of variables used in Operating and ITCAAN models.

Symbol	Description	Application
y	Subscript for year	Both
a	Subscript for age	Both
A	Subscript for oldest modeled age	Both
r	Subscript for region	Both
s	Subscript for spawning stock	Both
l	Subscript for release year	Both
F	Age-specific fishing mortality	Both
f	Apical fishing mortality	Both

v	Age-specific fishery selectivity	Both
V	Age-specific survey selectivity	Both
ρ	Autocorrelation coefficient in autoregressive process	Generation
μ	Mean fishing mortality in autoregressive process	Generation
σ	Variance of fishing mortality in autoregressive process	Generation
M	Natural mortality	Both
Z	Total mortality	Both
ϵ	Autocorrelated recruitment variation	Operating
δ	Random annual deviation in recruitment	Both
σ^2	Recruitment variance	Operating
α	Parameter of Ricker stock-recruit function	Operating
β	Parameter of Ricker stock-recruit function	Operating
w	Weight at age	Operating
m	Maturity at age	Operating
N	Population abundance	Both
$T_{s,r}$	Movement rate from stock s to region r	Both
S	Survival	Both
I	Survey abundance index	Both
q	Catchability coefficient	Both
C	Harvest	Both
ζ	Observation error in harvest	Operating
τ	Observation error in survey index	Operating
E	Observed fishing effort	Both
γ	Observation error in effort	Operating
σ_C^2	Variance of catch data observation error	Operating
σ_I^2	Variance of index data observation error	Operating
σ_E^2	Variance of effort data observation error	Operating
n	Number of tagged fish alive	Both
R	Number of tagged fish released	Both
ϕ^F	Expected proportion of tagged cohort to be harvested	Operating
ϕ^S	Expected proportion of tagged chort to survive	Operating
ϕ^M	Expected proportion of tagged cohort to die from natural mortality	Operating

Φ	Vector of expected proportions of the fates of tagged cohorts	Operating
t	Number of tags captured by fishery F, die naturally M, or survive S	Operating
r	Number of tags recovered	Both
$\omega_{s,r}$	Multinomial logit parameter for estimating movement from stock s to region r	ITCAAN
Γ	Mean abundance for first model year	ITCAAN
Δ	Age-specific abundance deviations in first year	ITCAAN
Λ	Recruitment in the first year	ITCAAN
λ	Annual recruitment deviation	ITCAAN
K	Catchability parameter in the first year	ITCAAN
κ	Annual catchability deviation	ITCAAN
P	Harvest age composition	ITCAAN
η	Survey age composition	ITCAAN
θ	Tag recovery proportions	ITCAAN
Υ	Angler reporting rate of tags	ITCAAN
ψ_C	Standard deviation for harvest data component	ITCAAN
ψ_I	Standard deviation for survey data component	ITCAAN
ψ_E	Standard deviation for fishery catchability random walk	ITCAAN
ψ_R	Standard deviation for recruitment deviations random walk	ITCAAN
ψ_N	Standard deviation for abundance deviations in first model year	ITCAAN
ESS_C	Effective sample size for harvest age composition	ITCAAN
ESS_S	Effective sample size for survey age composition	ITCAAN

Operating Model

Underlying equations for the data-generating model.

Ricker stock-recruit function with autocorrelated error for each spawning stock:

$$\text{(A.1)} \quad N_{y+2,a=1,s} = \alpha_s \sum_a (m_a w_a N_{y,a,s}) e^{-\beta_s \sum_a (m_a w_a N_{y,a,s}) \epsilon_y}$$

Total insantaneous mortality by year, age, and region:

$$Z_{y,a,r} = M + F_{y,a,r}$$

First-order autoregressive component for Ricker stock-recruit function:

$$\epsilon_y = \begin{cases} \rho_s \epsilon_y + \sqrt{1-\rho^2{}_s}\delta_{y-1} & \text{for} \quad y > 1 \\ \delta_y & \text{for} \quad y = 1 \end{cases} \tag{A.2}$$

$$\text{where} \qquad \delta_y \sim N\left(\frac{-\sigma_s^2(1-\rho_s)}{2\sqrt{1-\rho_s^2}}, \sigma_s^2\right)$$

Annual survivial rate by year,age, and region:

$S_{y,a,r} = e^{-Z_{y,a,r}}$

Annual change in abundance at age accounting for Box-Transfer movement (not including last age group):

$$N_{y+1,a+1,s} = \sum_r N_{y,a,s} T_{s,r} S_{y,a,r} \qquad \text{where} \qquad a < A-1 \tag{A.3}$$

Annual change in abundance for last age group accounting for Box-Transfer movement:

$$N_{y+1,A,s} = \sum r N_{y,A,s} T_{s,r} S_{y,A,r} + N_{y,A-1,s} T_{s,r} S_{y,A-1,r} \tag{A.4}$$

Apical instantaneous fishing mortality by region and year:

$$\begin{aligned} f_{1,r} &\sim \text{Trunc. Normal}\Big(\mu_r, \sigma_r, 0, \infty\Big) \\ f_{y+1,r} &= \Big(\mu_r(1-\rho_r)\Big) + \rho_r f_{y,r} + \delta_y \end{aligned} \tag{A.5}$$

$$\text{where} \qquad \delta_y \sim \text{Trunc. Normal}\Big(0, \sigma_r, -(\rho_r f_{y,r} + \mu_r(1-\rho_r)\Big), \infty)$$

Instantaneous fishing mortality by year, age, and region:

$$F_{y,a,r} = f_{y,r} v_{a,r} \tag{A.6}$$

Observed fishery effort by year and region accounting for observation error:

$$E_{y,r} = f_{y,r}/q_r \gamma_{y,r} \qquad \text{where} \qquad \gamma_{y,r} \sim LN\Big(-\sigma_E^2/2, \sigma_E^2\Big) \tag{A.7}$$

Survey index of abundance by year, age, and region:

$$I_{y,a,r} = \sum_s N_{y,a,s} T_{s,r} e^{-Z_{y,a,r}*10/12} q_r V_{a,r} \tag{A.8}$$

Number of tagged fish alive by spawning stock and age from a tagging cohort at year of release:

(A.9) $n_{l,s,y,a} = R_{l,s,a} \quad \text{for} \quad y = l$

Annual allocation of tagged fish by spawning stock and age to regions:

(A.10) $t_{l,s,y,a,r} \sim MN\Big(n_{l,s,y,a}, T_{s,r}\Big)$

Probability of tagged fish being harvested by year, age, and region:

(A.11) $\phi^{F}_{y,a,r} = \dfrac{F_{y,a,r}}{Z_{y,a,r}}\Big(1 - S_{y,a,r}\Big)$

Probability of tagged fish surviving by year, age, and region:

(A.12) $\phi^{S}_{y,a,r} = S_{y,a,r}$

Probability of tagged fish dying naturally by year, age, and region:

(A.13) $\phi^{M}_{y,a,r} = \dfrac{M}{Z_{y,a,r}}(1 - S_{y,a,r})$

Vectorizing probabilities of harvest, surviving, and dying naturally by year, age, and region:

(A.14) $\Phi_{y,a,r} = \Big(\phi^{F}_{y,a,r}, \phi^{S}_{y,a,r}, \phi^{M}_{y,a,r}\Big)$

Generation of actual number of tagged fish from a tagged cohort that are harvested, survive, and die naturally by year, age, and region:

(A.15) $t^{F,S,M}_{l,s,y,a,r} \sim MN\Big(t_{l,s,y,a,r}, \Phi_{y,a,r}\Big)$

Number of tagged fish from a tagged cohort that are recovered and reported:

(A.16) $r_{l,s,y,a,r} \sim BIN\Big(t_{l,s,y,a,r}, \Upsilon\Big)$

Annual change in number of tagged fish from a tagged cohort that are alive by year, age, and region:

$$(A.17)\quad n_{l,s,y+1,a+1} = \sum_r t^S_{l,s,y,a,r}$$

Number of tagged fish from a tagged cohort that are never recovered and reported:

$$(A.18)\quad r^{NR}_{l,s} = \sum_a R_{l,s,a} - \sum_y \sum_a \sum_r r_{l,s,y,a,r}$$

Harvest by year, age, and region:

$C_{y,a,r} = \sum_s \frac{F_{y,a,r}}{Z_{y,a,r}} N_{y,a,s} T_{s,r}(1 - S_{y,a,r})$

Observed total harvest by year and region accounting for observation error:

$$(A.19)\quad C_{y,r} = \sum_a C_{y,a,r}\zeta_{y,r} \qquad \text{where} \qquad \zeta_{y,r} \sim LN(-\sigma^2_C/2, \sigma^2_C)$$

Observed total survey index of abundance by year and region accounting for observation error:

$$(A.20)\quad I_{y,r} = \sum_a I_{y,a,r}\tau_{y,r} \qquad \text{where} \qquad \tau_{y,r} \sim LN\Big(-\sigma^2_I/2, \sigma^2_I\Big)$$

722 ## ITCAAN Model

Predicted annual change in abundance at age by year and region (not including last age group):

$$(A.21)\quad \widehat{N}_{y+1,a+1,s} = \sum_r \widehat{N}_{y,a,s}\widehat{T}_{s,r}\widehat{S}_{y,a,r}$$

Predicted annual change in abundance for last age group :

$$(A.22)\quad \widehat{N}_{y+1,A,s} = \sum r\widehat{N}_{y,A,s}\widehat{T}_{s,r}\widehat{S}_{y,A,r} + \widehat{N}_{y,A-1,s}\widehat{T}_{s,r}\widehat{S}_{y,A-1,r}$$

Predicted abundance at age of a tagged cohort by year and spawning stock:

$$(A.23)\quad \begin{aligned} \hat{n}_{l,s,y,a} &= R_{l,s,a} && \text{for} \quad y = l \\ \hat{n}_{l,s,y+1,a+1} &= \sum_r \hat{n}_{l,s,y,a}\widehat{T}_{s,r}\widehat{S}_{y,a,r} && \text{for} \quad a < A-1 \\ \hat{n}_{l,s,y+1,A} &= \sum_r \hat{n}_{l,s,y,A}\widehat{T}_{s,r}\widehat{S}_{y,A,r} + \hat{n}_{l,s,y,A-1}\widehat{T}_{s,r}\widehat{S}_{y,A-1,r} && \text{for} \quad a = A \end{aligned}$$

Predicted index of abundance by year, age, and region:

$\hat{I}_{y,a,r} = \sum_s \widehat{N}_{y,a,s}\widehat{T}_{s,r}e^{-10/12*\hat{Z}_{y,a,r}}\hat{q}_r\hat{V}_{a,r}$

Predicted index of abundance by year and region summed over ages:

$\hat{I}_{y,a} = \sum_r \hat{I}_{y,a,r}$

Predicted survey age composition by year and region:

$\hat{\eta}_{y,g,a} = \frac{\hat{I}_{y,a,r}}{\hat{I}_{y,r}}$

Predicted harvest by year, age, and region:

$\hat{C}_{y,a,r} = \sum_s \frac{\hat{F}_{y,a,r}}{\hat{Z}_{y,a,r}}\Big(1-\widehat{S}_{y,a,r}\Big)\widehat{N}_{y,a,s}\widehat{T}_{s,r}$

Predicted harvest by year and region summed over ages:

$\hat{C}_{y,r} = \sum_a \hat{C}_{y,a,r}$

Predicted harvest age composition by year and region:

$\widehat{P}_{y,a,r} = \frac{\hat{C}_{y,a,r}}{\hat{C}_{y,r}}$

Predicted recruitment by year and spawning stock

$$\begin{aligned} \widehat{N}_{y=1,a=1,s} &= \widehat{\Lambda} \qquad \text{for} \qquad y=1 \\ \widehat{N}_{y,a=1,s} &= \widehat{N}_{y-1,a=1,s}e^{\widehat{\lambda}_{y,s}} \qquad \text{for} \qquad 1>y\leq Y-2 \\ \widehat{N}_{y,a=1,s} &= (\widehat{N}_{Y-2,a=1,s}+\widehat{N}_{Y-3,a=1,s}+\widehat{N}_{Y-4,a=1,s})/3 \qquad \text{for} \qquad y=Y-1 \text{ or } Y \end{aligned} \tag{A.24}$$

Negative log penalty for recruitment random walk deviations:

$$-ln(L_{\mathrm{Rec}}) = \sum_y\sum_s ln(\psi_R\sqrt{2\pi}) + 0.5\Big(\frac{\Big(-\widehat{\lambda}_{y,s}\Big)}{\psi_R}\Big)^2 \tag{A.25}$$

Predicted abundance at age in first year by spawning stock for ages 3 and older:

$$\widehat{N}_{y=1,a,s} = exp(\hat{\Gamma}_s+\widehat{\Delta}_{s,a}) \qquad \text{for} \qquad a>2 \qquad \text{where} \sum_a \widehat{\Delta}_{s,a} = 0.0 \tag{A.26}$$

Negative log penalty for abundances at age for initial year:

$$-ln(L_{\mathrm{N0}}) = \sum_a\sum_s ln(\psi_N\sqrt{2\pi}) + 0.5\Big(\frac{\Big(-\widehat{\Delta}_{s,a}\Big)}{\psi_N}\Big)^2 \tag{A.27}$$

Estimated instantaneous fishing mortality by year, age, and region :

$$\hat{F}_{y,a,r} = \hat{q}_{y,r}E_{y,r}\hat{v}_{a,r} \tag{A.28}$$

Estimated instantaneous total mortality by year, age, and region:

$\hat{Z}_{y,a,r} = \hat{M} + \hat{F}_{y,a,r}$

Estimated annual survival rate :

$\widehat{S}_{y,a,r} = e^{-\hat{Z}_{y,a,r}}$ Predicted fishery catchability coefficient by year and spawning stock

$$\text{(A.29)} \quad \begin{aligned} \hat{q}_{y=1,r} &= K \qquad \text{for} \qquad y = 1 \\ \hat{q}_{y,r} &= \hat{q}_{y-1,r} e^{\hat{\kappa}_{y,r}} \qquad \text{for} \qquad 1 > y \leq Y - 2 \end{aligned}$$

Negative log penalty for fishery catchability coefficient random walk deviations:

$$\text{(A.30)} \quad -ln(L_{\text{q}}) = \sum_{y} \sum_{r} ln(\psi_E \sqrt{2\pi}) + 0.5 \Big(\frac{\Big(- \hat{\kappa}_{y,s} \Big)}{\psi_E} \Big)^2$$

Multinomial logit parameterization of movement

$$\text{(A.31)} \quad \widehat{T}_{r,s} = \frac{e^{\omega_{r,s}}}{\sum_{r=1}^{R-1} e^{\omega_{r,s}}}, \qquad \text{where} \quad \omega_{R,s} = 0.$$

Negative log likelihood for total harvest:

$$\text{(A.32)} \quad -ln(L_{\text{Harvest}}) = \sum_{y} \sum_{g} ln(\psi_C \sqrt{2\pi}) + 0.5 \left(\frac{\Big(ln(C_{y,g}) - ln(\hat{C}_{y,g}) \Big)}{\psi_C} \right)^2$$

Negative log likelihood for index survey of abundance:

$$\text{(A.33)} \quad -ln(L_{\text{Survey}}) = \sum_{y} \sum_{r} ln(\psi_I \sqrt{2\pi}) + 0.5 \left(\frac{\Big(ln(I_{y,r}) - ln(\hat{I}_{y,r}) \Big)}{\psi_I} \right)^2$$

Negative log penality for catchability random walk deviations:

$$\text{(A.34)} \quad -ln(L_{\text{Effort}}) = \sum_{y} \sum_{g} ln(\psi_E \sqrt{2\pi}) + 0.5 \left(\frac{\Big(ln(\text{mean}\hat{q}_{y,g}) - ln(\hat{q}_{y,g}) \Big)}{\psi_E} \right)^2$$

Calculated variances for log likelihoods and log penalties:

$$\begin{aligned} \psi_E &= \psi_C \\ \psi_I &= \sqrt{2 * \psi_C^2} \\ \psi_R &= 4 \\ \psi_N &= 4 \end{aligned} \tag{A.35}$$

Negative log likelihood for harvest age composition:

$$-ln(L_{\text{CP}}) = -ESS_C \sum_y \sum_g \sum_a P_{y,g,a} ln\left(\widehat{P}_{y,g,a}\right) \tag{A.36}$$

Negative log likelihood for survey age composition:

$$-ln(L_{\text{SP}}) = -ESS_S \sum_y \sum_a \sum_r \eta_{y,a,r} ln\left(\hat{\eta}_{y,a,r}\right) \tag{A.37}$$

Predicted reported recoveries of a tagged cohort by year, age, and region:

$$r_{l,s,y,r} = \sum_a \hat{n}_{l,s,y,a} \widehat{T}_{s,r} \frac{\hat{F}_{y,a,r}}{\hat{Z}_{y,a,r}} \left(1 - \widehat{S}_{y,a,r}\right) \widehat{\Upsilon_r} \tag{A.38}$$

Negative log likelihood for tagging cohorts:

$$-ln(L_{\text{TP}}) = -\sum_l \sum_s \left[\sum_y \left(\sum_g \sum_a \left(R_a^{l,s}\right) ln\left(\widehat{\theta}_{y,g}^{l,s}\right)\right) + \sum_a \left(R_a^{l,s}\right) ln\left(\widehat{\theta}_{NR}^{l,s}\right)\right] \tag{A.39}$$

Tagging age proportions:

$$\begin{aligned} \widehat{\theta}_{l,s,y,r} &= \frac{r_{l,s,y,r}}{\sum_a R_{l,s,a}} \\ \widehat{\theta}_{l,s}^{NR} &= \frac{r_{l,s}^{NR}}{\sum_a R_{l,s,a}} \end{aligned} \tag{A.40}$$

Table 1: List of simulation scenarios conducted exploring the performance of ITCAAN models. Abbreviation indicates the name of the scenario used in figures and manuscript text. (RC= regionally constant; RU=regionally unique; Known=Fixed at an assumed value; Estimated=Estimated in the ITCAAN model; EP=equal productivity across requions.

		ITCAAN model		Operating model	
Scenario group	Abbreviation	Reporting rate	Natural mortality	Emigration rate	# of tags released yearly
1	K1	Known	Known	1 %	2000 each region
1	M1	Known	Estimated RC	1 %	2000 each region
1	RR1	Estimated RU	Known	1 %	2000 each region
1	B1	Estimated RU	Estimated RC	1 %	2000 each region
1	K5	Known	Known	5 %	2000 each region
1	M5	Known	Estimated RC	5 %	2000 each region
1	RR5	Estimated RU	Known	5 %	2000 each region
1	B5	Estimated RU	Estimated RC	5 %	2000 each region
1	K10	Known	Known	10 %	2000 each region
1	M10	Known	Estimated RC	10 %	2000 each region
1	RR10	Estimated RU	Known	10 %	2000 each region
1	B10	Estimated RU	Estimated RC	10 %	2000 each region
1	K20	Known	Known	20 %	2000 each region
1	M20	Known	Estimated RC	20 %	2000 each region
1	RR20	Estimated RU	Known	20 %	2000 each region
1	B20	Estimated RU	Estimated RC	20 %	2000 each region
2	MS	Set at 1.5*True	Estimated RC	20 %	2000 each region
2	RSU	Estimated RU	Set at 0.5*True	20 %	2000 each region
2	RSO	Estimated RU	Set at 1.5*True	20 %	2000 each region
3	T1500	Estimated RU	Estimated RC	20 %	2000 each region
3	T1000	Estimated RU	Estimated RC	20 %	1000 each region
3	T500	Estimated RU	Estimated RC	20 %	500 each region
3	T250	Estimated RU	Estimated RC	20 %	250 each region
4	S1	Estimated RC	Estimated RC	20 %	2000 each region
4	S3	Estimated RC	Estimated RU	20 %	2000 each region
4	S4	Estimated RU	Estimated RU	20 %	2000 each region
5	EP	Estimated RU	Estimated RC	20 %	2000 each region

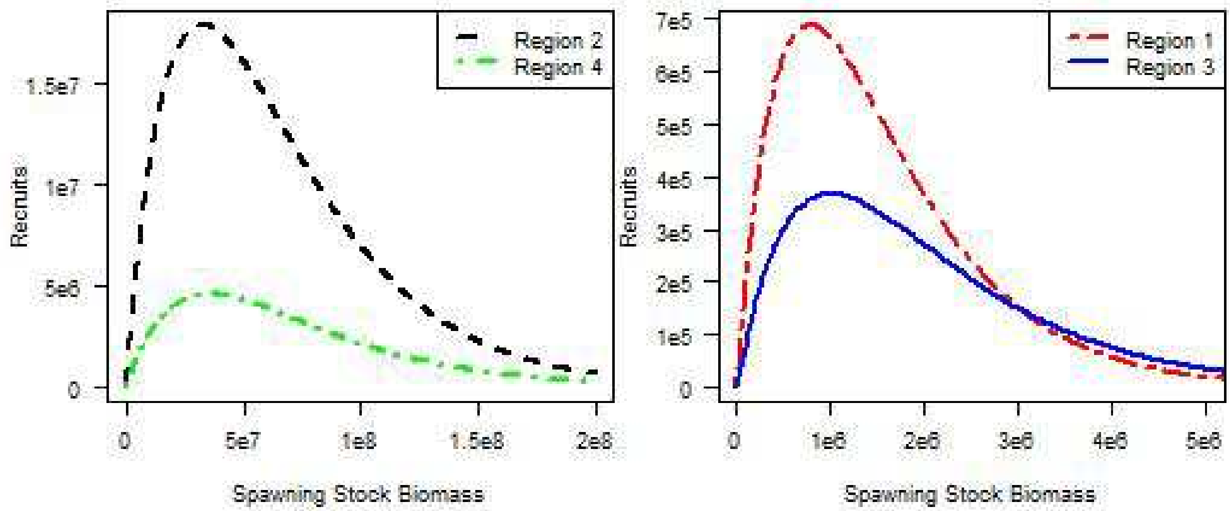

Figure 1: Ricker stock recruit relationships used to create the recruitment dynamics of the four regions in the simulation. Note the large difference in scale of axes between the two graphs.

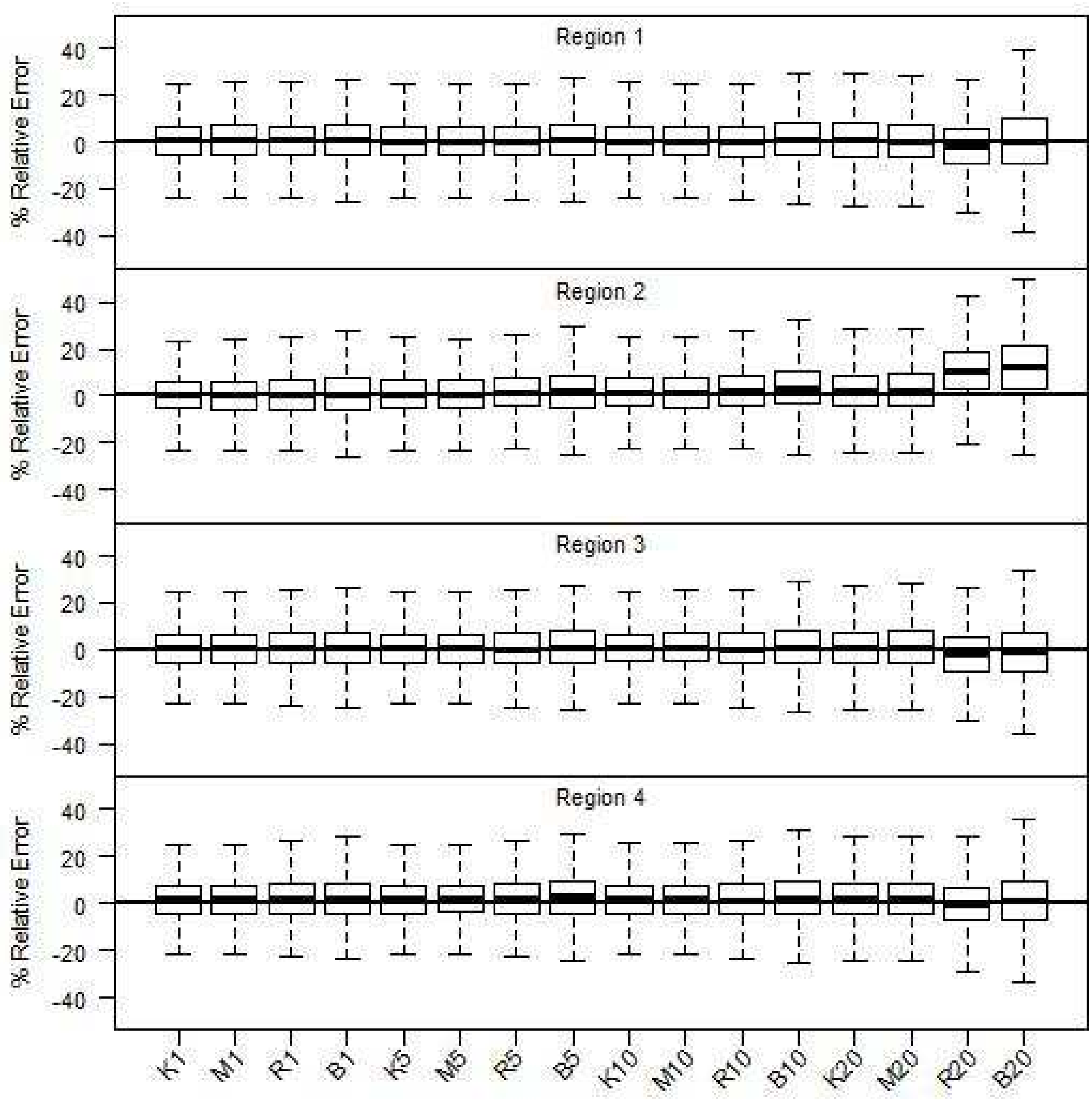

Figure 2: Relative error (%) of fishery catchabilities for each region of an ITCAAN model under different movement rates and parameter estimation assumptions (Scenario group 1) for 1000 simulation iterations. Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

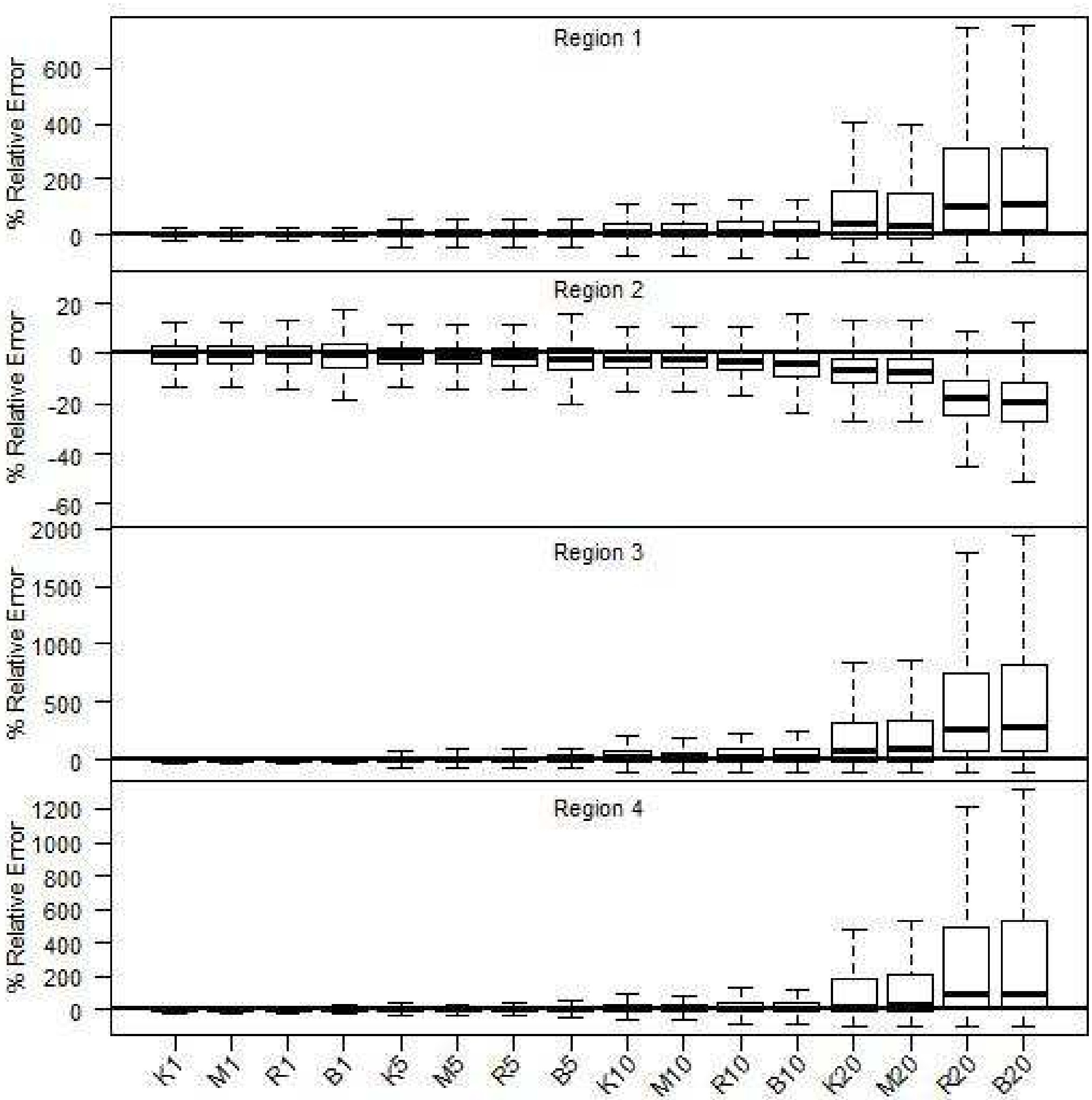

Figure 3: Relative error (%) of annual recruitment estimates for each region of an ITCAAN model under different movement rates and parameter estimation assumptions (Scenario group 1) for 1000 simulation iterations. Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded. Note the difference in y-axis scale between regions.

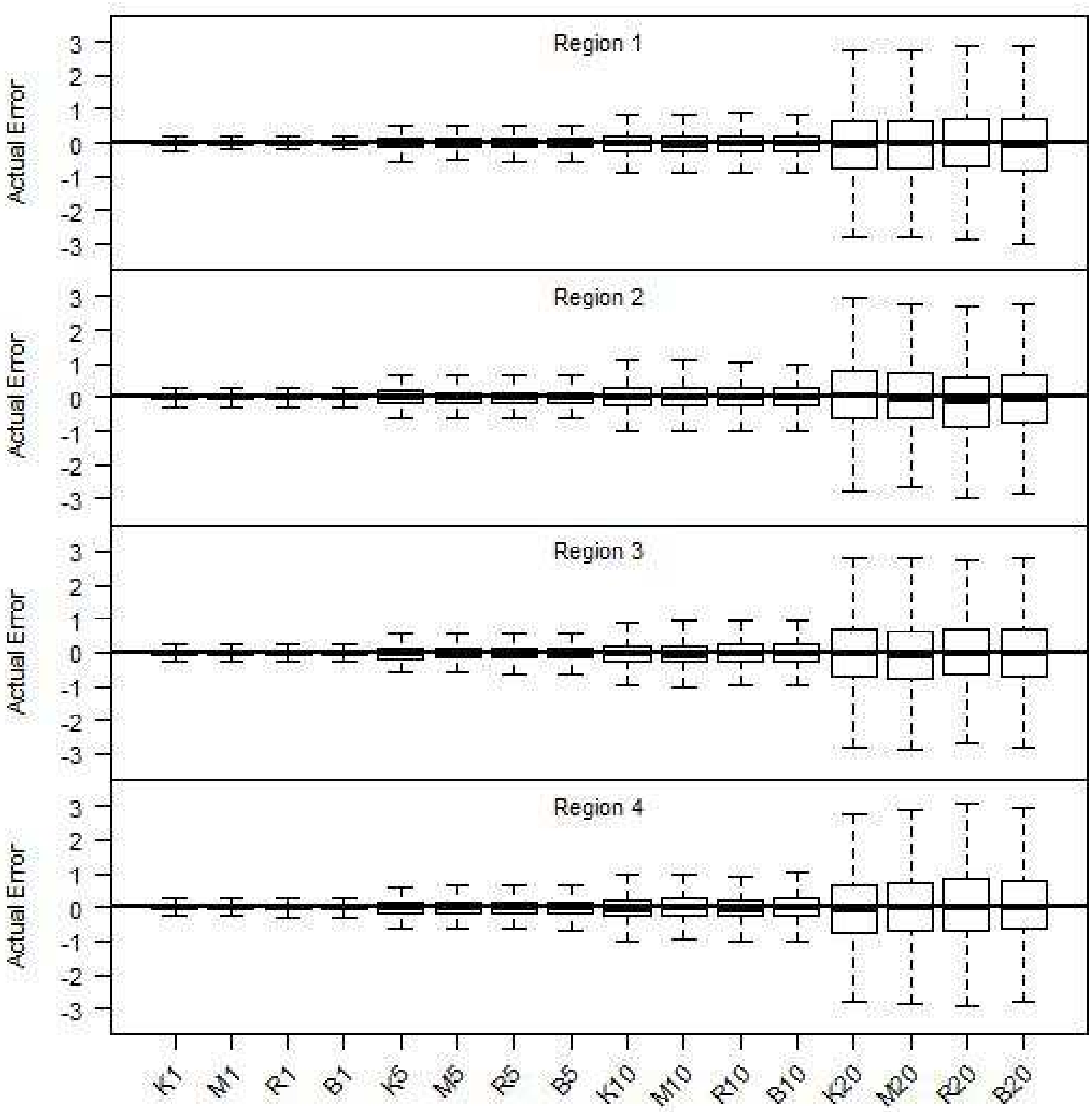

Figure 4: Actual error of movement ratess for each region of an ITCAAN model under different movement rates and parameter estimation assumptions (Scenario group 1) for 1000 simulation iterations . Table 1 lists the model abbreviations and differences in generating and estimating models. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

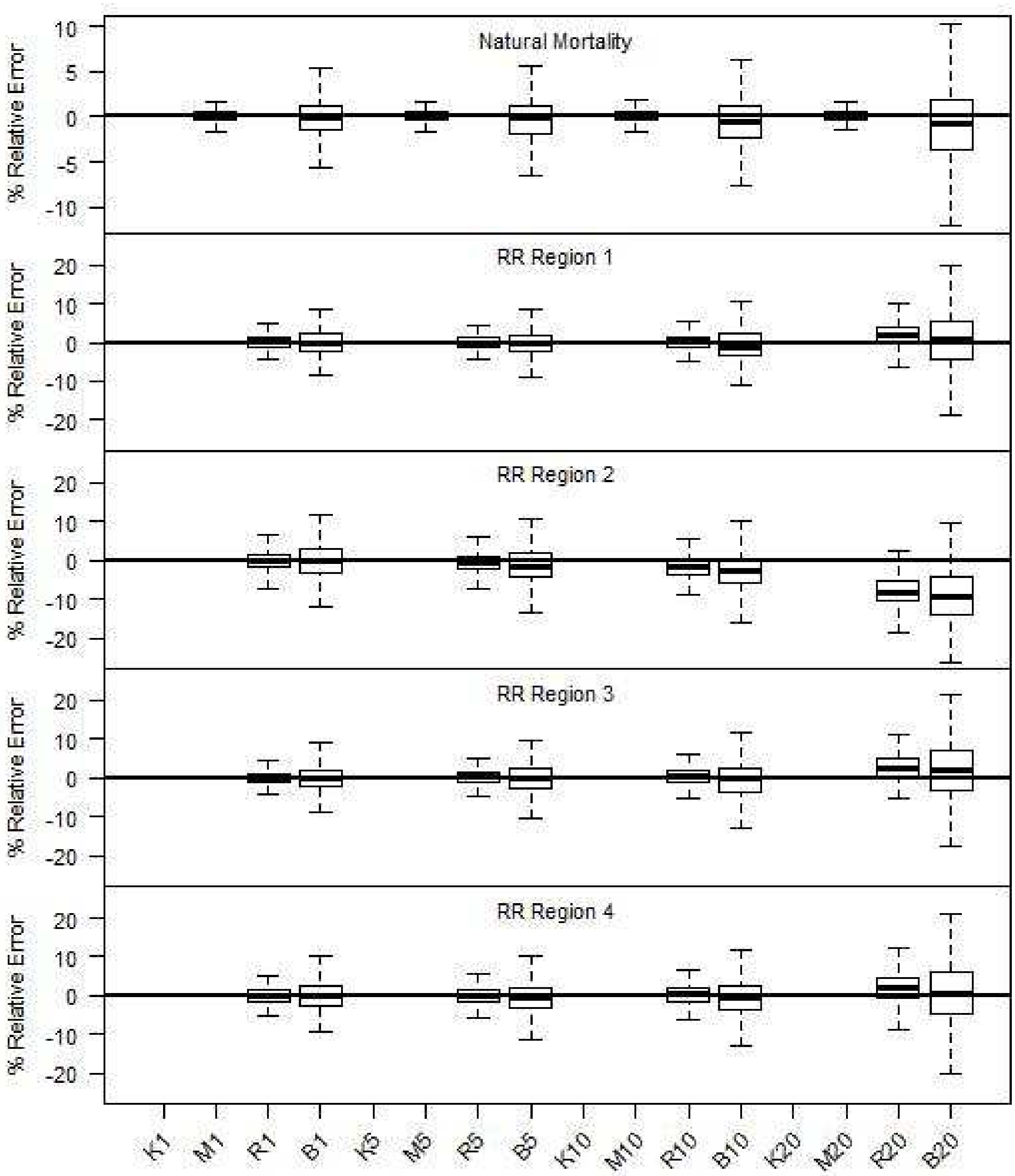

Figure 5: Relative error (%) of natural mortality and regional reporting rate estimates of an ITCAAN model under different movement rates and parameter estimation assumptions (Scenario group 1) for 1000 simulation iterations. Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

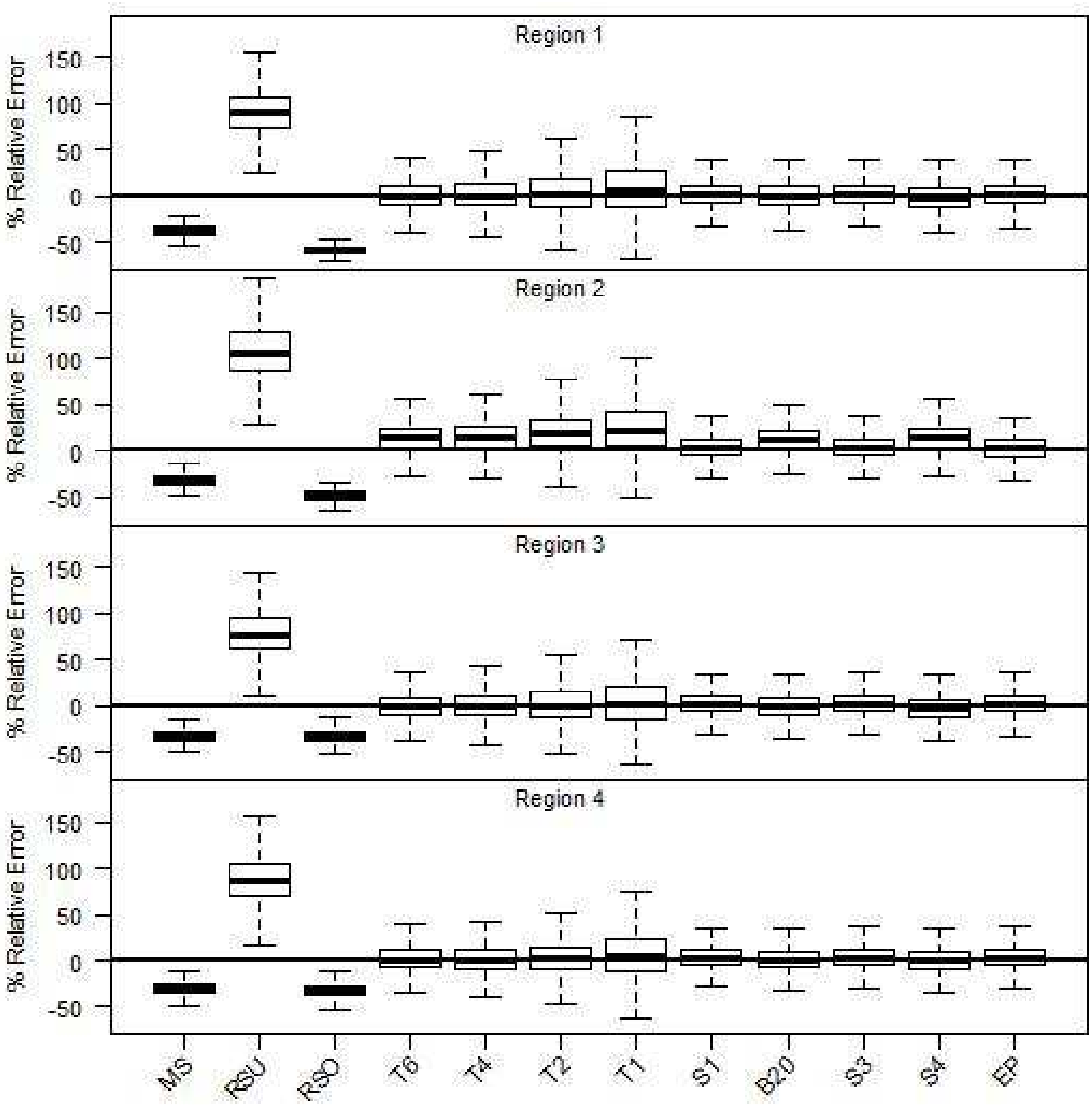

Figure 6: Relative error (%) in fishery catchabilities for each region of an ITCAAN model under misspecfied natural mortality and reporting rates, tag cohort size, spatial complexities, and equal productivities (Scenario groups 2-5). Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

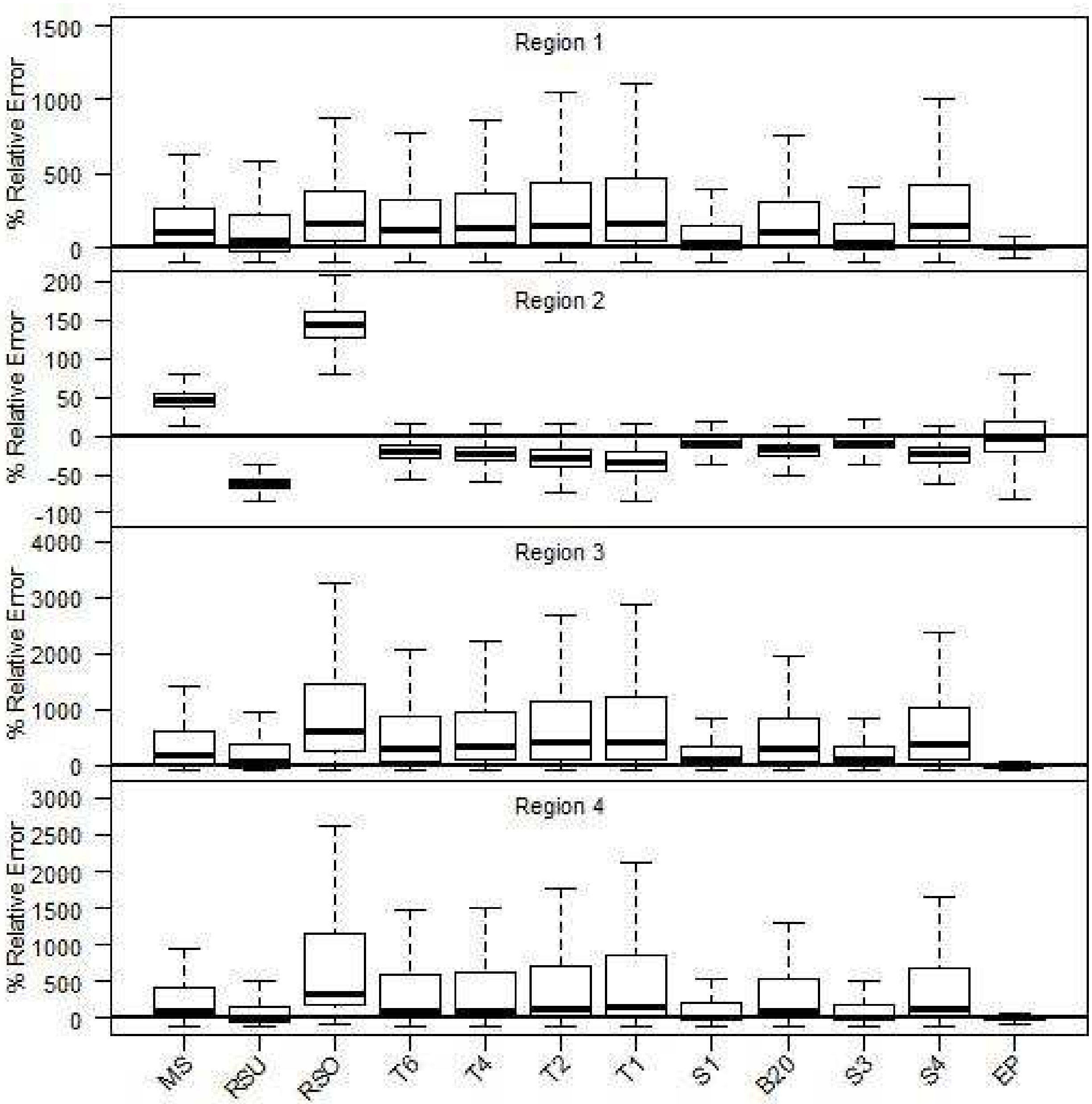

Figure 7: Relative error (%) of recruitment estimates for each region of an ITCAAN model under misspecfied natural mortality and reporting rates, tag cohort size, spatial complexities, and equal productivities (Scenario groups 2-5). Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded. Note the difference in y-axis scale between regions.

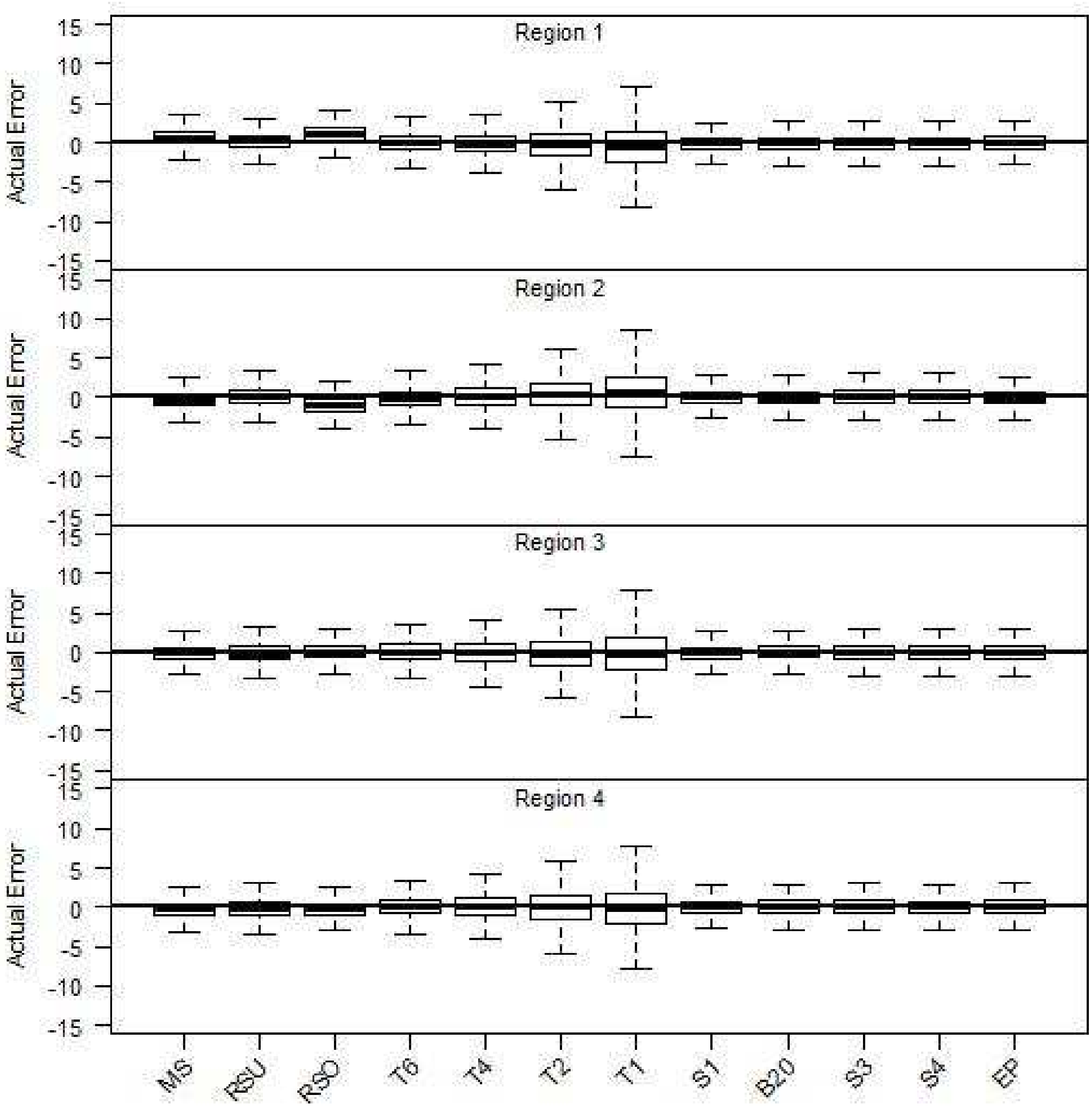

Figure 8: Actual error of movement rate estimates for each region of an ITCAAN model under misspecfied natural mortality and reporting rates, tag cohort size, spatial complexities, and equal productivities (Scenario groups 2-5). Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

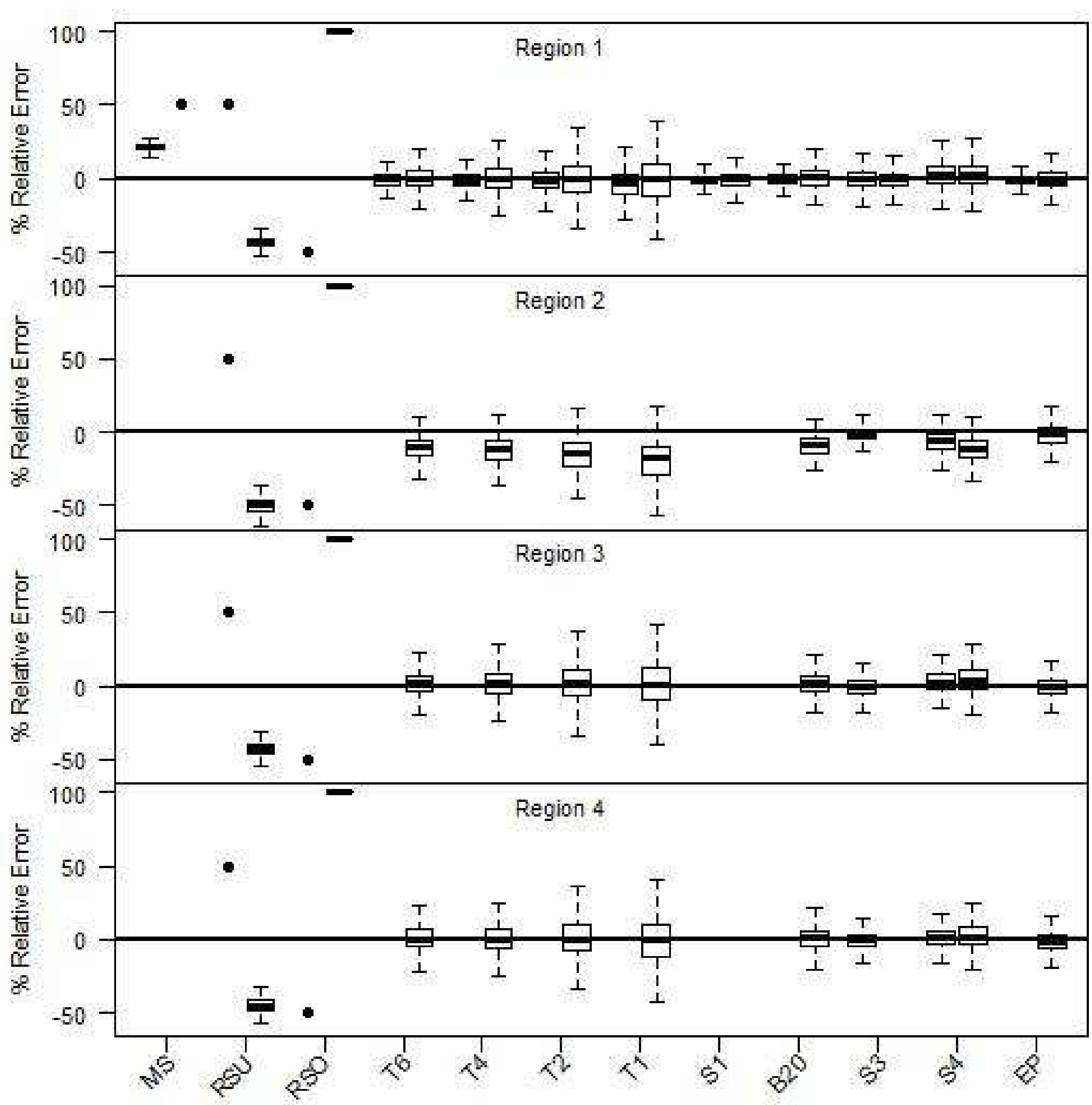

Figure 9: Relative error (%) of natural mortality (first plot in each pair) and reporting rate (second in pair) estimates for each region of an ITCAAN model under misspecfied natural mortality and reporting rates, tag cohort size, spatial complexities, and equal productivities (Scenario groups 2-5). Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded. Circles indicate a parameter specified in the assessment model and boxplots only presented in Region 1 were estimated as spatially constant.

Supplementary Material: Precision and bias of parameter estimates through simulation analysis of a multi-region tag-integrated catch-at-age assessment model

Matthew T. Vincent, Travis O. Brenden and James R. Bence

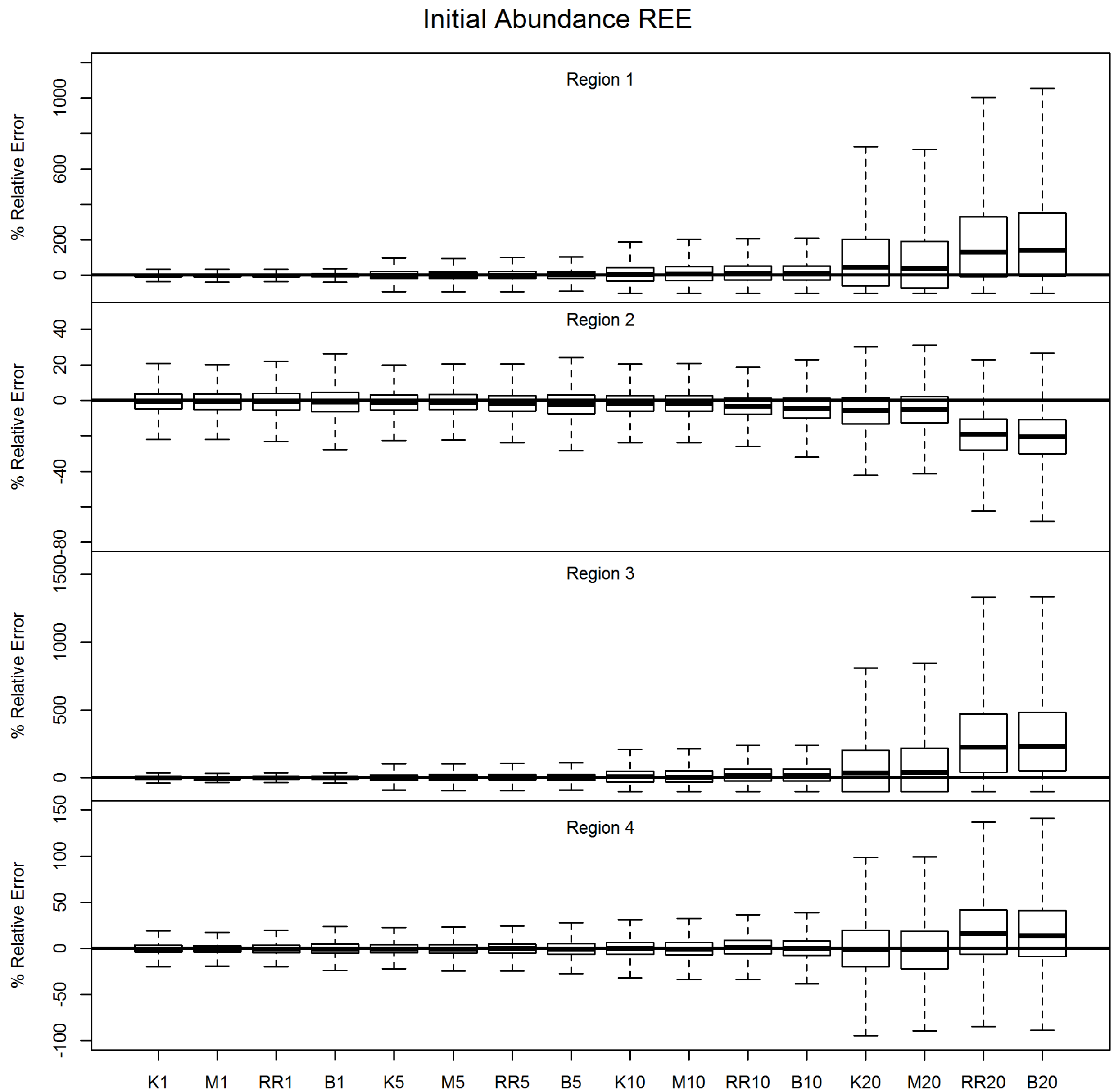

Figure 1: Relative error (%) of initial abundance at age for each region of an ITCAAN model under different movement rates and parameter estimation assumptions (Scenario group 1) for 1000 simulation iterations. Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

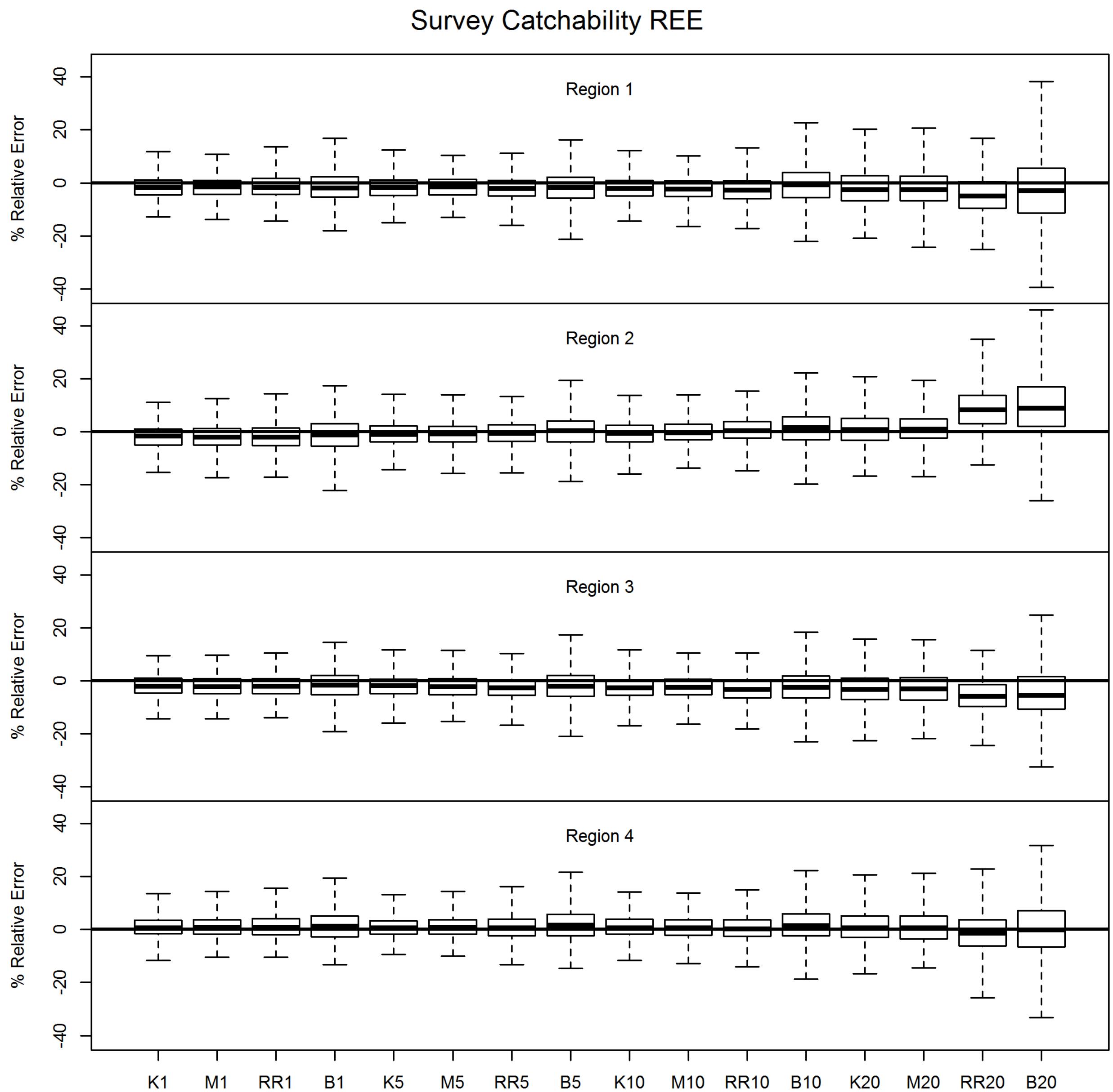

Figure 2: Relative error (%) of survey catchabilities for each region of an ITCAAN model under different movement rates and parameter estimation assumptions (Scenario group 1) for 1000 simulation iterations. Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

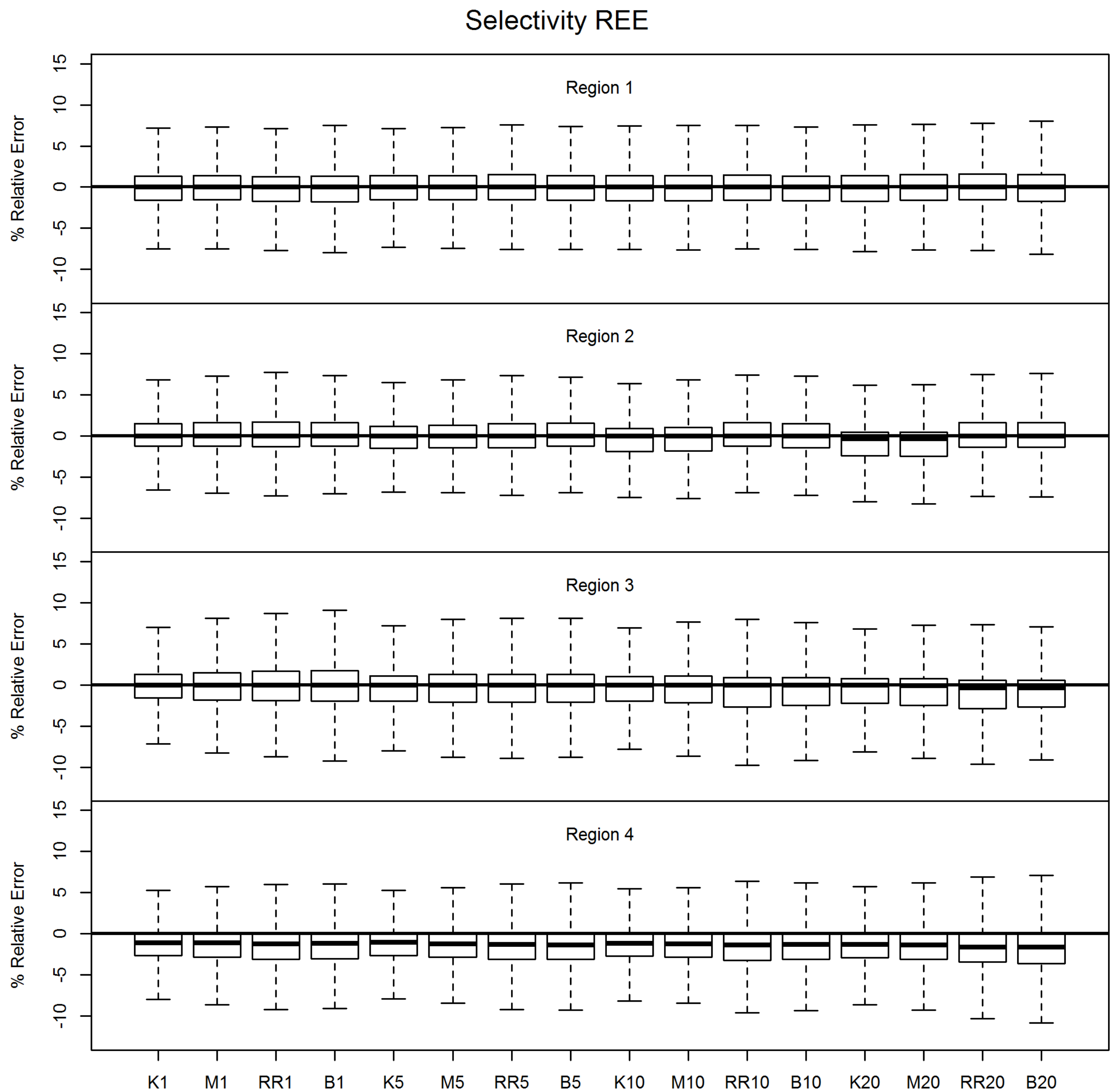

Figure 3: Relative error (%) of fishery selectivity for each region of an ITCAAN model under different movement rates and parameter estimation assumptions (Scenario group 1) for 1000 simulation iterations. Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

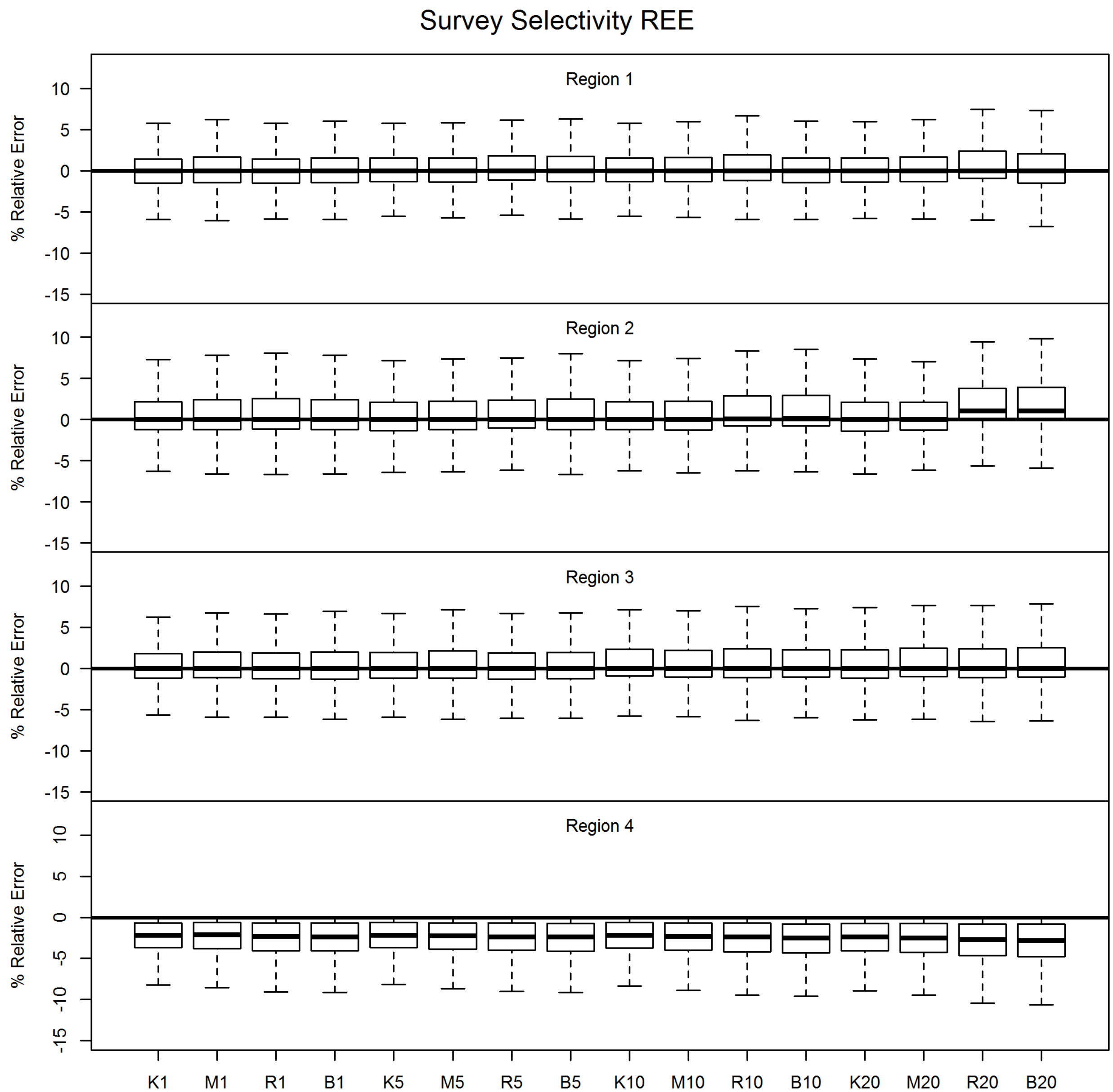

Figure 4: Relative error (%) of survey selectivity-at-age for each region of an ITCAAN model under different movement rates and parameter estimation assumptions (Scenario group 1) for 1000 simulation iterations. Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

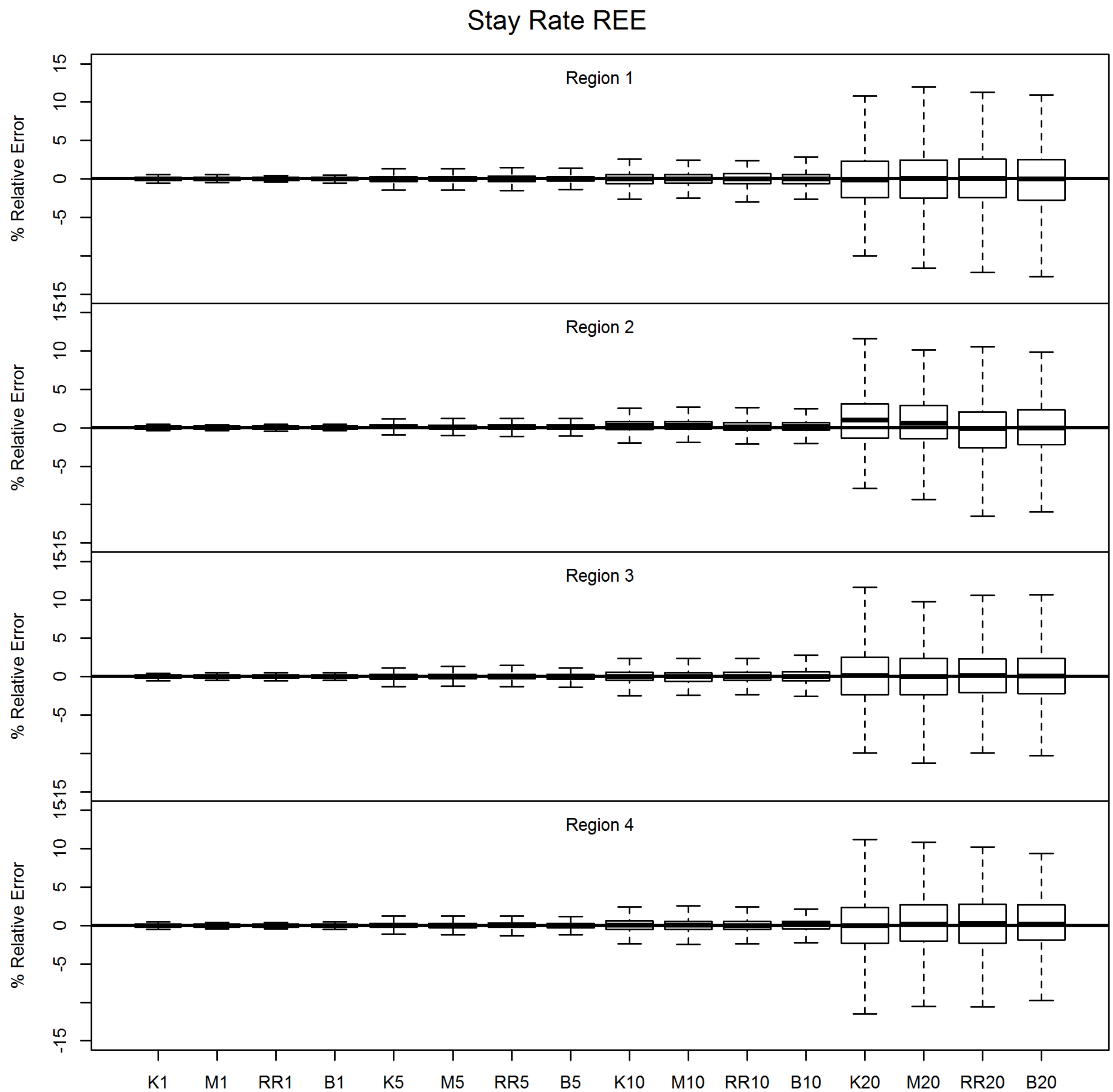

Figure 5: Relative error (%) of percent of population that remains in natal region for each region of an ITCAAN model under different movement rates and parameter estimation assumptions (Scenario group 1) for 1000 simulation iterations. Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

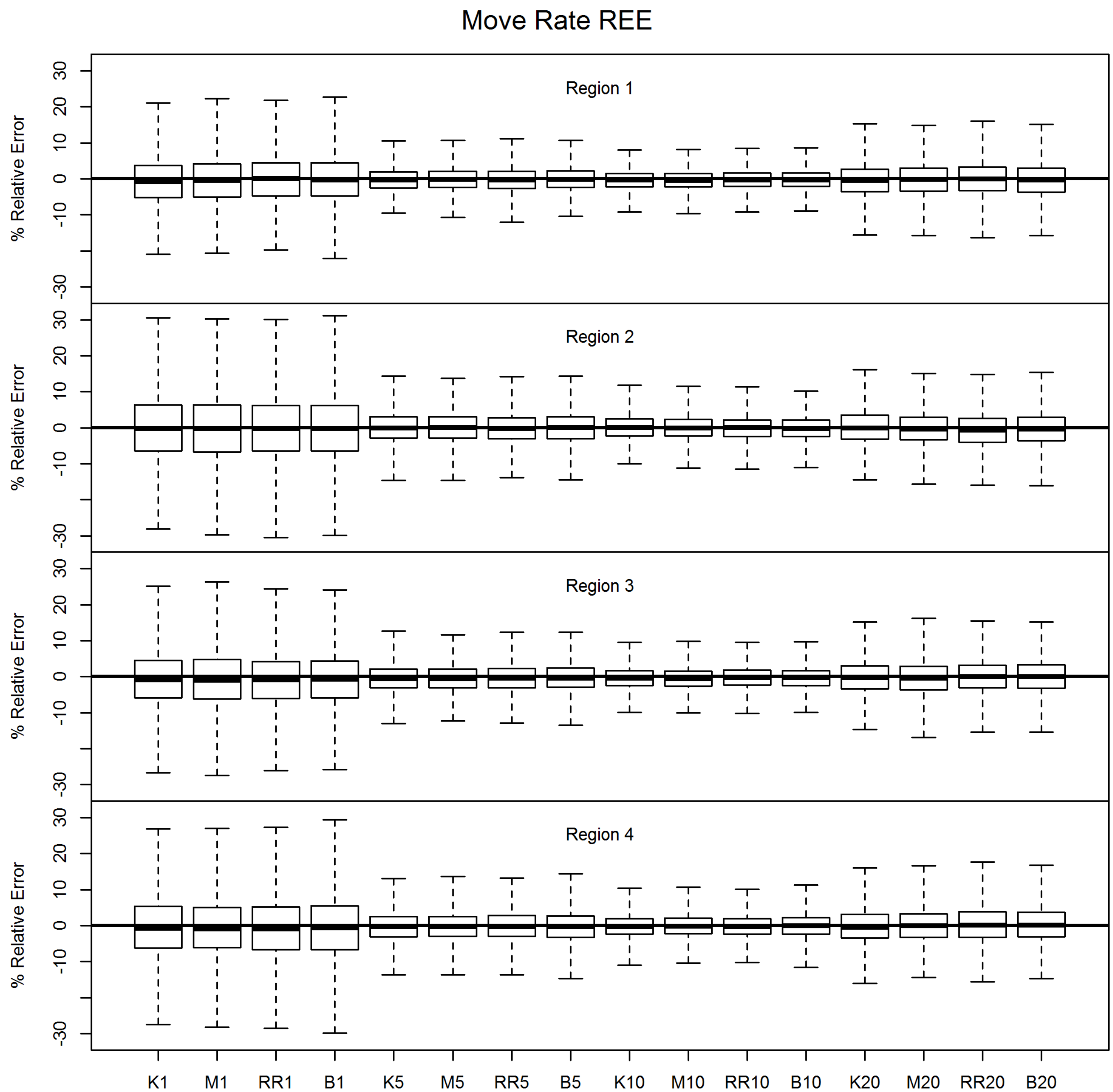

Figure 6: Relative error (%) of precent of population that move out of natal region for each region of an ITCAAN model under different movement rates and parameter estimation assumptions (Scenario group 1) for 1000 simulation iterations. Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

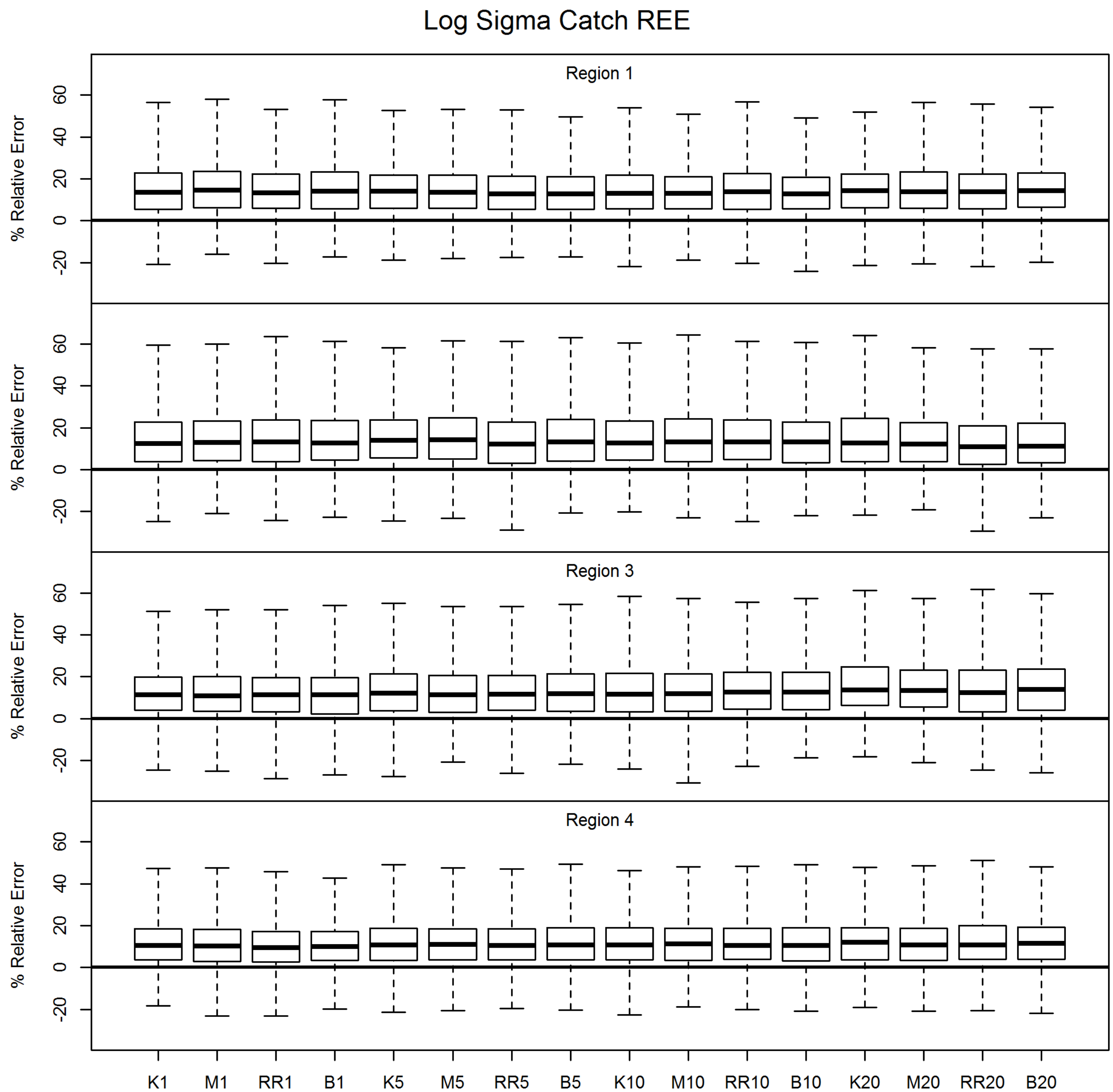

Figure 7: Relative error (%) of harvest variance of an ITCAAN model under different movement rates and parameter estimation assumptions (Scenario group 1) for 1000 simulation iterations. Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

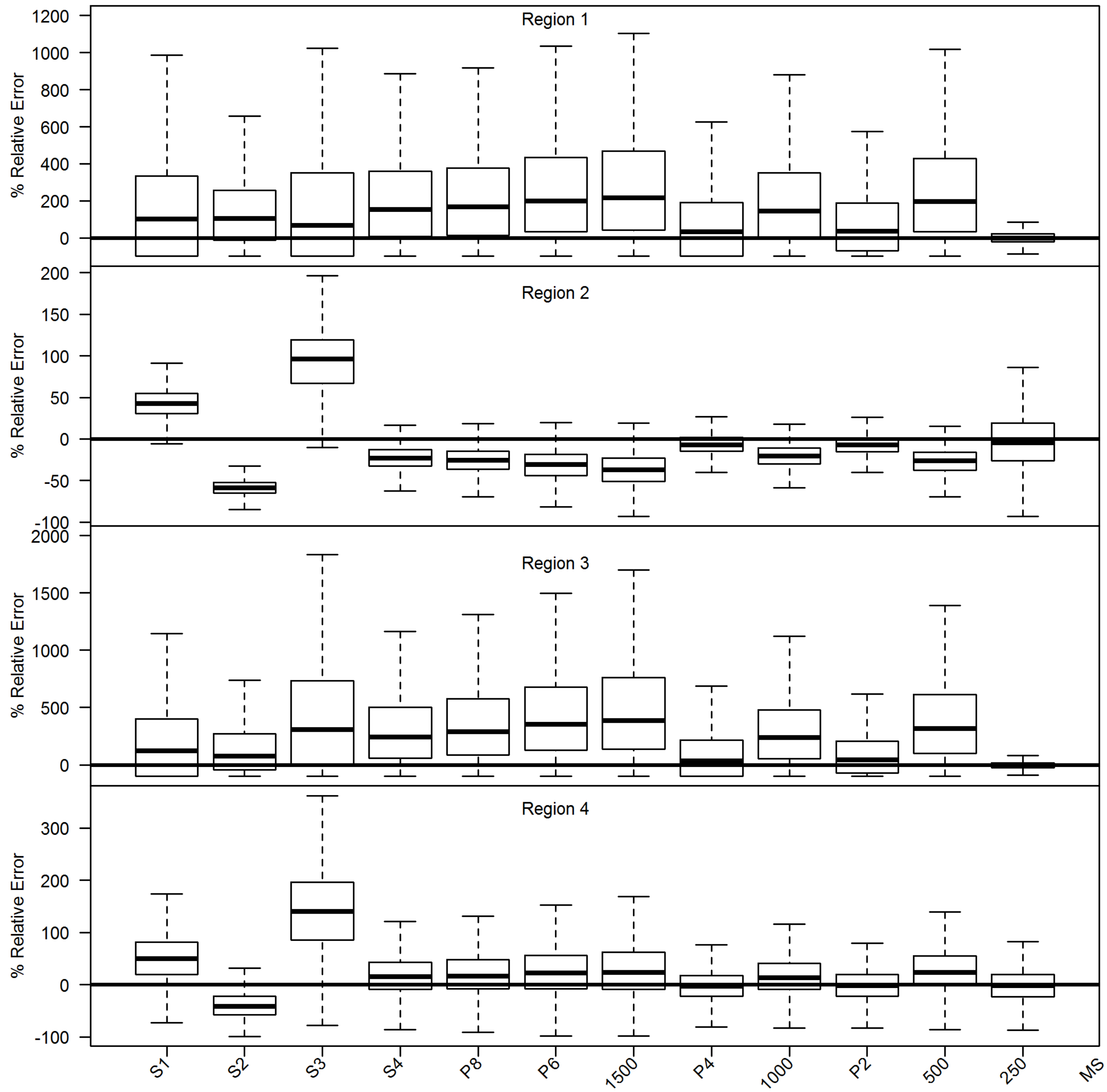

Figure 8: Relative error (%) of initial abundance estimates for each region of an ITCAAN model under misspecfied natural mortality and reporting rates, tag cohort size, spatial complexities, and equal productivities (Scenario groups 2-5). Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded. Note the difference in y-axis scale between regions.

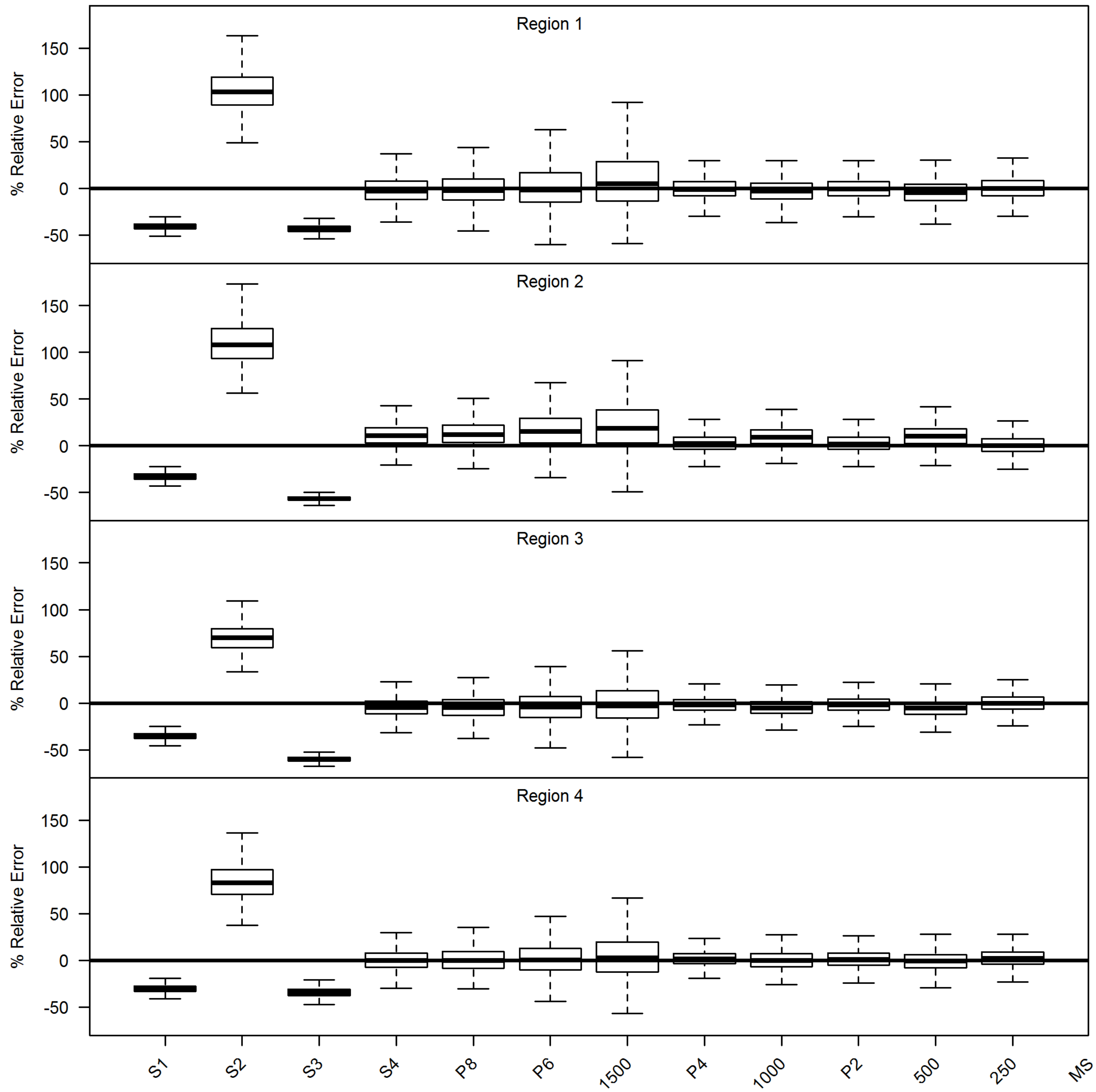

Figure 9: Relative error (%) of survey catchability estimates for each region of an ITCAAN model under misspecfied natural mortality and reporting rates, tag cohort size, spatial complexities, and equal productivities (Scenario groups 2-5). Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded. Note the difference in y-axis scale between regions.

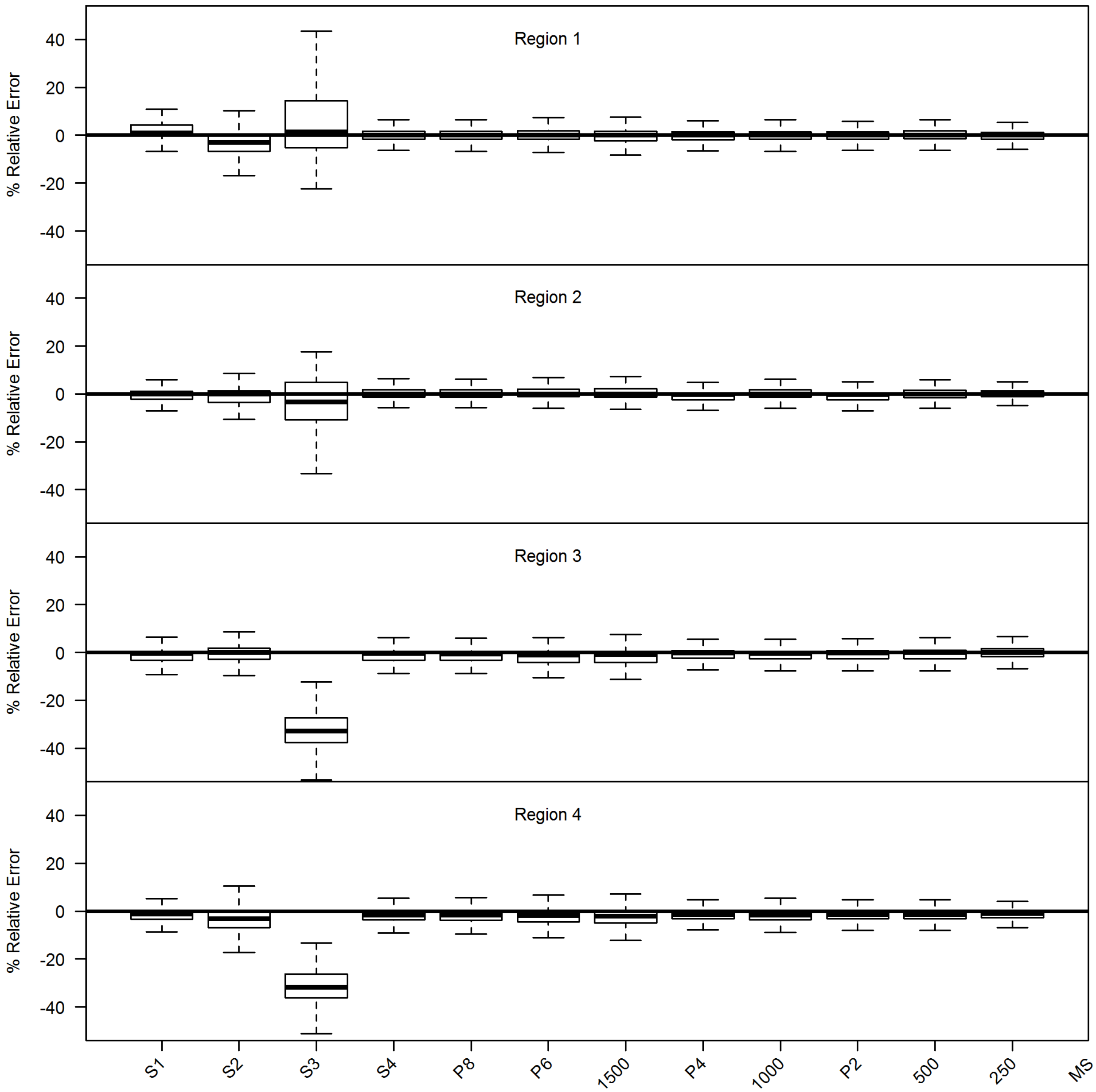

Figure 10: Relative error (%) of fishery selectivity estimates for each region of an ITCAAN model under misspecfied natural mortality and reporting rates, tag cohort size, spatial complexities, and equal productivities (Scenario groups 2-5). Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

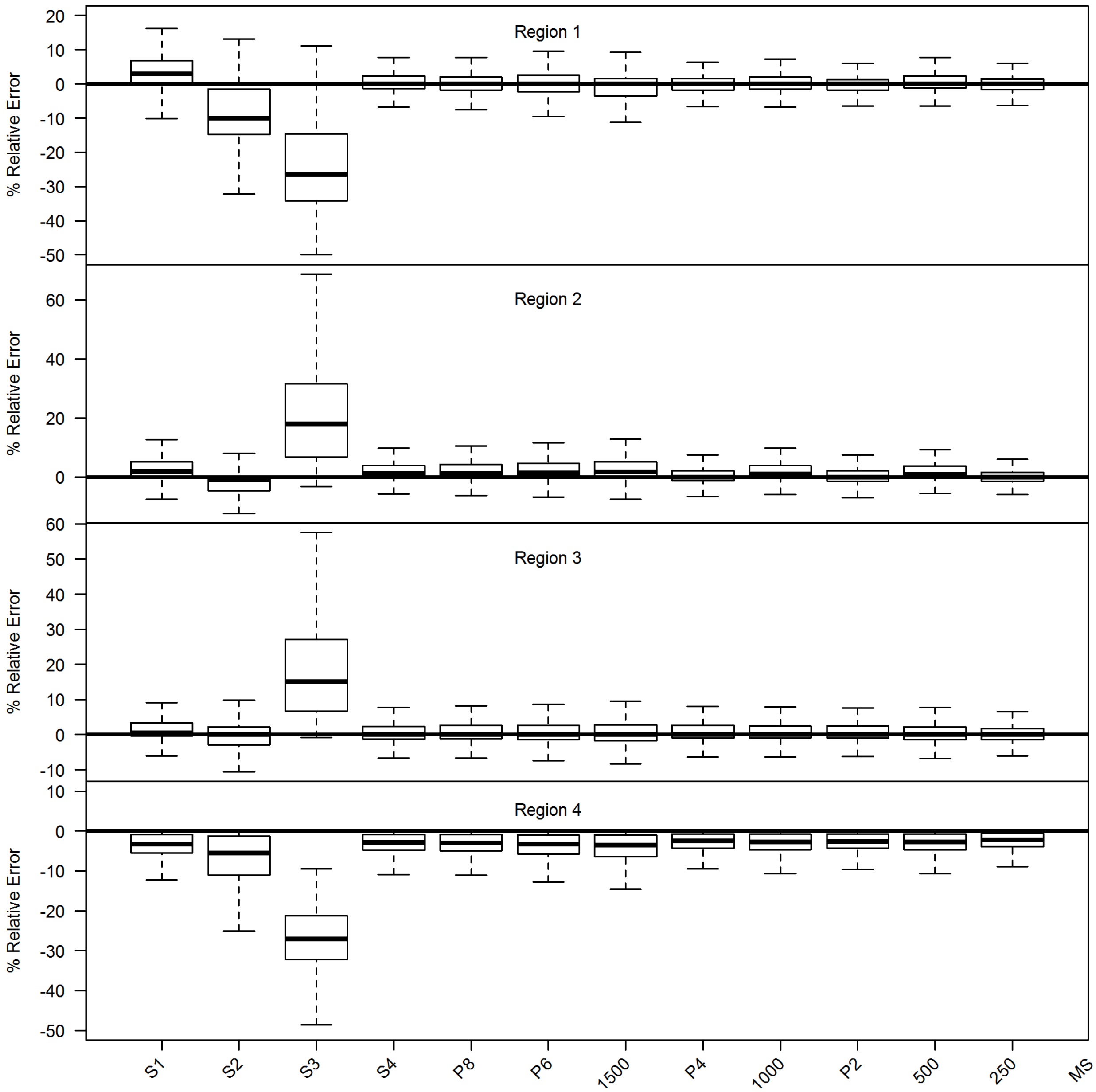

Figure 11: Relative error (%) of survey selectivity estimates for each region of an ITCAAN model under misspecfied natural mortality and reporting rates, tag cohort size, spatial complexities, and equal productivities (Scenario groups 2-5). Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded. Note the difference in scale between the different regions.

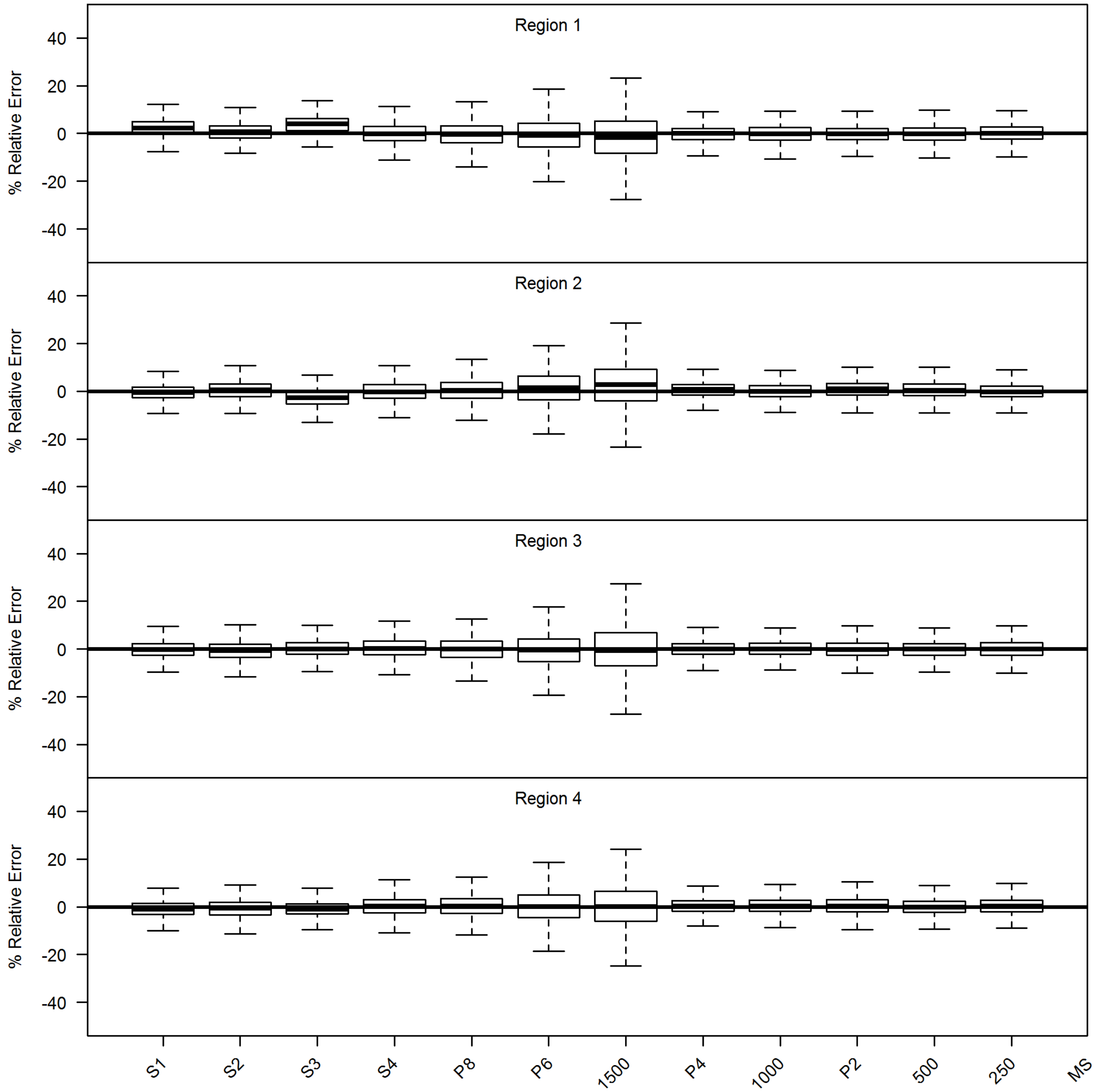

Figure 12: Relative error (%) of proportion of stock that stays in natal region estimates for each region of an ITCAAN model under misspecfied natural mortality and reporting rates, tag cohort size, spatial complexities, and equal productivities (Scenario groups 2-5). Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

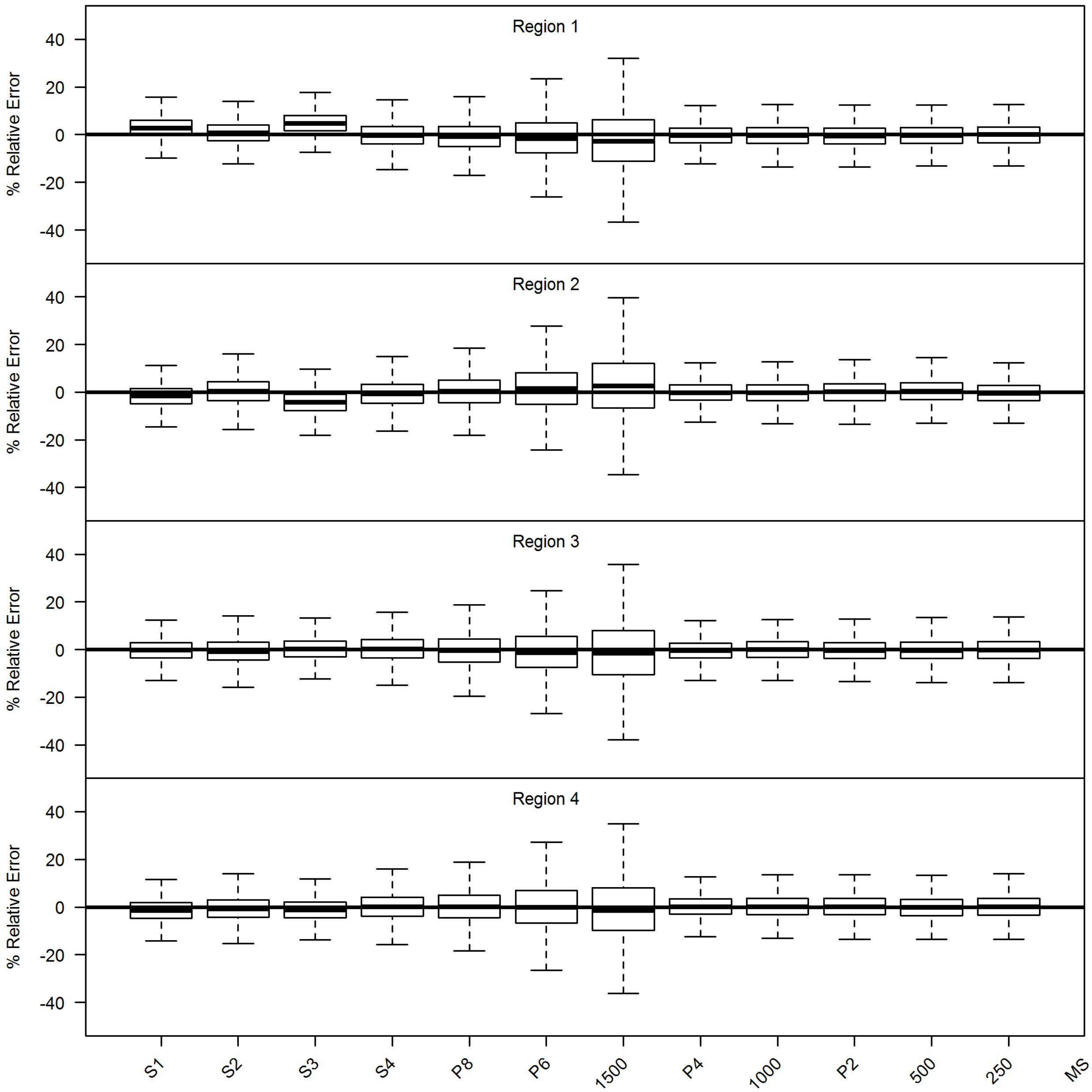

Figure 13: Relative error (%) of proportion of stocks that moves to all other non-natal region estimates for each region of an ITCAAN model under misspecfied natural mortality and reporting rates, tag cohort size, spatial complexities, and equal productivities (Scenario groups 2-5). Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

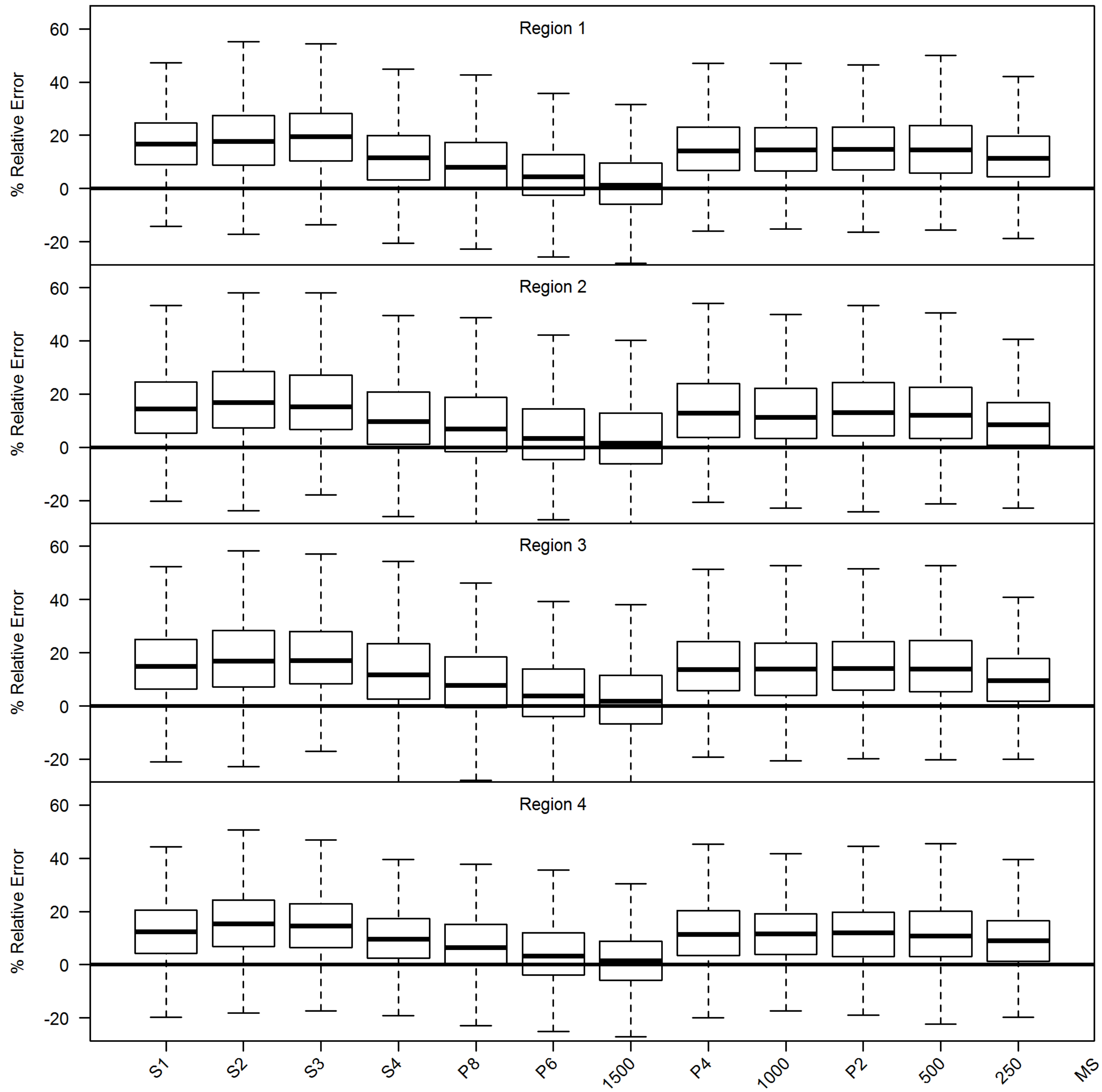

Figure 14: Relative error (%) of log variance of catch estimates for each region of an IT-CAAN model under misspecfied natural mortality and reporting rates, tag cohort size, spatial complexities, and equal productivities (Scenario groups 2-5). Table 1 lists the model abbreviations and corresponding model components. Whiskers on the boxplots extend to 1.5 times the inter-quartile range and points outside this range were excluded.

Table 1: Parameters in the data generating model that are unique for each age.

Parameter	Age 2	Age 3	Age 4	Age 5	Age 6	Age 7
Weight	0.8347	1.1659	1.4875	1.7687	1.944	2.3323
Maturity	0.308	0.824	0.914	0.935	0.978	1
Survey selectivity region 1	0.6	0.7	1	0.9	0.9	0.9
Survey selectivity region 2	1	0.8	0.6	0.55	0.55	0.3
Survey selectivity region 3	1	0.5	0.4	0.3	0.3	0.3
Survey relectivity region 4	1	1	1	1	1	1
Fishery selectivity region 1	0.35	0.98	1	0.7	0.5	0.5
Fishery selectivity region 2	0.4	1	0.9	0.8	0.8	0.7
Fishery selectivity region 3	0.1	0.6	0.65	0.7	0.8	1
Fishery selectivity region 4	0.01	0.13	0.35	1	1	1
Allocation of tags	5%	10%	20%	20%	20%	25%

Table 2: Parameters used to simulate data for each of the four regions in the simulation model.

Parameter	Region 1	Region 2	Region 3	Region 4
Ricker stock recruit α	2.41807	1.48449	1	0.34915
Ricker stock recruit β	1.29135e-6	3.0618e-8	10e-6	2.80287e-8
Survey catchability	1.5e-5	5e-6	2e-7	8e-7
Fishery catchability	2e-6	8e-6	3e-5	6e-5
Reporting rate	0.5	0.5	0.5	0.5
Natural mortality	0.32	0.32	0.32	0.32
μ	0.4382821	0.1941417	0.1941417	0.2317251
ρ	0.8441864	0.8441864	0.8441864	0.8441864
σ	0.06019978	0.06019978	0.06019978	0.06019978
Catch CV	0.1	0.1	0.1	0.1
Effort CV	0.1	0.1	0.1	0.1
Survey CV	0.2	0.2	0.2	0.2
Initial abundance CV	0.3	0.3	0.3	0.3
Data generating harvest age comp. samp size	1000	1000	1000	1000
Estimating model harvest age composition ESS	150	150	150	150
Data generating survey age comp. samp size	1000	1000	1000	1000
Estimating model survey age composition ESS	150	150	150	150
ρ_s	$\sim Trunc.Normal(0.466, 0.260, -0.99, 0.99)$			
σ_s	$\sim Trunc.Normal(0.777, 0.313, 0, \infty)$			

```
##This is code to run in R to generate a population of fish with characteristics like Lake Erie walleye with 4 spawning stocks and 4 regions of harvest. This script uses random number generators to create unique population dynamics and data sets everytime it is run. The out put from this script is a .dat file  and .pin file to be used in the release ADMB model. To run the code you must pass 9 arguments to the script using Rscript based on the scenario options described below.

library(truncnorm)
library(methods)
args=commandArgs(trailingOnly=TRUE)
args=as.numeric(args)
if (length(args)<9){stop ("You must put in atleast 9 arguments for the program to run successfully")}
if (length(args)==9){args[10]=4}
#Put in safety checks so that the correct simulation types are called
if (!(args[2]==1 | args[2]==2)) stop("RR Generation Type (args[2]) must equal 1 or 2")
if (!(args[3]==1 | args[3]==2 | args[3]==3)) stop("RR Estimation Type (args[3]) must equal 1,2 or 3")
if (!(args[6]==1 | args[6]==2)) stop("M Generation Type (args[6]) must equal 1 or 2")
if (!(args[7]==1 | args[7]==2 | args[7]==3)) stop("M Estimation Type (args[7]) must equal 1,2 or 3")
if (!(args[10]==4 | args[10]==8)) stop("Number of fisheries (args[6]) must equal 4 or 8")
#Read in commandline arguments there should be 9 of them the order of them is Movement Type, Reporting Rate Generation Type, Reporting Rate Estimation type, Reporting Rate Estimation Phase, Natural Mortality Generation Type, Natural Mortality Estimation Type, Natural Mortality Estimation Phase and an optional number of fisheries (8 fisheries has not been tested and does not currently work).

#This code is to create data to be used in the assessment model
#Data to be created includes: annual catch data, effort data/CPUE, index of abundance, tag return data, annual age composition data, tag returns

############### Scenario options

##Movement Scenarios 1: Base Case 70% stay 10% leave  2: 97% stay 1% leave 3: 85% stay 5% leave  4: 40% stay 20% leave  5: movement matrix 1  6: movement matrix 2  7:No movement  8: Lake Erie  9:Other?
MvmntType=args[1]
##Reporting Rate Scenarios 1: constant Reporting rate 50% spatially constant 2: randomly varying AR(1) process with mean 50% different in each region
RRType=args[2]
##Reporting Rate Estimation Type 1: constant value through time series 2: 5 Year block estimated 3: Random walk estimation with yearly estimates
RREst=args[3]
##Reporting Rate is it estimated?  If value is positive it is the phase that the parameter is estimated in the model, if negative and constant then it is a known value
PhaseRR=args[4]
##Time Varying Reporting Rate is it estimated?
```

```
RRVaryPhase=args[5]
##Natural Mortality generation options 1: Constant over the time series 2: Autoregressive 1 process generates yearly values
MType=args[6]
##Natural Mortality estimation options 1: Constant 2: 5Year Block estimation 3: Random walk estimation with yearly estimates
MEst=args[7]
#Natural mortality phase of estimation
PhaseM=args[8]
#Time Varying Natural mortality phase of estimation
MVaryPhase=args[9]
#How often the Natural Mortality Parameter varies Should keep this at 5 since tpl is set for this
YrsMVary=5


##Time Varying Movement Phase of estimation
MvmntVaryPhase=-10
## Time Varying movement scenarios 0:Not time varying  1: Randomly varying Movement  2: linealy increaseing movement out
MvmntTVType=0
#Number of fisheries.  If there are same number of fisheries as region then there is one in each, if there is double there is two in each. Commericial in each region and then recreational in each region if two fisheries.
fisheries=args[10]
#Tag Loss Scenarios 0: No tag loss  1: one tag loss  2: differnt tag loss in each region release
TagLossType=0
################ PARAMETERS

#Set up parameters for total number
years=40
regions=4
stocks=4
ages=6
if (MType==1){
    M=rep(0.32,years)   ## Constant M
} else if (MType==2){   ## AR(1) M
    M=numeric(years)
    #Set the autocorrealation for M based on simulating different values, these looked the most reasonable
    Mphi=0.8
    #Set the standard deviation that you want the stationary SD to be
    Msd=0.05
    #set the standard deviation of the random variable so that the stationary variance is equal to 0.05^2
    Msigma=sqrt(Msd^2*(1-Mphi^2))
    M[1]=rtruncnorm(1,mean=0.32,sd=Msigma,a=0,b=Inf)
    #Calculate constant so that the mean will be the Natural mortality that we want 0.32
    c=0.32*(1-Mphi)
    #Calculate an autoregressive trend for the natural mortality
    for (y in 2:years) M[y]=c+Mphi*M[(y-1)]+rtruncnorm(1, mean=0,sd=Msigma,a=-(Mphi*M[(y-1)]+c),b=Inf)
```

```
}

#Vector for proportion of tags lost for each stock could be fancier if need be
#For now assume there is no tag loss
TagLoss=numeric(length=stocks)

##Generate the Reporting rate based on the RRType. When RRType==1 then constant at 50% cross all regions and time.  When RRType==2 then generate random process for each region using an AR(1) process with mean 0.5 and stationary variance of 0.05

if (RRType==1){ReportingRate=matrix(0.5,nrow=years,ncol=fisheries)}
if (RRType==2){
   ReportingRate=matrix(0,nrow=years,ncol=fisheries)
   #Set the mean RR for all regions
   if (fisheries==4){   meanRR=rep(0.5,fisheries)}
   if (fisheries==8){   meanRR=c(rep(0.15,4),rep(0.43,4))}
   #set what the standard deviation of the stationary variance is
   sdRR=0.05
   #Set the autocorrelation level based on test plots
   RRphi=0.7
   #Calculate the SD for the white noise random error
   RRsigma=sqrt(sdRR^2*(1-RRphi^2))
   #Calculate constant so that mean is close to 0.5
   c=meanRR*(1-RRphi)
   #Generate Starting value based on a truncated normal distribution
   RR[1,]=rtuncnorm(fisheries,meanRR,RRsigma,a=0,b=1)
   #Generate the time series using an AR(1) process
   for (f in 1:fisheries){
       for (y in 2:years){
           RR[y,f]=c[f]+RRphi*RR[(y-1),f]+rtruncnorm(1,mean=0,sd=RRsigma,a=-(RRphi*RR[(y-1),f]+c),b=1-(c+RRphi*RR[(y-1),f]))
       }
   }
}

## Set the true vale of reporting rate
rr=ReportingRate[1,]

#Set the fishing mortality rate for each area and each fishery
effort = array(0, dim=c(years,fisheries))
##Create random fishing mortality trends using an AR(1) process that is estimated from the Western Basin fully selected fishing mortality. The means for the different regions are calculated based on the estimated values from the fully selected age and the central basin is assumed to have the same mean as the eastern Basin
meanFs=numeric(length=fisheries)

if (fisheries==4){
    #Calculate the means of the fishing mortalities for the 4 regions
    #Lake Huron Total Fishing mortality
    meanFs[1]=mean(c(0.174425,0.589382,0.0872441,0.0700667,0.0731005,0.566263,0.943108,0.107766,0.103648,0.
    )
```

```
    #Western Lake Erie Total Fishing mortality
    LakeErieF=c(0.375544, 0.496701, 0.378103, 0.45361, 0.360045, 0.239472,
   0.19751, 0.156237, 0.199569, 0.193022, 0.218706, 0.172667, 0.152607,
   0.133225, 0.16557, 0.224292, 0.198858, 0.239118, 0.295636, 0.248441,
   0.314397, 0.232081, 0.207921, 0.124245, 0.0870103, 0.0938314, 0.0748686,
   0.0846649, 0.090858, 0.0987179, 0.0811872, 0.0767647, 0.0718532,
   0.0668829, 0.107389, 0.116998, 0.15464)
    meanFs[2]=mean(LakeErieF)
    #Eastern Lake Erie Total Fishing mortality
    meanFs[3]=mean(c
   (0.176891,0.360011,0.181095,0.380324,0.186269,0.205295,0.252238,0.344163,0.239906,0.158
   )
    #Use the same mean for  Central Lake Erie as Western Lake Erie Total
   Fishing mortality
    meanFs[4]=meanFs[2]
    ##Calculate the AR(1) process from the Western basin fishing mortalities
    z=ar(LakeErieF,FALSE,order.max=1)
    #Set the autocorrelation for the processes
    Fphi=z$ar
    #Calculate the constant that needs to be added so that the mean is that of
     the regions
    cons=meanFs*(1-Fphi)
    #Set what the standard deviation of the process is based on wLE
    Fsigma=sqrt(z$var.pred)
    ## Randomly generate a starting F from a truncated normal distribution
   with lower bound 0
    effort[1,]=rtruncnorm(fisheries,meanFs,Fsigma,a=0,b=Inf)
    for (f in 1:fisheries){
        for (y in 2:years){
             #Randomly generate a Fishing mortality schedule using an AR(1)
   process but used truncated normal distributions so that negative values
   are not generated
             #Total Fishing mortality Lake Huron
          effort[y,f]=cons[f]+Fphi*effort[(y-1),f]+rtruncnorm(1,mean=0,sd=Fsigma,
   a=-(Fphi*effort[(y-1),f]+cons[f]),b=Inf)
        }
    }
}

if (fisheries==8){
    #Fishing mortality Trapnet Lake Huron
    meanFs[1]=mean(c
   (1.47358,0.0555759,0.171753,0.716091,0.0487051,0.0350967,0.0372921,0.691145,1.17786,0.0
   )

    #Fishing mortality commercial gillnet Ontario western Lake Erie
    meanFs[2]=mean(c
   (0.0470186,0.0709162,0.0728645,0.0815978,0.0897574,0.100577,0.0711576,0.0469349,0.06349
   )

    #Fishing mortality commercial central Lake Erie Use commercial fishery for
     Western Basin
    meanFs[3]=mean(c
   (0.0470186,0.0709162,0.0728645,0.0815978,0.0897574,0.100577,0.0711576,0.0469349,0.06349
```

```
)
#Fishing mortality commercial gillnet MU4  eastern Lake Erie
meanFs[4]=mean(c
(0.0520189,0.0383161,0.126938,0.185498,0.0544823,0.067996,0.139998,0.108291,0.0672252,0
)

#Fishing mortality Gillnet Lake Huron Used gill net because the mean F of
gill net and trapnet fisheries is closer to mean total F than would be
using the recreational F and gill net has a similar selectivity to the
recreational fishery
meanFs[5]=mean(c
(0.0133838,0.00971396,0.00622311,0.00744907,0.00612068,0.00611903,0.00483119,0.0085199
)

#Fishing mortality Ohio recreational fishery in western Lake Erie
meanFs[6]=mean(c
(0.341369,0.445157,0.325143,0.394301,0.294806,0.166369,0.14579,0.122123,0.143579,0.1395
)

#Fishing mortality recreational central Lake Erie    Use Ohio Fishing from
 the Western Basin
meanFs[7]=mean(c
(0.341369,0.445157,0.325143,0.394301,0.294806,0.166369,0.14579,0.122123,0.143579,0.1395
)
#Fishing mortality recreational NY and PA eastern Lake Erie
meanFs[8]=mean(c
(0.124872,0.321695,0.0541569,0.194826,0.131787,0.137299,0.11224,0.235872,0.172681,0.105
)

LakeErieTotalF=c(0.375544, 0.496701, 0.378103, 0.45361, 0.360045,
0.239472, 0.19751, 0.156237, 0.199569, 0.193022, 0.218706, 0.172667,
0.152607, 0.133225, 0.16557, 0.224292, 0.198858, 0.239118, 0.295636,
0.248441, 0.314397, 0.232081, 0.207921, 0.124245, 0.0870103, 0.0938314,
0.0748686, 0.0846649, 0.090858, 0.0987179, 0.0811872, 0.0767647,
0.0718532, 0.0668829, 0.107389, 0.116998, 0.15464)
##Calculate the AR(1) process from the Western basin fishing mortalities
z=ar(LakeErieTotalF,FALSE,order.max=1)
#Set the autocorrelation for the processes
Fphi=z$ar
#Calculate the constant that needs to be added so that the mean is that of
 the regions
cons=meanFs*(1-Fphi)
#Set what the standard deviation of the process is based on wLE
Fsigma=sqrt(z$var.pred)
## Randomly generate a starting F from a truncated normal distribution
with lower bound 0
effort[1,]=rtruncnorm(fisheries,meanFs,Fsigma,a=0,b=Inf)
for (f in 1:fisheries){
    for (y in 2:years){
        #Randomly generate a Fishing mortality schedule using an AR(1)
process but used truncated normal distributions so that negative values
are not generated
        #Total Fishing mortality Lake Huron
   effort[y,f]=cons[f]+Fphi*effort[(y-1),f]+rtruncnorm(1,mean=0,sd=Fsigma,a
```

```
    =-(Fphi*effort[(y-1),f]+cons[f]),b=Inf)
        }
    }

}

if (any(effort<=0)) stop("Apical Fishing mortality generated a value that is
    less than or equal to zero")

#Create an indicator variable for if a fishery is active in region
#For now will assume all years active
FisheryActive=array(0,dim=c(fisheries,regions))
#Create an loop instead of putting in data individuall will need to when doing
     seperate fisheries
for( i in 1:regions){FisheryActive[i,i]=1}
if (fisheries>regions){for( i in 1:regions){FisheryActive[i+fisheries,i]=1}}
#Set the selectivity at age for each fishery for each area

#selectivity for L Huron based on trapnet selectivity, western L Erie based on
     Ontario commercial, central L Erie based on Ohio2west (recreational
    fishery), eastern L Erie Ontario gill net if 4 fisheries active
if (fisheries==4)selectivity=array(c(0.35,0.98,1,0.7,0.5,0.5,
    0.4,1,0.9,0.8,0.8,0.7,  0.1,0.6,0.65,0.7,0.8,1,  0.01,0.13,0.35,1,1,1),dim
    =c(ages,fisheries))
if (fisheries==8) error("I did not set the selectivities for 8 fisheries!")

#selectivity for L Huron commercial, western L Erie commercial, central L Erie
     commercial, eastern L Erie commercial gillnet, L Huron recreational based
     on recreational scaled to 7 ages max, western L Erie recreational based
    on Ohio west 2, central L Erie recreational, eastern L Erie NYPA
    recreational anglers

#selectivity for Survey in L Huron, western L Erie based on Ontario CPUE
    survey, central L Erie is based on the Ohio cpue western basin survey and
    eastern L Erie assumes all ages are fully selected in that order
SurveySel=array(c(0.6,0.7,1,0.9,0.9,0.9,  1,0.8,0.6,0.55,0.55,0.3,
    1,0.5,0.4,0.3,0.3,0.3, 1,1,1,1,1,1),dim=c(ages,regions))


#Create vector of the ages that are fully recruited to the respective gears.
    Make sure that the age is one less than the acual age because age 2 is a=1
FisheryFullySelected=numeric(fisheries)
SurveyFullySelected=numeric(stocks)
#Calculate what the index for the maximum selectivity value in each row of the
     selectivity maxtrix  This will give a warning that the number of items
    to replace is not a multiple of replacement length if there is more than
    one age that is fully selected. This is okay because you just want the
    first age that is fully selected.
for(f in 1:fisheries) FisheryFullySelected[f]=which.max(selectivity[,f])
for(s in 1:stocks) SurveyFullySelected[s]=which.max(SurveySel[,s])

#Set the initial population abundance at age for each area
if (stocks>=1){N0R1 = c(9000000,7000000,5000000,3000000,1000000,1500000)}
if (stocks>=2){N0R2 = c(1000000,800000,600000,400000,200000,90000)}
```

```
if (stocks>=3){N0R3 = c(500000,300000,100000,80000,50000,50000)}
if (stocks>=4){N0R4 = c(500000,300000,100000,80000,60000,90000)}
if (stocks>=5){N0R5 = c(50000,25000,10000,5000,1000,2500)}

#Set the movement rate between each area
#Rows indicate the region fish are coming FROM
#Columns indicate the region fish are moving TO
#Movement[FROM,TO]
if (MvmntType==1) Movement=matrix(data=c(.7,.1,.1,.1,  .1,.7,.1,.1,
    .1,.1,.7,.1,  .1,.1,.1,.7), nrow=regions,byrow=TRUE)
if (MvmntType==2) Movement=matrix(data=c(.97,.01,.01,.01, .01,.97,.01,.01,
    .01,.01,.97,.01, .01,.01,.01,.97), nrow=regions,byrow=TRUE)
if (MvmntType==3) Movement=matrix(data=c(.85,.05,.05,.05, .05,.85,.05,.05,
    .05,.05,.85,.05, .05,.05,.05,.85), nrow=regions,byrow=TRUE)
if (MvmntType==4) Movement=matrix(data=c(.4,.2,.2,.2,  .2,.4,.2,.2,
    .2,.2,.4,.2,  .2,.2,.2,.4), nrow=regions,byrow=TRUE)
if (MvmntType==5) Movement=matrix(data=c
    (.95,.05,0,0,.15,.55,.25,.05,.02,.07,.8,.11,0,.02,.12,.86),nrow=regions,
    byrow=TRUE)
if (MvmntType==6) Movement=matrix(data=c
    (.97,.03,0,0,.08,.78,.12,.02,.01,.03,.9,.06,0,.01,.06,.93),nrow=regions,
    byrow=TRUE)
if (MvmntType==7) for(i in 1:regions)Movement=matrix(c(1,0,0,0, 0,1,0,0,
    0,0,1,0, 0,0,0,1),nrow=regions,byrow=TRUE)
if (MvmntType==8) Movement=matrix(data=c(.75,.12,.08,.05, .07,.8,.08,.05,
    .03,.06,.87,.04,  .02,.04,.06,.88), nrow=regions,byrow=TRUE)

#perform test to make sure that the rows sum to 1 so that no fish are created
checkSums=rowSums(Movement)
eps=1
while ((eps+1)>1){eps=0.5*eps}
for (r in 1:regions){if (!(checkSums[r]<(1+2*eps) && checkSums[r]>(1-2*eps))){
    stop("Movement does not sum to 1")}}


#Set the parameters for the recruitment curve
alpha = c(2.41807, 1.48449, 1,0.34915)
beta = c(1.29135e-6, 3.0618e-8, 1e-6,2.80287e-8 )


#Set the maturity schedule to use in the Ricker equation for recruitment
#Below are values from the  western basin assessment
maturity=c(0.308,0.824,0.914,0.935,.978,1)
weight=c(0.8347,1.1659,1.4875,1.7687,1.944,2.3323)

#set the CV for the Initial Abundance deviations
RandomCV=0.3

#set the CV for the observation error in the observed datasets
catchCV = 0.1
effortCV=0.1
processCV=0.04
surveyCV=0.2
```

```
#Create array to keep track of the temporal correlation for each stock
#This value comes from Thorson et al 2014 The estimate for perciformes the
    Autocorrelation from table 2 of 0.466 with a SD of 0.260
#Randomly simulate the autocorrelation based on the posterior distribution
    mean and sd
rho=rtruncnorm(4,mean=0.466,sd=0.260,a=-0.99,b=0.99)

#Calculate what the variability needs to be to get stationary variance with
    the autocorrelation term
logrecruitCV=list(mu=NA,sd=NA)
#Randomly generate the recruitment CV based on the estimated SD from Thorson
    et al 2014. This does not need to be bias corrected or transformed from a
    CV because it is estimated on the log scale as a standard deviation
logrecruitCV$sd=rtruncnorm(4,mean=0.777,sd=0.313,a=0,b=Inf)
logrecruitCV$mu=-(.5*logrecruitCV$sd^2*(1-rho)/sqrt(1-rho^2))


#Function to calculate what the mean and the standard deviation should be for
    the lognormal distribution given the mean and CV on the normal scale
lognormmusd <- function(mean,CV) {
     sigsq=log(CV^2+1)
     mu=log(mean)-(.5*sigsq)
     result=list(mu=mu,sd=sqrt(sigsq))
     return(result)
}

#Calculate the mean and sd for the random variables to be input into rlnorm
    functions

logcatchCV=lognormmusd(1,catchCV)
logeffortCV=lognormmusd(1,effortCV)
logprocessCV=lognormmusd(1,processCV)
logsurveyCV=lognormmusd(1,surveyCV)
logRandomCV=lognormmusd(1,RandomCV)

Test=array(0,dim=c(151,ages,stocks))
if (stocks>=1) Test[1,,1]=N0R1;
if (stocks>=2) Test[1,,2]=N0R2;
if (stocks>=3) Test[1,,3]=N0R3;
if (stocks>=4) Test[1,,4]=N0R4;
if (stocks>=5) Test[1,,5]=N0R5

for(y in 1:150){
     for(s in 1:stocks){
          Test[(y+1),1,s]= alpha[s]*((maturity*weight)%*%Test[y,,s])*exp(-beta[s
    ]*((maturity*weight)%*%Test[y,,s]))
          for(a in 1:(ages-1)){
               Test[(y+1),(a+1),s] = Test[y,a,s]*exp(-M[1])
          }
          Test[(y+1),ages,s]=Test[y,(ages-1),s]*exp(-M[1])+Test[y,ages,s]*exp(-M
    [1])
     }
}
StartPop=Test[151,,]
```

```

#Create an array for the abundance through time in each area
N=array(0, dim=c((years+2),ages,stocks))
#Set the initial population sizes as the equibilrium for the recruitment
    functions without movement but add in random variation to the ages
N[1,,]=StartPop*rlnorm(n=length(StartPop),meanlog=logRandomCV$mu, sdlog=
    logRandomCV$sd)
#Start Autocorrelation value for the second year of recruitment
Autocorrelation=array(dim=c(years+1,stocks))
Autocorrelation[1,]=rlnorm(n=stocks,meanlog=logrecruitCV$mu, sdlog=
    logrecruitCV$sd)
#Calculate the Recruitment for the second year with the first random value of
    autocorrelation. Need to do this here because of the two year lag on
    recruitment
N[2,1,]=StartPop[1,]*Autocorrelation[1,]

#set the sample size for the age composition data simulation
AgeCompSamples=array(1000,dim=c(years,ages,fisheries))
SurveyESS=array(1000,dim=c(years, ages,regions))

#Create array to store the fish in after they have moved
#The stock is the area from which the fish originated from and the region is
    the area to which is moves post spawning at the begining of the year
NMvmnt=array(0,dim=c(years,ages,stocks,regions))
#Create array for the total catch in each region
CatchAge=array(0,dim = c(years,ages,fisheries))
TotalCatch=array(0,dim=c(years,fisheries))

######################### Abundance Calculations
#calculate the population abundance for the 5 populations based upon the above
     parameters

#Let the following letter ber used for loops
# a  is the age of the fish 2:7 in reality but just use 1:6 for calculations
# y  is the year 1:40
# r  is the region 1:5 in which the fish is residing
# f  is the fisheries  (for now just one)
# s  is the stock from which the fish originates. For now we are assuming that
     the number of regions is the same as the number of stocks

#Calculate arrays for F, Z and Surv
F=array(0,dim=c(years,ages,regions,fisheries))
FTotal=array(0,dim=c(years,ages,regions))
Z=array(0,dim=c(years,ages,regions))
# FFull=array(0,dim=c(years,fisheries))
FFull=effort

for(f in 1:fisheries){
    #Apply process error to the underlying apical F
    # FFull[,f]=effort[,f]*rlnorm(length(effort[,f]),logprocessCV$mu,
    logprocessCV$sd)
     for(r in 1: regions){
         #Calculatethe age and region specific fishing mortality
         F[,,r,f]=(FFull[,f]%*%t(selectivity[,f]))*FisheryActive[f,r]
```

```
        #Calculate the total fishing mortality within each region by summing over active fisheries
        FTotal[,,r]=FTotal[,,r]+F[,,r,f]
    }
}


#Add natural mortality to fishing mortality
for(y in 1:years) Z[y,,]=FTotal[y,,]+M[y]
#Convert Z to survival for easier use
Survival=exp(-Z)


SurveyAge=array(0,dim=c(years,ages,regions))
#Survey Catchability coefficient for L Huron based on Saginaw Bay survey, western L Erie based on Ohio CPUE, central L Erie CPUE taken from western basin ontario gill net Q, eastern L Erie NY net CPUE survey
qSurvey=c(1.5e-5,5e-6,2e-7,8e-7)


#Begin loop over all of the years
for(y in 1:years){
    #Begin loop for each area
    for(s in 1:stocks){
        # simulate the recruitment for age 2 for each stock with a temporal autocorrelation so there is a 2 year time lag on recruitment but age 2 is the first age in model
        #This is y+1 because first value was filled in earlier from the equilibrium stock
        Autocorrelation[y+1,s]=rho[s]*Autocorrelation[y,s]+rnorm(n=1,mean=logrecruitCV$mu[s], sd=logrecruitCV$sd[s])*sqrt(1-rho[s]^2)
        N[(y+2),1,s]= alpha[s]*(maturity%*%N[y, ,s])*exp(-beta[s]*((maturity*weight)%*%N[y, ,s])+Autocorrelation[(y+1),s])
        if (N[y,1,s]<5) {
      message("This run through had a population that is less than 5")
      source("../DataSimulator.r")
      #Stop after rerunning to make sure that it doesn't rerun at the end
      stop()
      }
    }   #End stock loop

    #Begin loop over ages
    for(a in 1:ages){
        for(r in 1:regions){
            for(s in 1:stocks){
                #Calculate the number of fish that move to each area from spawning area and apply mortality
                NMvmnt[y,a,s,r]=N[y,a,s]*Movement[s,r]*Survival[y,a,r]
                SurveyAge[y,a,r]=SurveyAge[y,a,r]+N[y,a,s]*Movement[s,r]*exp(-Z[y,a,r]*10/12)*SurveySel[a,r]*qSurvey[r]
                #Calculate the catch for each area with ages seperate
                #Need to sum over the different spawning stocks
                #C=F/Z*(N*(1-surv))
                for(f in 1:fisheries){
```

```
                    CatchAge[y,a,f]=CatchAge[y,a,f]+((F[y,a,f,r]/Z[y,a,r])*(N[y,a,s]*Movement[s,r])*(1-Survival[y,a,r]))
                }
                #Calculate those that survive to the next year to spawn for each stock
                if(a<ages){
                    N[(y+1),(a+1),s]=N[(y+1),(a+1),s]+NMvmnt[y,a,s,r]
                } else{
                    N[(y+1),ages,s]=N[(y+1),ages,s]+NMvmnt[y,ages,s,r]
                }
            }                               #End stock loop
        }                                   #End region loop
        for(f in 1:fisheries){

            #Sum the catch over ages in each area
            #Need to do this outside of stock loop  and region loop or results in over counting the catch
            TotalCatch[y,f]=TotalCatch[y,f]+CatchAge[y,a,f]
        }
    }                           #End age loop
}                               #End year loop
######################### Data Simulation

#Add lognormal observation error to catch in each area
ObservedCatch=TotalCatch*rlnorm(n=length(TotalCatch),meanlog=logcatchCV$mu,sdlog=logcatchCV$sd)
#Fishery Catchability coefficient for L Huron based on gill net catchability , western L Erie based on commercial catchability , central L Erie fishery based on q for Ohio recreation fishery and eastern L Erie based on Mu4 commercial fishery in that order
if (fisheries==4) q=matrix(c(2e-6,8e-6,3e-5,6e-5),nrow=years,ncol=fisheries,byrow=TRUE)
if (fisheries==8) error("I never set the catchability for 8 fisheries")


#Create Arrays to store the observed CPUE survey and age composition proportion for each region
ObservedSurvey=array(NA,dim=c(years,regions))
ObservedSurveyAgeComp=array(NA,dim=c(years,ages,regions))
#Add lognormal observation error to the calculated survey index and apply catchability coefficient
for (y in 1:years){
    for (r in 1:regions){
        ObservedSurvey[y,r]=sum(SurveyAge[y,,r])*rlnorm(1,logsurveyCV$mu,logsurveyCV$sd)
        ObservedSurveyAgeComp[y,,r]=rmultinom(1,SurveyESS[y,1,r],SurveyAge[y,,r])/SurveyESS[y,,r]
    }
}

#Add lognormal observation error to the fishing mortality with a catchability coefficient
ObservedEffort=FFull/q*rlnorm(n=length(FFull), meanlog = logeffortCV$mu,sdlog = logeffortCV$sd)
```

```
#Simulate tag recoveries from multivariate distribution
ObservedAgeComp=array(0,dim=c(years,ages,fisheries))
#Simulate age composition from multivariate distribution of catches and turn
    into a proportion
for (y in 1:years){
    for (f in 1:fisheries){
        ObservedAgeComp[y,,f]=rmultinom(n=1,size=AgeCompSamples[y,1,f],prob=
    CatchAge[y,,f])/AgeCompSamples[y,,f]

    }
}



##########################################Tagging Data simulator

#Number released each year in each region
TagsReleased=matrix(2000,nrow=years,ncol=stocks,byrow=TRUE)

#Assume that there is the same proportion of ages from each release in each
    region
ProportionRelease=c(.05,.1,.2,.2,.2,.25)
ReleaseAge=array(0,dim=c(years,ages,stocks))
TagsAlive=array(0,dim=c(years,(years+1),ages,stocks))

#This keeps track of the tagged fish that are alive at the beginning of each
    year. Thus it starts out as the number of released by age in region for
    each year of release.
#year of release, year of recapture (or current year concerned about), age ,
    stock released from
for(y in 1:years){
      for(s in 1:stocks){
          ReleaseAge[y,,s]=round(TagsReleased[y,s]*ProportionRelease)
          ReleaseAge[y,ages,s]=TagsReleased[y,s]-sum(ReleaseAge[y,-ages,s])
          TagsAlive[y,y,,s]=ReleaseAge[y,,s]
      }
}

#Create matrix to calculate where fish are after movement each year
TagMvmnt=array(0,dim=c(years,years,ages,stocks,regions))

#Create vector to store fate of tagged fish in a region from each release
TagFate=array(0,dim=c(years,years,ages,stocks,regions,(fisheries+2)))
#Caught by fisheries, natural mortality, survival
#Create array to store the recaptured tags information
TagsRecaptured=array(0,dim=c(years,years,ages,stocks,regions,fisheries))
#release event year,recapture year, age, release stock, recapture region



#Create vector to temporarily store the probability of capture by fisheries
CaptureProb=numeric(length=(fisheries+2))


```

```
#begin loop over tagging year
for(ty in 1:years){
    #begin loop over recapture year
    for(ry in ty:years){
        #loop over ages
        for(a in 1:ages){
            #loop over release stocks
            for(s in 1:stocks){

                #Check to make sure that there are still fish alive for this release at this age
                    #Tag movement to new areas and apply tag loss by removing from the sample size of Tags Alive
                    #This needs to be outside of for loop for regions
                    #Tag movement using MULTINOMIAL distribution
                    TagMvmnt[ty,ry,a,s,]=rmultinom(n=1,size=TagsAlive[ty,ry,a,s]*(1-TagLoss[s]),prob=Movement[s,])
                    #check to make sure tags aren't created or destroyed during movement
                    if(round(TagsAlive[ty,ry,a,s]*(1-TagLoss[s])) != sum(TagMvmnt[ty,ry,a,s,])){stop("Something does not add up in the tag movement")}

                    #loop over recapture region
                    for(r in 1:regions){
                        if(TagMvmnt[ty,ry,a,s,r]<0)stop("negative movement!!!")
                        #Calculate probabilty of death by natural mortality and those that survive
                        CaptureProb[(fisheries+1)]=M[ry]/Z[ry,a,r]*(1-Survival[ry,a,r])
                        CaptureProb[(fisheries+2)]=Survival[ry,a,r]
                        #Loop over fisheries
                        for(f in 1:fisheries){
                            #Calculate the capture probability for each fishery
                            CaptureProb[f]=F[ry,a,f,r]/Z[ry,a,r]*(1-Survival[ry,a,r])
                        }       #End fisheries loop
                        #Determine tag fate using MULTINOMIAL distribution
                        TagFate[ty,ry,a,s,r,]=rmultinom(n=1,size=TagMvmnt[ty,ry,a,s,r],prob=CaptureProb)
                        #store the tags that are recaptured by fishery
                        TagsRecaptured[ty,ry,a,s,r,]=TagFate[ty,ry,a,s,r,1:fisheries]
                        #test to make sure tags aren't created or destroyed during tag fate calculations
                        if(sum(TagFate[ty,ry,a,s,r,])!=TagMvmnt[ty,ry,a,s,r])stop("something not adding up in movement 1")

                    }          #End regions loop
                    #test to make sure tags aren't created or destroyed anywhere
                    if(sum(TagFate[ty,ry,a,s,,])!=sum(TagMvmnt[ty,ry,a,s,]))
```

```
stop("something not adding up in movement 2")
                        #check to make sure that tags weren't created or destroyed
                        if(round(TagsAlive[ty,ry,a,s]*(1-TagLoss[s])) != sum(TagFate[ty,ry,a,s,,])){stop("Something does not add up in the tagging")}
                        #Progress those fish that survive to the next year and age
                        #remove those fish that die from the sample size of released fish i.e. only keep survivals
                        if(a<(ages-1)){
                            TagsAlive[ty,(ry+1),(a+1),s]=sum(TagFate[ty,ry,a,s,,(fisheries+2)])
                        } else if(a==ages){
                            TagsAlive[ty,(ry+1),a,s]=sum(TagFate[ty,ry,(ages-1),s,,(fisheries+2)]+TagFate[ty,ry,ages,s,,(fisheries+2)])
                        }    #End if else for plus group calculations
                }            #End stocks loop
            }                #End age loop
    }                        #End capture year loop
}                            #End tagging year loop

#Calculate the tag returns by summing over ages
TagReturns=colSums(aperm(TagsRecaptured,perm=c(3,1,2,4,5,6)),dim=1)

#reformat the Tag returns to get rid of the dimension for region of recapture
#This assumed that each fishery is only active in one region
#Also apply the reporting rate for that fishery
#This will only work if the fishery is active in only one region
TagsReported=array(0,dim=c(years,years,stocks,fisheries))
NeverRecovered=matrix(data = 0,nrow=years,ncol = stocks)
for(ty in 1:years){
    for(ry in ty:years){
        for(s in 1:stocks){
            for(f in 1:fisheries){
                tempr=which(FisheryActive[f,]==1)
                TagsReported[ty,ry,s,f]=rbinom(1,TagReturns[ty,ry,s,tempr,f],ReportingRate[y,f])
            }
        }
    }
}

for(y in 1:years){
    for(s in 1:stocks){
        NeverRecovered[y,s]=TagsReleased[y,s]-sum(TagsReported[y,,s,])
    }
}

##############################Calculate the parameters that need to be included in the data file for comparison to parameter estimates

LastYearN=numeric(stocks)
for (s in 1:stocks) LastYearN[s]=sum(N[years,,s])


##############################Create .dat file
```

```
#This puts in the first line description and creates the file or overwrites
    existing file since append=false
cat(c("#Simulated data to be read into the assessment model using ADMB","\n"),
    file="SimulatedData.dat",append=FALSE)

#Prints a bunch of variables
cat(c("#number of years",years,"#number of regions",regions,"#number of stocks
    ",stocks,"#number of fisheries",fisheries,"#Number of age classes",ages,"#
    Phase of Natural Mortality estimation",PhaseM,"#Phase of time-varying
    Natural Mortality",MVaryPhase,"#True Value of Natural Mortality",M[1]),sep
    ="\n",append=TRUE, file="SimulatedData.dat")


#Print out the Type of Natural Mortality estimation that will be  used 1==
    constant 2 == 5 year block 3 == random walk
cat(c("","#This is the M Estimation Type 1== constant 2 == 5 year block 3==
    random walk",MEst),file="SimulatedData.dat",append=TRUE,sep="\n")

if(MVaryPhase>0){
     cat("#This is the True Time Varying Natural Mortality \n \n",append=TRUE,
    file="SimulatedData.dat")
     write.table(t(M),append=TRUE,file="SimulatedData.dat",sep =" ",row.names =
     FALSE, col.names = FALSE)
}

#Prints out Tag Loss
cat(c("#This is the Tag Loss as a decimal yearly percentage lost","\n",TagLoss
    ),file="SimulatedData.dat",append = TRUE,sep=" ")
#write.table(TagLoss,file="SimulatedData.dat",append=TRUE,sep =" ",row.names =
     FALSE, col.names = FALSE)

#Prints out Reporting Rate  info
cat(c("","#Phase of Reporting Rate estimated", PhaseRR, "#Phase of time-
    varying Reporting Rate",RRVaryPhase,"#This is the initial guess for the
    reporting rate parameters or value if not estimated", t(rr)),file="
    SimulatedData.dat",append = TRUE,sep="\n")

#Print out the Type of reporting Rate estimation that will be  used 1==
    constant 2 == 5 year block 3 == random walk
cat(c("","#This is the RR Estimation Type 1== constant 2 == 5 year block 3==
    random walk",RREst),file="SimulatedData.dat",append=TRUE,sep="\n")

#Prints the True Mvmnt  matrix
cat(c("\n","#Matrix of True Movement parameters and used to calculate starting
     values","\n"),file="SimulatedData.dat",sep="",append=TRUE)
write.table(Movement,file="SimulatedData.dat",sep=" ",append=TRUE,row.names=
    FALSE,col.names=FALSE)

#Prints fishery active matrix
cat(c("\n","#Matrix of fishery active","\n"),file="SimulatedData.dat",sep="",
    append=TRUE)
write.table(FisheryActive,file="SimulatedData.dat",sep=" ",append=TRUE,row.
```

```
    names = FALSE, col.names = FALSE)

#Prints observed Catch Data
cat(c("\n","\n","#The observed Catch data for the fisheries","\n","\n"),file="
    SimulatedData.dat",append=TRUE,sep="")
write.table(round(ObservedCatch),"SimulatedData.dat",sep=" ",append=TRUE ,row.
    names = FALSE,  col.names = FALSE)

#prints Fishery Effort Data
cat(c("\n","#This is the observed Effort for the data","\n","\n"),file="
    SimulatedData.dat",append=TRUE,sep="")
write.table(round(ObservedEffort,2),"SimulatedData.dat",append=TRUE,sep=" ",
    row.names = FALSE,  col.names = FALSE)

#Print True Fishery Catchability coefficient
cat(c("\n","#This is the True fisheries  Catchability coefficient parameter
    TrueQ","\n \n"),file="SimulatedData.dat",append=TRUE,sep="")
write.table(q[1,],"SimulatedData.dat",append=TRUE,sep=" ",row.names = FALSE,
    col.names = FALSE)

#Print Survey Data
cat(c("\n #This is the observed Survey Data \n \n"),file="SimulatedData.dat",
    append=TRUE,sep="")
write.table(ObservedSurvey,"SimulatedData.dat",append=TRUE,sep=" ",row.names=
    FALSE,col.names = FALSE)

#Print True Survey catchability coefficient
cat(c("\n","#This is the True Survey Catchability Coefficient parameter
    TrueSurveyQ","\n \n"),file="SimulatedData.dat",append=TRUE,sep="")
write.table(qSurvey,"SimulatedData.dat",append=TRUE,sep = " ", row.names =
    FALSE,col.names = FALSE)

#Prints Observed Age Composition
#ObservedAgeComp1=aperm(ObservedAgeComp,perm=c(1,3,2))
cat(c("\n","#This is the simulated age composition","\n","\n"),file="
    SimulatedData.dat",sep="",append=TRUE)
write.table(ObservedAgeComp,file="SimulatedData.dat",append=TRUE,sep =" ",row.
    names = FALSE,  col.names = FALSE)

#Print out FisheryFullySelected age
cat("\n #This is the Age that is fully selected in the respective fishery to
    be used to set fully selected value \n \n",file="SimulatedData.dat",sep=""
    ,append=TRUE)
write.table(FisheryFullySelected,file = "SimulatedData.dat",append=TRUE,sep ="
     ",row.names = FALSE,  col.names = FALSE)

#Prints out the True Fishery Selectivity Parameters
cat(c("\n","#This is the True Selectivity Parameters excluding the fully
    selected","\n","\n"),file="SimulatedData.dat",sep="",append=TRUE)
TrueSelectivity=matrix(NA,nrow=(ages-1),ncol=fisheries)
for (f in 1:fisheries){TrueSelectivity[,f]=selectivity[-FisheryFullySelected[f
    ],f]}
write.table(TrueSelectivity,file="SimulatedData.dat",append=TRUE,sep =" ",row.
    names = FALSE,  col.names = FALSE)
```

```
#Print out Observed Survey Age Composition
cat(c("\n #This is the Observed Survey Age composition \n \n"),file="
    SimulatedData.dat",sep="",append=TRUE)
write.table(ObservedSurveyAgeComp,file="SimulatedData.dat",append=TRUE,sep ="
    ",row.names = FALSE, col.names = FALSE)

#Print out SurveyFullySelected
cat("\n #This is the age that is fully selected in the survey to be used to
    the fully selected age in the assessment \n \n",file="SimulatedData.dat",
    sep="",append=TRUE)
write.table(SurveyFullySelected,file = "SimulatedData.dat",append=TRUE,sep ="
    ",row.names = FALSE, col.names = FALSE)

#Prints out True Survey Selectivity Parameters
cat(c("\n","#This is the True Survey Selectivity Parameters excluding the
    fully selected","\n","\n"),file="SimulatedData.dat",sep="",append=TRUE)
TrueSurveySel=matrix(NA,nrow=(ages-1),ncol=fisheries)
for (r in 1:regions){TrueSurveySel[,r]=SurveySel[-SurveyFullySelected[r],r]}
write.table(TrueSurveySel,file="SimulatedData.dat",append=TRUE,sep =" ",row.
    names = FALSE, col.names = FALSE)

#Prints out the True Initial Abundance TrueN0
cat(c("\n","#This is the True values of the initial Abundance TrueN0","\n","\n
    "),file="SimulatedData.dat",sep="",append=TRUE)
write.table(N[1,2:ages,],file="SimulatedData.dat",append=TRUE,sep=" ",row.
    names = FALSE, col.names = FALSE)

#Calculate and Print out True Mean Recruitment
LogMeanRecruitment=colMeans(log(N[1:years,1,]))
cat(c("\n","#This is the True Mean Recruitment","\n \n"),file="SimulatedData.
    dat",sep="",append=TRUE)
write.table(LogMeanRecruitment,file="SimulatedData.dat",append=TRUE,sep=" ",
    row.names = FALSE, col.names = FALSE)

#Print out True Annual Recruitment
cat(c("\n","#This is the True Annual Recruitment","\n \n"),file="SimulatedData
    .dat",sep="",append=TRUE)
write.table(N[1:years,1,],file="SimulatedData.dat",append=TRUE,sep=" ",row.
    names = FALSE, col.names = FALSE)

#Print out the True Catch Sigma
cat(c("\n","#This is the True Sigma Catch","\n \n"),file="SimulatedData.dat",
    sep="",append=TRUE)
write.table(logcatchCV$sd,file="SimulatedData.dat",append=TRUE,sep=" ",row.
    names = FALSE, col.names = FALSE)

#Print out True Last Year's Abundance summed over ages
cat(c("\n #This is the True Last Years' Abundance \n \n "),file="SimulatedData
    .dat",sep="",append=TRUE)
write.table(LastYearN,file="SimulatedData.dat",append=TRUE,sep=" ",row.names =
     FALSE, col.names = FALSE)

#Print out test number 1
```

```
cat(c("\n","#This is the first test number","\n",1234567890),file="SimulatedData.dat",append = TRUE,sep="")

#Print out reported tag returns
cat(c("\n","\n","#This is the Tags Reported","\n","\n"),file="SimulatedData.dat",append=TRUE,sep="")
write.table(aperm(TagsReported,perm=c(1,4,2,3)),file="SimulatedData.dat",append=TRUE,row.names = FALSE,col.names = FALSE, sep=" ")

#Print out the True Reporting Rate only if it is estimated
if(PhaseRR>0){
  cat(c("\n","#This is the True Mean Reporting Rate","\n","\n"),file="SimulatedData.dat",append=TRUE,sep="")
  write.table(colMeans(ReportingRate),file="SimulatedData.dat",append=TRUE,row.names = FALSE,col.names = FALSE, sep=" ")
}

#Print out the True Time Varying Reporting rate only if it is estimated
if(RRVaryPhase>0){
  cat(c("\n","#This is the True Annual Reporting Rate","\n","\n"),file="SimulatedData.dat",append=TRUE,sep="")
  write.table(ReportingRate,file="SimulatedData.dat",append=TRUE,row.names = FALSE,col.names = FALSE, sep=" ")
}
#Print out test number 2
cat(c("","#This is the second test number",1234567890,""),file="SimulatedData.dat",append = TRUE,sep="\n")

#Prints out Tags released by age
cat(c("#This is the Tags Released by Age, year and stock",""),file="SimulatedData.dat",append = TRUE,sep="\n")
write.table(aperm(ReleaseAge,perm=c(1,3,2)),file="SimulatedData.dat",append=TRUE,sep =" ",row.names = FALSE, col.names = FALSE)

#Prints out Total Tags Released
cat(c("","#This is the Total Tags Released by year and stock",""),file="SimulatedData.dat",append = TRUE,sep="\n")
write.table(TagsReleased,file="SimulatedData.dat",append=TRUE,sep =" ",row.names = FALSE, col.names = FALSE)

#Print out test number 3
cat(c("#This is the third test number",1234567890),file="SimulatedData.dat",append = TRUE,sep="\n")

#Prints out Tags Never Recovered
cat(c("","#This is the number of tags that are never recovered for each release event",""),file="SimulatedData.dat",append = TRUE,sep="\n")
write.table(NeverRecovered,file="SimulatedData.dat",append=TRUE,sep =" ",row.names = FALSE, col.names = FALSE)

#Print out test number 4
cat(c("","#This is the fourth test number",1234567890),file="SimulatedData.dat",append = TRUE,sep="\n")
```

```
#Print out the True abundance at age for each of the stocks. This won't be read into the admb file but it might be useful later
cat("\n #This is the True Abundance at Age for the stocks \n \n", file="SimulatedData.dat",append=TRUE)
write.table(N, file="SimulatedData.dat",append=TRUE, sep =" ",row.names = FALSE, col.names = FALSE)

#Print out the Fishing mortality with the random variation. Won't be read into the admb file but might be important later
cat("\n #This is the True Fishing Mortality with random variation \n\n",file="SimulatedData.dat",append=TRUE)
write.table(FFull, file="SimulatedData.dat",append=TRUE, sep =" ",row.names = FALSE, col.names = FALSE)

###################################
#Code to write a stocastic starting value for release.pin

StartCV=.1



cat("# Log Recruits \n",file="release.pin",append=FALSE)
StartLogRec=t(rnorm(n=length(LogMeanRecruitment),mean=LogMeanRecruitment,sd=abs(LogMeanRecruitment)*StartCV))
write.table(StartLogRec,file="release.pin",sep=" ",append=TRUE,row.names = FALSE,col.names = FALSE)

cat("# Log N0 \n",file="release.pin",append=TRUE)
StartLogN0=matrix(rnorm(length(N[1,1,]),log(rowMeans(N[1,2:ages,])),abs(log(N[1,2:ages,])*StartCV)),nrow=1,ncol=stocks)
write.table(StartLogN0,file="release.pin",sep=" ",append=TRUE,row.names = FALSE,col.names = FALSE)

cat("# Log N0 Devs\n",file="release.pin",append=TRUE)
write.table(matrix(0,ncol=(ages-1),nrow=stocks),file="release.pin",sep=" ",append=TRUE,row.names = FALSE,col.names = FALSE)

cat("# Log Q \n",file="release.pin",append=TRUE)
StartLogQ=t(rnorm(length(q[1,]),mean=log(q[1,]),sd=abs(log(q[1,])*StartCV)))
write.table(StartLogQ,file="release.pin",sep=" ",append=TRUE,row.names = FALSE,col.names = FALSE)

cat("# LogSurveyQ  \n",file="release.pin",append=TRUE)
StartLogSrvyQ=t(rnorm(length(qSurvey),log(qSurvey),abs(log(qSurvey)*StartCV)))
write.table(StartLogSrvyQ,file="release.pin",sep=" ",append=TRUE,row.names = FALSE,col.names = FALSE)

cat("# slctvty \n",file="release.pin",append=TRUE)
StartSlctvty=matrix(rnorm(length(TrueSelectivity),TrueSelectivity,abs(TrueSelectivity*StartCV)),nrow=(ages-1))
StartSlctvty[StartSlctvty <= 0]= 0.001
StartSlctvty[StartSlctvty >= 5]= 4.99
StartSlctvty[is.nan(StartSlctvty)]=1
write.table(StartSlctvty,file="release.pin",sep=" ",append=TRUE,row.names =
```

```
FALSE, col.names = FALSE)

StartSrvySlctvty=matrix(rnorm(length(TrueSurveySel),TrueSurveySel,abs(TrueSurveySel*StartCV)),nrow=(ages-1))
StartSrvySlctvty[StartSrvySlctvty <= 0] = 0.001
StartSrvySlctvty[StartSrvySlctvty >= 5] = 4.99
StartSrvySlctvty[is.nan(StartSrvySlctvty)]=1
cat("# SrvySlctvty   \n",file="release.pin",append=TRUE)
write.table(StartSrvySlctvty,file="release.pin",sep=" ",append=TRUE,row.names = FALSE,col.names = FALSE)

cat("# LogRecruitmentDev1   \n",file="release.pin",append=TRUE)
write.table(matrix(0,nrow=(years-3),ncol=stocks),file="release.pin",sep=" ",append=TRUE,row.names = FALSE,col.names = FALSE)

cat("# LogEffortDev1  \n",file="release.pin",append=TRUE)
write.table(matrix(0,nrow=(years-1),ncol=fisheries),file="release.pin",sep=" ",append=TRUE,row.names = FALSE,col.names = FALSE)

StartLogCatchCV=rnorm(regions,log(logcatchCV$sd),abs(log(logcatchCV$sd)*StartCV))
cat("# LogSigmaCatch   \n",file="release.pin",append=TRUE)
write.table(StartLogCatchCV,file="release.pin",sep=" ",append=TRUE,row.names = FALSE,col.names = FALSE)

cat("# Mvmnt \n",file="release.pin",append=TRUE)
StartMvmnt=matrix(rnorm(length(Movement[,-4]),log(Movement[,-4]/(1-rowSums(Movement[,-4]))),abs(log(Movement[,-4]/(1-rowSums(Movement[,-4])))*StartCV)),nrow=4)
StartMvmnt[StartMvmnt <= -6] = -6
StartMvmnt[StartMvmnt >= 6] = 6
StartMvmnt[is.nan(StartMvmnt)]=rnorm(length(StartMvmnt[is.nan(StartMvmnt)]),0,1)

write.table(StartMvmnt,file="release.pin",sep=" ",append=TRUE,row.names = FALSE,col.names = FALSE)



cat("# RR \n",file="release.pin",append=TRUE)
StartRR=rnorm(length(rr),-log((1/rr)-1),abs(-log((1/rr)-1)*StartCV))
StartRR[StartRR <= -6] = -6
StartRR[StartRR >= 6] = 6
StartRR[is.nan(StartRR)]=6
write.table(t(StartRR),file="release.pin",sep=" ",append=TRUE,row.names = FALSE,col.names = FALSE)

cat("# LogM \n",file="release.pin",append=TRUE)
StartLogM=rnorm(1,log(M[1]),abs(log(M[1])*StartCV))
if (PhaseM<0)
{
    write.table(0,file="release.pin",sep=" ",append=TRUE,row.names = FALSE,col.names = FALSE)
} else{
```

```
    StartLogM=rnorm(1,log(M[1]),abs(log(M[1])*StartCV))
    write.table(StartLogM, file="release.pin",sep=" ",append=TRUE, row.names =
   FALSE, col.names = FALSE)
}

if (RREst==1)
{
    cat("# LogRRDevs \n", file="release.pin",append=TRUE)
    write.table(matrix(0,ncol=fisheries,nrow=1), file="release.pin",sep=" ",
   append=TRUE, row.names = FALSE, col.names = FALSE)
}

if (RREst==2)
{
    cat("# LogRRDevs \n", file="release.pin",append=TRUE)
    write.table(matrix(0,ncol=fisheries,nrow=((years/5))), file="release.pin",
   sep=" ",append=TRUE, row.names = FALSE, col.names = FALSE)
}

if (RREst==3)
{
    cat("# LogRRDevs \n", file="release.pin",append=TRUE)
    write.table(matrix(0,ncol=fisheries,nrow=(years-1)), file="release.pin",sep
   =" ",append=TRUE, row.names = FALSE, col.names = FALSE)
}

if (MEst ==1)
{
    cat("# LogMDevs \n", file="release.pin",append=TRUE)
    write.table(0, file="release.pin",sep=" ",append=TRUE, row.names = FALSE, col
   .names = FALSE)
}

if (MEst ==2)
{
    cat("# LogMDevs \n", file="release.pin",append=TRUE)
    write.table(t(rep(0,(years/5-1))), file="release.pin",sep=" ",append=TRUE,
   row.names = FALSE, col.names = FALSE)
}

if (MEst ==3)
{
    cat("# LogMDevs \n", file="release.pin",append=TRUE)
    write.table(t(rep(0,(years-1))), file="release.pin",sep=" ",append=TRUE, row
   .names = FALSE, col.names = FALSE)
}
```

```
//This is code to compile using ADMB to estimate population dynamics parameters from the simulated dataset using R.

TOP_OF_MAIN_SECTION
  arrmblsize = 1000000000; // use instead of gradient_structure::set_ARRAY_MEMBLOCK_SIZE
  gradient_structure::set_GRADSTACK_BUFFER_SIZE(10000000);
  gradient_structure::set_CMPDIF_BUFFER_SIZE(25000000);

GLOBALS_SECTION
  #include <admodel.h>
  #include <qfclib.h>

  //From Vandergoot walleye movement code
  //This function calculates the movement rate using a parameter for all but the last region and converts to logit scale so the values are between 0 and 1
  dvar_vector LogitProp(const dvar_vector& a)
  {
  int dim;
  dim=a.size()+1;
  dvar_vector p(1,dim);
  dvar_vector expa=exp(a);
  p(1,dim-1)=expa/(1.+sum(expa));
  //p(dim)=1.-sum(p(1,dim-1));
  p(dim)=1./(1.+sum(expa));
  return p;
  }

DATA_SECTION
  //change the name of the file that will contain the simulated data
  !! ad_comm::change_datafile_name("SimulatedData.dat");

  init_int years    //number of years
  init_int regions    //number of regions
  init_int stocks   //number of stocks
  init_int fisheries    //number of fisheries
  init_int ages     //number of ages modeled
    ///Variables that are not read in. creates variables from read in ones
  int yearsp1     //years plus 1
  int yearsm1     //years minus 1
  int yearsm2     //years minus 2
  int yearsby5      //Number of 5 year blocks in time series
  int agesm1      //ages minus 1
  int regionsm1     //Number of regions minus 1

 LOCAL_CALCS
  //Calculate variables to be used to create some parameter vectors
  yearsp1=years+1;
  yearsm1=years-1;
  yearsm2=years-2;
  agesm1=ages-1;
  yearsby5=years/5;
  regionsm1=regions-1;
```

```
END_CALCS

 //More read in data
 init_int PhaseM            //variable whether to estimate M or not. If it is negative do not estimate if positive it is estimated in that phase
 init_int MVaryPhase                 //Variable for if a time varying M is estimated or not. If it is negative do not estimate if positive it is estimated in that phase
 init_number TrueM                    //natural mortality value  if phaseM is positive then you need to transform this starting value so it is on the logistic scale
 init_number MEst              //Variable to determine which type of natural mortality estimation is going to be used 1==constant 2==5 year block 3==random walk
 int Mlength
LOCAL_CALCS
   if(MEst==2){Mlength=yearsby5;
   }else if (MEst ==3){ Mlength=yearsm1;
   }else{ Mlength=1;    }
END_CALCS
 !!if (MVaryPhase>0)
     init_vector TrueTVM(1,years)          //True value for the Time Varying natural mortality only if it is estimated
 init_vector TagLoss(1,stocks)          //Tag loss of tagged fish will be a percentage lost annually each year
 init_number PhaseRR               //Variable for if a reporting rate is estimated or not. If it is negative do not estimate if positive it is estimated in that phase
 init_number RRVaryPhase              //Variable for if a time-varying reporting rate is estimated or not. If it is negative do not estimate if postive it is estimated in that phase
 init_vector rr(1,stocks)              //Initial starting value for the Reporting rate or the value of the parameter if not estimated
 init_number RREst             //Variable to determine which type of reporting rate estimation is going to be used 1==constant 2==5 year block 3==random walk
 int RRlength
LOCAL_CALCS
   if(RREst==2){RRlength=yearsby5;
   }else if (RREst ==3){ RRlength=yearsm1;
   }else{ RRlength=1;    }
END_CALCS
 init_matrix TrueMvmnt(1,regions,1,regions)        //Matrix of the starting values to set for the Mvmnt. On the logit scale and calculates the last regions
 init_matrix FisheryActive(1,fisheries,1,regions)      //Indicator variable for if fisheries are active in a region
 init_matrix ObservedCatch(1,years,1,fisheries)        //Observed total Catch by fisheries
 init_matrix ObservedEffort(1,years,1,fisheries)       //Observed fishing effort by fishery
 init_vector TrueQ(1,fisheries)            //True Fishery Catchability Coefficient parameters
 init_matrix ObservedSurvey(1,years,1,regions)       //Observed Catch Per Unit
```

```
Effort from each region by a survey
init_vector TrueSurveyQ(1,regions)        //True Catchability coefficient for the surveys parameters
init_3darray ObservedAgeComp(1,years,1,fisheries,1,ages) //Observed age composition by fishery
init_vector FisheryFullySelected(1,fisheries)      //The age that is fully selected for each fishery
init_matrix TrueSel(1,agesm1,1,fisheries)      //True selectivity parameter matrix
init_3darray ObservedSurveyAgeComp(1,years,1,regions,1,ages) //Observed Age Composition from the survey for each region
init_vector SurveyFullySelected(1,regions)          //The age that is fully selected to the survey in each region
init_matrix TrueSurveySel(1,agesm1,1,regions)          //True Selectivity Parameters for the survey from each region
init_matrix TrueN0(2,ages,1,stocks)        //True Initial Abundance parameters
init_vector TrueMeanRecruits(1,stocks)       //True Mean Recruitment parameters
init_matrix TrueRecruits(1,years,1,stocks)       //True Annual Recruitment parameters
init_number TrueSigmaCatch          //True Catch Sigma to compare to LogSigmaCatch
init_vector TrueLastYearN(1,stocks)        //True Abundance summed over ages for all stocks
init_number test1           //test value
// test to see if age composition has been read in correctly
!!if (test1 != 1234567890){cout << "Test 1 not read correctly" << endl; exit(10);}
init_4darray TagsReported(1,years,1,stocks,1,years,1,fisheries) //Tags Reported for release year, recapture years, release stock, fishery of recapture
!!if (PhaseRR>0)
    init_vector TrueRR(1,fisheries)

!!if (RRVaryPhase>0)
    init_matrix TrueTVRR(1,years,1,fisheries)

init_number test2            //Test value 2
// test to see if the tag returns have been read in correctly
!!if(test2 != 1234567890){cout << "Test 2 not read correctly" << endl; exit(11);}
init_3darray ReleaseAge(1,years,1,ages,1,stocks) //Number of Tags released by age for calculations
init_matrix TagsReleased(1,years,1,stocks)     //Total number of tags released by year and stock
init_number test3          //Test value 3
// test to see if the tags released have been read in correctly
!!if(test3 != 1234567890){cout << "Test 3 not read correctly" << endl; exit(12);}
init_matrix NeverRecovered(1,years,1,stocks) //Number of tags that are never recovered in simulated data
init_number test4           //Test value 4
  // test to see if the tags never returned have been read in correctly
```

```
  !!if(test4 != 1234567890){cout << "Test 4 not read correctly" << endl; exit
    (13);}

  int y       //indice to keep track of years
  int s       //indice to keep track of stock
  int r       //indice to keep track of region
  int r2      //indice to keep track of second region for movement
    calculations
  int f       //indice to keep track of fishery
  int a       //indice to keep track of age
  int ty      //indice to keep track of tagging year
  int ry      //indice to keep track of recapture year
  vector TagsRetained(1,stocks) //vector of the probability that a tag remains
     on a fish at large i.e. 1-TagLoss
  // !!cout<<"Finished Data Section"<<endl;

PARAMETER_SECTION
  //Parameters to estimate
  init_bounded_vector LogRecruits(1,stocks,5.,25.,1)        //Log of mean
    recruitment for each stock
  init_bounded_vector LogN0_mean(1,stocks,5.,25.,1)
  init_bounded_dev_vector N01(2,ages,-10,10,5)
  init_bounded_dev_vector N02(2,ages,-10,10,5)
  init_bounded_dev_vector N03(2,ages,-10,10,5)
  init_bounded_dev_vector N04(2,ages,-10,10,5)
  init_bounded_vector LogQ(1,fisheries,-20.,-2.,1)        //Catchability
    coefficient for fisheries
  init_bounded_vector LogSurveyQ(1,regions,-20.,-2.,1)        //Catchability
    coefficient for surveys
  init_bounded_matrix slctvty(1,agesm1,1,fisheries,0.,5.,1)     //Selectivity
    parameters without the fully selected age
  init_bounded_matrix SrvySlctvty(1,agesm1,1,regions,0.,5.,1)     //Selectivity
     parameters for the survey without the fully selected ages which varies by
     region
  init_bounded_matrix LogRecruitmentDevs(2,yearsm2,1,stocks,-10.,10.,4) //
    Recruitment deviation vector for stock 1 will be put into matrix for
    calculations
  init_bounded_matrix LogEffortDevs(2,years,1,fisheries,-5.,5.,3)     //
    Catchability Coefficient deviation vector for stock 4 will be put into
    matrix for calculations
  init_bounded_vector LogSigmaCatch(1,regions,-6.,2.,6)         //Log SD for
    catch
  init_bounded_matrix Mvmnt(1,stocks,1,regionsm1,-6.,6.,1)       //Movement
    parameters for all but last region will be converted to logit scale
  init_bounded_vector RR(1,fisheries,-10,10,PhaseRR)          //Reporting Rate
    for each fishery will be converted to logit scale
  init_bounded_number LogM(-10,1,PhaseM)          //Natural Mortality estimated
     value
  init_bounded_matrix LogRRDevs(1,RRlength,1,fisheries,-10,10,RRVaryPhase)  //
    Deviations for annual reporting rate for each year
  init_bounded_vector LogMDevs(1,Mlength,-10,10,MVaryPhase)         //Natural
    Mortality deviation vector to calculate time-varying M
  objective_function_value nll             //Objective negative log likelihood
     value
```

```
//Variables that are calculated from the estimated parameters
matrix Selectivity(1,ages,1,fisheries)          //All Selectivity Parameters for the fisheries
matrix SurveySelectivity(1,ages,1,regions)          //All Selectivity Parameters for the surveys
3darray N(1,yearsp1,1,ages,1,stocks)          //Abundance of individuals by age and stock
4darray NMvmnt(1,years,1,ages,1,stocks,1,regions)      //Abundance of fish after movement and mortality in each region
matrix Movement(1,stocks,1,regions)          //Rate of movement between regions calculated from parameters
matrix Q(1,years,1,fisheries)             //matrix of log catchability deviations
4darray F(1,years,1,fisheries,1,ages,1,regions)      //Fishing mortality calculated from catchability, effort, and selectivity
3darray FTotal(1,years,1,ages,1,regions)         //Total Fishing mortality in a region summing over fisheries
3darray CatchAge(1,years,1,fisheries,1,ages)          //Number of fish caught in year by fisheries and age
matrix TotalCatch(1,years,1,fisheries)          //Total number of fish caught in a year by a fishery
3darray Z(1,years,1,ages,1,regions)          //Total Mortality in a region (Z=F+M)
3darray Survival(1,years,1,ages,1,regions)          //Survival in a region calculated from total mortality
3darray Deaths(1,years,1,ages,1,regions)          //Deaths in a region calculation from 1-survival
4darray Baranov(1,years,1,fisheries,1,ages,1,regions)          //matrix to store calculations of M/Z*(1-Survival) to be used in catch calculation and tag returns
3darray AgeComp(1,years,1,fisheries,1,ages)          //Proportions of age group in catch calculated from CatchAge
3darray SurveyAgeComp(1,years,1,regions,1,ages)      //Proportion of age group caught by each survey
matrix SurveyQMatrix(1,ages,1,regions)          //matrix to be filled with the estimated parameter to be used in survey calculations
3darray SurveyMortality(1,years,1,ages,1,regions)      //Array to store the calculation for the combination of survey selectivity, catchability and mortality in the year up to occurance (October)
matrix TotalSurvey(1,years,1,regions)             //matrix of the predicted survey CPUE for each year and region
3darray SurveyAge(1,years,1,regions,1,ages)          //Survey by age to be used to calculate proportions and totals
number CatchNLL              //negative log likelihood from catch
number EffortNLL               //negative log likelihood from catchability coefficient deviations
number AgeCompNLL              //negative log likelihood from age composition
number SurveyNLL               //negative log likelihood from the surveys
number SurveyAgeCompNLL             //negative log likelihood from the survey age composition
number TagNLL              //negative log likelihood from tagging
// Use variance ratio to calculate LogSigmaEffort in objective function from estimate of LogSigmaCatch
number EffortVarianceRatio              //Variance Ratio of the effort
```

```
  variance compared to the catch variance
  number SurveyVarianceRatio          //Variance Ratio of the survey compared to the catch variance
  vector LogSigmaEffort(1,regions)          //SD of catchability coefficient deviations for likelihood calculations
  vector LogSigmaSurvey(1,regions)          //SD of error in the survey data
  number LogSigmaRec              //SD of error in Recruitment Deviations used to weight likelihood
  number LogSigmaAbun             //SD of error in initial abundance
  number LogSigmaM              //SD of error in Natural Mortality deviations to weight random walk
  number LogSigmaRR             //SD of error in Reporting Rate deviations to weight random walk
  number RecruitmentNLL             //negative log likelihood from recruitment deviations
  number InitAbunNLL              //negative log likelihood for initial adundance deviations
  vector M(1,years)                           //vector for natural mortality
  4darray TagsAlive(1,years,1,years,1,ages,1,stocks)       //Number of Tags alive at the beginning of year  (year of tag release, year of tag recapture/alive, age of fish,stock of fish release)
  matrix TempNMvmnt(1,stocks,1,regions)               //Temporary number to not repeat the calculation of multiplying N and movement
  5darray TagMvmnt(1,years,1,stocks,1,years,1,ages,1,regions)    //Number of Tags that move to each region  (year of tag release, year of tag recapture/alive, age of fish,stock of fish release, region of )
  4darray TagsCaught(1,years,1,stocks,1,years,1,fisheries) //Fate of tagged fish. (year of tag releas, year of tag recapture, age of fish,stock of fish release, First f are captured by fisheries)
  matrix ReportingRate(1,years,1,fisheries)           //The reporting rate for each year and fishery value will be between 0 and 1
  matrix RRtemp(1,years,1,fisheries)            //Temporary matrix to calculate the random walk to convert to Reporting Rate when RREst==3
  4darray TagReturns(1,years,1,stocks,1,years,1,fisheries)     //Tags Returned by year and fishery they are summed over regions and ages
  matrix TotalReturned(1,years,1,stocks)          //Total number of tags returned for each release
  matrix NotReturned(1,years,1,stocks)          //Number of Tags that were never Recovered either not caught, shed or not reported
  vector LastYearN(1,stocks)  //vector of the sum of abundance over ages for the last year for report
  vector zerovec(2,yearsm2);
  vector zerovec2(2,years);
  vector zerovec3(2,ages);
  vector maxSel(1,fisheries);
  vector maxSurveySel(1,regions);
  // !!cout<<"Finished Parameter Section"<<endl;

PRELIMINARY_CALCS_SECTION
  //Set the starting values for various parameters
  if (PhaseM<0){
    M=TrueM;
  }
  if(PhaseRR<0){
```

```
      for (y=1;y<=years;y++)
          ReportingRate[y]=rr;
  }
  EffortVarianceRatio=1.;
  SurveyVarianceRatio=0.5;
  LogSigmaRec=log(4.0);
  LogSigmaAbun=log(4.0);
  LogSigmaRR=log(2);
  LogSigmaM=log(2);
  TagsRetained=1.-TagLoss;
  // cout<<"Finished Preliminary Calcs"<<endl;

PROCEDURE_SECTION
  CalculateParameters();
  CalculateFZ();
  CalculateN();
  CalculateTagReturns();
  CalculateObjectiveFunction();

FUNCTION CalculateParameters
  //Initialize the parameters that will be calculated by this function
  //Use logit function to calculate what the movement proportions will be
  Movement.initialize();
  for (s=1;s<=stocks;s++)
  {
      Movement(s)=LogitProp(Mvmnt(s));
  }
  //insert the parameter estimates into the correct location in the
    selectivity matrices using the known fully selected age
  for (a=1;a<=ages;a++)
  {
      for (r=1;r<=regions;r++)
      {
          if (a<SurveyFullySelected[r])
          {
        SurveySelectivity(a,r)=SrvySlctvty(a,r);
          }
          else if (a==SurveyFullySelected[r])
          {
              SurveySelectivity(a,r)=1;
          }
          else
          {
        SurveySelectivity(a,r)=SrvySlctvty((a-1),r);
          }
      }
      for (f=1;f<=fisheries;f++)
      {
    if (a<FisheryFullySelected[f])
          {
        Selectivity(a,f)=slctvty(a,f);
          }
          else if (a==FisheryFullySelected[f])
          {
```

```
                Selectivity(a,f)=1;
            }
            else
    {
        Selectivity(a,f)=slctvty((a-1),f);
    }

      }
  }
  if (PhaseRR>0 || RRVaryPhase>0)
  {        //If Reporting Rate is estimated
      if (RREst==1)
      {        //Reporting Rate is estimated but not time-varying
          for (y=1;y<=years;y++)
              ReportingRate[y]=1./(1.+exp(-RR));
      } else if (RREst==2)
      {        //Reporting Rate is estimated in 5 year blocks
          for (y=1;y<=yearsby5;y++)
      {
              for (int temps=1;temps<=5;temps++)
          {
              ReportingRate[(y-1)*5+temps]=1./(1.+exp(-(LogRRDevs[y])));

          }
      }

      } else if (RREst==3)
      {        //If Reporting Rate is estimated time-varying as a random walk
          ReportingRate[1]=1./(1.+exp(-RR));
    RRtemp[1]=RR;
          for (y=1;y<=yearsm1;y++)
        RRtemp[y+1]=RRtemp[y]+LogRRDevs[y];
              ReportingRate[y+1]=1./(1.+exp(-RRtemp[y+1]));
      } else
      {
    cout<<"You must specify RREst equal to 1, 2 or 3"<<endl;
    exit(21);
      }
  }
  //If not estimated is already done in preliminary calcs and does not change
  if (PhaseM>0 || MVaryPhase>0)
  {        //Natural Mortality estimated
      if (MEst==1)
      {
          M=exp(LogM);   //Natural mortality is estimated constant
      }
      else if (MEst==2)
      {        //Natural Mortality is estimated in 5 year blocks
          for (y=1;y<=yearsby5;y++)
      {
              for (int temps=1;temps<=5;temps++)
          {
              M[(y-1)*5+temps]=exp(LogMDevs[y]);

```

```
            }
        }
        }
        else if (MEst==3)
        {       //Natural mortality is estimated as a Random walk
            M[1]=exp(LogM);
            for (y=1;y<=yearsm1;y++)
     {
          M[y+1]=M[y]+exp(LogMDevs(y));

     }
        } else
        {
     cout<<"You must specify MEst equal to 1, 2 or 3"<<endl;
     exit(31);
        }
   }
   //If not estimated is already done in preliminary calcs and does not change
   // Fill in the Survey Q matrix to allow for elementwise calculations
   for (a=1;a<=ages;a++)
   {
        SurveyQMatrix[a]=mfexp(LogSurveyQ);
   }
   Q[1]=exp(LogQ);
   for (y=2;y<=years;y++)
   {
        Q[y]=elem_prod(Q[y-1],exp(LogEffortDevs[y]));
   }
   // cout<<"Finished Calculate Parameters"<<endl;

FUNCTION CalculateFZ
   FTotal.initialize(); F.initialize(); Z.initialize(); Survival.initialize();
   for (y=1;y<=years;y++)
   {       //Begin year loop
        for (a=1;a<=ages;a++)
        {       //Begin age loop
            for (f=1;f<=fisheries;f++)
     {    //Begin fisheries loop
          //Calculate fishery mortality from parameters
          F[y][f][a]=Q(y,f)*ObservedEffort(y,f)*Selectivity(a,f)*FisheryActive[f
     ];
          for (r=1;r<=regions;r++)
            {      //Begin region loop
        //Calculate total fishing mortality by summing over fisheries
        FTotal(y,a,r)+=F(y,f,a,r);
          }    //End regions loop
            }    //End fishery loop
        }      //End ages loop
        //Calculate Total mortality
        Z[y]=FTotal[y]+M[y];
   }       //End year loop
   // Calculate Survival
   Survival = mfexp(-1.0*Z);
   Deaths = 1-Survival;
```

```
  for (y=1;y<=years;y++)
  {       //Begin year loop
      for (f=1;f<=fisheries;f++)
      {
      // Calculate F/Z *(1-Survival) to be used for catch at age and tagging
          Baranov[y][f]= elem_prod(elem_div(F[y][f],Z[y]),Deaths[y]);
      }
      // Calculate the mortality, catchability and selectivity that occur for
    each survey assume it occurs in october so 10/12 is approximately 0.833333
      SurveyMortality[y]= elem_prod(elem_prod(mfexp(-0.8333333333*Z[y]),
    SurveySelectivity),SurveyQMatrix);
  }
  // cout<<"Finished FZ"<<endl;

FUNCTION CalculateN
  //Initialize variables used in this section
  N.initialize(); NMvmnt.initialize(); CatchAge.initialize(); TotalCatch.
    initialize(); AgeComp.initialize(); TotalSurvey.initialize(); SurveyAge.
    initialize(); TempNMvmnt.initialize();
  //Initialize abundance calculated from estimated parameters
  for (a=2;a<=ages;a++)
  {
      N[1][a][1]=exp(LogN0_mean(1)+N01(a));
      N[1][a][2]=exp(LogN0_mean(2)+N02(a));
      N[1][a][3]=exp(LogN0_mean(3)+N03(a));
      N[1][a][4]=exp(LogN0_mean(4)+N04(a));
  }
  N[1][1]=exp(LogRecruits);
  for (y=2;y<=(yearsm2);y++)
  {
      N[y][1]=elem_prod(N[y-1][1],exp(LogRecruitmentDevs[y]));
  }
  // Recruitment of last 2 years is equal to average of 3 previous years
  N[years-1][1]=(N[years-2][1]+N[years-3][1]+N[years-4][1])/3.0;
  N[years][1]=(N[years-2][1]+N[years-3][1]+N[years-4][1])/3.0;
  for (y=1;y<=years;y++)
  {       //Begin year loop
      for (a=1;a<=ages;a++)
      {       //Begin age loop
          for (s=1;s<=stocks;s++)
          {     //Begin stock loop
        //Calculate a row vector of fish that move to all the regions from one
     stock
              TempNMvmnt[s]=N(y,a,s)*Movement[s];
              //Calculate the area specific mortality for the fish in each
    region
              NMvmnt[y][a][s]=elem_prod(TempNMvmnt[s],Survival[y][a]);
              for (f=1;f<=fisheries;f++)
              {   //Begin fishery loop
            //Calculate the catch for each area summing over the different
    spawning stocks
                  CatchAge(y,f,a)+=sum(elem_prod(Baranov[y][f][a],TempNMvmnt[s
    ]));
        }   //End fishery loop
```

```
              //Calculate the Abundance at the next time step by summing
   survival over regions. Assumed a plus group calculation
        if((a<ages))
        {
            N((y+1),(a+1),s)=sum(NMvmnt[y][a][s]);
        }
        else{
                  N((y+1),ages,s)+=sum(NMvmnt[y][ages][s]);
              }      //End if/else ages
          }      //End stock loop
          for (r=1;r<=regions;r++)
          {      //Begin region loop
              SurveyAge[y][r][a]=sum(column(TempNMvmnt,r)*SurveyMortality(y,a,
   r));
    }      //End region loop
      }        //End ages loop
      for(f=1;f<=fisheries;f++)
      {        //Begin fisheries loop
          // Calculate the Total Catch and proportion in each age class in the
    catch
          TotalCatch(y,f)=sum(CatchAge[y][f]);
          AgeComp[y][f]=CatchAge[y][f]/TotalCatch(y,f);
      }        //End fisheries loop
      for(r=1;r<=regions;r++)
      {        //Begin region loop
          // Calculate the total Survey and the proportion in each age class
   of the fish caught
          TotalSurvey(y,r)=sum(SurveyAge[y][r]);
          SurveyAgeComp[y][r]=SurveyAge[y][r]/TotalSurvey(y,r);
      }        //End region loop
  }        //End year loop
  // cout<<"Finished N"<<endl;

FUNCTION CalculateTagReturns
 // This keeps track of releases by age, year and region of release for one
   release event and then which ones are recovered
  TagsAlive.initialize(); TagMvmnt.initialize();  TotalReturned.initialize();
   TagReturns.initialize(); NotReturned.initialize();
 for (ty=1;ty<(years-ages);ty++)
  {        //Loop over tag release years
  //Don't loop over the last ages of years so not exceeding the bounds of  the
    arrays. Will run another loop for the remaining years
      // Initialize the Tags Alive as the number of tags released
      TagsAlive[ty][ty]=ReleaseAge[ty];
      for (s=1;s<=stocks;s++)
      {        //Loop over stock of release
    for (ry=ty;ry<(ty+ages);ry++)
    {      //Loop over recapture years 1 starting from tag year and going only
   to the age where all ages are in the plus group so don't need to do all of
    these calculations. Will run another loop for just the plus group
        for (a=1;a<=ages;a++)
        {      //Loop over Ages
                    //Calculate the tags that move to each region after applying
    a tag shedding rate
```

```
            TagMvmnt[ty][s][ry][a]=TagsAlive(ty,ry,a,s)*(TagsRetained(s))*
  Movement[s];
                      if (a<ages)
                      {     //Begin If loop for ages
                          //Calculate the fish that are alive at the beginning
   of the next year
       TagsAlive(ty,(ry+1),(a+1),s)=sum(elem_prod(TagMvmnt[ty][s][ry][a],
  Survival[ry][a]));
                      }
                      else
                      { //Continue If statement
                           //Calculate the fish that are alive at the
  beginning of the next year in the plus group
                          TagsAlive(ty,(ry+1),ages,s)+=sum(elem_prod(TagMvmnt[
  ty][s][ry][ages],Survival[ry][ages]));
                      } //End If statement for ages plus group
              }     //End loop over ages
       for (f=1;f<=fisheries;f++)
       { //Start loop over fisheries
                  // Calculate the number of tags caught in each region for
  each fishery
     TagsCaught(ty,s,ry,f)=sum(elem_prod(TagMvmnt[ty][s][ry],Baranov[ry][f]))
  ;
       }     //End Loop over fisheries
   }     //End Loop over recapture years 1
        for (ry=(ty+ages);ry<=years;ry++)
   {     //Begin loop over recapture years 2 to loop over the years that just
   have tags in the plus group
      // Calculate the fish that move to each region after applying the tag
  shedding rate
      TagMvmnt[ty][s][ry][ages]=TagsAlive(ty,ry,ages,s)*(TagsRetained(s))*
  Movement[s];
      if (ry<years)
          // Calculate the tags alive at the beginning of the next years
  just for the plus group
          TagsAlive(ty,(ry+1),ages,s)+=sum(elem_prod(TagMvmnt[ty][s][ry][
  ages],Survival[ry][ages]));
      for (f=1;f<=fisheries;f++)
      { //Start loop over fisheries
                 //Calculate the tags that are caught in each region for just
   the plus group
     TagsCaught(ty,s,ry,f)=sum(elem_prod(TagMvmnt[ty][s][ry][ages],Baranov[ry
  ][f][ages]));
      }   //End Loop over fisheries
   }   //End loop over recapture years 2
   // Calculate the Tags that are returned and the total tags returned
   TagReturns[ty][s]=elem_prod(TagsCaught[ty][s],ReportingRate);
   TotalReturned(ty,s)=sum(elem_prod(TagsCaught[ty][s],ReportingRate));
     }       //End Loop over stock of release
 }       //End Loop over tagging years
 for(ty=(years-ages);ty<=years;ty++)
 {      //Loop over the last years to make sure that the array bounds are not
   exceeded
    // Initialize the Tags Alive as the number of tags released
```

```
        TagsAlive[ty][ty]=ReleaseAge[ty];
        for (s=1;s<=stocks;s++)
        {       //Loop over stock of release
            for (ry=ty;ry<=years;ry++)
     {      //Loop over recapture years starting from tag year
          for (a=1;a<=ages;a++)  //try getting rid of if statement
          {      //Loop over Ages
                  //Calculate the tags that move to each region after applying a tag shedding rate
             TagMvmnt[ty][s][ry][a]=TagsAlive(ty,ry,a,s)*(TagsRetained(s))*Movement[s];
        if (ry<years)
        {   //Begin If loop for recapture year
                        if (a<ages)
                        {      //Begin If loop for ages
                            //Calculate the fish that are alive at the beginning of the next year
                            TagsAlive(ty,(ry+1),(a+1),s)=sum(elem_prod(TagMvmnt[ty][s][ry][a],Survival[ry][a]));
                        }
                        else
                        { //Continue If statement
                            //Calculate the fish that are alive at the beginning of the next year in the plus group
                            TagsAlive(ty,(ry+1),ages,s)+=sum(elem_prod(TagMvmnt[ty][s][ry][ages],Survival[ry][ages]));
                        } //End If statement for ages plus group
        } //End If statement for recapture year
                }      //End loop over ages
          for (f=1;f<=fisheries;f++)
          { //Start loop over fisheries
              // Calculate the fish that are caught by each fishery
              TagsCaught(ty,s,ry,f)=sum(elem_prod(TagMvmnt[ty][s][ry],Baranov[ry][f]));
          }      //End loop over fisheries
            }      //End loop over recapture year
     // Calculate the Tags that are reported and the total tags returned
            TagReturns[ty][s]=elem_prod(TagsCaught[ty][s],ReportingRate);
     TotalReturned(ty,s)=sum(elem_prod(TagsCaught[ty][s],ReportingRate));
        }       //End loop over stock of release
  }             //End loop over tagging year
  NotReturned=TagsReleased-TotalReturned;
  // cout<<"Finished Calculate Tag Returns"<<endl;

FUNCTION CalculateObjectiveFunction
  CatchNLL.initialize(); EffortNLL.initialize(); AgeCompNLL.initialize(); nll.initialize(); TagNLL.initialize(); SurveyNLL.initialize(); SurveyAgeCompNLL.initialize(); InitAbunNLL.initialize();
  double myeps=1.e-60;
  double EPS=1.e-60;
  if (current_phase() ==1)   myeps=1.e-8;
  //Calculate Sigma associated with the Effort data and Survey data
  LogSigmaEffort=log(sqrt((1./EffortVarianceRatio)*square(mfexp(LogSigmaCatch))));
```

```
LogSigmaSurvey=log(sqrt((1./SurveyVarianceRatio)*square(mfexp(LogSigmaCatch))));
//Calculate the negative log likelihood for the total Catch
CatchNLL=nllNormal(log(column(ObservedCatch,1)),log(column(TotalCatch,1)),exp(LogSigmaCatch(1)));
CatchNLL+=nllNormal(log(column(ObservedCatch,2)),log(column(TotalCatch,2)),exp(LogSigmaCatch(2)));
CatchNLL+=nllNormal(log(column(ObservedCatch,3)),log(column(TotalCatch,3)),exp(LogSigmaCatch(3)));
CatchNLL+=nllNormal(log(column(ObservedCatch,4)),log(column(TotalCatch,4)),exp(LogSigmaCatch(4)));
//Calculate the negative log likelihood for the Survey
SurveyNLL=nllNormal(log(column(ObservedSurvey,1)),log(column(TotalSurvey,1)),exp(LogSigmaSurvey(1)));
SurveyNLL+=nllNormal(log(column(ObservedSurvey,2)),log(column(TotalSurvey,2)),exp(LogSigmaSurvey(2)));
SurveyNLL+=nllNormal(log(column(ObservedSurvey,3)),log(column(TotalSurvey,3)),exp(LogSigmaSurvey(3)));
SurveyNLL+=nllNormal(log(column(ObservedSurvey,4)),log(column(TotalSurvey,4)),exp(LogSigmaSurvey(4)));
//Calculate the negative log likelihood associated with the age composition
AgeCompNLL=-sum(150.*elem_prod(ObservedAgeComp,log(AgeComp+myeps)));
//Calculate negative log likelihood associated with the survey age composition
SurveyAgeCompNLL=-sum(150.*elem_prod(ObservedSurveyAgeComp,log(SurveyAgeComp+myeps)));
//Calculate negative log likelihood associated with Effort Deviations
EffortNLL=nllNormal(column(LogEffortDevs,1),zerovec2,exp(LogSigmaEffort(1)));
EffortNLL+=nllNormal(column(LogEffortDevs,2),zerovec2,exp(LogSigmaEffort(2)));
EffortNLL+=nllNormal(column(LogEffortDevs,3),zerovec2,exp(LogSigmaEffort(3)));
EffortNLL+=nllNormal(column(LogEffortDevs,4),zerovec2,exp(LogSigmaEffort(4)));
// Calculate the negative log likelihood associated with the tag returns
for (ty=1;ty<=years;ty++)
{        //Begin loop over tag years
    for (s=1;s<=stocks;s++)
    {        //Begin loop over stocks
        TagNLL-=sum(elem_prod(log((TagReturns[ty][s]+myeps)/(TagsReleased(ty,s))),TagsReported[ty][s]));
    }        //End loop over stocks
}        //End loop over tag years
TagNLL-=sum(elem_prod(log(elem_div(NotReturned+myeps,TagsReleased)),NeverRecovered));
//Add in a recruitment penalty to help make the model converge
RecruitmentNLL=nllNormal(column(LogRecruitmentDevs,1),zerovec,exp(LogSigmaRec));
RecruitmentNLL+=nllNormal(column(LogRecruitmentDevs,2),zerovec,exp(LogSigmaRec));
RecruitmentNLL+=nllNormal(column(LogRecruitmentDevs,3),zerovec,exp(LogSigmaRec));
RecruitmentNLL+=nllNormal(column(LogRecruitmentDevs,4),zerovec,exp(
```

```
  LogSigmaRec));
  //Calculated Process Error associated with Initial Abundance
  InitAbunNLL=nllNormal(N01,zerovec3,exp(LogSigmaAbun));
  InitAbunNLL+=nllNormal(N02,zerovec3,exp(LogSigmaAbun));
  InitAbunNLL+=nllNormal(N03,zerovec3,exp(LogSigmaAbun));
  InitAbunNLL+=nllNormal(N04,zerovec3,exp(LogSigmaAbun));

  //Calculate Negative Log Likelihood
  nll=CatchNLL+EffortNLL+AgeCompNLL+TagNLL+SurveyNLL+SurveyAgeCompNLL+
   RecruitmentNLL+InitAbunNLL;
  //Add a likelihood term for the random walk of natural mortality if MEst==3
  if (MEst==3)
  {
      nll+=(LogSigmaM*size_count(LogMDevs))+(1./2.*square(mfexp(LogSigmaM))*
   norm2(LogMDevs));
  }
  //Add a likelihood term for the random walk of Reporting Rate if RREst==3
  if (RREst==3)
  {
      nll+=(LogSigmaRR*size_count(LogRRDevs))+(1./2.*square(mfexp(LogSigmaRR))
   *norm2(LogRRDevs));
  }

RUNTIME_SECTION
  convergence_criteria 1.e-1,1.e-2,5.e-3
  maximum_function_evaluations 5000,10000,15000,25000,50000

REPORT_SECTION
  ofstream myreport ("release.txt");
  myreport<<objective_function_value::pobjfun->gmax<<endl;
  myreport<< "#Initial Abundance" <<endl;
  myreport<< N[1] <<endl;
  myreport<< "#True Initial Abundance" <<endl;
  myreport<< TrueN0 <<endl;
  myreport<< "#Initial Abundance Relative Error"  <<endl;
  for (a=2;a<=ages;a++)
      myreport<< elem_div((N[1][a]-TrueN0[a]),TrueN0[a])*100 <<endl;

  myreport<< "#Mean Recruitment" <<endl;
  myreport<< LogRecruits <<endl;
  myreport<< "#True Mean Recruitment" <<endl;
  myreport<< TrueMeanRecruits <<endl;
  myreport<< "#Mean Recruitment Relative Error" <<endl;
  myreport<< elem_div((LogRecruits-TrueMeanRecruits),TrueMeanRecruits)*100 <<
   endl;

  myreport<< "#Recruitment Estimate"  <<endl;
  for (y=1;y<=years;y++)
      myreport<< N[y][1] <<endl;
  myreport<< "#Recruitment True" <<endl;
  myreport<< TrueRecruits <<endl;
  myreport<< "#Recruits Relative Error"  <<endl;
  for (y=1;y<=years-2;y++)
     myreport<<elem_div((N[y][1] - TrueRecruits[y]),TrueRecruits[y])*100 <<endl;
```

```
myreport<< "#Catchability Coefficient" <<endl;
myreport<< Q <<endl;
myreport<< "#Catchability True" <<endl;
myreport<< TrueQ <<endl;
myreport<< "#Catchability Relative Error" <<endl;
for (f=1;f<=fisheries;f++)
{
    maxSel[f]=max(column(Selectivity,f));
}
for (y=1;y<=years;y++)
{
    myreport<< elem_div((elem_prod(Q[y],maxSel)-TrueQ),TrueQ)*100 <<endl;
}

myreport << "#Survey Catchability Coefficient" << endl;
myreport << mfexp(LogSurveyQ) << endl;
myreport << "#Survey Catchability True" << endl;
myreport << TrueSurveyQ << endl;
myreport << "#Survey Catchability Relative Error" << endl;
for (r=1;r<=regions;r++)
{
    maxSurveySel[r]=max(column(SurveySelectivity,r));
}
myreport << elem_div((elem_prod(mfexp(LogSurveyQ),maxSurveySel)-TrueSurveyQ)
  ,TrueSurveyQ)*100 << endl;

myreport<< "#Estimated Selectivity Matrix" <<endl;
myreport<< slctvty <<endl;
myreport<< "#Selectivity True" <<endl;
myreport<< TrueSel <<endl;
myreport<< "#Maximum Selectivity" << endl;
myreport<< maxSel << endl;
myreport<< "#Selectivity Relative Error" <<endl;
myreport<< elem_div((slctvty-TrueSel),TrueSel)*100 <<endl;
myreport<< "#Adjusted Selectivity Relative Error" << endl;
for (a=1;a<ages;a++)
{
    myreport << elem_div((elem_div(slctvty[a],maxSel)-TrueSel[a]),TrueSel[a
  ])*100 <<endl;
}
myreport<< ((1/maxSel)-1)/1*100 <<endl;

myreport << "#Estimated Survey Selectivity Matrix" << endl;
myreport << SrvySlctvty << endl;
myreport << "#Survey Selectivity True" << endl;
myreport << TrueSurveySel << endl;
myreport << "#Maxiumum Survey Selectivity" << endl;
myreport << maxSurveySel << endl;
myreport << "#Survey Selectivity Relative Error" << endl;
myreport << elem_div((SrvySlctvty-TrueSurveySel),TrueSurveySel)*100 << endl;
myreport << "#Adjusted Survey Selectivity Relative Error" << endl;
for (a=1;a<ages;a++)
{
```

```
    myreport << elem_div((elem_div(SrvySlctvty[a],maxSurveySel)-
  TrueSurveySel[a]),TrueSurveySel[a])*100 <<endl;
  }
  myreport << ((1/maxSurveySel)-1)/1*100 <<endl;

  myreport<< "#Movement Matrix"  <<endl;
  myreport<< Movement <<endl;
  myreport<< "#Movement True" <<endl;
  myreport<< TrueMvmnt <<endl;
  myreport<< "#Movement Relative Error" <<endl;
  myreport<< elem_div((Movement-TrueMvmnt),TrueMvmnt)*100 <<endl;

  myreport << "#Log Sigma Catch" <<endl;
  myreport << LogSigmaCatch << endl;
  myreport << "#SigmaCatch Relative Error assuming 0.1" << endl;
  myreport << (exp(LogSigmaCatch)-TrueSigmaCatch)/TrueSigmaCatch*100 << endl;

  for (a=1;a<=ages;a++)
      LastYearN += N[years][a];
  myreport<< "#Last Years' Abundance summed over ages" <<endl;
  myreport<< LastYearN <<endl;
  myreport<< "#Last Years' Abundance True" <<endl;
  myreport<< TrueLastYearN <<endl;
  myreport<< "#Last Years' Abundance Error" << endl;
  myreport<< elem_div((LastYearN-TrueLastYearN),TrueLastYearN)*100 <<endl;

  if (PhaseRR>0)
  {
      myreport<< "#Area Reporting Rate" <<endl;
      myreport<< RR <<endl;
      myreport<< "#Reporting Rate True" <<endl;
      myreport<< TrueRR <<endl;
      myreport<< "#Reporting Rate Relative Error" <<endl;
      myreport<< elem_div((RR-TrueRR),TrueRR)*100 <<endl;
  }

  if (RRVaryPhase>0)
  {
      myreport<< "#Time Varying Reporting Rate" <<endl;
      myreport<< ReportingRate <<endl;
      myreport<< "#Time Varying Reporting Rate True" <<endl;
      myreport<< TrueTVRR  <<endl;
      myreport<< "#Time Varying Reporting Rate Relative Error" <<endl;
      myreport<< elem_div((ReportingRate-TrueTVRR),TrueTVRR)*100 <<endl;
  }

  if (PhaseM>0)
  {
      myreport<< "#Natural Mortality" <<endl;
      myreport<< exp(LogM) <<endl;
      myreport<< "#Natural Mortality True" <<endl;
      myreport<< TrueM  <<endl;
      myreport<< "#Natural Mortality Relative Error" <<endl;
      myreport<< ((exp(LogM)-TrueM)/TrueM)*100 <<endl;
```

```
}

if (MVaryPhase>0)
{
    myreport<< "#Time Varying Natural Mortality" <<endl;
    myreport<< M <<endl;
    myreport<< "#Time-Varying Natural Mortality True" <<endl;
    myreport<< TrueTVM <<endl;
    myreport<< "#Time Varying Natural Mortality Relative Error" <<endl;
    myreport<< elem_div((M-TrueTVM),TrueTVM)*100 <<endl;
}

myreport.close();
```