SimSurvey: an R package for comparing the design and analysis of fisheries surveys by simulating spatially-correlated fish stocks

Paul M. Regular1\*, Gregory J. Robertson1, Keith P. Lewis1, Jonathan Babyn1, Brian Healey1, Fran Mowbray1

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

\*Corresponding author  
E-mail: [Paul.Regular@dfo-mpo.gc.ca](mailto:Paul.Regular@dfo-mpo.gc.ca) (PMR)

# Abstract

Fish populations often show complex spatial and temporal dynamics, creating challenges in designing and implementing effective surveys. Inappropriate sampling designs can potentially lead to both under-sampling (reducing precision and increasing the risk of bias) and over-sampling (through the extensive and potentially expensive sampling of correlated metrics). For assessments based on fisheries-independent surveys, the ability to estimate population parameters is affected by multiple levels of sampling, such as the number of sampling stations as well as the sub-sampling of fish captured to measure biological characteristics (e.g. lengths or ages). Population estimates are also affected by the pathway taken to analyze such data. Though simulations are a useful tool for exploring the efficacy of specific sampling strategies and statistical methods, there are a limited the number of tools that facilitate the simulation testing of a range of sampling and analytical pathways for fisheries-independent survey data. Here we introduce the R package **SimSurvey**, which has been designed to simplify the process of simulating surveys of age-structured and spatially-distributed fish populations. The package allows the user to simulate age-structured populations that vary in space and time and explore the efficacy of a range of built-in or user-defined sampling protocols to reproduce the population parameters of the known population. **SimSurvey** also includes a function for estimating the stratified mean and variance of the population from the simulated survey data. **SimSurvey** can serve as a convenient, accessible and flexible platform for simulating a wide range of sampling strategies for fish stocks that show complex structuring. Various statistical approaches can then be applied to the results to test the efficacy of different analytical approaches.

# Introduction

Fisheries-independent 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. While costly to obtain, this information forms the basis of many stock assessments throughout the world [1] and the quality of this information depends on 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. Notably, many fish stocks show size-specific patterns of aggregation, making it difficult to collect sufficient biological sub-samples (e.g. lengths and ages) to represent the entire target population [2–4]. This undermines the assumptions of many analyses as sub-samples from such populations tend to exhibit strong positive intracluster correlation [5]. These complexities may explain why simulations that test the full sampling and analytical pathway of fisheries-independent surveys are rare [6,7].

Here we document **SimSurvey**, an R package designed to simplify and facilitate realistic simulations of fisheries-independent trawl surveys. 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. The package has two main components: the first focuses on mimicking realistic fish stocks by simulating a spatially and age-correlated population distributed across a habitat gradient, and the second component focuses on simulating various surveys of these virtual fish stocks.

This simulation framework has similarities to those presented by Schnute and Haigh [7] and Puerta et al. [6], 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. First we outline the equations underlying the package in the **[Model structure](#model-structure)** section and then move into a vignette of the package under the **[Using SimSurvey](#using-simsurvey)**  section. While the core of this paper focuses on how to use this package, it is important to note that the defaults of the package are based on a case study. Output from default function calls are therefore relevant to the case study and these results are described and discussed in **[S1 Appendix](#s1-appendix-case-study)**.

# Model structure

In this section, we describe the framework currently implemented in the **SimSurvey** package. With this framework, we tried to strike a balance between realism, simplicity, generality and computational feasibility. The framework follows four general steps: 1) simulate a spatially-aggregated age-structured population; 2) distribute the population throughout a spatial grid, imposing correlation across space, time and age; 3) sample the population using random sampling; and, 4) obtain population estimates using a design-based analysis. Though there is a degree of flexibility in each of these steps, users can circumvent specific components by applying user defined equations, inputs and/or analyses. Details on how to use the package and, if desired, circumvent some aspects of its structure are outlined in the **[Using SimSurvey](#using-simsurvey)**  section.

## 

8], Note that a plus group is not modeled as the number of ages can easily be extended to include groups with zero fish. Abundance at age is then converted to abundance at length using the original von Bertalanffy growth curve [9]:

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, . Numbers at age are distributed across discrete length groups following a lognormal distribution by calculating the probability of being in a specific length group given age, . These probabilities are calculated using the standard normal cumulative density function for a sequence of length groups from length 0 to 10 times the maximum predicted length at an interval of :

stock-recruitment), this formulation facilitates the simulation of age-structured populations that are dynamic enough to assess survey performance across a range of abundance levels across years and ages.

## Simulate spatial distribution

Rather than developing a full spatially-explicit model, population and spatial dynamics are modeled as independent processes for simplicity. The complexities of spatial population dynamics - such as larval dispersal, spatial differences in growth and population connectivity - are not explicitly accounted for and, as such, the model is a necessary simplification of reality. Despite this limitation, the approach taken facilitates the simulation of spatial, age-structured populations with sufficient complexity for testing the efficacy of various survey designs. The simplicity also limits the number of unknown parameters that need to be specified to simulate a population. Parameter estimates from spatially-aggregated age-structured models, which are commonly used in stock assessments, can therefore be used to simulate a population using the cohort model and the resultant abundance at age values can be distributed across a spatial grid. Here, a grid of cells is generated where each cell has an area of and depth ; depth is defined using a sigmoid curve, applied across one spatial axisWe use depth as our main stratification variable, but note that any other appropriate stratification variables could be used. The grid can be divided into two hierarchical levels, management divisions and habitat-based survey strata. For demonstration purposes, we envision these levels as part of a stratified-random survey within international fishery divisions, i.e., depth-based strata within NAFO or ICES divisions, or any other geographically bounded area). 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. This noise term is scaled to sum to 1 to ensure that the total population of each age for each year through the grid equals the number simulated by the cohort model:

8], to control the level of similarity in distributions across ages and years. The rate at which point-to-point spatial correlation decays with distance is controlled by a smoothing () and a scaling parameter () [10

## 

The sampling is random or stratified random, emulating surveys conducted by many research institutions around the world. The area of each strata is calculated and this is used to define the number of sampling stations , hereafter referred to as sets, allocated to one or more strata under a particular set density, (i.e. ).

In cases where there are multiple sets in one cell, the population in that cell is divided across the sets. While this means that numbers caught in an isolated simulation cannot exceed the numbers in the population, keep in mind that the survey, no matter how intense, is assumed to have no impact on the population from one year to the next.

Once catches are simulated, lengths of the fish sampled by each set are simulated using the von Bertalanffy growth equation found above in the **[Simulate abundance](#simulate-abundance)** section. Sub-sampling is then conducted whereby a subset of fish are sampled for length measurements and a subset of this subset are sampled for age determination.

## 

11], design-based approaches, such as stratified analyses, are often used. Here we apply formula presented in Smith and Somerton [[12

## 

# Using SimSurvey

The **SimSurvey** package was written in the programming language R [13] and it holds a series of functions for 1) simulating the abundance and distribution of virtual fish populations with correlation across space, time and age (sim\_abundance, sim\_distribution), 2) simulating surveys with a range of sampling strategies and intensities (sim\_survey), and 3) estimating the stratified mean and variance of simulated survey data (run\_strat; Table 1). **SimSurvey** relies heavily on functions from the **data.table** [14], **raster** [15] and **plotly** [16] packages for their efficient data processing, geographic and plotting facilities, respectively. Package documentation has been published online using **pkgdown** (<https://paulregular.github.io/SimSurvey/>) and all the source R code behind **SimSurvey** is available on GitHub (<https://github.com/PaulRegular/SimSurvey>). **SimSurvey** can be installed via GitHub using the **remotes** package:

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

Table 1: Names and descriptions of the key functions of **SimSurvey**. Functions in bold font are core functions and those in medium font are designed for use inside the core functions. The latter are typically closures, which are functions that contain data and return functions [17]; here they are used to store parameter values and return functions that require dimensions, such as ages or years, to be supplied.

|  |  |
| --- | --- |
| Function | Description |
| **sim\_abundance** | Simulate a basic age-structured population dynamics model |
| sim\_R, sim\_Z, sim\_N0, sim\_vonB | Closures, to use inside sim\_abundance, for simulating recruitment, total mortality, initial abundance and growth, respectively |
| **sim\_distribution** | Simulate spatial and temporal distribution of an age-structured population |
| sim\_ays\_covar, sim\_parabola | Closures, to use inside sim\_distribution, for simulating age-year-space covariance and parabolic relationships with covariates (e.g. depth), respectively |
| make\_grid | Make a basic depth stratified square grid to use inside sim\_distribution |
| **sim\_survey** | Simulate a survey of a spatial, age-structured population |
| sim\_logistic | Closure, to use inside sim\_survey, for simulating age-specific catchability as a logistic curve |
| **run\_strat** | Run a stratified analysis on simulated survey data |
| **strat\_error** | Calculate the error of stratified estimates (e.g. root mean squared error of stratified estimates from true values) |
| **test\_surveys** | Test the sampling design of multiple surveys using a stratified analysis (internally loops over sim\_survey, run\_strat and strat\_error) |
| expand\_surveys | Create a data frame, for use in test\_surveys, with all combinations of supplied survey settings |

The equations behind the functions listed in Table 1 are detailed in the **[Model structure](#model-structure)** section. Note that several of the core equations are implemented using “closures”, which are functions that contain data and return functions [17]. For example, sim\_R returns a function that holds the supplied parameter values and requires a sequence of years to be supplied.

R\_fun <- sim\_R(log\_mean = log(500), log\_sd = 0.5)  
R\_vec <- R\_fun(years = 1:100)

Here, the R\_vec object holds 100 years of simulated recruitment values and each run of the R\_fun function will result in different simulated values using the internal formulation and the parameters supplied to sim\_R. The other closures included in the package operate in a similar way in that parameter inputs are supplied to the closure and the functions returned by the closure requires inputs such as ages and/or years. This was done to avoid the repeated specifications of key arguments, such as ages and years. Moreover, this approach provides an option for advanced R users to inspect and modify the closures implemented in the package to supply custom closures with alternate equations. Also note that each of the closures implemented in the package includes a plot argument such that quick visuals can be obtained using a line of code like this: sim\_R(log\_mean = log(500), log\_sd = 0.5, plot = TRUE)(years = 1:100)).

## sim\_abundance

## Table 2

: Default sim\_abundance function call, with descriptions, default values and associated parameter symbols of key arguments.

|  |  |  |
| --- | --- | --- |
| **Function call** | **Description** | **Symbol** |
| sim\_abundance( |  |  |
| ages = 1:20, | Ages |  |
| years = 1:20, | Years |  |
| R = sim\_R(log\_mean = log(30000000), | Mean recruitment1 |  |
| log\_sd = 0.5), | Standard deviation of log-recruitment |  |
| Z = sim\_Z(log\_mean = log(0.5), | Mean total mortality2 |  |
| log\_sd = 0.2, | Standard deviation of total mortality (log) |  |
| 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 in birth year (cm) |  |
| K = 0.1, | Growth rate |  |
| log\_sd = 0.1, | Standard deviation of the von Bertalanffy growth curve |  |
| length\_group = 3)) | Length group bin size for abundance at length (cm) |  |

1 Can be a vector of means with a length equal to the number of years in the simulation.  
2 Can be a matrix of means with number of rows and columns equaling the number of ages and years in the simulation, respectively.

Abundance at age and length is simulated using the sim\_abundance function and a default function call is described in Table 2 along with associated symbols from the equations outlined in the **[Simulate abundance](#simulate-abundance)** section. This function has a simple structure and requires the specification of a series of ages and years along with a series of closures, such as sim\_R, sim\_Z and sim\_vonB, for simulating recruitment (R), total mortality (Z) and growth (growth), respectively. Overall, the function provides a simple tool for simulating a range of dynamic age-structured populations. For instance, below we provide examples where we simulate a relatively long and short lived species (note that default variance, starting abundance and growth settings were used in both simulations).

set.seed(438)  
long <- sim\_abundance(ages = 1:20,  
 R = sim\_R(log\_mean = log(3e+07)),  
 Z = sim\_Z(log\_mean = log(0.2)))  
short <- sim\_abundance(ages = 1:6,  
 R = sim\_R(log\_mean = log(1e+10)),  
 Z = sim\_Z(log\_mean = log(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 functions for making quick plotly-based [16] 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. As an example, we display the abundance at age matrix (object named N in the list produced by sim\_abundance; Figure 1); other names can be supplied to the mat argument to visualize a different matrix from the sim\_abundance list, such as Z.

plot\_surface(long, mat = "N")  
plot\_surface(short, mat = "N")

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

## sim\_distribution

Table 3: Default sim\_distribution function call, with descriptions and associated parameter symbols of key arguments.

|  |  |  |
| --- | --- | --- |
| **Function call** | **Description** | **Symbol** |
| sim\_distribution( |  |  |
| sim, | Simulated population from sim\_abundance |  |
| grid = make\_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 in x and y dimensions (km) - i.e. cell area |  |
| shelf\_depth = 200, | Shelf depth (m) |  |
| shelf\_width = 100, | Shelf width (km) |  |
| depth\_range = c(0, 1000), | Depth range from coast to slope (m) |  |
| n\_div = 1, | Number of divisions |  |
| strat\_splits = 2, | Number of strata within each depth class |  |
| strat\_breaks = seq(0, 1000, 40)), | Series of depth breaks for defining strata |  |
| ays\_covar = sim\_ays\_covar(sd = 2.8, | Standard deviation of age-year-space distribution |  |
| range = 300, | Range of spatial correlation (km) |  |
| lambda = 1, | Smoothness of spatial correlation |  |
| 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 noise equal across these ages1 |  |
| group\_years = NULL), | Make space-age-year noise equal across these years1 |  |
| depth\_par = sim\_parabola(mu = 200, | Depth at which abundance is typically highest (m) |  |
| sigma = 70)) | Dispersion around depth of peak abundance (m) |  |

1 All ages or years are independent if argument values is NULL.

The equations outlined in the **[Simulate spatial distribution](#simulate-spatial-distribution)** section are used in the make\_grid, sim\_ays\_covar and sim\_parabola functions, and these functions are used within sim\_distribution to distribute a population simulated using sim\_abundance throughout a grid (Table 3). The output from make\_grid is a raster object [15] 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 as a raster to sim\_distribution if the same structure structure and correct projection is used. The package includes a manually constructed survey grid of NAFO Subdivision 3Ps off the southern coast of Newfoundland (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 a default run of sim\_distribution. Below we generate and plot (Figure 2) a default grid, another grid with the number of strata increased by increasing the number of strat\_splits, and another with four divisions and a linear depth gradient (the sigmoid curve is forced to be linear when shelf\_width is set to zero).

a <- make\_grid(n\_div = 1, strat\_splits = 2, shelf\_depth = 200,  
 shelf\_width = 100, depth\_range = c(0, 1000))  
b <- make\_grid(n\_div = 1, strat\_splits = 3, shelf\_depth = 200,   
 shelf\_width = 100, depth\_range = c(0, 1000))  
c <- make\_grid(n\_div = 4, strat\_splits = 1, shelf\_depth = 500,   
 shelf\_width = 0, depth\_range = c(0, 1000))  
plot\_grid(a)  
plot\_grid(b)  
plot\_grid(c)

Figure 2: Plots produced by plot\_grid when supplied a raster object produced by make\_grid a) using default settings, b) settings that increase the number of times depth strata are split, and c) settings that produce a more linear depth gradient and increase the number of divisions. In these plots, the yellow to purple color gradient represents depth, the thick grey lines delineate divisions and thin white lines delineate strata.

In addition to supplying objects produced by sim\_abundance and make\_grid, the sim\_distribution 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 adjusting a few parameters in these closures (Figure 3). 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 (i.e. lower phi\_year and higher phi\_age). Distributions can also be forced to be the same across ages and years by using the group\_ages and group\_years arguments, respectively, in the sim\_ays\_covar closure. Variance in the size of the clusters can also be modified by changing the sd argument in the sim\_ays\_covar function. 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 [18] friendly such that values from one function call are forwarded to the next function (i.e. output from the two calls below are functionally the same though the approach is slightly different).

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

The sim\_distribution function retains all the data simulated by sim\_abundance and adds a data.table [14], 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 merged 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 (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 (default 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.

## sim\_survey

Table 4: Default sim\_survey function call, with descriptions and associated parameter symbols of key arguments.

|  |  |  |
| --- | --- | --- |
| **Function call** | **Description** | **Symbol** |
| sim\_survey( |  |  |
| sim, | Simulated spatial population from sim\_distribution |  |
| n\_sims = 1 | Number of times to repeat the survey |  |
| q = sim\_logistic(k = 2, | Steepness of logistic curve of catchability |  |
| x0 = 3), | Midpoint of logistic curve of catchability (age) |  |
| trawl\_dim = c(1.5, 0.02), | Trawl dimensions (distance towed, trawl width; km) - i.e. area trawled |  |
| min\_sets = 2 | Minimum number of sets to conduct per strata |  |
| set\_den = 2/1000, | Set density (km-2) |  |
| lengths\_cap = 500, | Maximum number of lengths to collect / set |  |
| length\_group = 1, | Length group bin size for age sampling (cm) |  |
| ages\_cap = 10, | Maximum number of ages to sample / length group / division |  |
| age\_sampling = "stratified") | Controls whether age sampling is length "stratified" or "random" |  |

The function sim\_survey can be used to simulate data from one survey over a population created using sim\_distribution. A default function call is described in Table 4. The sim\_survey 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. Also note that the min\_sets argument imposes a minimum of number of sets to conduct per strata, regardless of its allocation given strata area and set density. This argument imposes a useful constraint for generating data to be analyzed using design-based approaches that require more than one value for the calculation of a mean.

Like sim\_abundance and sim\_distribution, custom closures can be supplied to sim\_survey to impose alternate parametric curves for catchability at age (i.e. a closure including an equation for a dome can be constructed and used in lieu of sim\_logistic to impose a dome-shaped catchability). Multiple simulations of the same survey can be run using the n\_sims argument, however, requesting large numbers of simulations can be computationally demanding depending on the processing capacity available. Below we use sim\_survey to simulate two surveys over a default population, of which one is set-up to have higher set density (set\_den) 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 (abundance at age matrix multiplied by survey catchability), 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 [16] and crosstalk [19] 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 (Figure 4).

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

Figure 4: Bubble plots of abundance and histograms of set catches from a simulated stratified-random survey of a default population under relatively a) low and b) high sampling effort. Point size and color are scaled by abundance in the bubble plots. Histograms of length and age composition include the distribution of all fish caught next to those sampled overlaid with a line of the true distribution of lengths 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, available RAM 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 this process. This function is set-up to run multiple sim\_survey calls in parallel using the **doParallel** package [20] 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 in a given computer. 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.

## run\_strat

Stratified estimates of abundance are obtained 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 survey over a population are obtained using the following code (using default values):

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).

## test\_surveys

Assuming a stratified analysis as the default method for obtaining an index of abundance, a series of survey protocols can be tested using the test\_surveys function. Provided a simulated population from sim\_distribution and a series of survey protocols 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 less demanding 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 code below will replicate the results from our case study (see **[S1 Appendix](#s1-appendix-case-study)** for more detail). The expand\_surveys function sets up a series of 175 surveys to test (i.e. all possible combinations of the set\_den, lengths\_cap and ages\_cap vectors) 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 = 3)

Processing time will be 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 [21], which details percent completion and will also include an estimate time of arrival (eta) after the first step of the loop completes. 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 to 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 (e.g. 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 thick black line indicates the true trend in the total population available to the survey and the yellow to purple 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 the maximum number of length samples, . The thick black line indicates the true trend in the total population available to the survey and the yellow to purple 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 the maximum number of length samples, . Number of ages sampled per length group, , was 10 in all scenarios. The thick black line indicates the true trend in the total population available to the survey and the yellow to purple 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 which\_strat argument, the plot can be focused on total, length or age based stratified results (Figure 8). The plot\_error\_surface displays 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; Figure 9).

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 protocols (set density [] and maximum number of lengths measured per set []). Records are ranked by lowest to highest RMSE score. Within each plot, a yellow to purple color gradient is applied from lowest to highest value as an additional visual aid. Note that the exponent format of the axes defaults to SI unit symbols (e.g. M for million).

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 RMSE scales are different across panels. Note that the exponent format of the axes defaults to SI unit symbols (e.g. M for million).

# Parameterisation

The default parameter values used in this package were chosen to emulate survey data from a specific cod population (see **[S1 Appendix](#s1-appendix-case-study)** for details) and, as such, these values ought to provide a good place to start for another population of cod or similar groundfish. In this section we outline a series of recommended steps to take in order to tailor the default settings to suit other systems. Throughout this outline, we also provide some context on how default parameter values were chosen as this information may help guide a users’ choices.

## 1. Define survey area and protocol

Parameterising the survey design should be the easiest place to start as the area and protocol should be clearly defined. The survey area can be defined using make\_grid or it can be manually constructed. For the case study, make\_grid arguments were modified to create a grid with similar dimensions to the stratified design actually applied to the case study area (i.e. total area, shelf depth, number of strata, etc.). Also note that the resolution (res argument) of the grid was defined to be similar to the resolution at which sets are allocated in the survey. A square grid was chosen in lieu of the actual survey grid to create an area that avoids potential issues associated with land as a barrier while being sufficiently complex to test survey designs. When setting up a grid, be careful not to drastically increase the dimensions (x\_range or y\_range arguments) or resolution (res argument) as simulating the spatial noise over a large field can be computationally demanding.

With a survey grid defined, the settings of sim\_survey can be modified to match the protocol of a users’ survey of interest. For the case study, the trawl dimensions (trawl\_dim) were modified to match the dimensions of a standard trawl conducted in the region. Likewise, area is used to define the number of sets to allocate to a strata and the value used by default for set\_den matches the protocol, as does the min\_sets setting (i.e. at least two sets are conducted in each strata to facilitate stratified analyses). Finally, length and age sub-sampling settings were modified to match the protocol for the survey of cod in the case study area. The most difficult parameter to define in the sim\_survey function is q. A logistic curve is used by default to define these unknown values and the sim\_logistic parameters were tweaked to approximate the catchability curve supplied to the survey-based model that was used to assess the case study population [22]. A similar tactic can be applied to other cases with estimates of survey catchability.

## 2. Define abundance and growth parameters

Abundance and growth parameters are more difficult to define than the survey protocol as the true values are unknown. Nevertheless, spatially-aggregated age-structured models are frequently applied to assess fish populations and estimates from such models can be used to define sim\_abundance argument settings. Alternatively, key values from these models (e.g. Z and N matrices) can be used instead of simulated values. Default parameters used in sim\_abundance were largely inspired by estimates from the survey-based model used to assess the stock [22]. Based on this work, average total mortality was roughly centered around 0.5 (; argument log\_mean in sim\_Z closure) and we allowed it to make similar changes in magnitude through time and across ages by setting the standard deviation of log total mortality to 0.2 (; argument log\_sd in sim\_Z closure) and the correlations to 0.9 and 0.5 in the age and year dimensions ( and ; phi\_age and phi\_year), respectively. Likewise, sim\_R values were modified to generate similar patterns in recruitment as those estimated by the model. Ages groups to simulate across ranged from age 1 to 20 (; ages argument) to capture the tails of the population, and 20 years were simulated (; years argument) to capture some dynamics. Finally, von Bertalanffy growth parameters were roughly based on estimates from cod in the region [23]. While the choices made are obviously imperfect and ad hoc, it is not necessary to generate a perfect representation of the population as the goal here is to generate a dynamic population to survey. What is key is that we work with a known population from which we can evaluate the performance of specific survey designs and analyses.

## 3. Define spatial parameters

Ideally this step would be no more difficult than step 2 as the inputs required could come from a spatially-explicit age-structured model fit to real data. Such estimates, however, are rare and we can confirm that no such model has been developed and fit to data from our case study population. Unless a user has access to such estimates, parameter estimates will need to manually defined using visual comparisons of real and simulated data. A user will have a first-cut simulation of some survey data by following steps 1 and 2, and the plotting functions provided with the package will provide a means of visually exploring these data. For comparison purposes, actual survey data will need to be processed and visual patterns in the spatial distribution of each age will need to be inspected. It was in this step that we noticed that ages 1 to 4 tend to occupy different areas while ages 5+ occupy similar space for the cod population we focused on. This observation motivated the coupling of the spatial noise of ages 5+ (group\_ages argument in the sim\_ays\_covar closure), which forces this component of the population to occupy the same space, and the moderate value of 0.5 for the correlation in the spatial distribution across ages (; phi\_age argument in the sim\_ays\_covar closure). Year-to-year plots of the real data also revealed considerable inertia in the space occupied by each age group; this observation motivated a fairly high value of 0.9 for the correlation in the spatial distribution across years (; phi\_year argument in the sim\_ays\_covar closure). Finally, the range of the spatial correlation (; range argument in the sim\_ays\_covar closure) was iteratively modified until the simulated population exhibited a similar level of patchiness as the real data, and the standard deviation of the age-year-space process (; sd argument in the sim\_ays\_covar closure) was increased to generate set catches with tails similar to those observed in the real data (cod catches in the area tend to be zero-inflated and heavy-tailed). Finally, visual inspection of the real data revealed that the largest catches tended to be centered on sets conducted in 200 m of water; a parabolic relationship, defined using the sim\_parabola closure, with a mean of 200 m and standard deviation of 70 m approximated this pattern.

# Research opportunities

The case study described in **[S1 Appendix](#s1-appendix-case-study)** provides one example of how **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 [24]. Nevertheless, there has long been interest in using model-based approaches to improve abundance estimates [11,25,26]. **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 [27]. Results from the case study (see **[S1 Appendix](#s1-appendix-case-study)**) reiterate this point and **SimSurvey** may serve as a platform for testing potential model-based solutions to this problem [27].

## 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. [6].

# 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 will 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 and correlation across ages, years and space can be defined. 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 steps 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.

A large number of statistical models that may be tested using these simulated data, but, 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 hinge 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

This work has benefited from valuable feedback from numerous colleagues, including Aaron Adamack, Alejandro Buren, Noel Cadigan, Karen Dwyer, Geoff Evans, Paul Higdon, Danny Ings, Mariano Koen-Alonso, Joanne Morgan, Derek Osborne, Pierre Pepin, Dwayne Pittman, Don Power, Craig Purchase, Martha Robertson, Mark Simpson, Brad Squires, Don Stansbury and Peter Upward. We also thank Dave Cote and Joanne Morgan for providing constructive comments on a previous version of this manuscript. Finally, the package and the manuscript were greatly improved by feedback from two anonymous reviewers. This work was supported by the NSERC visiting-fellow program and Fisheries and Oceans Canada.

# References

1. Pennington M, Strømme T. Surveys as a research tool for managing dynamic stocks. Fisheries Research. 1998;37: 97–106. doi:[10.1016/S0165-7836(98)00129-5](https://doi.org/10.1016/S0165-7836(98)00129-5)

2.

3.

4.

5. Aanes S, Vølstad JH. Efficient statistical estimators and sampling strategies for estimating the age composition of fish. Canadian Journal of Fisheries and Aquatic Science. 2015;72: 938–953. doi:[10.1139/cjfas-2014-0408](https://doi.org/10.1139/cjfas-2014-0408)

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

7. Schnute JT, Haigh R. A simulation model for designing groundfish trawl surveys. Canadian Journal of Fisheries and Aquatic Sciences. 2003;60: 640–656.

8.

9.

10.

11.

12.

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

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

15. Hijmans RJ. Raster: Geographic data analysis and modeling [Internet]. 2016. Available: <https://CRAN.R-project.org/package=raster>

16. Sievert C. Plotly for r [Internet]. 2018. Available: <https://plotly-book.cpsievert.me>

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

18. Bache SM, Wickham H. Magrittr: A forward-pipe operator for r [Internet]. 2014. Available: <https://CRAN.R-project.org/package=magrittr>

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

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

21. Csárdi G, FitzJohn R. Progress: Terminal progress bars [Internet]. 2016. Available: <https://CRAN.R-project.org/package=progress>

22. Ings D, Rideout R, Rogers R, Healey B, Morgan M, Robertson G, et al. Assessing the status of the cod (*Gadus morhua*) stock in NAFO Subdivision 3Ps in 2018. DFO Can Sci Advis Sec Res Doc. 2019;2019/069: v + 75p.

23. Cadigan N, Konrad C. A cohort time-series Von Bertalanffy growth model for Northern cod (*Gadus morhua*), and estimation of the age of tagged cod. DFO Can Sci Advis Sec Res Doc. 2016;2016/017: v + 37p.

24. Cochran WG. Sampling techniques. 3rd ed. John Wiley & Sons; 1977. p. 428.

25. Smith SJ. Use of statistical models for the estimation of abundance from groundfish trawl survey data. Canadian Journal of Fisheries and Aquatic Sciences. 1990;47: 894–903.

26. Berg CW, Nielsen A, Kristensen K. Evaluation of alternative age-based methods for estimating relative abundance from survey data in relation to assessment models. Fisheries Research. 2014;151: 91–99. doi:[10.1016/j.fishres.2013.10.005](https://doi.org/10.1016/j.fishres.2013.10.005)

27. Berg CW, Kristensen K. Spatial age-length key modelling using continuation ratio logits. Fisheries Research. 2012;129-130: 119–126. doi:[10.1016/j.fishres.2012.06.016](https://doi.org/10.1016/j.fishres.2012.06.016)

28. Sutherland WJ. Ecological census techniques: A handbook. Cambridge University Press; 2006.

29. Dahlgren CP, Eggleston DB. Ecological processes underlying ontogenetic habitat shifts in a coral reef fish. Ecology. 2000;81: 2227–2240.

30. Galaiduk R, Radford BT, Saunders BJ, Newman SJ, Harvey ES. Characterizing ontogenetic habitat shifts in marine fishes: Advancing nascent methods for marine spatial management. Ecological Applications. 2017;27: 1776–1788.

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

32. Booth AJ. Incorporating the spatial component of fisheries data into stock assessment models. ICES Journal of Marine Science. 2000;57: 858–865.

33. Bogstad B, Pennington M, Vølstad JH. Cost-efficient survey designs for estimating food consumption by fish. Fisheries Research. 1995;23: 37–46. doi:[10.1016/0165-7836(94)00341-S](https://doi.org/10.1016/0165-7836(94)00341-S)

34. Coggins LG, Gwinn DC, Allen MS. Evaluation of Age-Length Key Sample Sizes Required to Estimate Fish Total Mortality and Growth. Transactions of the American Fisheries Society. 2013;142: 832–840. doi:[10.1080/00028487.2013.768550](https://doi.org/10.1080/00028487.2013.768550)

35. Zhang Y, Cadrin SX. Estimating Effective Sample Size for Monitoring Length Distributions: A Comparative Study of Georges Bank Groundfish. Transactions of the American Fisheries Society. 2013;142: 59–67. doi:[10.1080/00028487.2012.722167](https://doi.org/10.1080/00028487.2012.722167)