SimSurvey: an R package for simulation testing the design and analysis of fisheries-independent trawl surveys

Paul M. Regular, Fran Mowbray, et al.?

Fisheries and Oceans Canada, Northwest Atlantic Fisheries Center, 80 East White Hills, St. John’s, Newfoundland and Labrador, A1C 5X1, Canada

# Summary

**1.** Fisheries scientists are often faced with two critical questions: (i) how many samples are needed to adequately represent a population? And, (ii) how should samples from multi-stage sampling be analyzed?

**2.** For assessments based on fisheries-independent surveys, our ability to estimate population characteristics, such as abundance at age, is affected by multiple levels of sampling (i.e. the number of sets, lengths and ages sampled in a survey) as well as the methods used to analyze the data.

**3.** Though simulations are a useful tool for exploring the efficacy of specific sampling strategies and statistical methods, system and survey complexity have limited the number of frameworks that simulation test the full sampling and analytical pathway of fisheries-independent survey data.

**4.** Here we introduce the R package **SimSurvey**, which has been designed to simplify the process of simulating surveys of spatial fish populations. The packages includes methods for simulating age-structured populations that varies in space and time as well as multi-stage surveys that sample the virtual population. A series of parameters can be used to control the characteristics of the population and the sampling intensity of the survey. **SimSurvey** also include a function for running a stratified-random analysis of the simulated data.

**5.** **SimSurvey** can serve as a convenient, accessible and flexible platform for simulating a wide range of sampling strategies of a range of species. Various design- or model-based approaches can be applied to the results to test the efficacy of different approaches.

**Key-words:** length and age distribution; simulation; spatial correlation; stratified analysis; survey design

# Introduction

It is well known that the accuracy and reliability of results from any population monitoring program hinges on good survey design and analysis (Thomas *et al.*, 2010). Ideally, the design and analysis of large and costly survey should be simulation tested before embarking on expensive field programs. This is especially true for the large-scale and costly fisheries-independent trawl surveys conducted by numerous regional fisheries management organizations. These surveys have become a mainstay in the management of dynamic fish stocks as they provide indices of population abundance as well as estimates of various population characteristics such as length and age frequencies (Pennington and Strømme, 1998). This information forms the basis of many stock assessments throughout the world. Efficient, effective and informed management decisions therefore require surveys and analyses that maximize information while minimizing the expense of data collection. While simulations provide a platform for exploring solutions to this optimization problem, building the necessary simulation framework is not a trivial task given the multi-stage nature of the typical sampling program and the complexity of the population processes we aim to represent. The complexity of this problem is perhaps why simulations that test the full sampling and analytical pathway of fisheries-independent surveys are rare (but see Puerta *et al.*, 2019; Schnute and Haigh, 2003).

Here we document **SimSurvey**, an R package designed to facilitate realistic simulations of fisheries-independent trawl surveys. This simulation framework has similarities to those presented by Schnute and Haigh (2003) and Puerta et al. (2019), however, efforts were focused on developing a series of general and accessible functions to simplify the process of testing multiple sampling scenarios and analytical pathways. The steps taken to simulate surveys of spatial, age-structure populations are outlined below, with specific emphasis of a case study that motivated this work.

# The SimSurvey package

The **SimSurvey** package was written in the programming language R (R Core Team, 2017) and, in short, the package allows for the simulation of random or stratified-random surveys of an age-structured population that varies in space and time. **SimSurvey** relies heavily on functions from the **data.table** (Dowle and Srinivasan, 2017), **raster** (Hijmans, 2016) and **plotly** (Sievert, 2018) packages for their efficient data processing, geographic and plotting facilities, respectively. All the source R code behind **SimSurvey** is available on GitHub (<https://github.com/PaulRegular/SimSurvey>) and the package can be installed via GitHub:

install.packages("devtools")  
devtools::install\_github("PaulRegular/SimSurvey")

## Case study

The construction of the **SimSurvey** package was motivated by a regional need for assessing the sampling design of trawl surveys conducted along the Newfoundland and Labrador shelf by Fisheries and Oceans Canada. As a tangible first step, we focused our efforts on emulating data from the survey of cod in NAFO division 3Ps since it is one of the most dynamic and heavily sampled stocks in the region. Below we outline the equations behind a series of **SimSurvey** functions that were used to simulate survey data with similar features to data obtained by the survey of 3Ps cod. The settings were iteratively modified until it was difficult to distinguish the simulated data from real survey data; these values were set as the defaults for the main functions in the package. Here we evaluate the efficacy of alternate sampling protocol by assessing deviation between stratified estimates of abundance (Smith and Somerton, 1981) from the true abundance available to the survey. While assessing model-based analyses was an option, we focused on the design-based stratified analysis for simplicity and because of its widespread use. See Table 1 for a description of the code used to generate the case study results. Text below provides more detail on each function and also demonstrates a series of functions for visualizing the results. See Appendix 1 for an interpretation and discussion of the case study results.

Table 1 - Main function calls, with argument descriptions and associated parameter symbols, used to simulate the cod-like population used in the case study presented here. Note that the settings below largely correspond with the defaults used in the functions.

|  |  |  |
| --- | --- | --- |
| **Function call** | **Description** | **Symbol** |
| > set.seed(438) | Set random number generator seed to replicate results |  |
| > |  |  |
| > abundance <- sim\_abundance( | *Simulate abundance* |  |
| + ages = 1:20, | Ages |  |
| + years = 1:20, | Years |  |
| + R = sim\_R(mean = 30000000, | Mean recruitment |  |
| + log\_sd = 0.5), | Standard deviation of log-recruitment |  |
| + Z = sim\_Z(mean = 0.5, | Mean total mortality |  |
| + log\_sd = 0.2, | Standard deviation of total mortality |  |
| + phi\_age = 0.9, | Correlation across ages in error around total mortality |  |
| + phi\_year = 0.5), | Correlation across years in error around total mortality |  |
| + growth = sim\_vonB(Linf = 120, | Mean asymptotic length (cm) |  |
| + L0 = 5, | Length at birth (cm) |  |
| + K = 0.1, | Growth rate |  |
| + log\_sd = 0.1, | Standard deviation of the von Bertalanffy growth curve |  |
| + length\_group = 3)) | Length group for abundance at length (cm) |  |
| > |  |  |
| > grid <- make\_grid( | *Make a grid* |  |
| + x\_range = c(-140, 140), | Range of grid in the x dimension |  |
| + y\_range = c(-140, 140), | Range of grid in the y dimension |  |
| + res = c(3.5, 3.5), | Grid resolution (km) |  |
| + shelf\_depth = 200, | Shelf depth (km) |  |
| + shelf\_width = 100, | Shelf width (km) |  |
| + depth\_range = c(0, 1000), | Depth range (m) |  |
| + n\_div = 1, | Number of divisions |  |
| + strat\_breaks = seq(0, 1000, by = 40)) | Series of depth breaks for defining strata |  |
| > |  |  |
| > distribution <- sim\_distribution( | *Simulate distribution* |  |
| + sim = abundance, | Object produced by sim\_abundance function |  |
| + grid = grid, | Object produced by make\_grid function |  |
| + ays\_covar = sim\_ays\_covar(sd = 2.8, | Standard deviation of age-year-space distribution |  |
| + range = 300, | Range of spatial correlation (km) |  |
| + phi\_age = 0.5, | Correlation across ages in spatial distribution |  |
| + phi\_year = 0.9, | Correlation across years in spatial distribution |  |
| + group\_ages = 5:20), | Make space-age-year variance equal across these ages |  |
| + depth\_par = sim\_parabola(mu = 200, | Depth at which abundance is typically highest (m) |  |
| + sigma = 70)) | Dispersion around depth of peak abundance (m) |  |
| > |  |  |
| > surveys <- expand\_surveys( | *Define a series of surveys to test* |  |
| + set\_den = c(0.0005, 0.001, 0.002, 0.005, 0.01), | Set density (km-2) |  |
| + lengths\_cap = c(5, 10, 20, 50, 100, 500, 1000), | Maximum number of lengths to collect / set |  |
| + ages\_cap = c(2, 5, 10, 20, 50)) | Maximum number of ages to sample / length group / division |  |
| > |  |  |
| > tests <- test\_surveys( | *Simulate and analyze surveys* |  |
| + sim = distribution, | Object produced by sim\_distribution function |  |
| + surveys = surveys, | Object produced by expand\_surveys function |  |
| + n\_loops = 1000, | Number of times to simulate each survey |  |
| + q = sim\_logistic(k = 2, | Steepness of logistic curve of catchability |  |
| + x0 = 3), | Midpoint of logistic curve of catchability |  |
| + trawl\_dim = c(1.5, 0.02), | Trawl dimensions (km) - i.e. area covered by a trawl |  |
| + length\_group = 1) | Length group for age sampling (cm) |  |

*Caution:* as written, the test\_surveys function call may take days or weeks to run. Run duration can be decreased by running multiple simulations per loop (n\_sims argument) and by running the loops in parallel (cores argument). See help file for more details (?test\_surveys).

## Simulate abundance

The simulation starts with a common cohort model where the abundance at age in year () equals the abundance of that cohort in the previous year multiplied by the survival rate, which is expressed in terms of total mortality ():

Here, numbers at age in the first year are filled via exponential decay, , numbers at age 1 (i.e. recruits) vary around a baseline value, , and total mortality is set to a baseline level plus process error, . The error around the recruitment process was set to follow a random walk, , and the process error was simulated using the covariance structure described in Cadigan (2016), . The covariance across ages and years is controlled by a process error variance parameter () along with age and year correlation parameters ( and , respectively). This structure allows for autocorrelation in process errors across ages and years (i.e. total mortality can be made to be more similar for fish that are closer together in age and time). Abundance at age is then converted to abundance at length using the original von Bertalanffy growth curve (Cailliet *et al.*, 2006):

Where is the mean asymptotic length, is length at birth, is the growth rate parameter and the error is assumed to follow the normal distribution, . Abundance in discrete length groups, , was determined by using this formula to calculate the probability of being within a specific length group given age and then using the resultant length-age-key to convert abundance at age to abundance at length. Overall, this formulation allows for the simulation of an array of population dynamics that one might expect to observe.

Abundance is simulated using the sim\_abundance function and the settings used for our case study are shown in Table 1. These settings are also the defaults of the function. This function has a simple structure and requires the specification of a series of ages and years along with functions for simulating recruitment (R), total mortality (Z) and growth (growth). The package includes the functions sim\_R, sim\_Z and sim\_vonB for simulating recruitment, total mortality and growth, respectively. These functions are “closures” (i.e. functions that contain data and return functions; Wickham, 2014) and this structure was chosen to avoid the repeated specifications of ages and years. The use of closures also allows users to use their own closures with a similar structure but different underlying formula. Overall, the function provides a simple tool for simulating a range dynamic age-structured populations. For instance, below we simulate a relatively long (default case study settings) and short lived species.

set.seed(438)  
a <- sim\_abundance(ages = 1:20,  
 R = sim\_R(mean = 3e+07),  
 Z = sim\_Z(mean = 0.5))  
b <- sim\_abundance(ages = 1:6,  
 R = sim\_R(mean = 1e+10),  
 Z = sim\_Z(mean = 0.8))

The sim\_abundance function returns a list with the sequence of ages (ages), sequence of years (years), sequence of lengths (lengths), numbers of recruits across all years (R), numbers at age in the first year (N0), total mortality matrix (Z), abundance at age matrix (N), abundance at length matrix (N\_at\_length) and the function supplied to the growth argument (sim\_length). The growth function is retained for later use in sim\_survey to simulate lengths given simulated catch at age in a simulated survey.

The package also includes several plotting function for making quick plotly-based (Sievert, 2018) interactive visuals of the simulated population. For instance, the plot\_surface function can be used to make quick visuals of matrices contained within the list returned by sim\_abundance. The plots output from the below code are combined into Figure 1. Here we display the abundance at age matrix (object named N in the list produced by sim\_abundance); other names can be supplied to the mat argument to visualize a different matrix from the sim\_abundance list, such as Z.

plot\_surface(a, mat = "N")  
plot\_surface(b, mat = "N")

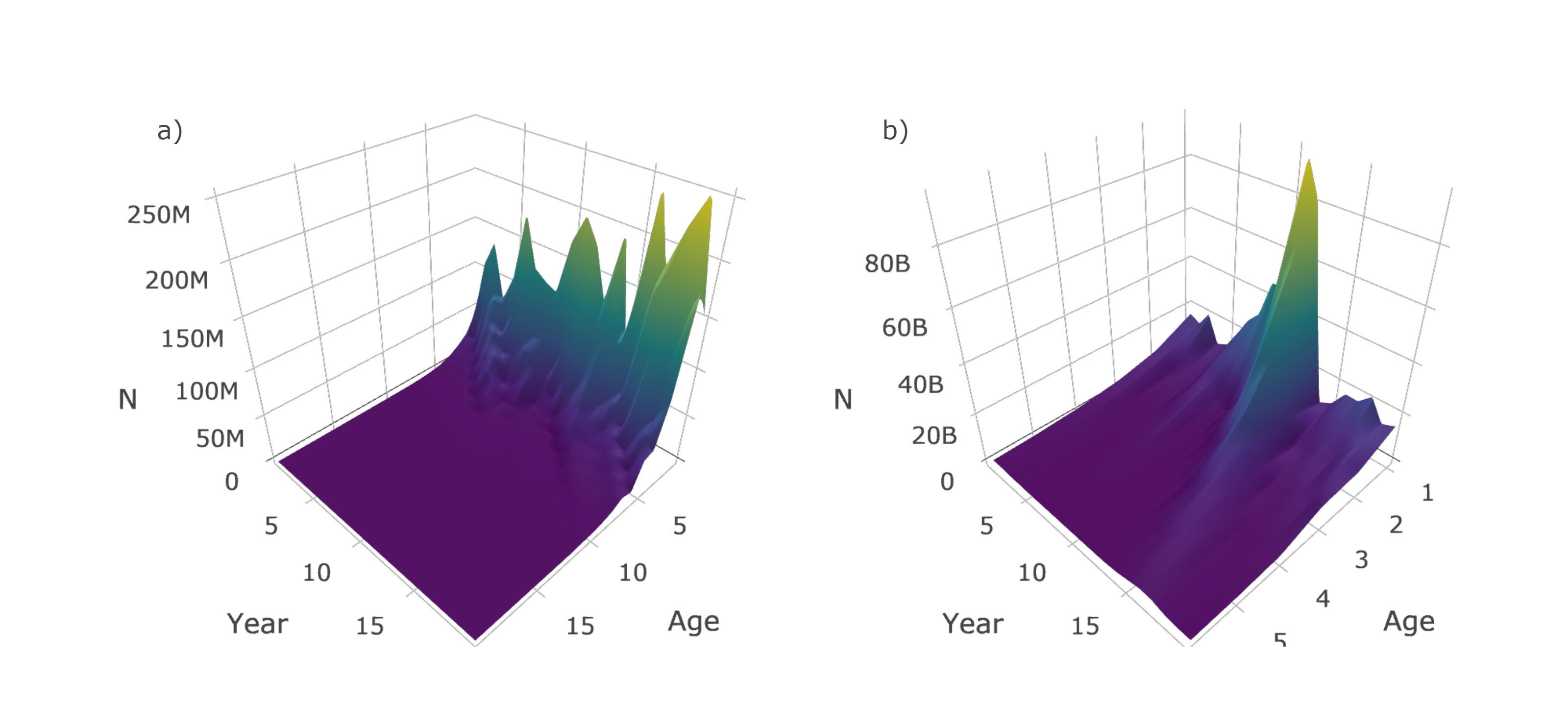


Figure 1 - Surface plots of simulated abundance at age of a relatively a) long lived (case study settings) and b) short lived species. These plots were produced by plot\_surface when supplied a list produced by sim\_abundance.

## Simulate spatial distribution

The next step in the simulation is to distribute the abundance at age matrix simulated using the cohort model throughout a spatial field. Here, a grid of locations was generated with a resolution of and a depth gradient. The depth gradient was set-up to have a sigmoid shape with a depth range of , shelf depth of and a shelf width of . The grid is divided into divisions and depth-based strata. The simulated population is distributed through the grid by simulating spatial-temporal noise controlled by a parabolic relationship with depth and covariance between ages, years and space

Where is the depth in a specific cell of the grid, is the mean depth where abundance is typically highest and controls the width or dispersion of abundance around the mean. This helps simulate depth preferences of particular fish. Residual noise is added to this depth relationship using a combination of Matérn covariance, to control the level of spatial aggregation, and the age-year covariance described in Cadigan (2016), to control the level of similarity in distributions across ages and years. Spatial correlations was controlled by a smoothing () and scaling parameter () (here we approximate from a range parameter (), ; Blangiardo and Cameletti, 2015) and correlation across ages and years was controlled by and , respectively. The overall variance of the process was controlled by . In short, this formulation allows control of depth preferences, the level of spatial aggregation and the amount of age and year specific clustering. Note that is forced to sum to one in each year to ensure that the total population of each age for each year through the grid equals the number simulated by the cohort model. Also note that numbers were rounded such that a discrete number of fish are in each cell of the grid. A more detailed description of the space-age-year covariance is included in Appendix 2. Note that the the distributions of specific length groups were not simulated because of the greater computational cost of storing spatially dis-aggregated abundance across all length groups.

The make\_grid function is used to generate a survey grid for use in the sim\_distribution function. The output from make\_grid is a raster object (Hijmans, 2016) with four layers: depth, cell, division and strat. If a more detailed and realistic grid is required, users can manually generate their own survey grid using real data and this grid can be supplied to sim\_distribution if the same structure is used. The package includes a manually constructed survey grid of NAFO division 3Ps (named survey\_grid) and the data-raw folder in the GitHub directory includes the data and code used to construct this grid. However, for simplicity, we use make\_grid to construct a square grid for our case study and the settings used are shown in Table 1. Below we generate and plot (combined in Figure 2) a default case study grid and another grid with four divisions and a more linear gradient in depth.

a <- make\_grid(n\_div = 1, shelf\_depth = 200,  
 shelf\_width = 100, depth\_range = c(0, 1000))  
b <- make\_grid(n\_div = 4, shelf\_depth = 500,   
 shelf\_width = 0, depth\_range = c(0, 1000))  
plot\_grid(a)  
plot\_grid(b)

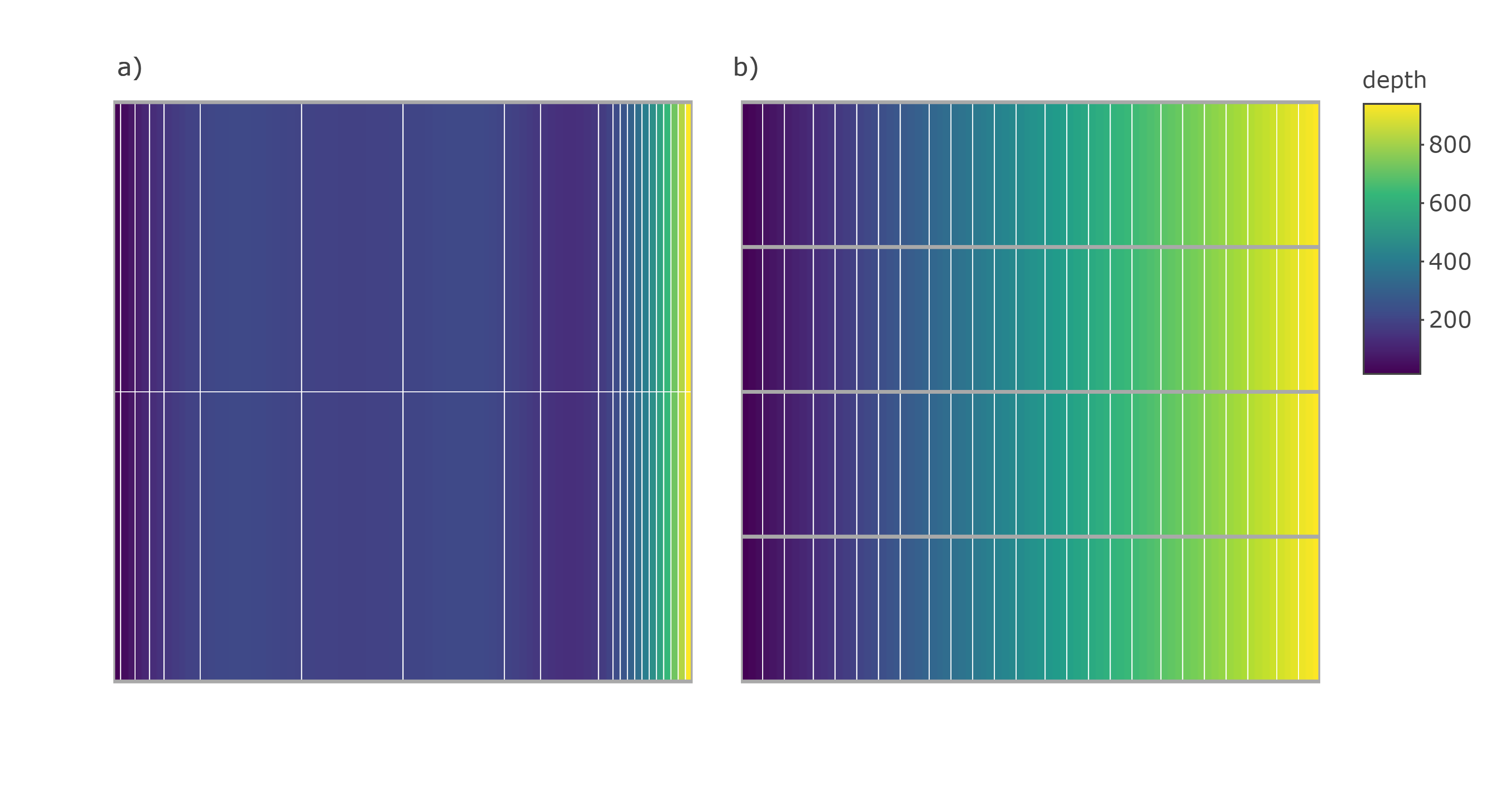


Figure 2 - Plots produced by plot\_grid when supplied an raster object produced by make\_grid a) using default case study settings and b) settings generates a grid with four divisions and a more linear depth gradient. In these plots, the color gradient represents depth, the grey lines delineate divisions and white lines delineate strata.

The sim\_distribution function distributes the population simulated using sim\_abundance throughout the grid generated using make\_grid. Table 1 outlines the default settings used for the function and the case study. In addition to supplying objects produced by sim\_abundance and make\_grid, this function requires two closures that describe the age-year-space covariance and the relationship with depth. Here we use sim\_ays\_covar and sim\_parabola to control these relationships and a wide range of age and year specific distributions can be obtained by tweaking a few parameters in these closures. Below we run a default sim\_distribution call, which generates a population that forms tight clusters that are more strongly correlated across years than ages, and another call that generates a population that is more diffuse (i.e. wider range) and exhibits stronger correlation across ages than years. Distributions can also be forced to be the same across ages and years by using the group\_ages and group\_years, respectively, arguments in the sim\_ays\_covar closure. In other words, these parameters can be modified to control the degree of age-specific clustering and inter-annual site-fidelity exhibited by the simulated population. Note that the resolution of the default grid is high and, as such, the simulations below may take minutes to complete. Also note that the key functions in the **SimSurvey** package have been set-up to be pipe (%>%; Bache and Wickham, 2014) friendly.

set.seed(438)  
a <- sim\_abundance() %>%   
 sim\_distribution(ays\_covar = sim\_ays\_covar(range = 300,  
 phi\_year = 0.9,  
 phi\_age = 0.5))  
b <- sim\_abundance() %>%   
 sim\_distribution(ays\_covar = sim\_ays\_covar(range = 2000,  
 phi\_year = 0.2,  
 phi\_age = 0.9))

This function retains all the data simulated by sim\_abundance and adds a data.table (Dowle and Srinivasan, 2017), named sp\_N, with abundance (N) split by age, year and cell. The function also retains the grid object and converts these data into a data.table, named grid\_xy, with headers x, y, depth, cell, division and strat. The sp\_N object can be merge with the grid\_xy data by cell to associate abundance with specific locations, depth, divisions or strata. The plot\_distribution function can be used to provide a quick visual of the distribution across ages and years. The code below will generate interactive plots with an Age-Year slider, however, for this paper we present a facet plot of the simulated data in Figure 3.

plot\_distribution(a, ages = 1:3, years = 1:3, type = "heatmap")  
plot\_distribution(b, ages = 1:3, years = 1:3, type = "heatmap")

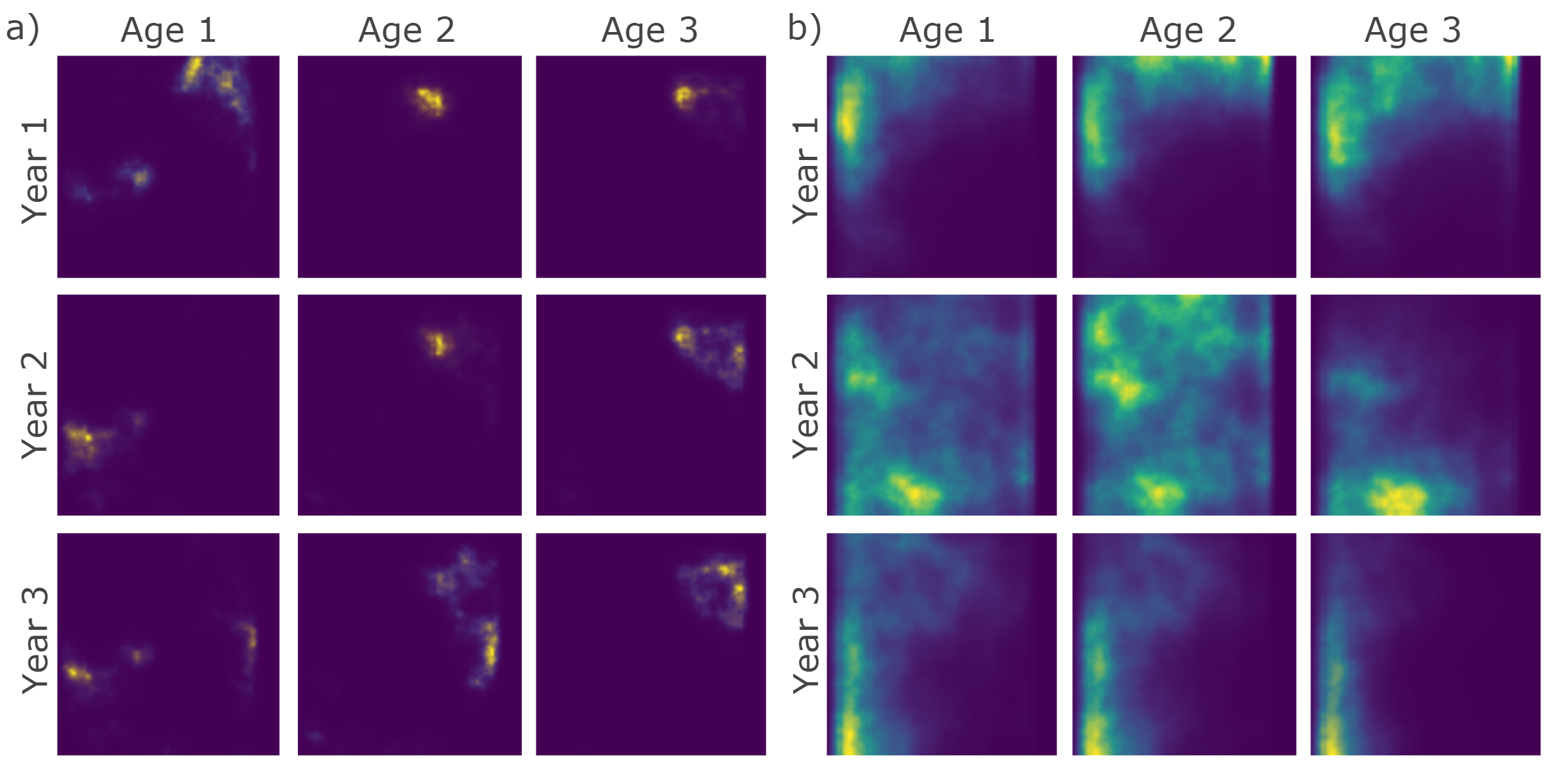


Figure 3 - Distribution plots of simulated populations that form a) tight clusters with stronger correlation through years than ages (case study settings), and b) relatively diffuse clusters with stronger correlation through ages than years. This plot is a facet of plots produced by plot\_distribution when supplied simulations from sim\_distribution.

## Simulate survey

The final step in the simulation is to conduct a stratified random survey over the age-year-space array generated using formula described above. The area of each strata is calculated and this is used to define the sets to allocate to each strata under a particular set density, . The allocated number of cells are randomly selected in each strata and the number of fish caught in each set is calculated by applying binomial sampling of the fish in each sampled cell by the proportion of the area covered by the trawl and the catchability of each age:

Where is the number of fish sampled by a set, indicates the area covered by the trawl, is the area of a grid cell (i.e. ), and is the catchability coefficient of each age. Here, catchabilities were defined using a logistic curve controlled by a steepness, , and midpoint parameter, . The lengths of the fish sampled by the set are then simulated using the von Bertalanffy growth equation found above in the [Simulate abundance](#simulate-abundance) section. Depending on the number of fish caught, sub-sampling is then conducted. Specifically, a maximum number of lengths are measured per set, , and a maximum number of ages, , are sampled per length group, , per division. Such sub-sampling is common in fisheries-independent surveys as it is costly to sample every fish sampled. Age determination is especially time-consuming, which is why age sampling tends to be length-stratified.

The function sim\_survey can be used to simulate data from one survey over a population simulated using sim\_distribution. This function simulates the sampling process of the survey and, as such, requires a closure for defining catchability as a function of age and definitions of the design of the survey. Specifically, the q argument requires a closure, such as sim\_logistic, for defining the probability of catching specific age groups, trawl dimensions are defined in the trawl\_dim argument, and set, length and age sampling effort are defined using the set\_den, lengths\_cap and ages\_cap arguments, respectively. Multiple simulations of the same survey can be run using the n\_sims argument, however, it is important not to request large numbers of simulations because this function can quickly deplete a computers RAM. Defaults used in the function and for the case study roughly correspond to the assumed catchability in NAFO division 3Ps as well as the sampling protocol used in the survey of this stock (Table 1). Below we use sim\_survey to simulate two surveys over our case study population, of which one is set-up to have higher sampling effort than the other.

set.seed(438)  
pop <- sim\_abundance() %>%   
 sim\_distribution()  
a <- pop %>%   
 sim\_survey(n\_sims = 5,  
 set\_den = 1 / 1000,  
 lengths\_cap = 100,  
 ages\_cap = 5)  
b <- pop %>%   
 sim\_survey(n\_sims = 5,  
 set\_den = 5 / 1000,  
 lengths\_cap = 500,  
 ages\_cap = 25)

Again, this function retains all the objects listed in the output of sim\_distribution and adds data.tables that detail the set locations (setdet) and sampling details (samp). Catchability corrected abundance matrices, named I and I\_at\_length, are also produced and added to the output; these matrices are useful for comparing the true abundance available to the survey to abundance estimates obtained using design-based or model-based analyses of the simulated survey data. Specific surveys can be explored using the plot\_survey function, which uses plotly (Sievert, 2018) and crosstalk (Cheng, 2016) in the background to link the bubble plot of aggregate set catch to the histogram of lengths and ages sampled to facilitate explorations of set-specific catches. The plots generated below have been combined here into a static plot for this paper (Figure 4).

plot\_survey(a, which\_sim = 1, which\_year = 20)  
plot\_survey(b, which\_sim = 1, which\_year = 20)

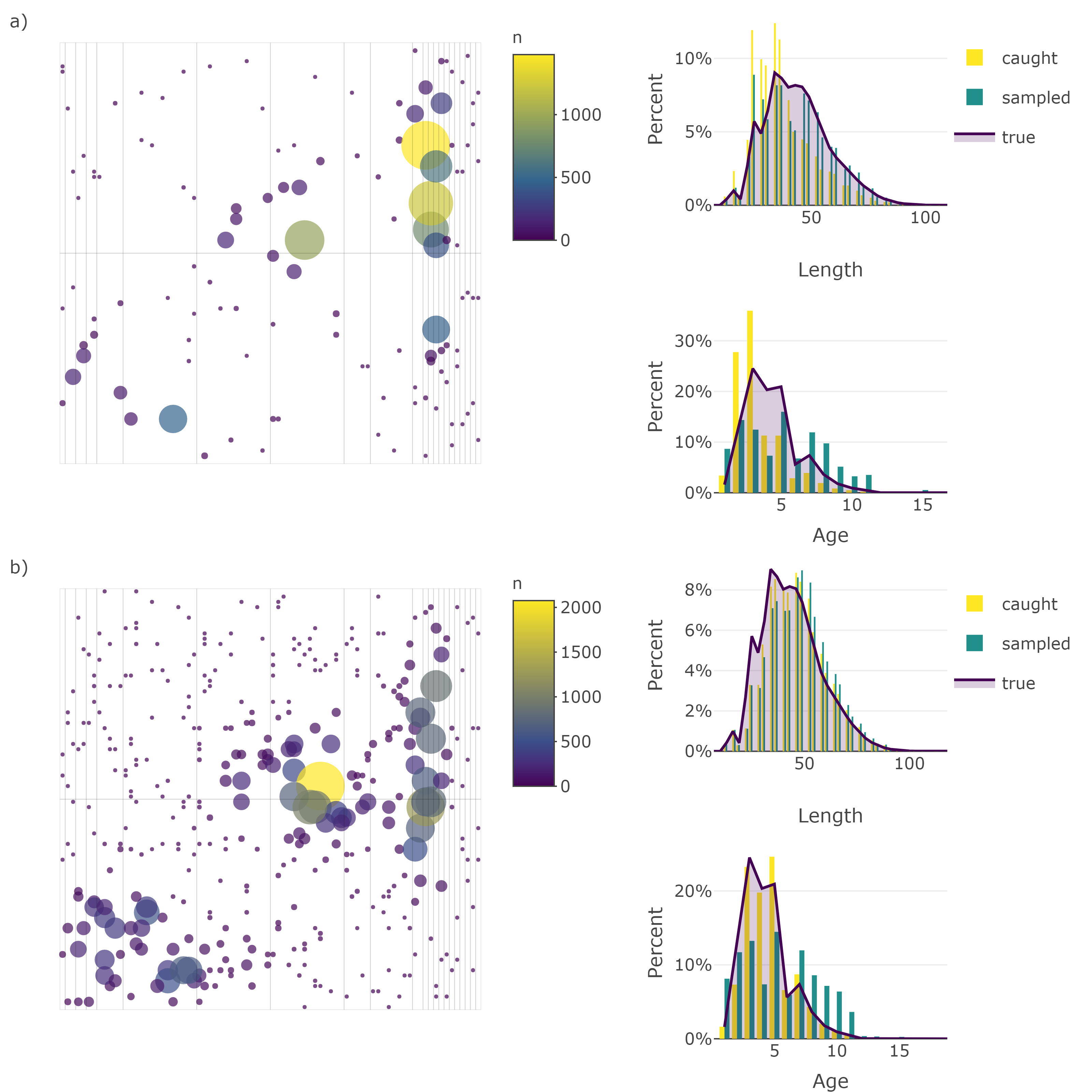


Figure 4 - Bubble plots and histograms of set catches from a simulated stratified-random survey of the case study population under relatively a) high and b) low sampling effort. Histograms of length and age composition include the distribution caught and sampled overlaid with a line of the true distribution of length and ages available to the survey. Note that the first simulation of the survey in year 20 is depicted here. These plots are produced by plot\_survey when supplied survey data simulated using sim\_survey.

As noted above, the amount of RAM a user’s computer has may limit the utility of the sim\_survey function for running thousands of simulations of the same survey. The sim\_survey\_parallel function was therefore constructed to facilitate the running of thousands of simulations without running into RAM limitations. This function is set-up to run multiple sim\_survey calls in parallel using the **doParallel** package (Revolution Analytics and Weston, 2015) and, as such, multiple loops can be run using the n\_loops argument and within each loop multiple simulations can be run (controlled using the n\_sims argument). Total simulations will be the product of n\_loops and n\_sims arguments. If more than one core (cores) is specified, then the simulations will be run in parallel to speed up the process. Low numbers of n\_sims and high numbers of n\_loops will be easier on RAM, but may be slower. The optimum ratio of n\_sims to n\_loops will depend on the amount of RAM and number of cores the user’s computer has. In any case, this function simplifies the process of running thousands of simulations of the same survey and the simulated data can then be supplied to survey-based or model-based analyses that require simulation testing.

## Stratified analysis

Once the survey data are simulated, they then need to be analyzed to obtain an index of abundance. While there are many model-based options for obtaining an index (e.g.  Thorson *et al.*, 2015), survey-based approaches, such as stratified analyses, are often used. Here we apply formula presented in Smith and Somerton (1981) to calculate stratified estimates of total abundance (), abundance at length () and abundance at age (). Note that estimates of total abundance are based on aggregate catch while abundance at length requires length frequencies to be bumped up using set-specific ratios of measured to caught fish, and these length frequencies are converted to age by applying a division-level age-length-key. We used root-mean-squared error (RMSE) as a measure of the precision and bias of the abundance at age estimates from each survey:

Where , , and are the number of years, ages and simulations, respectively, and is the true abundance available to the survey (i.e. catchability corrected abundance; ). RMSE was also calculated for abundance at length estimates, where the above formula is indexed by length groups , and total abundance, which lacks a group index of or .

Stratified estimates of abundance obtained are by supplying the output from sim\_survey to the run\_strat function. RMSE of the stratified estimates can then be calculated using the strat\_error function. Results and error of a stratified analysis of one default survey over a default population are obtained using the following code:

set.seed(438)  
sim <- sim\_abundance() %>%   
 sim\_distribution() %>%   
 sim\_survey() %>%   
 run\_strat() %>%   
 strat\_error()

The returned object will include all the objects accumulated through the sim\_abundance to strat\_error. The run\_strat function adds three data.tables called total\_strat, length\_strat and age\_strat that include stratified estimates of total abundance, abundance at length, and abundance at age, respectively. To this, strat\_error adds data.tables ending with \_strat\_error or \_strat\_error\_stats. The \_strat\_error objects simply contain stratified estimates of abundance (column named I\_hat) with corresponding true values of abundance available to the survey (column named I) and the strat\_error\_stats data.frame includes metrics of mean absolute error (MAE), mean-squared error (MSE) and root-mean-squared error (RMSE).

## Testing survey protocol

Assuming a stratified analysis as the default method for obtaining an index of abundance, a series of survey protocol can be tested using the test\_surveys function. Provided a simulated population from sim\_distribution and a series of survey protocol from expand\_surveys, this function will simulate and analyze data from each survey using the sim\_survey, run\_strat and strat\_error functions. Like sim\_survey\_parallel, this function operates in parallel and allows the specification of n\_sims and n\_loops, and the product of these two arguments equals the number of times each survey is simulated. Keep in mind that low numbers of n\_sims and high numbers of n\_loops will be easier on RAM, but may be slower, especially if the work is spread across few cores. Because most of the default settings of the functions match the case study settings, the below code will replicate the results from our case study (see Table 1 for a more detail). The expand\_surveys function sets up a series of 175 surveys to test (i.e. all possible combinations of set\_den, lengths\_cap and ages\_cap protocol) and the test\_surveys function will run 1000 simulations of each survey and compare stratified estimates of abundance to the true abundance available to the survey.

set.seed(438)  
pop <- sim\_abundance() %>%  
 sim\_distribution()  
  
surveys <- expand\_surveys(set\_den = c(0.0005, 0.001, 0.002, 0.005, 0.01),  
 lengths\_cap = c(5, 10, 20, 50, 100, 500, 1000),  
 ages\_cap = c(2, 5, 10, 20, 50))  
  
tests <- test\_surveys(pop, surveys = surveys,  
 n\_sims = 5, n\_loops = 200, cores = 2)

Processing time will system (i.e. amount of RAM and number of cores) and setting (i.e. n\_loops and n\_sims ratio) dependent. The test\_survey function will print a progress bar, generated using the **progress** package (Csárdi and FitzJohn, 2016), which details percent completion and will also include an eta after the first step of the loop completes. Keep in mind that thousands of simulations of hundreds of survey designs may take days to run. The test\_surveys function therefore includes an option for exporting intermediate results to a local directory, via the export\_dir argument, and the resume\_test function can be used resume a test\_surveys run that had to be stopped part way through the process. The final object produced will be a list that includes all objects from sim\_abundance and sim\_distribution with the table of survey designs tested (named surveys) and tables produced by strat\_error that end with the names \_strat\_error and \_strat\_error\_stats. These tables include a survey column to allow merging of the survey protocol table with the error tables. Objects produced by sim\_survey (set and sampling details) and run\_strat (full stratified analysis results) are not retained to minimize object size. Like other core functions, some convenience functions are included in **SimSurvey** for creating interactive plots of the results from test\_surveys. For instance a series of plotting functions ending in \_fan produces fan charts where stratified estimates of abundance from each simulated survey are converted into a series of quantiles to depict the probability that estimates fall within a particular range. True values of abundance available to the survey are overlaid on the series of probability envelopes. These plots help visually assess the level of precision and bias from a specific set of survey protocol. The three lines of code below will produce interactive fan charts for stratified estimates of total abundance, abundance at length and abundance at age, respectively. A subset of these plots have been combined into static plots (Figure 5, 6, 7).

plot\_total\_strat\_fan(tests)  
plot\_length\_strat\_fan(tests, years = 1:20, lengths = 1:100)  
plot\_age\_strat\_fan(tests, years = 1:20, ages = 1:10)

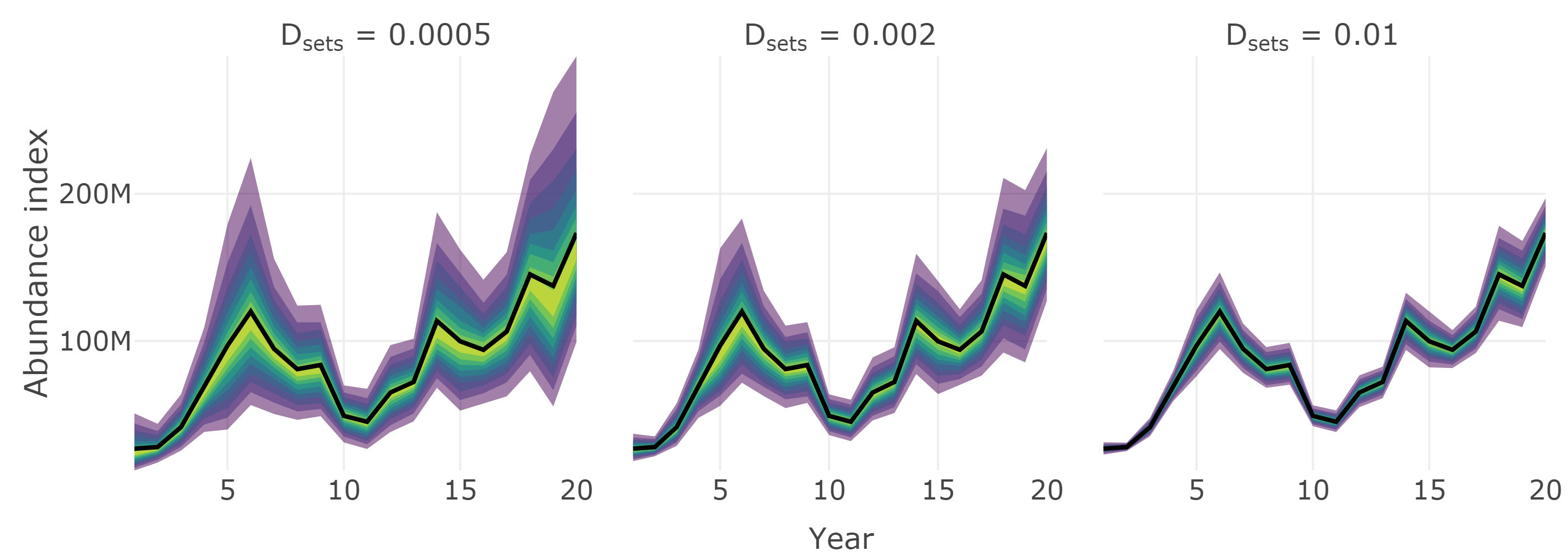


Figure 5 - Fan chart of stratified estimates of the trend in total abundance from surveys with different set densities, . The heavy dark line indicates the true tred in the total population available to the survey and the color gradient represents a range of probability envelopes from 10% to 90%. This plot is a facet of plots produced by plot\_total\_strat\_fan when supplied results from test\_surveys.

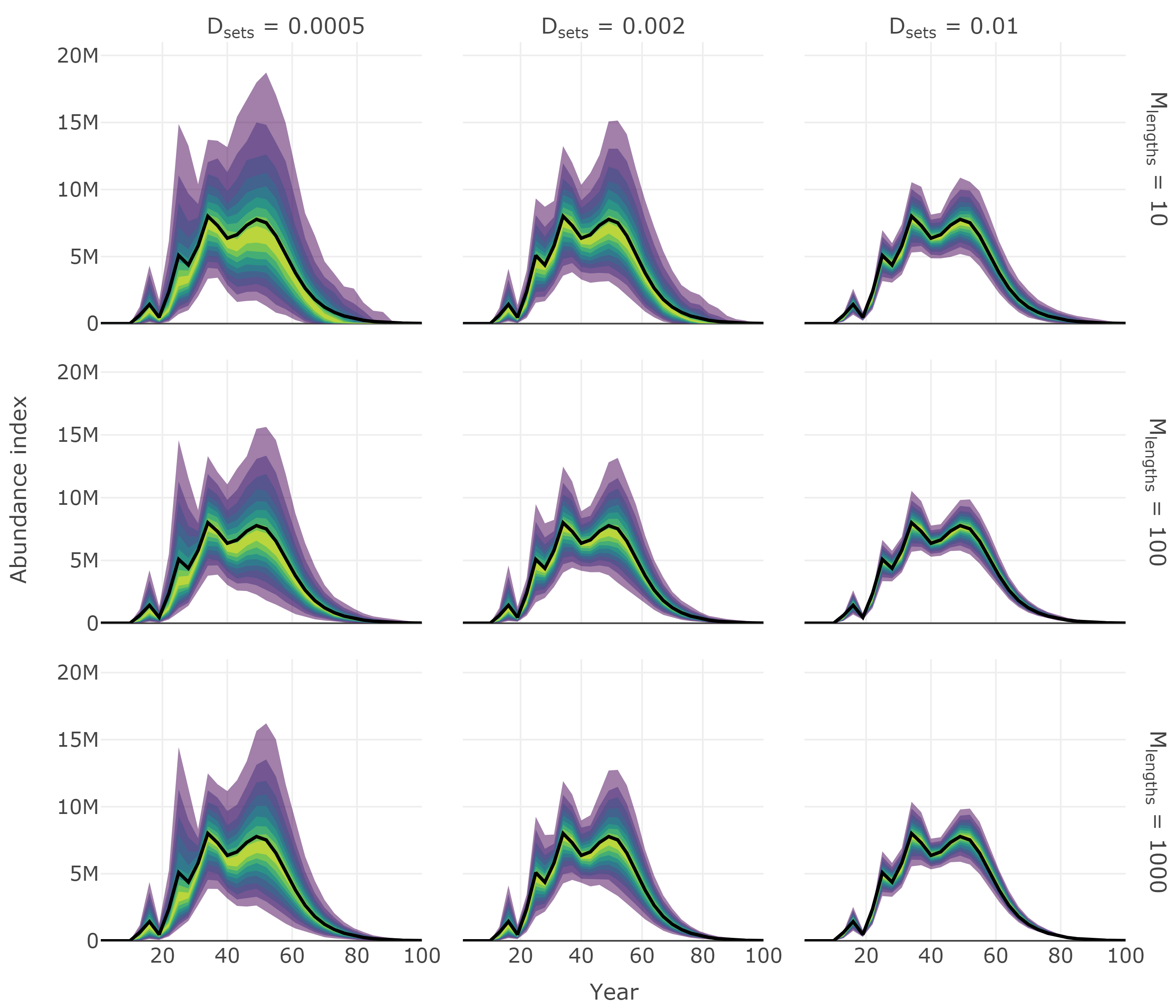


Figure 6 - Fan chart of stratified estimates of abundance at length from year seven of the simulation from surveys with different set densities, , and length sampling protocol, . The heavy dark line indicates the true tred in the total population available to the survey and the color gradient represents a range of probability envelopes from 10% to 90%. This plot is a facet of plots produced by plot\_total\_strat\_fan when supplied results from test\_surveys.

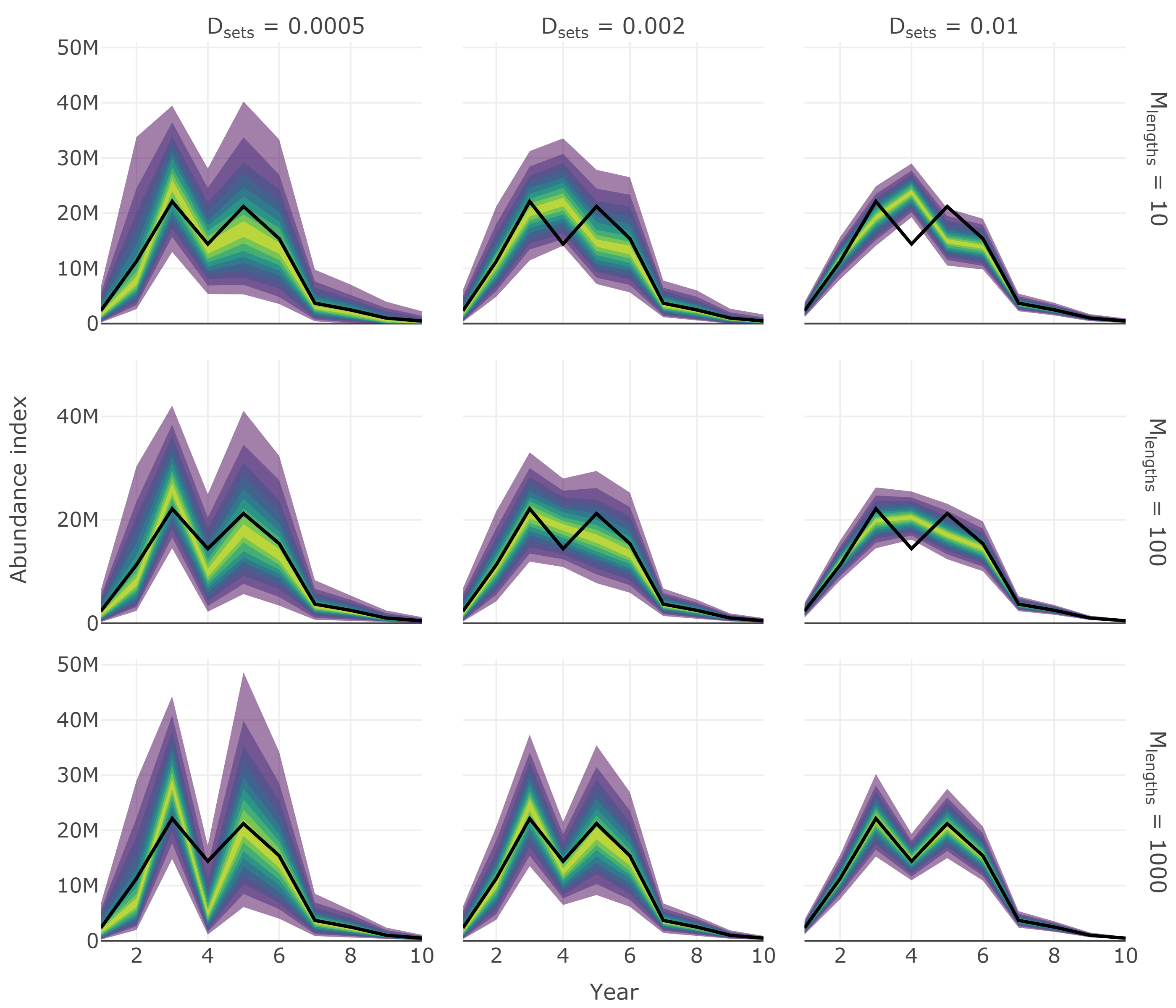


Figure 7 - Fan chart of stratified estimates of the trend in abundance at age four from surveys with different set densities, , and length sampling protocol, . Number of ages sampled per length group, , was 10 in all scenarios. The heavy dark line indicates the true tred in the total population available to the survey and the color gradient represents a range of probability envelopes from 10% to 90%. This plot is a facet of plots produced by plot\_total\_strat\_fan when supplied results from test\_surveys.

The relative performance of the surveys tested can be compared using plot\_survey\_rank and plot\_error\_surface. The plot\_survey\_rank function produces a divergent dot plot of the results which ranks the surveys by RMSE. Using the wich\_strat argument, the plot can be focused on total, length or age based stratified results. The plot\_error\_surface focuses solely on the age based stratified results by plotting a surface of RMSE (z-axis) by set (drop down selection), length (y-axis) and age (z-axis) sampling effort. The sampling effort axes can be rule or sample size based (plot\_by = "rule" or plot\_by = "samples", respectively). See figures 8 and 9 for examples of the output from the code below.

plot\_survey\_rank(tests, which\_strat = "length")  
plot\_error\_surface(tests, plot\_by = "rule")

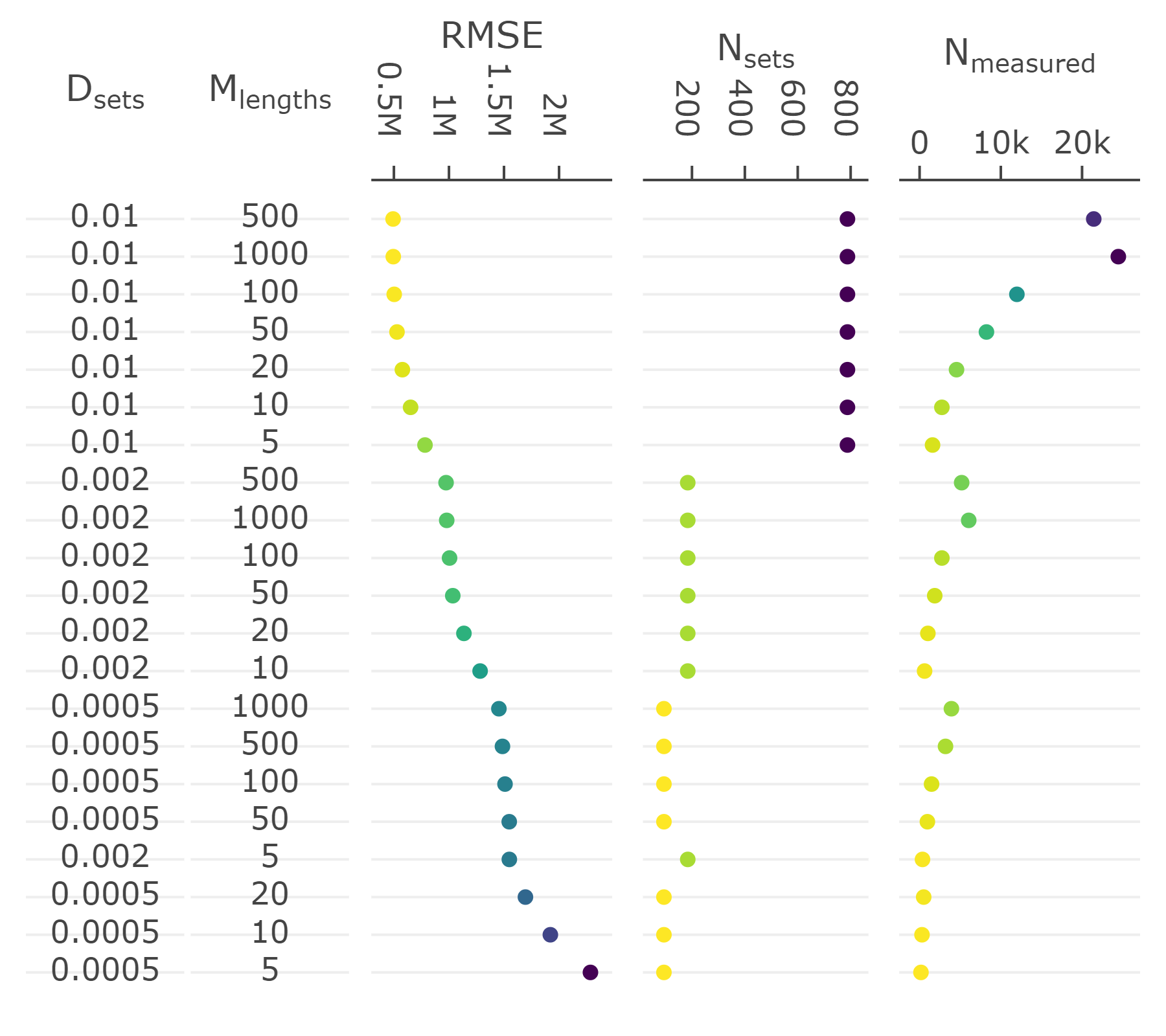


Figure 8 - Divergent dot plot of the precision and accuracy (RMSE) of length based stratified estimates of abundance, and total sampling effort (number of sets [] and length measurements []), under various sampling protocol (set density [] and maximum number of lengths measured per set []).

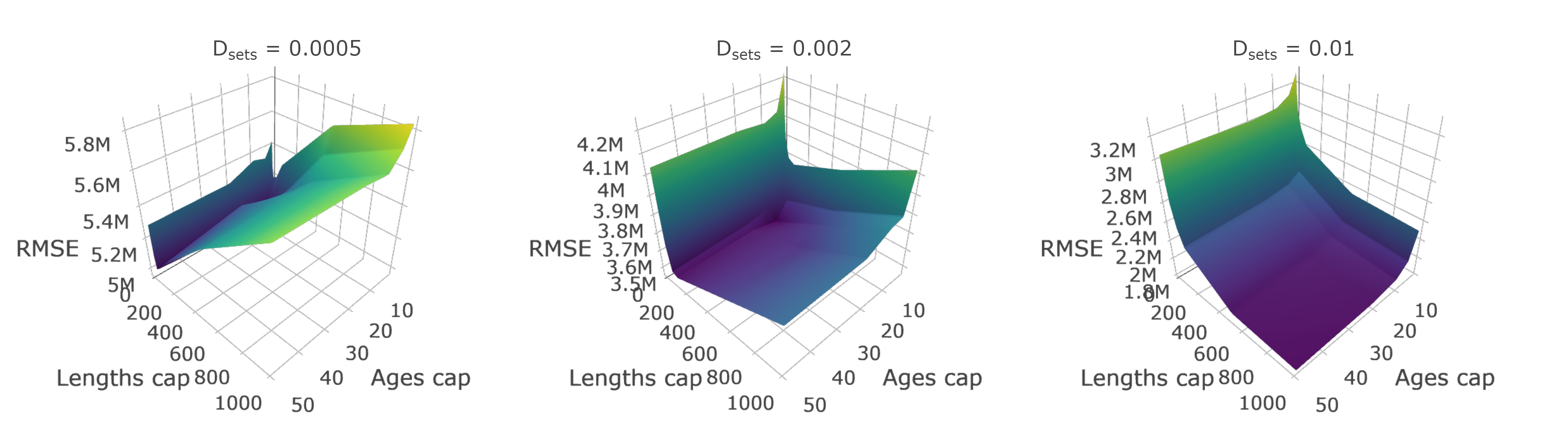


Figure 9 - Surface plots of RMSE from an array of surveys with different sampling protocol. Panels represent surveys with different set densities (), x-axes represent the maximum sampling effort of lengths per set (), and y-axes represent the maximum number of ages to collect per length group (). This plot is a facet of plots produced by plot\_error\_surface when supplied results from test\_surveys. Note that the z-axes scales are different across the facets.

## Assumptions

Like any model, this simulation is a simplification of a much more complex reality. For instance, the population is assumed to aggregate by age-class and be uniformly distributed within a cell when, in the real-world, they may aggregate by length and form finer-scale clusters. The survey is also an instantaneous snapshot of the population, meaning that the population is assumed to be in the same location from the beginning to the end of the survey. Also, fish are aged at random within length bins and ages are estimated without error. Finally, area trawled is assumed to be perfectly standard. This is a short, non-inclusive, list of additional variability that likely occurs in the real-world but was not implemented here for simplicity. Nevertheless, the **SimSurvey** package provides the most realistic operating model that we are aware of for simulating stratified-random survey data from a population that varies across age, year and space dimensions.

# Research opportunities

The case study used here provides one example of ways in which **SimSurvey** can be used to simulation test the design of fisheries-independent trawl surveys. There are multiple layers to the sampling design of such surveys and the end results are linked with the way such data are analyzed. Below we outline some examples where the **SimSurvey** package may aid future research efforts.

## Design or model-based approach

The analysis of data from fisheries-independent surveys have generally been confined to design-based mean and variance estimates of abundance using standard formula for stratified-random designs (e.g.  Cochran, 1977). Nevertheless, there has long been interest in using model-based approaches to improve abundance estimates (e.g.  Smith, 1990; Berg *et al.*, 2014; Thorson *et al.*, 2015). **SimSurvey** can serve as a convenient tool for simulation testing mean and variance estimates provided by a range of different approaches (design-based analyses, bootstrap estimates, generalized additive models, geostatistical models, etc.). Moreover, the full analytical pathway for obtaining age-disaggregated estimates of abundance has rarely been simulation tested. Existing and future approaches for calculating age-based indices of abundance can be simulation tested using **SimSurvey**.

## Growth analyses

Assessing ages for a large number of fish is very time-consuming and, as such, length-stratified sampling is often used to estimate age frequencies of fish populations. The resultant sub-sample is used to construct an age-length key (i.e. the probability a fish is a specific age given length) and age frequencies are obtained by applying this key to length-frequencies obtained via more expansive random sampling. One age-length key is typically assumed to be representative of the whole stock area, however, spatial variability in the relationship may introduce bias in abundance-at-age estimates (Berg and Kristensen, 2012). Results from the case study presented here (Appendix 1) reiterate this message and **SimSurvey** may serve as a platform for testing potential model-based solutions to this problem (e.g.  Berg and Kristensen, 2012).

## Random or stratified sampling

**SimSurvey** can be used to compare the precision and bias of population estimates obtained using random or stratified sampling. Simple random sampling can be implemented using a grid with one strata (e.g. make\_grid(depth\_range = c(0, 1000), strat\_breaks = c(0, 1000), strat\_split = 0)). Sub-sampling of ages can also be random rather than length-stratified by setting the age\_sampling argument in the sim\_survey function to "random" rather than "stratified". This can facilitate research similar to work presented in Puerta et al. (2019).

# Future directions

Up to now, the package has focused on the effects of sampling design on the precision and bias of population estimates obtained from fisheries-independent surveys, however, the costs associated with sampling has yet to be considered. In future iterations of **SimSurvey**, we hope to add options for integrating data on the time and monetary costs associated with each level of sampling (sets, length measurements, age determination) to facilitate cost-benefit analyses. We also realize that a single fisheries-independent survey may have multiple goals as data obtained are often used to assess multiple species or to conduct community analyses. We therefore endeavor to add functions for simulating multi-species surveys. Finally, it would be useful to add an option for testing the consequences of surveys with partial coverage of a population as survey coverage is a frequent concern in stock assessment.

# Summary

The **SimSurvey** package serves as a tool for simulating stratified random surveys of dynamic populations that vary across ages, time and space. The core of the simulation is based on the widely used cohort equation and, even though the processes that define recruitment and total mortality are simple, a wide range of stock dynamics can be simulated by changing a few parameters. This base population can then be distributed through a grid and relationships with depth can be defined as can the nature of the correlation across ages, years and space. Together, two functions (sim\_abundance and sim\_distribution) are capable of simulating a wide range of populations with different life histories, depth associations and spatial properties. The next necessary step to generating data similar to actual observations is to conduct a survey. In this package we implement a function, sim\_survey, that conducts a stratified random survey of the population. The sampling process is governed by the area covered by the trawl as well as age-specific catchability. Sub-sampling protocol (length and age sampling) can also be varied. As such, data from a wide range of surveys can be simulated.

There are a plethora of statistical models that may be tested using these simulated data, however, implementing a variety of analytical approaches was outside the scope of this package. Instead we focus on analyzing simulated stratified-random survey data using a design-based stratified analysis. A stratified analysis is facilitated using the run\_strat function and the precision and accuracy of the results (e.g. RMSE) can be calculated using strat\_error. This is a simple and widely-used analysis, and the speed at which it runs allows for a wide range of survey designs to be tested, via the test\_surveys function, in a reasonable time-frame.

Simulation testing is an important tool in the field of fisheries science as the inferred status of fish stocks hing on the data and models used to assess fish populations. Simulations provide an opportunity to explore survey and model performance, and such explorations are becoming increasingly important as model complexity increases. It is also important to continually assess the efficacy and efficiency of sampling programs given their costs and the constant scrutiny of the value added by such surveys. These are some of the reasons multiple simulation frameworks, including **SimSurvey**, have been developed to test the design and analyses of complex surveys. We have made **SimSurvey** as open and accessible as possible to allow the broader community to validate, reuse and improve this package. We hope that open-source sharing will extend the value of such simulation frameworks and we encourage users to extend the package for their own needs and contribute to future versions.

# Acknowledgements

**Feedback and advice:** Aaron Adamack, Jonathan Babyn, Alejandro Buren, Noel Cadigan, Karen Dwyer, Geoff Evans, Brian Healey, Paul Higdon, Danny Ings, Mariano Koen-Alonso, Joanne Morgan, Keith Lewis, Derek Osborne, Pierre Pepin, Dwayne Pittman, Don Power, Craig Purchase, Greg Robertson, Martha Robertson, Mark Simpson, Brad Squires, Don Stansbury, Peter Upward

**Package contributions:** Jonathan Babyn, Greg Robertson

**Review of previous draft:** Dave Cote, Keith Lewis, Greg Robertson

**Funding:** NSERC visiting-fellow program

# Appendix 1: case study results and discussion

## Results

Results from the case study presented provide one example of how this package can be used to explore the efficacy of various survey designs. A dynamic population with a patchy and age-clustered distribution was simulated for this case study (Figure 1a, 3a). In general, the parameter settings used (function defaults) dictate that all ages tend to aggregate in fairly dense clusters and that low-density zones are relatively wide-ranging. Surveys, therefore, tend to be characterized by regions with large catches interspersed with regions of no catch (Figure 4). The samples within each successful set are also dynamic and rarely include all age groups. This is because only moderate correlation ( of 0.5) was imposed on the distributions of ages 1-5+, allowing individuals of different ages to occupy different locations. These settings were chosen to simulated data that roughly correspond to actual survey data of cod from NAFO division 3Ps, where cod between the ages of 1-5 tend to be caught at different locations.

A series of surveys were run over the simulated population (see expand\_surveys call in Table 1) and stratified analyses were run on the data obtained. The stratified analyses provided estimates of total abundance, abundance at length and abundance at age, and different stages of sampling affect these estimates. Specifically, set density affects all estimates, length sampling affects estimates of abundance at length and abundance at age, and age sampling affects abundance at age estimates. This is because stratified estimates of total abundance is based on aggregate catch while abundance at length estimates requires both total and length frequency data; finally, abundance at age estimates requires set totals, and length and age data to construct an age-length key.

The effect of set density on estimates of total abundance, abundance at length and abundance at age are clear across all figures generated from the test\_surveys results (Figures 5, 6, 7, 8, 9). Across all fan plots, it is clear that the probability envelopes tighten (i.e. estimates are more precise) as set density is increased (Figures 5, 6, 7). Clear declines in RMSE are also apparent in estimates of abundance at length as set density is increased (Figure 8) and there are notable differences in the scale of RMSE in estimates of abundance at age as set density is increased (Figure 9).

Compared to increases in set density, the effects of increased length sampling effort on abundance at length estimates are less clear. Regarding estimates of abundance at length, RMSE appears to reach a plateau at when length sampling effort is increased 100 measurements per set (Figure 8). Increasing the sampling rule above 100 measurements per set results in many more fish being measured, however, the increase in sampling effort is not matched with substantive declines in RMSE (Figure 8). Interestingly, it appears that measuring fewer total fish at more locations appears to be more beneficial than many fish at fewer locations (Figure 8).

All levels of sampling affect the abundance at age estimates as set catches define the magnitude of the population, length sampling define the length distribution and data from the length-stratified age sampling is used to construct an age-length-key to convert length frequencies to age frequencies. Like the abundance at length estimates, the greatest improvements to RMSE come from increasing set density rather than sub-sampling effort (Figure 9). Specifically, decreasing the age sampling protocol below 10 ages sampled per length group per division appears to result in relatively small increases to RMSE (Figure 9). Length sampling effort, in contrast, has an uneven impact depending on the set density scenario. At low set densities ( = 0.0005 sets / km2), RMSE declines when length sampling effort is increased from around 5 to around 100 measurements per set, and RMSE starts to increase as length sampling effort is increased beyond ~100 measurements per set; in fact, RMSE values appear higher at the highest length sampling scenario (1000 measurements / set) than the lowest (5 measurements / set; Figure 9). A similar pattern is apparent under the medium set density scenario ( = 0.002 sets / km2), however, RMSE values under the lowest and highest length sampling scenarios are of similar magnitude. Finally, RMSE continues to decline with increased length sampling effort under the high set density scenarios ( = 0.01 sets / km2; Figure 9).

## Discussion

The simple case study presented here revealed some expected and unexpected patterns. First, there were clear improvements to precision in all population estimates as the number of sets were increased. This result was expected as it aligns with a basic principle of survey design that the precision and representativeness of a survey increases as the number of sampling units are increased (Sutherland, 2006). Second, stratified estimates of abundance at age were often biased and, in some cases, estimates were poorer when sub-sampling effort was increased. This result was unexpected because design-based estimators, such as the stratified analysis applied here, are expected to be unbiased (Smith, 1990) and increases to sub-sampling effort was expected to be relatively ineffective, not detrimental. Results from stratified estimates of abundance at length, in contrast, better align with these expectations. By deduction, these results indicate that the issue stems from the intervening age-length key and not the design-based estimator.

Using an age-length key in conjunction with the length distribution to estimate abundance at age is standard procedure in the analysis of fisheries-independent survey data as only a small fraction of the catch is typically aged. This is because the aging procedure is costly and time-consuming whereas length measurements are relatively easy to obtain. Ages are generally determined from length-stratified sub-samples of the catch and raw proportions of age-at-length are used to assign ages to fish in specific length groups. Age-length keys are usually constructed at larger spatial scales because there are rarely enough samples to stratify these data. Here we construct one age-length key for the division, as is done for the analysis of 3Ps cod. There is, however, a potential cost to the spatial scale of the key. Namely, it is unlikely that one age-length key is representative for the whole region because the probability of being a specific age given length varies in space (Berg and Kristensen, 2012). This would not be an issue if there was no size (age) specific clustering, however, this may not always be the case because different size groups often occur in different places because of ontogenetic habitat shifts (Dahlgren and Eggleston, 2000; Galaiduk *et al.*, 2017). For instance, it is not uncommon for populations to form distinct nursery and spawning areas (Booth, 2000; Marteinsdottir *et al.*, 2000). Because different age groups sometimes occur in different places, the translation of lengths to ages may be biased by the samples used to generate the age-length key. This bias is perhaps compounded by the length-stratified sampling of ages such that the ages sampled may be skewed towards sets with the most catch, especially under scenarios where length sampling effort is high. If this is the case, then the representativeness of the age-length key to the whole population could be diminished by excessive length sampling.

The negative consequences of excessive length sampling on abundance at age estimates are most apparent under the low set density scenarios. The precision of the estimates are also the poorest when fewer sets are conducted. Sets, however, are the most costly sample to obtain and, as such, the set densities of fisheries-independent surveys tend to be on the lower end of the scenarios presented in this paper. The set density of the multi-species survey conducted by Fisheries and Oceans Canada in the Newfoundland region, for instance, ranges between 0.001 and 0.002 sets / km2. Length sampling effort, in contrast, tends to be on the higher end of the scenarios presented here because lengths are relatively cheap and easy to obtain. Again, using Fisheries and Oceans Canada sampling protocol from the Newfoundland region as an example, it is not uncommon for length sampling effort to be capped at 500 measurements per set. The results presented in this paper paradoxically suggest that better abundance at age estimates are obtained by lowering the length sampling cap. Results also indicate that reductions to length sampling effort would not significantly impact abundance at length estimates. Nonetheless, it is hard to be prescriptive because this simulation is far from a perfect reflection of reality and it focuses on one case study. Further research is required on the consequences of some of the assumptions of this simulation as well as the interaction between age-specific clustering, length-stratified sampling and the age-length key. Ideally, the simulation would also include multiple species with different life-histories and distributions, and also integrate a cost component to assess the trade-offs between information and cost. Finally, it would be interesting to test alternate analyses of these data that may account for the spatial structure of the age-length key (e.g.  Berg and Kristensen, 2012).

While much work has yet to be done, results from the simple simulation echo the growing body of literature which concludes that extra sub-sampling is an ineffective means of improving estimates relative to sampling more locations (Pennington and Vølstad, 1994; Bogstad *et al.*, 1995; Pennington *et al.*, 2002; Coggins *et al.*, 2013; Zhang and Cadrin, 2013; Stewart *et al.*, 2014). In general, the most significant source of variation in fisheries-independent surveys stems from set-to-set variation, and not variability from individual sub-samples. This is largely because fish caught together tend to be more similar than those in the general population (Pennington and Vølstad, 1994). Therefore, if the goal of a trawl survey is to maximize information, it is likely better to stop collecting psudoreplicates and, instead, focus efforts on other species or the next set.

# Appendix 2: age-year-space covariance

The simulation applied in this paper was set-up to control covariance across ages, years and space. To do this we used a combination of Matérn covariance, to control the level of spatial aggregation, and the age-year covariance described in Cadigan (2016), to control the level of similarity in distributions across ages and years. As described in Appendix A in Cadigan (2016), the age-year covariance can be broken down into a series of AR(1) processes. We integrate Matérn covariance into this series of equations:

Where indicates the multivariate normal distribution, controls the variance of the process, and control correlation in the age and year dimension and is defined by Matérn correlation:

where is the Euclidean distance between two locations, is the gamma function, denotes the modified Bessel function of the second kind, and and control the smoothness and scale of the spatial process (Blangiardo and Cameletti, 2015). With this structure, simulated error is correlated across ages, years and space.

# References

Bache, S. M., and Wickham, H. 2014. Magrittr: A forward-pipe operator for r. <https://CRAN.R-project.org/package=magrittr>.

Berg, C. W., and Kristensen, K. 2012. Spatial age-length key modelling using continuation ratio logits. Fisheries Research, 129-130: 119–126. Elsevier B.V. <http://dx.doi.org/10.1016/j.fishres.2012.06.016>.

Berg, C. W., Nielsen, A., and Kristensen, K. 2014. Evaluation of alternative age-based methods for estimating relative abundance from survey data in relation to assessment models. Fisheries Research, 151: 91–99. Elsevier B.V. <http://dx.doi.org/10.1016/j.fishres.2013.10.005>.

Blangiardo, M., and Cameletti, M. 2015. Spatial and spatio-temporal bayesian models with r-inla. John Wiley & Sons.

Bogstad, B., Pennington, M., and Vølstad, J. H. 1995. Cost-efficient survey designs for estimating food consumption by fish. Fisheries Research, 23: 37–46.

Booth, A. J. 2000. Incorporating the spatial component of fisheries data into stock assessment models. ICES Journal of Marine Science, 57: 858–865. Oxford University Press.

Cadigan, N. G. 2016. A state-space stock assessment model for northern cod, including under-reported catches and variable natural mortality rates. Canadian Journal of Fisheries and Aquatic Sciences, 73: 296–308. NRC Research Press.

Cailliet, G. M., Smith, W. D., Mollet, H. F., and Goldman, K. J. 2006. Age and growth studies of chondrichthyan fishes: The need for consistency in terminology, verification, validation, and growth function fitting. Environmental Biology of Fishes, 77: 211–228. Springer.

Cheng, J. 2016. Crosstalk: Inter-widget interactivity for html widgets. <https://CRAN.R-project.org/package=crosstalk>.

Cochran, W. G. 1977. Sampling techniques. John Wiley & Sons.

Coggins, L. G., Gwinn, D. C., and Allen, M. S. 2013. Evaluation of Age-Length Key Sample Sizes Required to Estimate Fish Total Mortality and Growth. Transactions of the American Fisheries Society, 142: 832–840.

Csárdi, G., and FitzJohn, R. 2016. Progress: Terminal progress bars. <https://CRAN.R-project.org/package=progress>.

Dahlgren, C. P., and Eggleston, D. B. 2000. Ecological processes underlying ontogenetic habitat shifts in a coral reef fish. Ecology, 81: 2227–2240. Wiley Online Library.

Dowle, M., and Srinivasan, A. 2017. Data.table: Extension of ‘data.frame‘. <https://CRAN.R-project.org/package=data.table>.

Galaiduk, R., Radford, B. T., Saunders, B. J., Newman, S. J., and Harvey, E. S. 2017. Characterizing ontogenetic habitat shifts in marine fishes: Advancing nascent methods for marine spatial management. Ecological Applications, 27: 1776–1788. Wiley Online Library.

Hijmans, R. J. 2016. Raster: Geographic data analysis and modeling. <https://CRAN.R-project.org/package=raster>.

Marteinsdottir, G., Gudmundsdottir, A., Thorsteinsson, V., and Stefansson, G. 2000. Spatial variation in abundance, size composition and viable egg production of spawning cod (gadus morhua l.) in icelandic waters. ICES Journal of Marine Science, 57: 824–830. Oxford University Press.

Pennington, M., and Vølstad, J. H. 1994. Assessing the effect of intra-haul correlation and variable density on estimates of population characteristics from marine surveys. Biometrics, 50: 725–732.

Pennington, M., and Strømme, T. 1998. Surveys as a research tool for managing dynamic stocks. Fisheries Research, 37: 97–106.

Pennington, M., Burmeister, L.-M., and Hjellvik, V. 2002. Assessing the precision of frequency distributions estimated from trawl-survey samples. Fisheries Bulletin, 100: 74–80.

Puerta, P., Ciannelli, L., and Johnson, B. 2019. A simulation framework for evaluating multi-stage sampling designs in populations with spatially structured traits. PeerJ, 7: e6471. <https://doi.org/10.7717/peerj.6471>.

R Core Team. 2017. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.

Revolution Analytics, and Weston, S. 2015. DoParallel: Foreach parallel adaptor for the ’parallel’ package. <https://CRAN.R-project.org/package=doParallel>.

Schnute, J. T., and Haigh, R. 2003. A simulation model for designing groundfish trawl surveys. Canadian Journal of Fisheries and Aquatic Sciences, 60: 640–656. NRC Research Press.

Sievert, C. 2018. Plotly for r. <https://plotly-book.cpsievert.me>.

Smith, S., and Somerton, G. 1981. STRAP: A User-Oriented Computer Analysis System for Groundfish Research Trawl Survey Data. Canadian Technical Report of Fisheries; Aquatic Sciences No. 1030.

Smith, S. J. 1990. Use of statistical models for the estimation of abundance from groundfish trawl survey data. Canadian Journal of Fisheries and Aquatic Sciences, 47: 894–903. NRC Research Press.

Stewart, I. J., Hamel, O. S., and Rose, K. 2014. Bootstrapping of sample sizes for length- or age-composition data used in stock assessments. Canadian Journal of Fisheries & Aquatic Sciences, 71: 581–588.

Sutherland, W. J. 2006. Ecological census techniques: A handbook. Cambridge University Press.

Thomas, L., Buckland, S. T., Rexstad, E. A., Laake, J. L., Strindberg, S., Hedley, S. L., and Bishop, J. R. *et al.* 2010. Distance software: Design and analysis of distance sampling surveys for estimating population size. Journal of Applied Ecology, 47: 5–14. Wiley Online Library.

Thorson, J. T., Shelton, A. O., Ward, E. J., and Skaug, H. J. 2015. Geostatistical delta-generalized linear mixed models improve precision for estimated abundance indices for West Coast groundfishes. ICES Journal of Marine Science, 72: 1297–1310. <https://doi.org/10.1093/icesjms/fsu243>.

Wickham, H. 2014. Advanced r. Chapman; Hall/CRC.

Zhang, Y., and Cadrin, S. X. 2013. Estimating Effective Sample Size for Monitoring Length Distributions: A Comparative Study of Georges Bank Groundfish. Transactions of the American Fisheries Society, 142: 59–67.