

# The nepva R package User Manual

## The nepva R package, Version 4.13: User Manual

Adam Butler (Biomathematics and Statistics Scotland), Deena Mobbs, Francis Daunt & Kate Searle (UK Centre for Ecology and Hydrology)

21 January 2021

Developed under contracts for Natural England and JNCC

## Contents

Contents.....	1
Note.....	3
1 Introduction .....	3
2 Overview of methodology.....	4
3 Dependencies .....	5
4 nepva.fullrun .....	5
4.1 Inputs: model structure and dimensions.....	5
4.2 Inputs: format for specifying PVA .....	9
4.3 Inputs: baseline demographic rates .....	9
4.4 Inputs: initial population sizes.....	10
4.5 Inputs: scenarios of anthropogenic impacts .....	11
4.6 Inputs: format of outputs .....	12
4.7 Generating projections .....	13
4.8 Outputs .....	13
5 nepva.simplescenarios.....	14
6 nepva.validation .....	15
7 nepva.sensitivity.local .....	15
8 nepva.sensitivity.global .....	16
9 Calculating default values for inputs.....	16
9.1 Age at first breeding and maximum brood size .....	17
9.2 Survival rates .....	17
9.3 Productivity.....	18
10 Updating the package.....	19
10.1 Adding additional models for density dependence.....	19

## Description of the nepva R package, Version 4.13

10.2	Updating look up tables with demographic information.....	20
10.2.1	Species meta-data .....	20
10.2.2	Survival data.....	20
10.2.3	Breeding success.....	20
10.2.4	General notes on updating files.....	20
References.....		21

## Note

Although this is referred to as an “R package” throughout this documentation, this is not strictly true – it is currently a suite of R functions, and associated examples, that could readily be compiled into an R package, and we envisage that this will be done when possible (the only changes required are standardisation of the function documentation to meet the requirements for R packages). For all practical purposes, however, it operates in the same way as an R package.

## 1 Introduction

The **nepva** R package is designed to undertake population viability analysis (PVA) for seabird species. The development of the package constitutes part of project (NE: ITT\_4555 and JNCC: C19-0287-1259 PVA tool Phase II); the R package underpins the functionality of the user-friendly web-based Shiny tool that has been developed within this project. The technical functionality of the Shiny tool implements a subset of the functionality within the R package: the Shiny tool provides an interactive web-based mechanism for users to provide inputs to the R package, and to display and save the results obtained by running the R package (without the need for users to install or use R). To see a full description of the reduced functionality of the Shiny tool when compared to the **nepva** R package, see the guidance documentation for the Shiny tool available on the GitHub repository [Seabird\\_PVA\\_Tool](https://github.com/naturalengland/Seabird_PVA_Tool)<sup>1</sup>. .

The **nepva** R package operates as any other R package, requiring users to be proficient in the use of the R software environment, running code and entering required inputs to the various functions within the package itself. This is not the case for users with the Shiny tool, where no knowledge of R is required, and users interact with the tool entirely through the user interface.

The R package is motivated by the use of PVAs to assess the impacts of proposed offshore renewable energy developments (ORDs) upon seabird populations, but the functionality within the package is sufficiently general that the package could also be used for assessing the population-level consequences of other anthropogenic activities that impact on population demographic rates, including for example, positive management effects.

This document outlines the methodology used within the R package, the inputs that the tool requires, and the outputs that it produces. The R package contains five main functions that are called directly by the user:

- a) a function **nepva.fullrun** that is used to generate PVAs, based upon demographic rates, initial population size(s), and scenarios of anthropogenic impact;

---

<sup>1</sup> [https://github.com/naturalengland/Seabird\\_PVA\\_Tool](https://github.com/naturalengland/Seabird_PVA_Tool)

- b) a function **nepva.simplescenarios** which is similar to “nepva.fullrun” but includes only the options that are available with the corresponding Shiny tool;
- c) a function **nepva.getdefaults** that is used to calculate default values of demographic rates (productivity and survival) for specific species and regions/breeding colonies (within the British Isles), using existing data.
- d) a function **nepva.validation** that can be used to run PVAs retrospectively, for comparison against observed counts
- e) a function **nepva.sensitivity.local** that runs local sensitivity analyses associated with the PVA.
- f) a function **nepva.sensitivity.global** that runs global sensitivity analyses associated with the PVA.

The remainder of this document is structured as follows:

- Section 2 provides a brief introduction to the methods that are implemented within the tool
- Section 3 describes the dependencies of the R package
- Section 4 describes the inputs to the function **nepva.fullrun**
- Section 5 explains the calculations that are used in generating the PVA given these inputs
- Section 6 outlines the outputs that are produced by this function
- Section 7 describes the function **nepva.getdefaults**: how the function works, and how the default values of the inputs are calculated from existing data.
- Section 8 describes the function for performing validation
- Section 9 describe the functions for running sensitivity analyses.
- Section 10 outlines how the R package could be updated in future.

## 2 Overview of methodology

The **nepva** tool is designed to generate projections of the future size of a specific population of seabirds, under one or more scenarios of anthropogenic change (typically, the impact of an offshore renewable energy development, although the framework permits other impacts - including positive impacts - to also be considered).

Underpinning the tool is a Leslie matrix model. The tool allows either deterministic or stochastic Leslie matrix models to be used, allows either environmental or demographic stochasticity, or both, to be considered, and allows a range of potential distributions for dealing with both forms of stochasticity. The models also allows models of density dependence to be used.

The Leslie matrix model has a relatively general structure, but it does make some simplifying assumptions:

- a. it assumes that reproductive rates are independent of the age of the adult, once the adult has reached the “age at first breeding” (which is assumed to be a fixed and known value); and
- b. it assumes an equal sex ratio.

A graphical summary of the basic Leslie matrix structure is shown in Table 1.

The model assumes that the scenarios involve a shift in the demographic rates (both productivity and age-specific survival), relative to their baseline values, for years throughout a user-specified period of impact. Note that impacts upon only survival or productivity are easily accommodated within this approach, by setting the impacts relating to the other demographic rate equal to zero.

Simulations under impact scenarios are matched with simulations under the baseline, in order to accurately represent the uncertainty associated with the impact of each scenario. “Matching” means that the stochastic decisions within baseline and impacted scenarios are the same (i.e., technically speaking, it means that the same simulated random numbers are used in generating PVAs under all scenarios).

## 3 Dependencies

The **nepva** package was developed using Version 3.5.1 of R, and utilises two add-on packages: **mvtnorm** (it was developed using Version 1.0-8) and **popbio** (it was developed using Version 2.4.4).

## 4 nepva.fullrun

The main functions for running PVAs within the R package are **nepva.fullrun** and **nepva.simplescenarios**. They are identical except that “nepva.simplescenarios” excludes options (e.g. the specification of correlations between demographic rates) that are not included in the Shiny tool. **nepva.fullrun** provides the most flexible framework for running PVAs.

The inputs to the tool are divided into six blocks.

### 4.1 Inputs: model structure and dimensions

The first block of inputs that the tool requires relate to the basic structure and dimensions of the Leslie matrix model.

The first four inputs within this set define the parametric model that is used for simulating the values of the demographic rates (productivity and age-specific survival) in each year. The inputs specify the model used to describe density dependence (**trans.dd** and **model.dd**), the model used to describe environmental stochasticity (**model.envstoch**), and whether the model constrains productivity rates to be less than or equal to the maximum brood size (**model.prodmax = TRUE**) or allows them

to be unconstrained (**model.prodmax = FALSE**). Five possible models for density dependence are currently implemented (for technical details see Table 2); these are:

1. **model.dd = "nodd"** : applies density independence (i.e. no density dependence);
2. **model.dd = "ddulinear"** : implies that the effects of density dependence on the demographic rates are linearly proportional to population size;
3. **model.dd = "dduloglin"** : implies that the effects of density dependence on the demographic rates are a linear function of  $\log_{10}(\text{population size})$ ;
4. **model.dd = "dduweibull"** : this is a generalization of the "ddulinear" model in which density dependence is assumed to be equal to a power of population size, where the value of this power is one of the parameters specified by the user;
5. **model.dd = "dduthresh"**: this model assumes that the effect of density dependence is linearly proportional to population size above a threshold, and is independent of population size below this threshold. The threshold is one of the unknown parameters of the model, whose value can be specified by the user.

The input **trans.dd** specifies how these effects operate. If **trans.dd = FALSE** these models all apply to untransformed demographic rates, but this means that they may potentially simulate invalid rates (e.g. survival rates greater than 1, or less than 0, or productivity rates less than zero). If **trans.dd = TRUE** density dependence is applied to transformed versions of the demographic rates – this guarantees that the model will only ever simulate valid demographic rates. Specifically, when **trans.dd = TRUE** the density dependence model is assumed to apply to  $\text{logit}(\text{survival rate for each age})$ , and to apply to  $\text{logit}(\text{productivity})$  if **model.prodmax = TRUE** or  $\text{log}(\text{productivity})$  if **model.prodmax = FALSE**. There are theoretical reasons to prefer the transformed version (**trans.dd = TRUE**), but the parameters of the model are much easier to interpret when **trans.dd = FALSE**; this is why we provide the option to use either version.

Three possible models for environmental stochasticity are implemented:

- a. **model.envstoch = "deterministic"** : implies that environmental stochasticity is ignored (i.e. the magnitude of variation due to environmental stochasticity is assumed to be zero);
- b. **model.envstoch = "betagamma"** : implies that survival rates have a beta distribution; implies that the productivity rate has a beta distribution if **model.prodmax = TRUE** and a gamma distribution if **model.prodmax = FALSE**;

- c. **model.envstoch = "logitnlogn"** : implies that survival rates have a logit-normal distribution; implies that the productivity rate has a logit-normal distribution if **model.prodmax = TRUE** and a log-normal distribution if **model.prodmax = FALSE**. Note that the structure of this model (i.e. the fact that the logit-normal model is not easily parameterized in terms of a mean and variance on the untransformed scale) means that it can only be used with the transformed version of density dependence (**trans.dd = TRUE**).

We also allow two possible sets of distributions to be considered for environmental stochasticity (beta/gamma or logit-normal/lognormal) because all of these models are widely used in practice to represent the distributions of positive variables (gamma, lognormal) or the distribution of variables constrained to lie on an interval (beta, logit-normal). These seem likely to be the most popular choices to use in practice, which is why we have included them as options here; including two potential models for each type of variable also potentially allows users to assess the sensitivity of their results to the choice of distribution. It is not straightforward to predict the likely sensitivity of the PVA outputs to the choice of distribution.

Note that the beta and gamma distributions are parameterized in terms of a mean  $\mu$  and dispersion parameter  $\phi$ ; these are the parameterizations that are usually used in beta regression and gamma regression, but differ from the standard parameterizations used in R. They can easily be converted back to the standard parameterizations, however, via:

**Beta distribution:**  $\theta = (\mu\phi, (1 - \mu)\phi)$

**Gamma distribution:**  $\theta = (\mu/\phi, \phi)$

where  $\phi$  represents the scale parameter of each distribution.

The fourth input, **model.demostoch**, specifies whether demographic stochasticity is included in the model. If **model.demostoch = FALSE** then the survival and productivity rates are translated into actual numbers of birds surviving and born each year by simply multiplying the rates by the relevant population sizes, and then rounding to the nearest whole number. If **model.demostoch = TRUE** then the number of birds surviving (of each age) is simulated from a binomial distribution, and the number of chicks fledged is simulated either from a binomial distribution (if **model.prodmax = TRUE**) or a Poisson distribution (if **model.prodmax = FALSE**).

Users also specify the maximum brood size (**mbs**). If **model.prodmax = TRUE** then productivity values are constrained to be less than or equal to the value of **mbs**; if **model.prodmax = FALSE** then the value of **mbs** is ignored within the modelling, and can safely be left blank (i.e. **mbs = NULL**). The next input, **afb**, represents the age at first breeding – beyond this age all demographic rates (survival and productivity) are regarded as being independent of age (i.e., 'adults').

Users must specify the number of subpopulations to produce projections for, **npop**. This will be equal to one, by default, but by specifying **npop** greater than one it is possible for users to construct projections for multiple subpopulations – the tool will then combine these together before producing PVA metrics and other outputs. This functionality may be useful, for instance, if interest involves generating a PVA for an SPA or region, but the SPA/region contains multiple count units (e.g. Seabird

Monitoring Programme (SMP) sites), potentially with different demographic rates across units – it is likely to be particularly useful in situations where the most recent counts for the individual populations relate to different years, making it difficult to sum the initial population sizes of the populations to generate an initial population size for an SPA or region.

The user also needs to specify the number of impact scenarios, **nscen**, that they are interested in. Each “scenario” is assumed to represent a hypothesised scenario for a specific change in demographic rates over a specified future period, for instance a change in survival rate or productivity. Scenarios are assumed to apply in an identical way to all sub-populations being considered. “Scenarios” are likely to relate, in reality, to the impacts of different ORD footprints, but they could relate, in principle, to any scenario of change in future demographic rates (e.g. due to changes in other anthropogenic impacts, e.g. culling, bycatch, as well as management interventions designed to improve population status). Users also need to specify, using **include.baseline**, whether a “baseline” scenario should be included, in addition to the scenarios specified by the user. If “nscen = 0” and “include.baseline = TRUE” then a baseline-only PVA is generated: this will be useful, for example, for validation purposes.

Finally, within this block of inputs, the user needs to specify the number of simulations **sim.n** and can, optionally, specify the seed to be used for random number generation **sim.seed** (if the value of **sim.seed** is left at the default value, NULL, it is chosen automatically) The values of the seed determines the exact sequence of random numbers that R chooses during random number generation, so the option to specify **sim.seed** manually is designed to ensure that the exact results of the PVA can be reproduced again in future. If **sim.seed** is specified by the user then it will be possible for the exact results to be reproduced again, simply by specifying the same value of **sim.seed** again; if **sim.seed** is allowed to be chosen automatically then it will be effectively impossible to reproduce the results of a stochastic PVA again automatically). The actual values that are allocated to **sim.seed** are essentially arbitrary, and have no meaningful interpretation – the sole point of the seed is to ensure that the results can be reproduced by running them again with the same seed.

The package automatically determines whether a deterministic PVA is being run (if **model.envstoch** = “**deterministic**”, **model.demostoch** = **FALSE** and **scens.sds** are NULL or all equal to zero); if it is, the user-specified value of **sim.n** is ignored and **sim.n** is forced to be equal to one (for deterministic PVAs it is not necessary to produce multiple simulations, as every simulation will lead to an identical result). If a stochastic PVA is being run, then the user can select the value of **sim.n** themselves; it is not recommended to use small values of **sim.n** because PVAs based on small numbers of simulations are likely to be unreliable (using a value of less than 1000 will generate a warning message, but in practice the minimum number of simulations may need to be substantially higher than this in order to achieve reliable results), and PVAs based on very large numbers of simulations will make the tool very slow to run (so using a value of more than 100000 also generates a warning message). The ability to specify **sim.seed** explicitly is intended so that the results of previous PVAs can be reconstructed exactly; the values produced by a stochastic PVA should be exactly equal to those produced previously.



## 4.2 Inputs: format for specifying PVA

The next block of inputs specify the way in which subsequent inputs will be provided. The inputs in this block are all binary (yes/no).

**demobase.splitpops** determines whether users provide baseline demographic rates separately for each subpopulation (**demobase.splitpops = TRUE**) or assume that the same rates hold for all subpopulations (**demobase.splitpops = FALSE**). If `npop = 1` this input has no effect. Similarly, **impacts.splitpops** specifies whether impacts are specified separately for each subpopulation, or whether they are specified as an effect that is common to all subpopulations.

The next three inputs – **demobase.splitimmat**, **inipop.splitimmat** and **impacts.splitimmat** - determine whether users provide baseline survival rates, initial population sizes and impacts separately for immatures, or whether immatures are assumed to have the same values as breeding adults. If **impacts.splitimmat** is set to be **FALSE** then the same proportional impact on demographic rates is assumed for immatures as for adults.

## 4.3 Inputs: baseline demographic rates

The next block of inputs involve specifying the current, baseline demographic rates.

The demographic rates themselves are: productivity (**demobase.prod**), adult survival (**demobase.survadult**, corresponding to annual survival for adults of age **afb** or above), if **demobase.splitimmat = TRUE**, immature survival rates (for each age from 0 to **afb**).

The user has the option to specify the values for each of these demographic rates in terms of either (a) estimates from a parametric model (**demobase.specify.as.params = TRUE**), or (b) empirical summaries (**demobase.specify.as.params = FALSE**). For the latter option, the user specifies the mean and inter-annual standard deviation of each rate, and, where relevant, the effect of density dependence on the mean rate, and the tool calculates the parameter estimates associated with these values via moment matching. Note that this option is only available when **dd.trans = FALSE**, since the parameters of the transformed versions of the models cannot readily be derived from the empirical summaries.

Moment matching involves calculating the parameter values that are associated with a particular moment of the data – in our context we focus upon the first two moments (the mean and variance – or, equivalently, the mean and standard deviation). For the deterministic model (**model.envstoch = “deterministic”**) no moment matching is necessary. For the beta and gamma distributions (**model.envstoch = “betagamma”**) there are simple formulae for calculating the parameter values associated with particular values for the mean and SD (Table 4).

The inputs that need to be specified, and the order in which they need to be provided, are described in Table 5.

Users have the option to specify correlations between demographic rates, through the input **demobase.cormat**, although they can also leave this option blank (by specifying **demobase.cormat = NULL**) in which case all correlations between demographic rates are assumed to be zero. If users do specify correlations, then they specify the full correlation matrix between all demographic rates: i.e. a matrix of dimension  $[amax+2, amax+2]$ . The first demographic rate is productivity; the remaining rates are age-specific survival rates, arranged in ascending order of age. The tool checks whether this is a valid correlation matrix by examining whether all of the eigenvalues of the matrix are positive; if any eigenvalues are negative, indicating an invalid correlation matrix, then the tool returns an error message.

Users also have the option to specify the percentage of adult breeding birds that skip breeding in any particular year, **demobase.bskippc**. If this is NULL (the default) then this percentage is assumed to be zero; otherwise, the user needs to specify two values, corresponding to the mean and standard deviation of this percentage.

## 4.4 Inputs: initial population sizes

The tool allows users to specify the initial population size for each age (with numbers of ages greater than  $A$  being aggregated together), by specifying **inipop.agetype = "separate"** and then specifying values for each age within **inipop.vals**.

We allow users to specify initial population sizes using one of three possible ways: they can use **inipop.inputformat = all.individuals** to specify the overall initial population size,  $I$ , that has been aggregated across all ages, they can specify **inipop.inputformat = breeding.adults** and give the overall total number of breeding adults, or they can specify **inipop.inputformat = breeding.pairs** and give the total number of breeding pairs.

It is possible for users to specify initial population sizes separately for each immature age, by specifying **inipop.splitimmat = TRUE** and then providing these values in **inipop.immatvals**. In many situations, however, it will not be feasible for users to specify the breakdown of the initial population size by age. In this situation the tool automatically calculates the breakdown of this population into ages, using the following approach:

- a deterministic Leslie matrix model which is approximately equal to the actual model being used is constructed; for models containing density dependence the population size  $x$  is assumed equal to the total initial population size  $I$ ;
- the stable age structure associated with this Leslie matrix is calculated to be the eigenvector of the Leslie matrix associated with the largest eigenvalue, rescaled so that the proportions sum to one (Caswell, 2000)
- the stable age structure is assumed to give the proportion of individuals that are of each age; this is then multiplied by the total initial population size (if "inipop.agetype = all") or multiplied by the total population size divided by the total proportion of the population belonging to breeding adults (if

“`inipop.agetype = adults`”), and rounded to the nearest integer, in order to give the initial population size for each age.

The calculation of the stable age structure – step (b) – is a standard calculation in population modelling (Caswell, 2000), and is implemented via the **stable.stage** in the R package **popbio**. For step (a) we approximate the Leslie matrix of interest by a simplified Leslie matrix model in which demographic stochasticity is ignored, and environmental stochasticity is averaged (i.e. where the probability distribution associated with environmental stochasticity is used to derive the expected demographic rate, rather than being simulated from). The calculations involved in averaging across environmental stochasticity in this way follow directly from standard formulae for the expected value of the beta, gamma and log-normal distributions. An analytic formulae for the expectation of the logit-normal distribution does not exist, so in this case we calculate the expected value by repeatedly sampling from the distribution, and calculating the mean of these values.

The use of the stable age structure means that the initial age structure may differ from the age structure that emerges from running the PVA simulations, because the stable age structure only applies asymptotically, and so will not correspond to the age structure in a finite simulation/projection. In addition, for stochastic PVAs the stable age structure is calculated for a rather different model from that being considered for the PVA calculations (e.g. deterministic rather than stochastic). To deal with any potential bias these issues may create in the age structure within the simulations, we include the functionality for a “burn-in” period to be used – users specify this by providing **nburn**, the length of the burn-in period, in years. If **nburn = 0** no burn-in is applied.

The burn-in involves running baseline PVA simulations for **nburn** years, and outputting the age structures that are obtained at the end of this period. This age structure is then used as the initial age structure within the main PVA runs. The burn-in run, and main PVA run are identical except in the way that the initial age structure is specified.

If the PVA model simulates that the population will go extinct during the burn-in period an error message is produced, because in this case it is impossible to calculate the age structure using the burn-in period specified by the user.

Users also need to specify the year associated with the initial count for each population (**inipop.years**)

Users can specify initial values for multiple populations (if **npop** is greater than one) – the outputs from the projections for these populations will be summed together before producing an overall PVA for the combined population.

## 4.5 Inputs: scenarios of anthropogenic impacts

The next block of inputs relates to the scenarios of change in demographic rates when impacts are applied to the population. Users begin by specifying the year in which impacts are assumed to begin (**impacts.year.start**); users may also, optionally, specify the year in which impacts are assumed to end (**impacts.year.end**), but if this is left blank then the impacts are assumed to continue until the end of the period being considered within the PVA.

Users then specify **impacts.scennames**: these are simply labels to be used for referring to each of the **nscen** scenarios within the outputs. Any text strings can be used for these, including spaces and punctuation, although it is important that they do not contain commas (as this will cause problems in the generation of CSV files).

Users do not explicitly need to include a “baseline” scenario: the package will automatically add this (so long as **include.baseline = TRUE**).

Users next specify the type of impact, via **impacts.relative**. If **impacts.relative = FALSE** users specify the absolute number of birds that will be killed annually, within each scenario – this corresponds to a fixed number killed per year (during the period for which the impact is specified), regardless of the population size. This option may be relevant for, e.g., mechanisms such as culling, but would only be relevant for ORD impacts in very specific circumstances. If **impacts.relative = TRUE** users specify the change in annual demographic rates (productivity and survival) associated with each scenario, i.e. the value of (demographic rate under baseline– demographic rate under scenario) for each demographic rate. This has the effect of scaling the effect of the impact according to total population size in each year which has been the approach typically taken in ORD assessments. Note that it is (baseline – scenario), so a positive impact corresponds to a reduction in the demographic rates under the scenario (this makes it consistent with the direction of effects when **impacts.relative = FALSE**). If, for example, the baseline adult survival rate is 0.86, and the impact on adult survival is specified to be 0.01, then the impacted survival rate will be 0.85. If, by contrast, the baseline survival rate is 0.86 and the impact on adult survival is specified to be -0.01, then the impacted survival rate will be 0.87.

At the next step users specify the impacts of each scenario upon productivity (**impact.prod.mean**), upon adult survival (**impacts.adultsurv.mean**, and, if **impact.splitimmats = TRUE**, upon immature survival (**impacts.immatsurv.mean**). Impacts need to be specified separately for each population if **impacts.splitpops = TRUE**, or as a single value relating to all populations if **impacts.splitpops = FALSE**.

If **use.baseline = TRUE** the tool will automatically add an additional scenario, “baseline”, in which the values of **impacts.adultsurv.mean**, **impacts.immatsurv.mean** and **impacts.prod.mean** are zero for all demographic rates.

Finally, users also have the option to specify the standard error associated with each impact under each scenario, if they specify **impacts.provideses = TRUE**, via **impacts.adultsurv.se**, **impacts.prod.se** and **impact.immatsurv.se**. As with the mean impacts, users need to specify values for SEs under each scenario, for each demographic rate. If **impacts.provideses = FALSE** the uncertainty associated with the magnitude of the impacts of each scenario is ignored (or, equivalently, assumed to have a magnitude of zero).

## 4.6 Inputs: format of outputs

The final block of inputs specify the format of the outputs to be produced by running the PVA; these are described in the section below, alongside the description of outputs.

Users also specify whether error messages and graphs should be produced when running the tool (**silent = FALSE**), or whether these should be suppressed (**silent = TRUE**). We envisage that the former option will usually be used, but the latter option may be useful in situations where a large number of PVAs are being run automatically.

Users can either opt to produce only tabular outputs (**output.raw = FALSE**) or to also output the raw simulations (**output.raw = TRUE**).

Finally, users can choose to either change the column names within the output to match those used in the Shiny tool (**changetablenames = TRUE**) or to retain the shorter names that are used internally (**changetablenames = FALSE**). The documentation here refers to the column names specified with **changetablenames = TRUE**; the **changetablenames = FALSE** option is primarily used for internal testing.

## 4.7 Generating projections

The pre-processed inputs yield the set of input values for the main calculations.

The code then loops forward in time, up until **output.year.end**, in order to generate projected values of population size for each age (from 0 to **afb**, and for the set of all individuals of age greater than **afb**), for each population, under each scenario, for each simulation, in each year.

These values are then summed across ages and populations, in order to obtain:

- a) the total number of individuals (of all ages) for each scenario, for each simulation, in each year; and
- b) the total number of breeding adults (of age **afb** or greater), also for each scenario, for each simulation, in each year.

Note that the code for generating projections is designed in such a way as to try and maximise computational speed: this involves vectorising the calculations over **sim.n** (to avoid looping over **sim.n**, which would be likely to be computationally intensive since [a] R is relatively slow at running loops and [b] **sim.n** will often be large, and sometimes very large). The code is also set up in such a way that random number generation is all performed at one place within the code, to avoid running into issues of matching (e.g. if the code is not structured carefully then setting the seed may not be sufficient to ensure that scenario-based runs are correctly matched to baseline runs).

## 4.8 Outputs

The function produces a data frame, containing one row for each combination of scenario, year (from **output.year.first** to **output.year.last**) and age (only relevant if **output.agetype = output.agetype == "ages.separated"**). The columns contain:

- a. an indication of whether the impact is currently active (**Currently.Impacted**), and, if so, the number of years since the impact began (**Impact.year**);
- b. mean, median and standard deviation of projected population sizes;
- c. selected quantiles of projected population sizes: 1%, 2.5%, 5%, 10%, 20%, 25%, 33%, 50%, 66%, 75%, 80%, 90%, 95%, 97.5%, 99%.

- d. six metrics of impacts (Table 6), with mean, median and SDs given for the CGR and CPS metrics. These values will be missing (NA) if `Currently.Impacted=FALSE`.
- e. annualised population growth rate (`Annual_GR`) since the “baseline reference year”: mean, median, standard deviation, and the limits of a 95% confidence interval.
- f. overall population growth (`PGR`) since the “baseline reference year”: mean, median, standard deviation, and the limits of a 95% confidence interval.
- g. percentage population change (`pc_Pop_Change`) since the “baseline reference year”: mean, median, standard deviation, and the limits of a 95% confidence interval.

If `output.agetype = 'breeding.pairs'` then the outputs relate to breeding pairs, if `output.agetype = 'breeding.adults'` the outputs relate to breeding adults (of age `afb` or greater), if “`output.agetype = 'whole.population'`” the outputs relate to the entire population (all age classes combined), and if `output.agetype = 'ages.separated'` the outputs are produced separately for each age, and well as being reported for the whole population.

The impact metrics are summarized in Table 6. Some of the metrics relate to user-specified target population sizes (which the user may optionally specify using `output.popsizetarget`) or a user-specified threshold for quasi-extinction risk (which the user may optionally specify using `output.popsizeqe`). If the user leaves these blank by specifying `output.popsizetarget = NULL` and/or `output.popsizeqe = NULL` then the metrics associated with these thresholds are allocated a missing value within the output table.

## 5 nepva.simplescenarios

The function `nepva.simplescenarios` is a simplified version of `nepva.fullrun`, which only contains the options used in the Shiny tool. Specifically:

- a) it fixes “`trans.dd = FALSE`”, so density dependence can only be specified on an untransformed scale;
- b) it fixes “`include.baseline = TRUE`”, so a baseline run is always included in the PVA;
- c) it fixes “`inipop.splitimmat = FALSE`”, so population sizes of immatures cannot be directly provided by the user;
- d) `demobase.cormat` is `NULL` – i.e. not specified – so variations in demographic rates are assumed to be independent;
- e) `demobase.bskippc` is `NULL` – i.e. not specified – so skipped breeding is not considered.



## 6 nepva.validation

The tool can be used to produce “historical” PVAs, for validation purposes. If “nepva.validation” is used, the inputs are similar to those for “nepva.fullrun”, except that:

- a. In this case only a single subpopulation can be considered (**npop = 1**), a baseline must be included (**include.baseline = TRUE**) and no other scenarios can be considered (**nscen = 0**);
- b. users can specify a historical time series **output.validation.years** and **output.validation.counts**, which the outputs of the PVA are plotted against.

## 7 nepva.sensitivity.local

The package also contains two functions that can be used to run sensitivity analyses. Note that these functions perform sensitivity analysis in the context of the most basic set of options within the nepva package: i.e. a single population (**npop = 1**), a single scenario (**nscen = 1**), and initial population size specified as number of breeding adults. This function essentially runs the PVA for multiple combinations of five key input parameters - initial population size, baseline productivity, baseline adult survival (above **amax**), impact of scenario upon productivity, and impact of scenario upon adult survival – and examines how changes in the values of the input parameters are linked to changes in the value of the PVA outputs (impact metrics). Note that although the sensitivity analysis technically operates on a single scenario, it is effectively looking at the impact of different scenarios because it tests the sensitivity of the impact metrics to variations in the magnitude of impact associated with the scenario.

The first function, **nepva.sensitivity.local**, runs a “local” sensitivity analysis in which the values of each input parameter are altered in turn, and the resulting impact metrics and impacted population sizes calculated. The graph of input against impact metrics and impacted population sizes are shown for each parameter, and associated tables produced. The outputs that are presented can be used to calculate both “absolute” sensitivity (i.e. the actual change in the impact metrics), and “elasticity” (sensitivity, rescaled so as to be comparable across input parameters that are on different scales). The user can specify the number of points to consider for each parameter and direction (**sens.npvlocal**): the default is 1 (i.e. generated one point **sens.pcr%** above the current value of the parameter, and another point **sens.pcr%** below the current value of the parameter), but larger numbers can be used if desired. The range for the sensitivity analysis to explore, **sens.pcr**, is specified by the user, for each of the five key input parameters, as a percentage of the “standard” value of each parameter.

The outputs from **nepva.sensitivity.local** have a similar format to those from the main runs of the R package, but now contain outputs (projected population sizes and impact metrics for scenario versus baseline) for each percentage change in the five input parameters.

## 8 nepva.sensitivity.global

Users also have the option to run a “global” sensitivity analysis, using the function **nepva.sensitivity.global**. This has the advantage over the “local” sensitivity analysis that it accounts for the effects of interactions between input parameters, but the disadvantage that it is typically much more computationally intensive to run.

The global sensitivity analysis involves the user specifying the total number of points to simulate (**sens.npvglobal**), and specifying the ranges (**sens.pcr**) as for the local sensitivity analysis. The function then, internally:

- 1) Independently simulates two set of input parameters  $\theta_A$  and  $\theta_B$  with each parameter in each set being simulated uniformly over the range given for that parameter by **sens.pcr**;
- 2) Forming further sets of inputs  $\theta_k$  for  $k = 1, \dots, 5$  in which the values of the  $k$ -th parameter are taken from  $\theta_B$  and the values of all other parameters from  $\theta_A$ ;
- 3) Run each of these sets of inputs through the PVA in order to obtain values  $f(\theta_A)$ ,  $f(\theta_B)$  and  $f(\theta_k)$  for the output metric being considered.

An estimate of the “first-order sensitivity index” (FOI) of the output metric to each parameter  $k$  is then given by:

$$f(\theta_B)(f(\theta_k) - f(\theta_A))$$

and an estimate of the “total-order index” (TOI, which averages across the effect of higher-order interactions) by

$$\frac{1}{2}(f(\theta_k) - f(\theta_A))^2$$

In practice, this simulation process is repeated a large number of times, specified by **sens.npvglobal**, and the results averaged across simulations.

The output from **nepva.sensitivity.global** is a list, with two elements:

- a. **tab**: a data frame, with a similar structure to that produced by **nepva.sensitivity.local**;
- b. **decomposition**: a data frame with 5 rows, containing the variance decomposition for each of the five input parameters considered in the sensitivity analysis (one per row), for each possible output (metric or projected population size statistic). TEI and FOI values are both given.

## 9 Calculating default values for inputs

The R package also has the functionality to create “default” values for some of the PVA inputs, using empirical data from the British Isles, for each of fifteen species (Arctic Skua, Atlantic Puffin, Black-Legged Kittiwake, Common Guillemot, Common Tern, European Shag, Great Black-Backed Gull, Great Cormorant, Herring Gull,



Lesser Black-Backed Gull, Little Tern, Northern Fulmar, Northern Gannet, Razorbill, Sandwich Tern).

The default survival rates (mean and standard deviation) are derived from Horswill & Robinson (2015), and are taken to be the estimates for the 'National' survival rate, or where available, additional colony-level estimates are included (see Section 9.2).

All breeding success (productivity rates per pair) data were taken from a 2018 snapshot the JNCC's Seabird Monitoring Programme (SMP) dataset.

In order to calculate these values, the user specifies the species of interest, the regional classification and region to be used for pooling information on productivity, and the population to be used for deriving information on survival.

## 9.1 Age at first breeding and maximum brood size

A single default value of **afb** (mean age at first breeding, rounded) and **mbs** (maximum brood size) is selected for each species; these values are given in Table 7, derived from Horswill & Robinson (2015).

## 9.2 Survival rates

Default survival rates (age-specific means and standard deviations) are based upon the overall, national values given in Horswill & Robinson (2015); however, rates are recalculated from the source data, rather than using the published values, in order to adjust for some issues in the calculations of the standard deviations reported in Horswill & Robinson (2015). All differences with Horswill & Robinson (2015) are reported in the 'lookup-surv-metadata.xlsx' file, which can be accessed on the GitHub repository [Seabird\\_PVA\\_Tool](https://github.com/naturalengland/Seabird_PVA_Tool)<sup>2</sup>. The selection of studies is identical to that in Horswill & Robinson (2015), and we also follow the decision of that report to calculate the mean as a weighted mean. Means are weighted by length of time series for some species and ages (adults for: great cormorant, black-legged kittiwake, herring gull, lesser black-backed gull and common tern), whilst a simple (unweighted) mean is used for all other combinations of species and age. Standard deviations are, similarly, either a weighted or an unweighted mean of the standard deviations associated with individual studies. In some cases, the individual studies used by Horswill & Robinson (2015) report standard errors, rather than standard deviations. Where this occurred, we converted these into standard deviations by multiplying the standard error by the square root of the length of the time series (because, at least in simple situations, the standard error of the mean is equal to the standard deviation divided by the square root of the sample size).

For species with sufficient data, the R package also allows the option for default values (mean and standard deviations) to be produced for specific populations. Note that we only allow this option to be available for populations that have a standard deviation associated with them.

---

<sup>2</sup> [https://github.com/naturalengland/Seabird\\_PVA\\_Tool](https://github.com/naturalengland/Seabird_PVA_Tool)

## 9.3 Productivity

The default values for productivity are determined using breeding success data from the Seabird Monitoring Programme. The mean productivity value is determined as in Horswill & Robinson (2015) by:

- a. finding the region associated with the colony (or colonies) of interest, using the pooling region classification specified by the user;
- b. calculating the mean breeding success for each colony that has productivity data within this region; and
- c. taking the mean of these mean values.

The inter-annual standard deviation of productivity is determined as in Horswill & Robinson (2015) to be the maximum of [i] the mean of colony-specific inter-annual standard deviation and [ii] the standard deviation of colony-specific means. Year-by-colony combinations with very small sample sizes (less than 5 nests) are excluded from the calculations. Default values of productivity are only produced for regions if they contain at least 5 years of productivity data (from any colonies: i.e. 5 years of data for a single colony within the region, or 1 year of data for each of 5 colonies within the region, would both be sufficient).

Eight regional classification are considered: “Site” involves solely using data for the site of interest, “global” involves using available data for all Seabird 2000 colonies within the British Isles (England, Northern Ireland, Scotland, Wales, Channel Islands and Isle of Man) and the remaining classifications are based upon pooling within regions – regional definitions that are considered are ICES Regions, JNCC Regional Seas, MSFD regions, Countries (England, Wales, Scotland, Northern Ireland, Channel Islands, Isle of Man), and the overall “abundance” and “breeding success” regional classifications from Cook & Robinson 2010 (which we denote “CRA” and “CRB” respectively).

Region Name	Description	Reference
Country	Country level data for: England, Northern Ireland, Scotland, Wales, Channel Islands, Isle of Man	JNCC Seabird 2000: <a href="http://jncc.defra.gov.uk/seabird2000">http://jncc.defra.gov.uk/seabird2000</a>
CRA	Cook & Robinson 2010 BTO report: Abundance clusters	Cook & Robinson 2010: <a href="https://www.bto.org/sites/default/files/sharred_documents/publications/research-reports/2010/rr573.pdf">https://www.bto.org/sites/default/files/sharred_documents/publications/research-reports/2010/rr573.pdf</a>
CRB	Cook & Robinson 2010 BTO report: Breeding success clusters	Cook & Robinson 2010: <a href="https://www.bto.org/sites/default/files/sharred_documents/publications/research-reports/2010/rr573.pdf">https://www.bto.org/sites/default/files/sharred_documents/publications/research-reports/2010/rr573.pdf</a>

Global	All Seabird 2000 subsites	JNCC Seabird 2000: <a href="http://jncc.defra.gov.uk/seabird2000">http://jncc.defra.gov.uk/seabird2000</a>
ICES	International Council for Exploration of the Sea	<a href="https://www.ices.dk/marine-data/maps/Pages/default.aspx">https://www.ices.dk/marine-data/maps/Pages/default.aspx</a>
MSFD	Marine Strategy Framework Directive	<a href="https://www.eea.europa.eu/data-and-maps/data/europe-seas">https://www.eea.europa.eu/data-and-maps/data/europe-seas</a>
JNCC Regional Seas	JNCC Regional Seas	<a href="http://jncc.defra.gov.uk/page-1612">http://jncc.defra.gov.uk/page-1612</a>
Site	Individual Seabird 2000 subsites	JNCC Seabird 2000: <a href="http://jncc.defra.gov.uk/seabird2000">http://jncc.defra.gov.uk/seabird2000</a>

## 10 Updating the package

The package is structured in such a way that it is straightforward to update it to include additional parametric models for environmental stochasticity and/or density dependence, without needing to alter the basic functions used by the package.

### 10.1 Adding additional models for density dependence

Models for density dependence can be added by providing a list containing two things:

- “npdd”: the number of parameters associated with the density dependence model; and
- “fn”: a function that calculates the effect of population size (“popsize”) upon the (transformed or untransformed) demographic rates, given a vector of parameter values (“pars”).

#### 10.2 Additional models for environmental stochasticity

Additional models can be added by providing three things (within **models.R**), for both “constrained” and “unconstrained” versions of the model:

- [compulsory] a function which generate simulations from the model;
- [optionally] a function which determines parameter estimates for the model, given the mean and standard deviation; and
- [optionally] a function which generated expected values generated by the model.

The user also needs to specify the number of parameters associated with environmental stochasticity (npes).

If [b] is missing, then for two parameter models without density dependence, moment matching will be performed by numerical optimisation rather than using analytical

formulae. For models with more than two parameters, or that include density dependence, moment matching will not be available (i.e. it will be necessary to have **specify.as.params = TRUE**). If [c] is missing then the expected values are calculated via simulation.

## 10.2 Updating look up tables with demographic information

### 10.2.1 *Species meta-data*

Within the file 'lookup-spmeta.csv' the values in the columns "AFB" (age at first breeding) and "MBS" (maximum brood size) can be changed, and the file re-saved.

### 10.2.2 *Survival data*

In order to change this file, edit the file 'lookup-surv.xlsx', and then, when you have finished editing, go to "Save as > comma separated CSV file", and over-write the file lookup-surv.csv. Within lookup-surv.xlsx the values in all columns can be edited; note, however, that only the following six columns are actually used in the tool:

- Species
- Source
- Age.lo
- Age.hi
- Surv.mean
- Surv.SD

so all other columns are purely for reference (i.e. to aid anyone updating the file).

### 10.2.3 *Breeding success*

This file ('lookup-BS.csv') is not designed to be updated directly, and doing so is not recommended – an R script needs to be re-run in order to generate it.

### 10.2.4 *General notes on updating files*

When editing the files, it is important not to:

- delete or rename any columns
- amend the species names, or add additional species
- insert any non-numeric entries in the columns that must be numeric ("AFB" and "MBS" in lookup-spmeta.csv and "Age.lo", "Age.hi", "Surv.mean" and "Surv.SD" in lookup-surv.csv).

## References

- Caswell, H. 2000. Matrix Population Models: Construction, Analysis and Interpretation. Oxford University Press Inc. 328pp.
- Cook, A. S. C. P., and R. A. Robinson. 2010. How Representative is the Current Monitoring of Breeding Seabirds in the UK? BTO Research Report No. 573.
- Freeman, S., Searle, K., Bogdanova, M., Wanless, S. & Daunt, F. (2014) Population dynamics of Forth & Tay breeding seabirds: review of available models and modelling of key breeding populations (MSQ – 0006). Contract report to Marine Scotland Science.
- Horswill, C. & Robinson R. A. 2015. Review of seabird demographic rates and density dependence. JNCC Report No. 552. Joint Nature Conservation Committee, Peterborough.
- Jitlal, M., Burthe, S. Freeman, S. & Daunt, F. (2017) Testing and validating metrics of change produced by Population Viability Analysis (PVA) (Ref CR/2014/16). Draft report to Scottish Government
- Maclean, I.M.D., Frederikson, M & Rehfisch, M.M. (2007) Potential use of population viability analysis to assess the impact of offshore windfarms on bird populations. BTO Research Report No. 480 to COWRIE. BTO, Thetford
- Schreiber, E. A. & J. Burger (Eds). 2001. Biology of Marine Birds. CRC Marine Biology Series.

**Table 1.** Structure of Leslie matrix, in relation to demographic rates: productivity  $r_0$  and age-specific survival rates  $r_1, \dots, r_{A+1}$ , where  $A$  is “age at first breeding”.

	Age 0	Age 1	Age 2		Age afb- 1	Age afb	Age afb+1		Age A-1	Age A	Age > A
Age 0	0	0	0		0	$r_0$	$r_0$	...	$r_0$	$r_0$	$r_0$
Age 1	$r_1$	0	0		0	0	0	...	0	0	0
Age 2	0	$r_2$	0		0	0	0	...	0	0	0
								...			
Age afb-1	0	0	0		0	0	0	...	0	0	0
Age afb	0	0	0		$r_{afb}$	0	0	...	0	0	0
Age afb+1	0	0	0		0	$r_{afb+1}$	0		0	0	0
Age A-1	0	0	0		0	0	0		0	0	0
Age A	0	0	0		0	0	0	...	$r_A$	0	0
Age > A	0	0	0		0	0	0	...	0	$r_{A+1}$	$r_{A+1}$

**Table 2.** Parametric statistical models for density dependence (model.dd) that are allowed within the R package. These models relate the untransformed (if **trans.dd** = **FALSE**) or transformed (if **trans.dd** = **TRUE**) demographic rates  $\lambda$  to the current total population size (number of breeding adults)  $x$ , via unknown parameters  $(a, b, c)$ .

Model name (model.dd)	Model specification	Number of parameters, $d$
nodd	$\lambda = a$	1 ( $a$ )
ddulinear	$\lambda = a + bx$	2 ( $a, b$ )
dduloglin	$\lambda = a + b \log_{10}(x)$	2 ( $a, b$ )
dduweibull	$\lambda = a + bx^c$	3 ( $a, b, c$ )
dduthresh	$\lambda = a + H(x - c)$	3 ( $a, b, c$ )

**Table 3.** Specifications of models for environmental stochasticity in each of the demographic rates  $r$ , specified in terms of options for **model.envstoch** and **trans.dd**, and the results of the density dependence calculations in Table 2 ( $\lambda$ ).

Model name		Number of parameters	Function to generate simulations	
model.envstoch	trans.dd		<ul style="list-style-type: none"> <li>for age-specific survival rates</li> <li>for productivity rate if model.prodmax = TRUE</li> </ul>	for productivity rate if model.prodmax = FALSE
deterministic	FALSE	$d$	$r = \lambda$	$r = \lambda$
	TRUE	$d$	$r = \text{logit}^{-1}(\lambda)$	$r = \exp(\lambda)$
betagamma	FALSE	$d + 1$	$r \sim \text{Beta}(\mu, \phi)$ $\mu = \lambda$	$r \sim \text{Gamma}(\mu, \phi)$ $\mu = \lambda$
	TRUE	$d + 1$	$r \sim \text{Beta}(\mu, \phi)$ $\mu = \text{logit}^{-1}(\lambda)$	$r \sim \text{Gamma}(\mu, \phi)$ $\mu = \exp(\lambda)$
logitnlogn	TRUE	$d + 1$	$\text{logit}(r) \sim \text{Normal}(\lambda, \sigma^2)$	$\log(r) \sim \text{Normal}(\lambda, \sigma^2)$

**Table 4.** Moment matching formulae for the gamma and beta distributions: this derives the distributional parameters  $a$  and  $\phi$  (Tables 2 and 3) from the observed mean  $m$  and standard deviation  $s$ .

Distribution	Moment matching formulae
Beta	$a = m; \phi = \frac{m(1-m)}{s^2} - 1$
Gamma	$a = m; \phi = s^2/m$

**Table 5.** Formats in which users need to specify baseline demographic rates (demobase.prod, demobase.survadult, demobase.survimm), in terms of the quantities defined in Tables 2-4. The parameters in brackets are only specified if the selected model for model.dd contains these parameters.

demobase. specify.as. params	model.envstoch	Inputs required			
		1 <sup>st</sup> column	2 <sup>nd</sup> column	3 <sup>rd</sup> column	4 <sup>th</sup> column
FALSE	deterministic or betagamma	$m$	$s$	$(b)$	$(c)$
TRUE	deterministic	$a$	$(b)$	$(c)$	Not used
	Betagamma	$a$	$\phi$	$(b)$	$(c)$
	Logitnlogn	$a$	$\sigma$	$(b)$	$(c)$



**Table 6.** Mathematical definition of the impact metrics reported by the tool, , in terms of the projected population sizes  $n_{iy}^{(l)}$  for each simulation  $i = 1, \dots, S$  and scenario  $l = 1, \dots, L$  (where  $l = 0$  denotes the “baseline” scenario). Metrics are defined in relation to each output year ( $F$ ), year associated with baseline population size ( $B$ , the population size in the year before the impact starts), and user-specified thresholds for quasi-extinction risk ( $u\_QE$ ) and target population size ( $u\_TPS$ ). Note that the ‘output year’,  $F$ , relates to each year within the simulation over the whole simulation period for which output is produced, regardless of whether the impact is currently in effect.

Metric	Name	Calculation of metric for a single simulation run	How are simulations combined to derive an overall metric?
<b>CGR</b>	Ratio of annual growth rate under scenario over the period $[B, F]$ to be equivalent growth rate under the baseline for this period	$\left( \frac{n_{iF}^{(l)}}{n_{iB}^{(l)}} \right)^{\frac{1}{F-B}} / \left( \frac{n_{iF}^{(0)}}{n_{iB}^{(0)}} \right)^{\frac{1}{F-B}}$	Three possible summaries across simulations are reported: median, mean and SD
<b>CPS</b>	Ratio of final population size in scenario to final population size in baseline	$\left( \frac{n_{iF}^{(l)}}{n_{iF}^{(0)}} \right)$	Three possible summaries across simulations are reported: median, mean and SD

Description of the nepva R package, Version 4.13

<b>QuantileUNIMP 50pcIMP</b>	Percentage of simulations in which final baseline population size is lower than the median impacted final population size	$I\left(n_{iF}^{(0)} < \text{median}_i(n_{iF}^{(l)})\right)$	Mean, multiplied by 100: this gives the percentage of simulations
<b>QuantileIMP 50pcUNIMP</b>	Percentage of simulations in which final impacted population size is lower than the median final population size under the baseline	$I\left(n_{iF}^{(l)} < \text{median}_i(n_{iF}^{(0)})\right)$	Mean, multiplied by 100: this gives the percentage of simulations
<b>Quasi_ Extinction</b>	Quasi-extinction risk: percentage of simulations in which impacted final population size is lower than the threshold for quasi-extinction, $u_{QE}$	$I\left(n_{iF}^{(l)} < u_{QE}\right)$	Mean, multiplied by 100: this gives the percentage of simulations
<b>pc_ImpSims_ above_TPS</b>	Percentage of simulations in which impacted final population size exceeds the user-specified target population size $u_{TPS}$	$I\left(n_{iF}^{(l)} > u_{TPS}\right)$	Mean, multiplied by 100: this gives the percentage of simulations

**Table 7.** Default values of `afb` (age at first breeding) and `mbs` (maximum brood size) for each species. Values were taken from Horswill & Robinson (2015) with the exception of Roseate tern (Schreiber & Berger 2001)

Species	Age at first breeding	Maximum brood size
Northern gannet	5	1
Northern fulmar	9	1
Black-legged kittiwake	4	3
Common guillemot	6	1
Razorbill	5	1
Atlantic puffin	5	1
Great cormorant	3	6
Sandwich tern	3	2
Little tern	2	3
Common tern	3	4
Arctic skua	4	2
European shag	2	4
Herring gull	5	3
Lesser black-backed gull	5	3
Great black-backed gull	5	3