Journal of
open research software

bbsBayes: An R Package for Hierarchical Bayesian Analysis of North American Breeding Bird Survey Data

SOFTWARE METAPAPER

]u[ubiquity press

BRANDON P. M. EDWARDS

ADAM C. SMITH

*Author affiliations can be found in the back matter of this article

ABSTRACT

The North American Breeding Bird Survey (BBS) is the primary ecological monitoring program used to assess the population, status, and trends of North American birds. As such, accessible analysis of BBS data is crucial to wildlife conservation/management and ecological science in North America. The R package *bbsBayes* was developed as a wrapper for the analysis of BBS data using hierarchical Bayesian models, including the models currently used by the Canadian Wildlife Service and the United States Geological Survey. The goal of *bbsBayes* is to provide an accessible package for anyone in the conservation community to estimate population trajectories (time-series) and trends (rates of change) for any of the 400+ bird species monitored by the BBS, and to allow more advance users to easily access the data and model-templates necessary to customize an analysis for their research.

CORRESPONDING AUTHOR:
Brandon P. M. Edwards
Carleton University, Department of Biology, Ottawa, Ontario, Canada; Canadian Wildlife Service, Environment and Climate Change Canada, Ottawa, Ontario, Canada

brandonedwards3@cmail.carleton.ca

KEYWORDS:
United States Geological Survey; Canadian Wildlife Service; Breeding Bird Survey; demographics; mcmc; Bayesian; conservation

TO CITE THIS ARTICLE:
Edwards BPM, Smith AC 2021 bbsBayes: An R Package for Hierarchical Bayesian Analysis of North American Breeding Bird Survey Data. *Journal of Open Research Software,* 9: 19. DOI: *https://doi.org/10.5334/jors.329*

(1) OVERVIEW

1.1 INTRODUCTION

The North American Breeding Bird Survey (BBS) is a long-term, broad-scale survey of bird populations in North America [1, 2]. It was designed to monitor changes in bird populations at relatively broad scales, such as continental, national, or states/provinces. It was started in 1966 and the database now includes annual counts of over 500 species of birds across more than 5000 road-side routes in Canada, the United States, and a few in Mexico [3]. The data collected by the BBS have been the primary source of data in more than 700 scientific publications to date [4].

The BBS provides highly structured data for generating annual indices of relative abundance through time to monitor population change [2, 5]. The relevant structure in the data come from the spatial design of the survey locations and the standardized field protocols that control effort among years. Each BBS route is established using a spatially stratified, random design, and consists of a sequence of 50 count locations ("stops"), approximately 800 m apart, along a secondary road. BBS observers, both professional and volunteer, are assigned one or more routes, which they survey annually, following a standardized protocol. One morning each year, starting at 30 minutes before local sunrise, the observer conducts a 3-minute point count at each of the 50-stops, counting all the birds they can see and birds they can hear within 400 m [2]. The highly structured data from the BBS can be combined with less structured data, such as those from eBird [6], and statistical approaches to integrating data across programs are an area of active research [7, 8]. With model-based controls for variations in effort, eBird data may also support inference about population change over the short-term [9]. However, the hierarchical Bayesian modelling of the BBS data that we discuss here still provide the most reliable estimates of broad-scale and long-term changes in bird populations.

Estimates of population trends (rates of change in population size) and the associated trajectories (series of annual estimates of relative abundance) are derived from the BBS data annually by federal government agencies in the U.S.A. [2] and Canada [10]. These trends and trajectory estimates are used by government, academic, and public organizations to monitor bird populations, set priorities for conservation, identify species at risk, and explore fundamental ecological relationships [3, 10]. For example, BBS status and trend estimates were a major component of the study that estimated a loss of almost 3 billion birds from North America since 1970 [11]. BBS status and trend estimates also provide much of the information used in State of Birds reporting in the U.S.A. and Canada [12, 13]. Additionally, in Canada, estimates of trends are used by the Committee on the Status of Endangered Wildlife in Canada (COSEWIC) to inform decisions as to whether species should be added or removed from the list of species at risk [14]. The models and modelling frameworks used to produce estimates from the BBS are continually evolving [10, 15, 16, 17]. Currently, these trend and trajectory estimates are generated using a hierarchical Bayesian, over-dispersed Poisson regression model that accounts for variation among and within observers, routes, and geo-political strata [10, 16, 17].

The CWS and USGS jointly manage the BBS data and publish the entire data set each year in an accessible online format [18]. However, the detailed model structures and computer code necessary to replicate these estimates are not nearly as accessible. Although the models have been described in the literature, there does not yet exist a streamlined method for a researcher to download the entire data set and replicate the estimates of trends and trajectories, or to customize the standard hierarchical Bayesian models for a species of interest. Additionally, the process of properly subsetting and preparing the raw BBS data for modelling (for example, properly distinguishing zero-counts and missing data) may be a daunting task to a researcher who may not have much experience using Bayesian models or using the BBS data.

The R package *bbsBayes* was developed to address these issues: to provide a clear and open workflow within which a user could replicate the agency-based analyses of trends and trajectories and to customize the standard models for specific research questions. *bbsBayes* is an R package that contains the standard BBS models and tools to download and prepare the raw data, and functions that can help researchers customize their own analyses using hierarchical Bayesian models. In this paper, we detail the development of the package itself, as well as the data prepping, modelling, and subsequent summaries and visualisations that can be performed with *bbsBayes*. Using the functionality described, we provide a worked example of a full analysis run on Wood Thrush (*Hylocichla mustelina*), a near-threatened and declining passerine bird that breeds across much of Eastern North America [19].

1.2 IMPLEMENTATION AND ARCHITECTURE

bbsBayes seeks to mirror the typical workflow of BBS data analysis for a given species. As such, the package provides functionality for 4 main processes:

1. Data import
2. Data preparation
3. Markov chain Monte Carlo (MCMC) simulation
4. Model summary

The user has access to functionality under all four processes to analyse data as needed for their research. The following subsections detail the implementation of each of these processes and describe the functionality available to the user from each process. We will also give a brief discussion on how to modify the models to add covariates, change priors, etc. A summary of this typical workflow can be found in ***Figure 1***.

1.2.1 Data Import

When the package is first installed, the user must download the data from the USGS, which can be accomplished within the *bbsBayes* package using the `fetch_bbs_data()` function. When called, this function downloads raw BBS data from the USGS's data repository website ScienceBase (*https://www.sciencebase.gov/catalog/*) using functions from the *sbtools* package [20] and saves it to an application directory on disk. Note that this function must be run to make the raw data accessible to *bbsBayes*, and it should be rerun to update the local copy of the data after the release of the annual data sets.

The user can optionally bring the raw, unstratified data into R, using the function `load_bbs_data()`however it is not necessary. Calling the `load_bbs_data()`creates a list of three data frames: `route`, `bird`, and `species`. These raw BBS data are well documented by the USGS [18], and most users need not directly interact with them, so below we highlight the key variables that are used in data preparation and simulation.

1.2.1.1 Route Data

The `route` data frame contains over 116,000 rows across 32 variables, and provides the location of each route and information on each unique sampling event: i.e., the survey conducted on each route in each year. Each row in `route` corresponds to route information per year, such that route *x* run in year *y* will have its own row unique from route *x* run in year *y*+1. Each route has corresponding

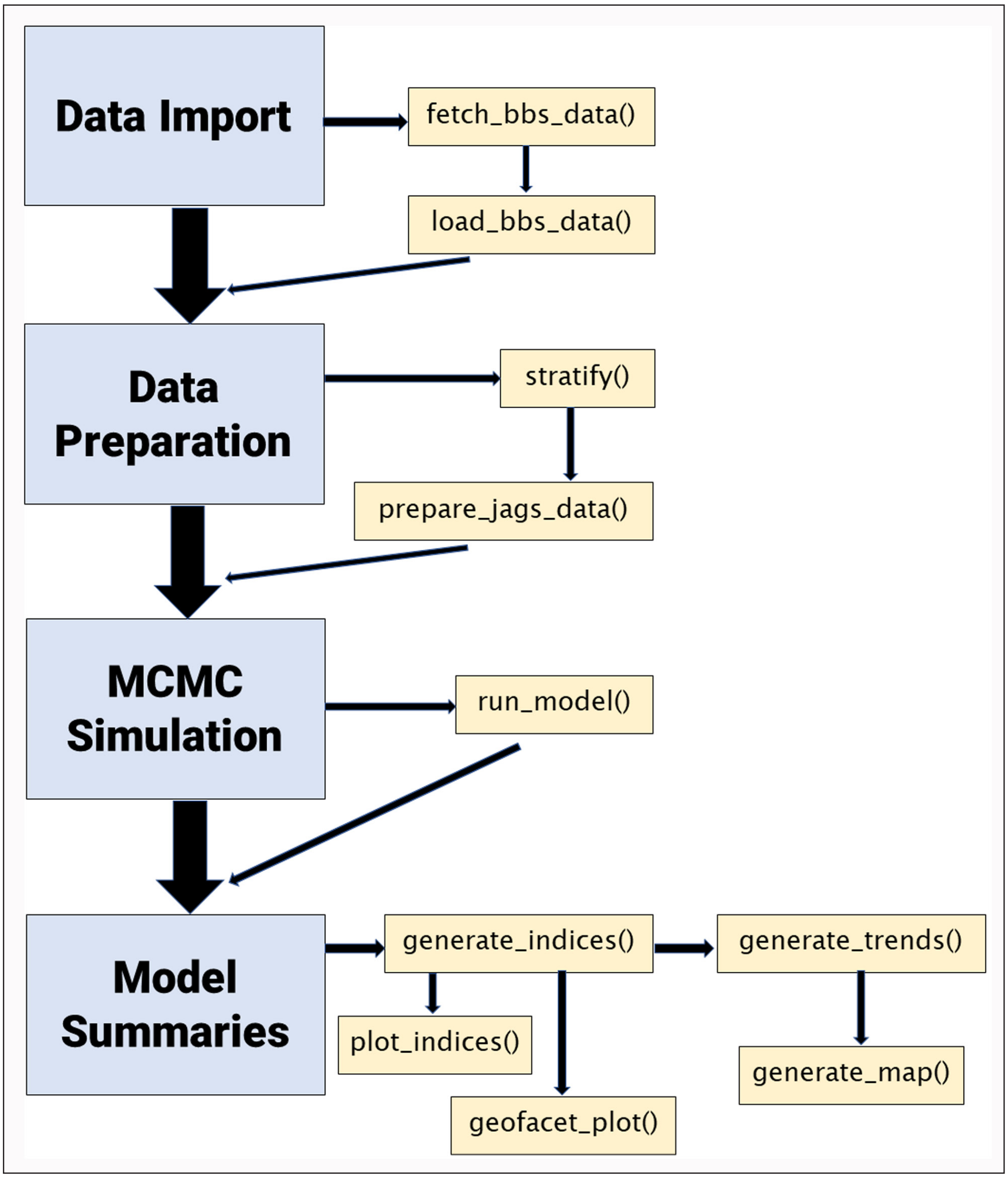

Figure 1 Flowchart of the typical workflow of generating status and trend estimates using *bbsBayes* for any species of North American bird covered by the North American Breeding Bird Survey.

locational data with it, such as state/province, country, latitude/longitude, and Bird Conservation Region (BCR, [21]) number.

Each route is uniquely identified based on the political jurisdiction (state, province or territory) in which it is located (hereafter "state"), plus a unique numerical identifier that is nested within state. Each route is also designated a unique text-based name, usually referring to the general location of the route, however the package does not directly rely on the text name. The combination of state, route number, and year, provides a unique identifier for each sampling event.

Each entry in `route` also contains information about the weather conditions during the survey, the date and time the survey was conducted, and a unique numerical identifier for the observer who conducted the survey. The observer identifiers are used in the models. The date, time and weather data are not explicitly used in the modelling, however they are used to identify which observations meet the acceptable conditions for inclusion in the models. All sampling events that are included in the models are identified with the value 1, in column titled "RunType".

1.2.1.2 Bird Data

The `bird` data frame contains over 6,500,000 rows across 15 variables, and provide counts of species by route. Each row of the `bird` data frame corresponds to the number of individuals observed of a given species on a given route, in a given year. For example, the number of mallards observed on route *x* in year *y* will be its own row separate from the number of mallards observed on route *x* in year *y*+1. Similarly, if 50 species were observed on route *x* in year *y*, then 50 separate rows will exist with count data for each species on route *x* in year *y*. Note: this data frame does not include zero counts; if a species is not observed on a given route, in a given year, there will be no record of it in this file, even if that species has been observed on that route in other years.

The `route` variable in the `bird` data frame corresponds to the numerical representation of the route as described in 1.2.1.1. Additionally, each entry in `bird` is assigned general locational information, such as state/province, country, and Bird Conservation Region (BCR) number. Indeed, the combination of route number, state, and year can be used to cross reference species count data to the route data contained in the `route` data frame.

1.2.1.3 Species Data

The `species` data frame contains 756 entries across 10 variables, and provides a list of all North American bird species in their English, French, and Spanish names, their taxonomic classifications, and a unique numeric code originally assigned by the American Ornithological Union (AOU; now the American Ornithological Society, AOS). Users should familiarize themselves with the taxonomic groupings in the BBS database [22], as some taxonomic revisions made since the start of the surveys require careful treatment of the historical data. For example, it is not possible to retroactively assign all observations of Traill's Flycatcher made before the species was split into the two separate species, Alder and Willow Flycatcher [22].

1.2.2 Data Preparation

Data preparation functions are provided for functionality to prepare raw BBS data into data used as input for modelling. As such, data preparation is heavily dependent on what analyses the user is looking to perform.

1.2.2.1 Stratification

All of the models available in *bbsBayes* require a stratification to estimate trends and trajectories for different geographical areas of North America. There are a number of stratification options within the package, all of which are based on distinct geographic regions. The function `stratify()` is used to create an intermediate `bird` and `route` data frame that contain references to the geographic stratum-allocation for each data point. There are two ways to use the function. In the first way, the user can load raw BBS data into the R session using the `load_bbs_data()` function and save it to a variable. This variable would then be passed into the `stratify()` function with the `bbs_data` argument. This approach would be most useful if an advanced user wished to make some modification to the dataset to suit a customized model or a custom subset of the data.

However, since `fetch_bbs_data()` saves raw BBS data directly to the user's disk, functions from *bbsBayes* can therefore access this data. Indeed, the less memory-intensive and recommended way to use the `stratify()` function is to simply not provide any input data to the function; the function will access the raw data from the user's disk, stratify the data, and return the stratified data as a large list.

To specify how the data should be stratified, the user provides a string to the `by` argument, which can be given 5 possible options:

1) "latlong" – Stratify by degree blocks (i.e. 1 degree of latitude by 1 degree of longitude, the sampling strata with which the BBS routes are defined)
2) "state" – political jurisdictions only (i.e. by province, state, and territory)
3) "bcr" – Bird Conservation Region (BCR) only
4) "bbs_usgs" – Intersection of political jurisdictions and BCR (USGS method)
5) "bbs_cws" – Intersection of political jurisdictions and BCR (CWS method)

Figure 2 illustrates each of these stratification options on maps of North America.

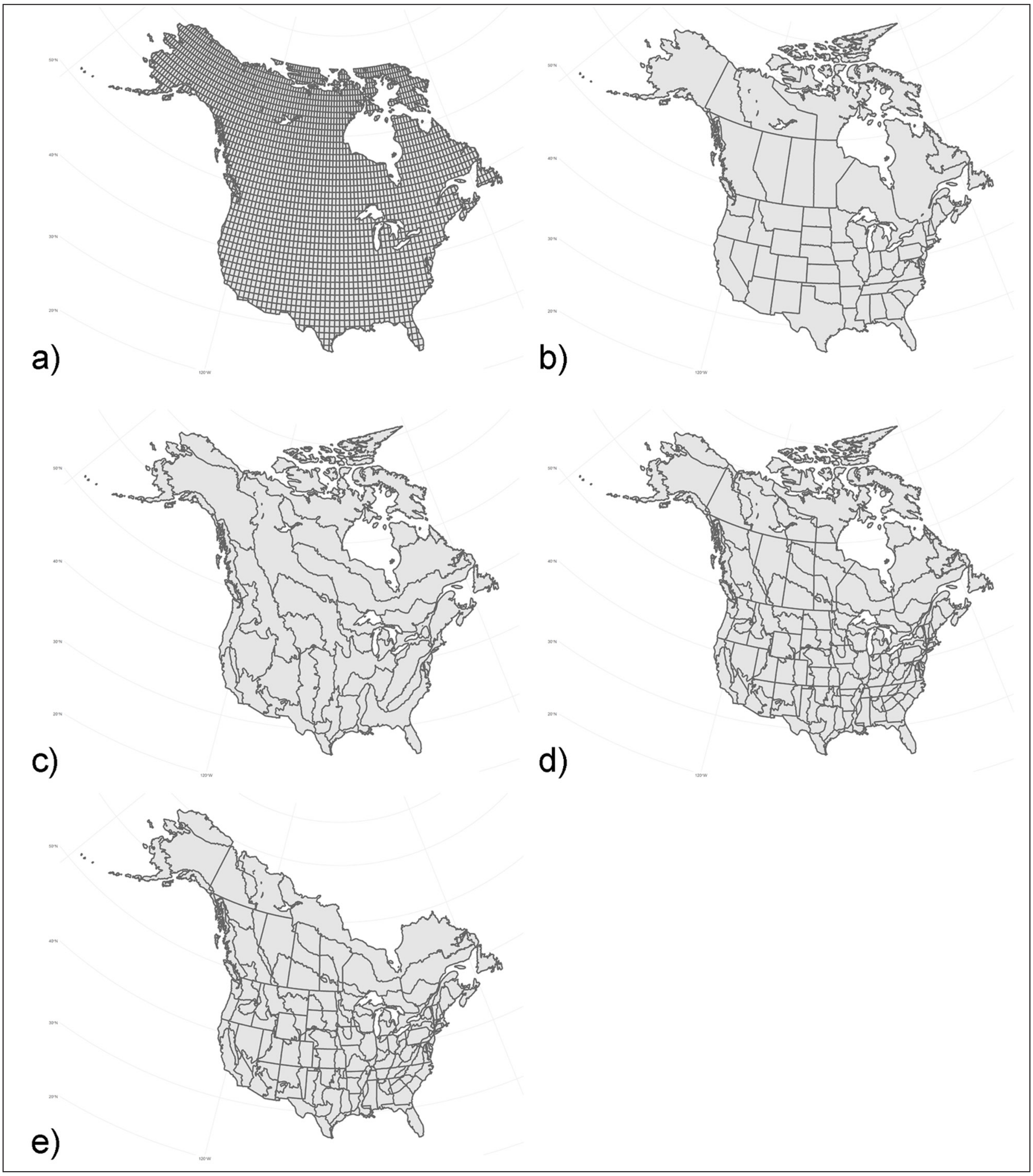

Figure 2 Maps of the 5 stratification options offered by *bbsBayes*. A user can specify to stratify by degree blocks (**a**), state (**b**), Bird Conservation Region (BCR; **c**), BCR × state (USGS method; **d**), or BCR × state (CWS method; **e**).

In current hierarchical Bayesian analyses of BBS data, option 4 ("bbs_usgs") or option 5 ("bbs_cws") are typically used. The USGS and CWS methods of these intersections vary slightly. In both, the strata are defined by the intersection of political borders (i.e. state, provincial, and territorial borders) and BCR borders. For example, for a route located in Toronto, Ontario, Canada, the political boundary is specified by the province of Ontario, and the BCR boundary is specified by BCR 13: Lower Great Lakes/St. Lawrence Plain, so all counts conducted in this intersection would be assigned to the stratum "Ontario-BCR13". The CWS stratification is identical, except for two small modifications: 1) The provinces of Prince Edward Island (PEI) and Nova Scotia are treated as a single province, because PEI is so small that it only contains 4 BBS routes and therefore often fails to meet the minimum data criteria [16]; 2) The entirety of BCR 7: Taiga Shield and Hudson Plains is treated as one stratum, because this large portion of Canada's Boreal forest contains a large portion of many species' populations but it includes very few routes and would often be excluded from analyses if separated by province/territory [14].

The `stratify()` function returns a large list that contains modified versions of the `bird` and `route` data frame that contain references to their strata. Each of the `bird` and `route` data frame gain a unique identifier for each entry that is a combination of the state, route, and year (as mentioned in 1.2.1). If the user selected to stratify using the CWS method of BCR and state intersection, both the `route` and `bird` data frames will be modified to reflect the combined provinces of PEI and Nova Scotia and the combined routes for BCR 7.

1.2.2.2 Data Preparation

The function `prepare_data()` creates the species-specific data used for fitting the model. The models supplied by *bbsBayes* can only be used for one species at a time, so this function must be called separately for each species to be modelled. This function provides the user with options to select the species to model (by setting the `species` argument), as well as data exclusion criteria (e.g., minimum number of routes in a stratum with the species present) and options for model-specific required data (e.g., the distribution used to model overdispersion).

`prepare_data()` uses the stratified data set created by `stratify()`. The function starts by subsetting the `bird` data frame, keeping only observations of the specified species. This subset of data is then merged with the `route` data frame. As mentioned in 1.2.1, the `bird` data frame and `route` data frame both contain references to the state/province and BCR the data were recorded in (i.e. what state/province and BCR the route is in), the year for each route run, and a numerical representation of the route itself. Therefore, these two data frames can be merged by these 4 variables.

The merging of these two data frames serves two purposes. For one, since the strata information is contained in the `route` data frame (among other qualitative data about the route), merging these two data frames now allows for a stratum reference for the species of interest, and allows a reference for qualitative route data as possible covariates for the species count. Additionally, merging these two data frames adds the zero-counts to the data. Because the list of species not observed on any given route *x* year combination is extremely long, zero-counts are not explicitly stored in the BBS data set: an entry for species *s* on route *x* in year *y* will appear in the `bird` data frame only if it was actually observed. This creates the case where if species *s* was only observed for 10 out of the 50 years that route *x* has been run, there are implied yet missing data points that shows 0 count for the remaining 40 years. With this merge, we ensure that all the years a given route was run appear in the merged data frame, even if the species was not observed in a particular year.

`prepare_data()` provides a number of arguments for the user to exclude data from the analysis that do not meet a threshold. These are:

- `min_year`: Set the minimum year to keep in the analysis
- `max_year`: Set the maximum year to keep in the analysis
- `min_n_routes`: specify the minimum number of routes with non-zero observations of the species in a stratum
- `min_max_route_years`: specify the minimum of the maximum number of years with non-zero observations of the species on each route in a stratum
- `min_mean_route_years`: specify the minimum of the mean number of years with non-zero observations of the species on the routes in a stratum
- `strata_rem`: specify specific stratum or strata to remove from the analysis

The authors of *bbsBayes* have set defaults for these variables based on what is done for analyses performed by CWS [10].

bbsBayes incorporates four Bayesian models into the package. When the user calls `prepare_data()`, they will need to specify for which model the data will be prepared using the `model` argument. As of this version (additional models will be added in the future), there are 4 model options, reflecting 4 different approaches to modelling the temporal patterns of population change (***Table 1***). Each of the models have the following same basic structure:

$$\log\left(\lambda_{s,j,t}\right) = \theta_s + \Delta_s\left(t\right) + \eta\mathrm{I}\left[j,t\right] + \omega_j + \varepsilon_{s,j,t}$$

where each count in geographic stratum *s* by obsever *j* in year *t* is treated as a Poisson random variable with mean $\lambda_{s,j,t}$, with log-linear functions of stratum-specific intercepts θ_s, observer-route effects (ω_{jj}), first-year startup effects for a given observer ($\eta I[j,t]$), and a count-level random effect to model overdispersion ($\varepsilon_{s,j,t}$). Variations among and within observers on the probability of detection of each species are modeled using the random effects of observers and the first-year startup effects, and controlled by the field protocol that limits changes in observers within each route. Variations in detectability are also controlled by the field protocol that strictly limits any variation in time of day, weather conditions, or season [3]. The models vary in their temporal components, estimated using a function of year ($\Delta_s(t)$, see ***Table 1***). Priors are set following [17] and [23]. For each model, the user can also specify whether the extra-Poisson error distribution should be modelled as a normal distribution or a heavy-tailed *t*-distribution [10, 17], the latter of which is accomplished by setting `heavy_tailed = TRUE`. For users that simply want to replicate the status and trend results of the 2019-data version of the CWS analysis, the GAMYE model should be used. To replicate the 2019-data version of the USGS analysis,

MODEL	TEMPORAL PARAMETERS	DESCRIPTION	REFERENCE
Slope `model = "slope"`	Random-effect log-linear slopes (overall long-term rate of population change) with random year-effect deviations (yearly fluctuations around the overall long-term slope).	Based on the model used by the CWS and USGS since 2011, but with slopes and intercepts fit as random effects, so that slopes and intercepts for data-sparse strata are shrunk towards the survey-wide means.	[16]
First-difference `model = "firstdiff"`	Year-effects follow a random walk, where for each stratum, the differences between year-t and year-t-1 is a zero-mean normal distribution with an estimated variance.	Based on the first-difference model described in Link and Sauer 2020. [33] The year-effects are shrunk towards the value in the previous year, so that the long-term trajectory is relatively flexible (e.g., can follow cyclical population patterns well) but annual fluctuations are dampened.	[17]
GAM `model = "gam"`	Year-effects follow a penalized thin-plate spline (i.e., a GAM smooth), with a number of knots chosen by the user. The parameters linking the basis function to the yearly values are estimated as random effects, centred on a survey-wide mean, so that the shape of the trajectory in a data-sparse stratum is shrunk towards a survey-wide mean trajectory.	GAM basis structure based on Crainiceanu et al. 2005. [24] For a number of knots, similar to the defaults (0.25 * number of years), the estimated trajectories are relatively smooth in the short-term (i.e., show no annual fluctuations) but are quite flexible over the long- and medium-term (e.g., population cycles on a 3-10 year period and change points in medium-term trends are modelled well).	[10]
GAMYE `model = "gamye"`	Combines the GAM components of the above model with the random year-effects of the slope model.	Trajectories are quite flexible over the long- and medium-term (e.g., population cycles on a 3–10 year period and change points in medium-term trends are modelled well), and include yearly fluctuations around the smoothed trajectory.	[10]

Table 1 Comparison of the 4 models provided by *bbsBayes*.

depending on the species, either the Slope model or the First Difference model should be used.

The Slope model is effectively a regression model that uses a slope parameter and annual fluctuations to model population change, sharing information among strata on the slope parameter [14, 16]. The GAM and GAMYE models use a Generalized Additive Model to smooth population trajectories and the GAMYE version includes year-effects to model annual fluctuations around the smooth. These two models also share information among strata on the pattern of population change [10]. The First Difference model uses a random-walk time-series approach to model the first-order differences between subsequent years to model the changes in population within a stratum [17]. In contrast to the other three models, the First Difference does not share information among strata on the pattern or rate of population change, but instead estimates changes in each stratum independently of other regions in the species' range. In general, the choice of which model to use is a complicated decision that goes far beyond the scope of this paper. We encourage users to carefully consider the ongoing discussions of model selection for the BBS in the literature [3, 10, 17].

The generalized additive model (GAM) and GAM year effect (GAMYE) require a basis function with *n* number of knots [24]. The user can specify the number of knots to be used with the `n_knots` argument, but `prepare_data()` will default to $[\frac{Y}{4}]$, rounded to the nearest integer, where Y is the total number of years.

In all four models, the data list returned by `prepare_data()` will contain the number of counts, number of strata used, the area-weight of each stratum, minimum and maximum years, the count data per route per year, the stratum for each count, the observer-route combination for each count, the year for each count, the number of observers for each count, and an indicator variable of whether it was the observer's first year of counting. If the user chooses the slope model, a fixed year, which is simply the median of all the years, is added into the list. If the user chooses the GAM or GAMYE models, the basis function matrix will be returned in this list.

`prepare_data()` includes an additional `sampler` argument, which is currently set to `sampler = "jags"` by default. As of this version of the package, *bbsBayes* only has capability for modelling using the JAGS sampling software [25]. However, future versions will include the ability to use models written in Stan, a more efficient MCMC sampler that uses Hamiltonian Monte Carlo sampling [26].

1.2.3 MCMC Simulation

MCMC simulation with JAGS is accomplished using functionality from the *jagsUI* package [27]. The function `run_model()` acts as a wrapper for creating, adapting, and sampling from the model.

For simplicity, `run_model()` provides a variety of defaults: the only required argument to the function is the list of data produced by `prepare_data()`. The model to be run is extracted from that list. `run_model()`, by default, will run a model with 3 chains, 20000 burn-in steps per chain, 10000 sampling iterations per chain. The model will thin each chain by 10 steps and will save a total of 2000 steps per model run. No specific initialization values are supplied. The function defaults to only tracking the derived parameter *n* (i.e., the trajectories of stratum-level annual indices of abundance), which is necessary for later trend analysis. *bbsBayes* offers a total of four possible ways to calculate this annual index of abundance:

1. `n`: Index of abundance as used in [14]. If the GAMYE model is chosen, this index includes the added random year effects.
2. `n2`: Index of abundance as used in [16]. If the GAMYE model is chosen, this index includes the added random year effects.
3. `n3`: Same index as `n`; however, does not include the random year effects if the GAMYE model is chosen (i.e., this is just the smooth component of the trajectory).
4. `n4`: Same index as `n2`; however, does not include the random year effects if the GAMYE model is chosen (i.e., this is just the smooth component of the trajectory).

By tracking both `n` and `n3` (or `n2` and `n4`), the user can decompose the GAMYE into a trajectory with random year effects and a trajectory that is just the GAM smooth [10, 28]. The differences between `n` and `n2` or `n3` and `n4` are often small. Conceptually, `n` and `n3` estimate the mean expected counts from among the existing collection of observer-route combinations in a given stratum, whereas `n2` and `n4` represent the mean expected count from a hypothetical new observer-route combination somewhere in the species' range [10].

The user has the option to provide their own values for any of the model parameters mentioned above. When running models in JAGS, advanced users may specify a character vector of JAGS modules to load before analysis. By default, no extra modules are loaded (other than "basemod" and "bugs"). To force "glm" or other modules to load, use `modules = "glm"`. The authors note, however, that including the "glm" module may cause problems with the BBS data and models.

The MCMC simulation is, by far, the most time consuming and computationally expensive process in the analysis of BBS data, with model runs of data rich species (such as Barn Swallow, Mourning Dove, or American Robin), and particularly with more complicated models (e.g., GAMYE), taking 72 hours or more to complete. If the user has more than 1 processor core available, the user may specify the argument `parallel = TRUE` to run one chain per processor to cut down significantly on this processing time. Users should pay close attention to the system requirements outlined in section 2.3.

1.2.4 Model Summary

A number of model summary and visualization tools are available from *bbsBayes* to allow the user to generate meaningful metrics given the simulated posterior generated by `run_model()`.

1.2.4.1 Convergence

In this package, we use Gelman and Rubin's $\hat{R}$ ("R-hat") statistic, also known as potential scale reduction factor (PSRF), to assess convergence [29]. R-hat is a ratio of the posterior variance estimates for the pooled traces of the parameter and the within-chain variance. When converged, both variance estimates will be equal, giving an R-hat value of 1. Values of R-hat greater than 1 imply that some chains may not have converged, and more sampling may be necessary [29].

The `run_model()` function will send a warning if $\hat{R} > 1.1$ for any of the monitored parameters. From here, the user should decide as to how to proceed with model summary. To possibly improve convergence, one may consider taking more samples from the posterior; *bbsBayes* provides a `get_final_values()` function that will return the final values of a model which can be used as initial values to a new call to `run_model()`, avoiding the need to wait for an additional burn-in period (see 1.3 Worked Example: Wood Thrush).

However, the seriousness of a convergence failure is something the user must interpret for themselves. In some cases, some parameters of the model may not be separately estimable, but if there is no direct inference drawn from those separate parameters, their convergence may not be necessary. If all or almost all of the `n` parameters have converged (e.g., the user receives a warning message for other monitored parameters), then inference on population trajectories and trends from the model are reliable.

1.2.4.2 Generating Indices and Trends

Given the model output by `run_model()` and the prepared data output by `prepare_data()`, the function `generate_indices()`will output a data frame of strata-weighted indices as well as a vector of quantiles sampled from the posterior (to be used for plotting credible intervals about a trajectory). By default, the function will output indices for the continent (survey-wide) and for individual strata. However, the user can set the `regions` argument to output indices for composite regions such as countries, provinces and states, Bird Conservation Regions, etc.

Once the annual indices of abundance are calculated, they can be used to generate population trends. In *bbsBayes*, and in analyses performed by CWS and USGS, the trends are calculated as the geometric mean rates of change (%/year) between two points in time [2]. This calculation is performed using the `generate_trends()` function, which can simply take the indices generated previously as input. The user can also specify a minimum year and maximum year for which to generate trends (by setting the `min_year` and `max_year` argument, respectively). The function will return a data frame with 1 row for each unit of the region types that were requested in the `generate_indices()` call (i.e., 1 row per stratum, 1 for continental, 1 per any other region selected). Additionally, the data frame contains other information related to each trend, including the start and end year of the trend, lists of included strata, total number of routes used, among others.

`generate_trends()` also allows for an alternative trend calculation. As mentioned, the default trend calculation is to calculate the geometric mean annual change in population between the start and end years. However, by setting the argument `slope = TRUE`, the function will also fit a log-linear slope to the series of all annual indices between the two end points. For some models that contain strong annual fluctuations for which no decomposition is possible (for example, the first difference model), the slope trend may be a better measure of average population change, as it integrates the pattern of change between the two end points.

Finally, `generate_trends()` provides functionality for the user to make statements about percent change and the probability of change for a given species. This is accomplished by setting the `prob_decrease` and/or `prob_increase` arguments with a vector of probabilities (i.e., a vector of numbers between 0 and 100, inclusive). Then, the function will output (for each row in the data frame) the probability of that change. For example, if the user set `prob_increase = c(0,100)`, then the data frame will contain columns related to the probability of the species increasing (that is, a > 0% increase over the time period) and the probability of the species increasing more than 2-fold (that is, a > 100% increase over the time period). Section 1.3 will cover this in more detail.

1.2.4.3 Model Visualizations

With any modelling of large data sets, and especially when talking about an increasing or decreasing population of an animal species in its habitat, it is crucial to have clear and easy-to-interpret visualizations of what the model produces for indices and trends.

bbsBayes comes with a variety of functions to visualize the indices and trends produced by the model. The `plot_indices()` function is used to create a time series of the indices for each region specified in the initial call to `generate_indices()`. That is, given the data frame of indices, `plot_indices()` will return a list of plots (created with ggplot2 [30]), one per stratum, one continent-wide, and one for each additional region specified. The user has a variety of additional visual options for the trajectory plots: they can set `add_observed_means = TRUE` to overlay the observed mean counts for each year, they can set `add_number_routes = TRUE` to superimpose a dotplot of the number of BBS routes included for each year, they can change the minimum and maximum year to plot by setting the `min_year` and/or `max_year` arguments, or they can change the sizes of axes, title, text, etc. The `plot_indices()`function returns a ggplot2 object, allowing for the user to further customize their plot using the ggplot2 library.

These trajectories can also be visualized on a geofacet plot using the `geofacet_plot()` function, which is a graphic that plots the state/province level population trajectories in facets arranged approximately in a geographical arrangement [31]. Again, this function simply takes in the indices generated by `generate_indices()`. This plotting can only be accomplished if the data was stratified by one of “state”, “bbs_cws”, or “bbs_usgs”.

Finally, the stratum-specific trends created by `generate_trends()` can be mapped out using the `generate_map()` function which creates a stratified continental heat map.

1.2.5 Modifying Models

Often with Bayesian models, new research is published that lead to more informative or useful priors on estimated parameters. Or, researchers may be interested in parameterizing an already-existing model differently to experiment with model selection. It therefore makes sense that the models supplied with *bbsBayes* should be reasonably simple to customize, and that the customized model files can easily be used with the rest of the functions of *bbsBayes*.

Indeed, the function `model_to_file()` allows the user to save the model files supplied by *bbsBayes* as plain text files. Suppose the user wanted to save the slope model file to disk. They could use this function setting the arguments `model = “slope”` and `filename = “slope_model.txt”`, saving the model file to the current working directory as a file called “slope_model.txt”. The user could then customize some portion of the model (e.g. changing priors) and rename the file to “slope_model_modified.txt”. The user could then run the custom model by setting the `model_file_path` argument in `run_model()` as `model_file_path = “slope_model_modified.txt”.`

1.3 WORKED EXAMPLE: WOOD THRUSH

We will now provide an example of using *bbsBayes* to generate a meaningful quantitative analysis of Wood Thrush trends over the time period of the BBS data collection. The Wood Thrush is a medium-sized neotropical migrant that occurs in eastern North America during its breeding season. Since the late 1970s, habitat destruction in both wintering and breeding grounds have contributed to a severe loss of abundance in this species [19]. BBS data on Wood Thrush is reasonably data rich, so it makes it a good species to model and pick up these trends over time. The script for this example, as well as the prepared data generated from `prepare_data()` and the model output generated from `run_model()` is available at *https://github.com/BrandonEdwards/bbsBayes-Wood-Thrush-Worked-Example*.

1.3.1 Data Fetching, Preparation, and Modelling

We start with calling the *bbsBayes* library, then making a call to download the BBS data from the USGS data repository. As mentioned, this step is required for the first use of *bbsBayes*, and should only be used once each year as USGS releases updated data sets.

```
# Use bbsBayes library
library(bbsBayes)

# Download the data (requires internet connection)
fetch_bbs_data()
```

When `fetch_bbs_data()` is called, the user must agree to the terms and conditions (see *https://www.pwrc.usgs.gov/BBS/RawData/*) of the data usage by typing "yes" in the console. Otherwise, the data will not be downloaded. These terms and conditions overview some best practices for using BBS data, point the user to review the metadata for the BBS data, and encourage the user to involve BBS staff in their analyses where needed. Once downloaded, the BBS data is saved on disk to a package-specific directory that can be accessed by functions of *bbsBayes*.

Now that we have the raw data downloaded to disk, we can bring the data into R and stratify it. The function `stratify()` will accomplish both these tasks. For this worked example, we choose to stratify by the state X BCR intersections (CWS method), thus our argument for `by` will be the string "bbs_cws". We recommend assigning the string "bbs_cws" (or which ever stratification the user is choosing) to a variable as it will eventually be used later in the analysis.

```
# Stratify the data
s <- "bbs_cws"
stratified_data <- stratify(by = s)
```

The variable `stratified_data` contains the stratified data which can now be prepared for use in modelling. In this example, we will model Wood Thrush counts using the hierarchical Bayesian generalized additive model with year effects, requiring the string "gamye" to be passed as an argument. Additionally, we must now specify which species to subset and prepare for modelling. As previously mentioned, the version included in this manuscript only allows the use of JAGS for sampling; for posterity, we are setting the `sampler = "jags"` argument to explicitly show that this example is using the JAGS sampler, and future versions will allow the use of Stan models.

```
# Prep the data for JAGS modelling
jags_data <- prepare_data(strat_data = bbs_strat,
                          species_to_run = "Wood
                          Thrush",
                          model = "gamye",
                          sampler = "jags")
```

We are now ready to run the model. As mentioned, this will typically end up being the most time- and memory-consuming process in an analysis, so we recommend that the user pay close attention to 2.3 Additional system requirements before running any models. As mentioned, the prepared data and model output for this worked example have been made available at *https://github.com/BrandonEdwards/bbsBayes-Wood-Thrush-Worked-Example*, for readers that may want to skip over this step and try out the model summary and visualization features.

We could simply run the following code to run a model with default settings:

```
jags_mod <- run_model(jags_data = jags_data)
```

However, for this example, we will modify a number of default settings for the model. Here, we will run the model with 1000 adaptation steps, 10,000 burn-in iterations, saving 1000 steps, 3 MCMC chains, and thinning each chain by a factor of 10. By not specifying the number of iterations, we run the model with the default 10,000 iterations. We will also give `run_model()` a list of other parameters to track. Here, we will track the parameters `n`, `n3` `taunoise`, `strata`, `B.X`, and `beta.X`. By default, run_model() will always track the parameter `n`, no matter what other parameters are specified (even if n is not explicitly specified), as this parameter is needed to calculate annual indices and trends later in the analysis. In this case, we also track `n3`, which we could later used to decompose the trajectory into both the GAM smooth trajectory and GAM smooth with yearly fluctuations.

```
# Run JAGS
jags_mod <- run_model(jags_data = jags_data,
                      n_adapt = 1000,
                      n_saved_steps = 1000,
                      n_burnin = 10000,
                      n_chains = 3,
                      n_thin = 10,
                      parallel = FALSE,
                      parameters_
                      to_save = c("n",
                                  "n3",
                                  "taunoise",
                                  "strata",
                                  "B.X",
                                  "beta.X"))
```

Depending on what species is being modelled and what model is being used, this step in the analysis process can take up to a few days to run on a single processer. If multiple processor cores are available on the user's computer, this processing time can be reduced with the argument `parallel = TRUE`, where each MCMC chain will be run on a separate core. The user could also consider taking advantage of one of the several cloud computing options that are available to further reduce computational time by running several species in parallel on multiple cores.

1.3.2 Model summary and Visualizations

When the model is complete, the `run_model()` function will send a warning if any of the tracked parameters have an $\hat{R} > 1.1$. When the authors ran this worked example, convergence warnings were issued for only one `n` value and two `n3` values; the remaining convergence warnings were for other parameters. As mentioned before, if all or most of the `n` values are converged, then that is sufficient for generating indices and trend estimates. In any case, if the user were wanting to run the model for more iterations in an attempt to improve convergence, they can make use of the `extract_final_values(jags_mod = jags_mod)` function and use these values as initial values for a new model.

Once the user is happy with convergence, indices and trends can be calculated. For this example, our goal is to use *bbsBayes* to create 4 figures: 1) a plot of the continental annual indices of abundance with observed means for Wood Thrush between 1966 and 2019, 2) a 2-panel plot of the national annual indices for Canada and US, 3) a geofacet plot, and 4) a heat map of population trends for Wood Thrush from 2009 – 2019 (i.e., a 10-year trend) for each stratum. We would also like to make a statement regarding the probability that the species has declined across its range by 0% (i.e., any decrease at all), by 50%, and by 100% from 2009 – 2019.

Let us begin by generating the indices of abundance for the continental, national, and stratum levels.

```
# Generate indices at the continental, national,
and stratum level
indices <- generate_indices(jags_mod = jags_mod,
                            jags_data = jags_
                            data,
                            regions =
                            c("continental",
                              "national",
                              "stratum"))
```

Already, we can generate the first two figures that we specified above. Let us first create the list of trajectory plots with observed means. We will also specify some different sizing for titles and axes:

```
# Create a list of trajectory plots, with
observed means
plot_list <- plot_indices(indices_list = indices,
                          species = "Wood Thrush",
                          add_observed_means =
                          TRUE,
                          title_size = 18,
                          axis_title_size = 14,
                          axis_text_size = 12)
```

We can then access the continental plot list with `plot_list$Continental`. The resulting plot is shown in ***Figure 3***.

Generating the 2-panel plot of the national indices requires slightly more work and requires the use of the

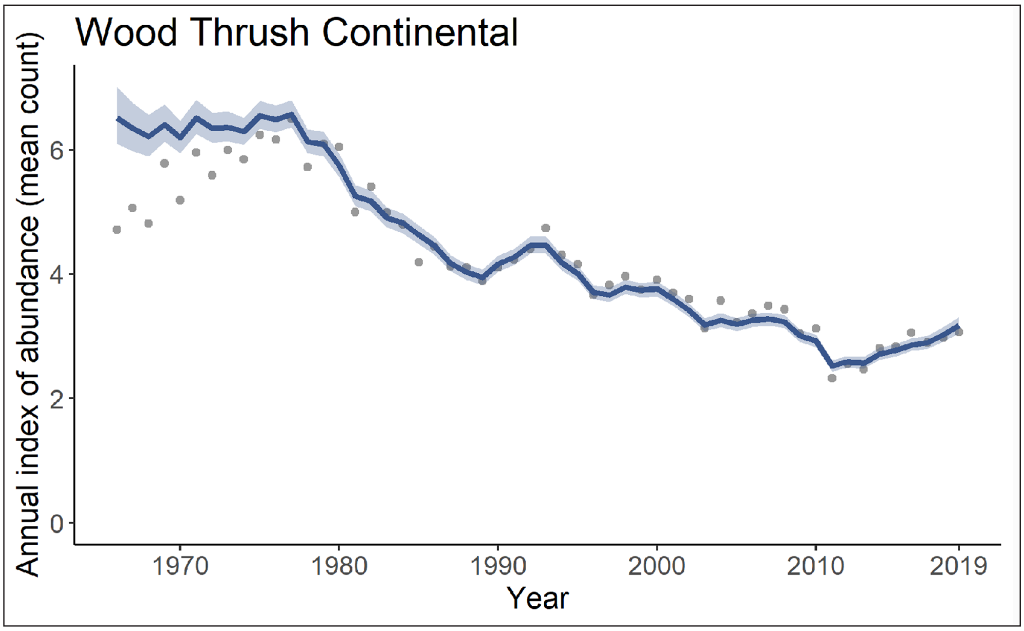

Figure 3 Plot of continental annual index of abundance for Wood Thrush from 1966 to 2019 with 95% credible band and observed means (grey dots). This plot was generated using the `plot_indices()` function.

library gridExtra [32]. For the national trends, we can access the Canada and US plot with `plot_list$Canada` and `plot_list$United_States_of_America`, respectively. Thus, by using the `grid.arrange()` function from gridExtra, we can generate a 2-panel national trajectory plot with

```
library(gridExtra)
grid.arrange(plot_list$Canada,
             plot_list$United_States_of_America,
             nrow = 2, ncol = 1)
```

which can be seen in ***Figure 4***.

To create the geofacet plot given the stratification we used (recall we set `s = "bbs_cws"`), we can run

```
geofacet_plot(indices_list = indices,
              stratify_by = s,
              select = TRUE,
              multiple = TRUE,
              species = "Wood Thrush")
```

In this case, we must set `select = TRUE` to indicate that the function must select the stratum-level data out of a data frame that contains other region types; in our case, our indices data frame contains indices at the stratum level, national level, and continental level. Additionally, we must set `multiple = TRUE` to indicate that each province/state facet may be made up of multiple strata-level trajectories that must be combined. The resulting figure can be seen in ***Figure 5***.

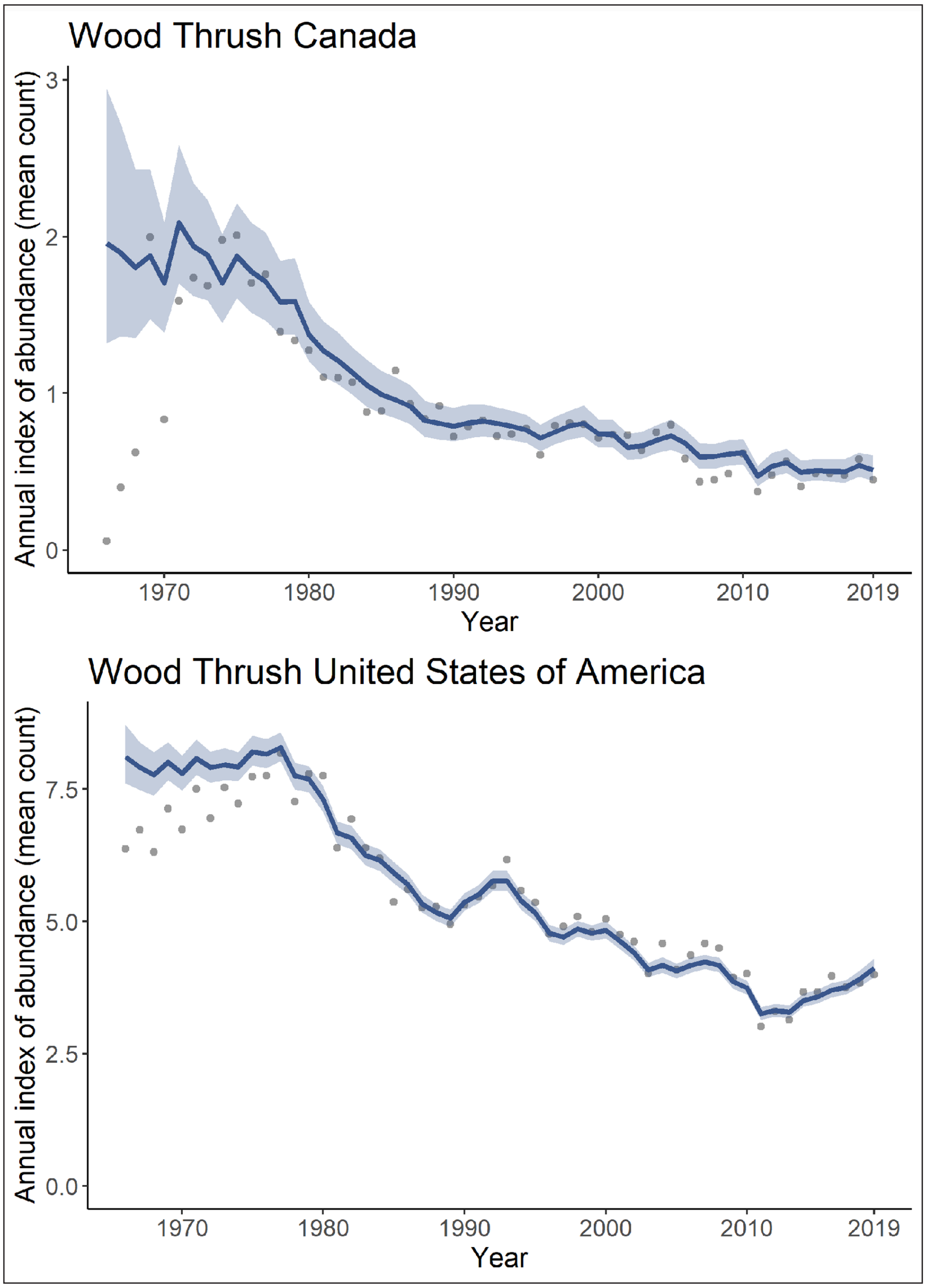

Figure 4 Plots of national annual indices of abundance for Wood Thrush from 1966 to 2019 for Canada and USA. These plots were generated using the `plot_indices()` function.

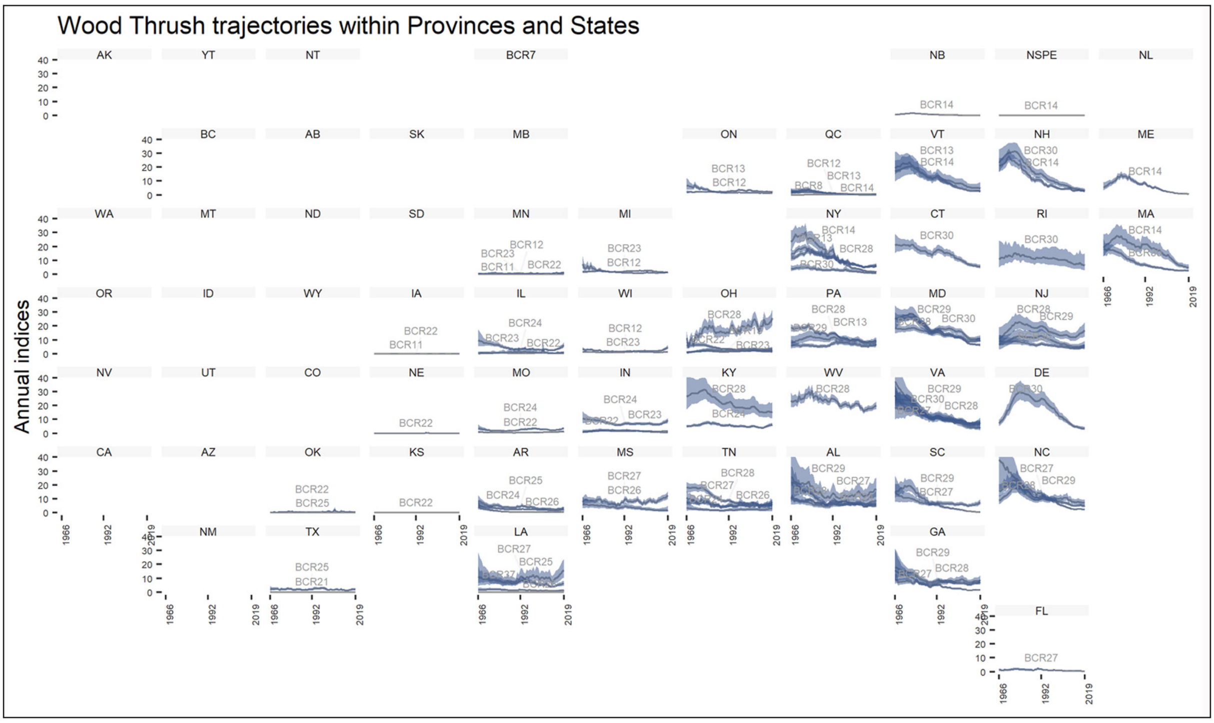

Figure 5 Geofacet plot of Wood Thrush trajectories for each province, state, and territory, created using the `plot_geofacet()` function. Each line within a state represents the trend (and 95% credible interval) for each BCR within the state.

Shifting gears slightly, let us now do some analysis of trends. As mentioned, we want to create a heat map of the trend by each stratum, and make some quantitative statements about the trends. First, we will generate the trends from 2009 to 2019

```
trends <- generate_trends(indices = indices,
                          Min_year = 2009,
                          Max_year = 2019,
                          prob_decrease  =  c(0,
                          50, 100))
```

Right away, we can generate a heat map of the trends for each stratum over this 10-year period with

```
generate_map(trend = trends,
                        select = TRUE,
                        stratify_by = s,
                        species = "Wood Thrush",
                        col_viridis = TRUE)
```

Similar to the geofacet plot, we must set `select = TRUE` to specify that we are providing a data frame that contains trends for more than just the stratum regions, and we must also set `stratify_by = s` to indicate the original stratification used. *bbsBayes* offers two different colour palettes that are both colourblind-friendly; for this example, we have opted to use colours from the viridis palette [33] by setting the argument `col_viridis = TRUE`. The resulting figure is seen in ***Figure 6***.

When we generated the trends, we have also specified `prob_decrease = c(0, 50, 100)`; our `trends` data frame will therefore contain the posterior probabilities that Wood Thrush has decreased by at least 0%, by at least 50%, and by at least 100% for this 10 year period for each region initially specified. ***Table 2*** summarizes these percent changes and probabilities of change for the continent, Canada, USA, and four select strata. With these data, we can make a statement such as "our model shows a 97% probability of any population decrease at in Canada", or "our model shows a 100% probability of the population decreasing by 50% in the US-SC-27 (South Carolina × BCR 27) stratum".

1.4 QUALITY CONTROL

The original code of *bbsBayes*, prior to being used in this package, has been continuously developed and used for several years to provide yearly trend estimates of BBS data [10, 16, 17]. *bbsBayes* also underwent 6 months of beta testing where users could submit bugs or suggestions through GitHub issues.

The R package *testthat* [34] was used in a test harness for data fetching. Finally, 1.3 provides a worked example that a user can work through to ensure the package is functioning correctly.

A small amount of sample data is provided for the users to test the toy examples given in the documentation.

The authors of the package will continue to review bug fixes and suggested changes made in pull requests to the Github repository.

(2) AVAILABILITY

2.1 OPERATING SYSTEM

Any system that can run R (obtained from *https://www.r-project.org/*) and JAGS (obtained from *http://mcmc-jags.sourceforge.net/*).

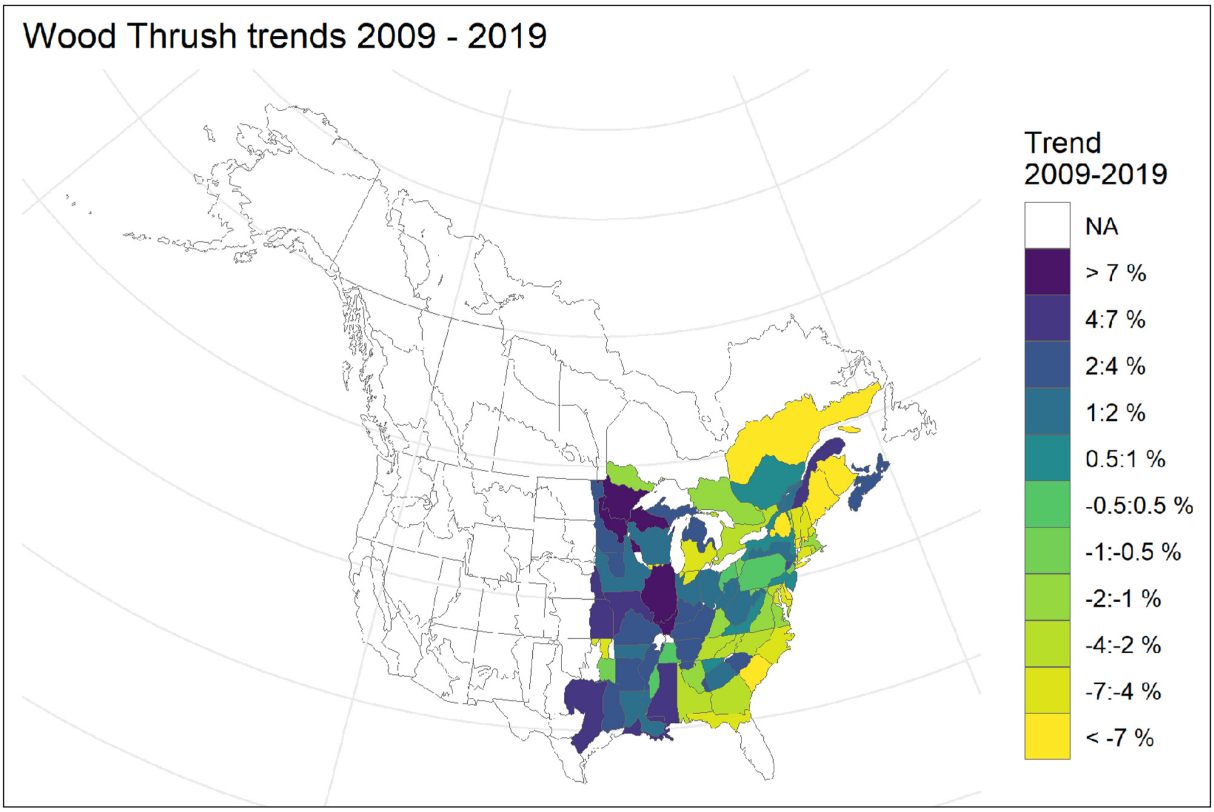

Figure 6 Heat map of Wood Thrush trends for each stratum for the 10-year period between 2009 and 2019. This map was generated using the `generate_map()` function.

REGION	PERCENT CHANGE [LOWER LIMIT, UPPER LIMIT]	PROBABILITY OF DECREASING BY 0%	PROBABILITY OF DECREASING BY 50%	PROBABILITY OF DECREASING BY 100%
Continental	+5.23% [+0.26%,+10.6%]	0.02	0.00	0.00
Canada	–16.4% [–30.7%,+0.56%]	0.97	0.00	0.00
United States	+6.45% [–1.30%,+12.0%]	0.01	0.00	0.00
US-SC-27	–74.5% [–84.2%,–57.7%]	1.00	1.00	0.00
CA-ON-12	–16.4% [–39.5%,+14.6%]	0.87	0.00	0.00
CA-QC-12	+6.71% [–36.1%,+80.2%]	0.40	0.00	0.00
US-LA-25	+13.6% [–23.3%,+66.2%]	0.26	0.00	0.00

Table 2 Percent change (and 95% credible interval) and probability of changes for the continent-wide trend, national trends, and stratum-level trends for select strata for Wood Thrush between 2009–2019. Based on the function that was run, these probabilities show the probability of the Wood Thrush population decreasing by 0%, 50%, and 100% in each of the geographical regions.

2.2 PROGRAMMING LANGUAGE

R version 2.10 or higher. JAGS version 4.3.0.

2.3 ADDITIONAL SYSTEM REQUIREMENTS

An internet connection is required to install the *bbsBayes* package, to install JAGS, and to download the BBS data. Since the BBS data is extremely data rich, it is recommended to have 8 GB or more of RAM to handle the large process created by the model for a given species.

2.4 DEPENDENCIES

R package: *progress* [35], *jagsUI* [27], *ggrepel* [36], *geofacet* [31], *ggplot2* [30], *stringr* [37], *grDevices* [38], *rgdal* [39], *dplyr* [40], *sf* [41], *tools* [38], *latticeExtra* [42], *rappdirs* [43], sbtools [20].

2.5 LIST OF CONTRIBUTORS

This package was created by Adam C. Smith and Brandon P.M. Edwards.

2.6 SOFTWARE LOCATION

Archive (e.g. institutional repository, general repository) (required – please see instructions on journal website for depositing archive copy of software in a suitable repository)

Name: CRAN

Persistent identifier: https://cran.r-project.org/package=bbsBayes
Licence: MIT
Publisher: Brandon P.M. Edwards
Version published: 2.3.7.2020
Date published: 22/06/21

Code repository (e.g. SourceForge, GitHub etc.) (required)
Name: GitHub
Identifier: https://github.com/BrandonEdwards/bbsBayes
Licence: MIT
Date published: 22/06/21

2.7 LANGUAGE

R [38], JAGS [25]

(3) REUSE POTENTIAL

The goal of *bbsBayes* is to make the analysis of BBS data using hierarchical Bayesian models more accessible to the conservation community, allowing users who are familiar with R to generate population trends and trajectories for any of the species monitored by the BBS. Because the BBS represents the best population monitoring information for most species of birds in North America [1, 2, 3], *bbsBayes* will have reuse potential in many settings, including academic, government, and conservation NGOs. This package allows an easy-to-use function to download route-level for researchers to use in custom analyses, or to run the models provided by the package. This package also gives the user the capability to download stop-level data for researchers requiring that type of data. However, *bbsBayes* does not support any analysis for stop-level data; it is for convenience only.

For users who are already familiar with analysing BBS data, *bbsBayes* will streamline the process of fetching updated BBS data, and running additional models. For users of the trend and trajectory estimates published by the CWS and USGS, *bbsBayes* makes accessible the models and data-preparation steps underlying the estimates. Advanced users may also consider expanding the use of the models provided by *bbsBayes* to other data sets. Although the models provided by *bbsBayes* were built for the route-level counts of the BBS and estimate trends and trajectories at the stratum-level, one could imagine a scenario where a user could modify the model files to generate trends at a route-level. Alternatively, a user could attempt to modify their own point-level dataset into one that resembles counts at the route level, to be fed into the models provided by this package. Finally, for researchers, the customization options that *bbsBayes* includes will be extremely useful for modelling the almost infinite set of ecological mechanisms and hypotheses that could be explored with one of the greatest long-running, rigorously-collected, continental-scale, ecological monitoring datasets in the world.

ACKNOWLEDGEMENTS

We thank the thousands of skilled volunteers who have contributed to the Breeding Bird Survey over the years, as well as those who have served as provincial and territorial coordinators. Thank you to Dr Gregory M. Mitchell, Dr Scott Wilson, Dr Charles Francis, and Dr Marie-Anne Hudson (Environment and Climate Change Canada) for their comments and suggestions in the initial development phase. Thank you to Dr Maxwell B. Joseph (Earth Lab) for contributions made to data fetching functionality, and to Dr Jessica Burnett (United States Geological Survey) for contributions and suggestions to the package. Thank you to the anonymous reviewer for the useful suggestions to improve the manuscript. We acknowledge that the National Wildlife Research Centre, on the Carleton University campus, resides on the traditional and unceded territory of the Algonquin nation.

FUNDING STATEMENT

This software did not result from funded research.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Brandon P. M. Edwards *orcid.org/0000-0003-0865-3076*
Carleton University, Department of Biology, Ottawa, Ontario, Canada

Adam C. Smith *orcid.org/0000-0002-2829-4843*
Canadian Wildlife Service, Environment and Climate Change Canada, Ottawa, Ontario, Canada

REFERENCES

1. **Hudson MAR, Francis CM, Campbell KJ, Downes CM, Smith AC, Pardieck KL.** The role of the North American Breeding Bird Survey in conservation. *The Condor.* 2017; 119(3): 526–545. DOI: https://doi.org/10.1650/CONDOR-17-62.1
2. **Sauer JR, Pardieck KL, Ziolkowski DJ, Jr., Smith AC, Hudson MAR, Rodriguez V, Berlanga H, Niven DL, Link, WA.** The first 50 years of the North American Breeding Bird Survey. *The Condor.* 2017; 119(3): 576–593. DOI: https://doi.org/10.1650/CONDOR-17-83.1

3. **Downes CM, Hudson MAR, Smith AC, Francis CM.** The Breeding Bird Survey at 50: scientists and birders working together for bird conservation. *Avian Conservation and Ecology*. 2016; 11(1): 8. DOI: *https://doi.org/10.5751/ACE-00855-110108*
4. United States Geological Survey, 2019. *BBS Bibliography*. Available at *https://www.pwrc.usgs.gov/BBS/Bibliography/.*
5. **Miller DAW, Pacifici K, Sanderlin JS, Reich BJ.** The recent past and promising future for data integration methods to estimate species' distributions. *Methods in Ecology and Evolution*. 2019; 10: 22–37. DOI: *https://doi.org/10.1111/2041-210X.13110*
6. **Sullivan BL,** et al. The eBird enterprise: An integrated approach to development and application of citizen science. *Biological Conservation*. 2014; 169: 31–40. DOI: *https://doi.org/10.1016/j.biocon.2013.11.003*
7. **Robinson OJ, Ruiz-Gutierrez V, Reynolds MD, Golet GH, Strimas-Mackey M, Fink D.** Integrating citizen science data with expert surveys increases accuracy and spatial extent of species distribution models. *Diversity and Distributions*. 2020; 26(8): 976–986. DOI: *https://doi.org/10.1111/ddi.13068*
8. **Maxwell MB, Pavlacky DC, Jr., Bartuszevige AM.** Data fusion for abundance estimation: community science augments systematically collected removal-in-time distance sampling data. *bioRxiv*. 2021; 1–26. DOI: *https://doi.org/10.1101/2021.05.02.442379*
9. **Fink D, Auer T, Johnston A, Ruiz-Gutierrez V, Hochachka WM, Kelling S.** Modelling avian full annual cycle distributions and population trends with citizen science data. *Ecological Applications*. 2019; 30(3): e02056. DOI: *https://doi.org/10.1002/eap.2056*
10. **Smith AC, Edwards BPM.** North American Breeding Bird Survey status and trend estimates to inform a wide range of conservation needs, using a flexible Bayesian hierarchical generalized additive model. *Ornithological Applications*. 2020; 123(1): 1–16. DOI: *https://doi.org/10.1101/2020.03.26.010215*
11. **Rosenberg KV, Dokter AM, Blancher PJ, Sauer JR, Smith AC, Smith PA, Stanton JC, Panjabi A, Helft L, Parr M, Marra PP.** Decline of the North American avifauna. *Science*. 2019; 366(6461): 120–124. DOI: *https://doi.org/10.1126/science.aaw1313*
12. **North American Bird Conservation Initiative.** The State of North America's Birds 2016. Ottawa: Environment and Climate Change Canada 2016; 1–8.
13. **North American Bird Conservation Initiative Canada.** The State of Canada's Birds 2019. Ottawa: Environment and Climate Change Canada 2019; 1–12. DOI: *https://doi.org/10.22621/cfn.v128i2.1565*
14. **Smith AC, Hudson MAR, Downes C, Francis CM.** Estimating breeding bird survey trends and annual indices for Canada: how do the new hierarchical Bayesian estimates differ from previous estimates? *The Canadian Field-Naturalist*. 2014; 128(2): 119–134. DOI: *https://doi.org/10.22621/cfn.v128i2.1565*
15. **Sauer JR, Link WA.** Hierarchical models and the analysis of bird survey information. *Ornis Hungarica*. 2003; 12–13(1–2): 217–222.
16. **Sauer JR, Link WA.** Analysis of the North American Breeding Bird Survey using hierarchical models. *Auk*. 2011; 128(1): 87–98. DOI: *https://doi.org/10.1525/auk.2010.09220*
17. **Link WA, Sauer JR, Niven DK.** Model selection for the North American Breeding Bird Survey. *Ecological Applications*. 2020; 30: e02137. DOI: *https://doi.org/10.1002/eap.2137*
18. **Pardieck KL, Ziolkowski DJ, Jr., Lutmerding M, Aponte VI, Hudson MAR.** North American Breeding Bird Survey Dataset 1966–2019: U.S. Geological Survey data release. 2020. DOI: *https://doi.org/10.5066/P9J6QUF6*
19. **Evans M, Gow E, Roth RR, Johnson MS, Underwood TJ.** Wood Thrush (Hylocichla mustelina), version 2.0. In *The Birds of North America* (A. F. Poole, Editor), 2011. Ithaca, NY, USA: Cornell Lab of Ornithology. DOI: *https://doi.org/10.2173/bna.246*
20. **Winslow LA, Chamberlain S, Appling AP, Read JS.** sbtools: A package connecting R to cloud-based data for collaborative online research. *The R Journal*. 2016; 8: 387–398. DOI: *https://doi.org/10.32614/RJ-2016-029*
21. **Bird Studies Canada and North American Bird Conservation Initiative.** Bird Conservation Regions. Published by *Bird Studies Canada on behalf of the North American Bird Conservation Initiative* 2014. *http://www.birdscanada.org/research/gislab/index.jsp?targetpg=bcr* Accessed 4 January 2019.
22. North American Breeding Bird Survey. 2021. *https://www.pwrc.usgs.gov/bbs/.* Accessed 15 June 2021.
23. **Smith AC, Hudson MAR, Aponte V, Francis CM.** North American Breeding Bird Survey – Canadian Trends Website, Data-version 2017. *Environment and Climate Change Canada* 2019.
24. **Crainiceanu CM, Ruppert D, Wand MP.** Bayesian Analysis for Penalized Spline Regression Using WinBUGS. *Journal of Statistical Software*. 2005; 14(14). DOI: *https://doi.org/10.18637/jss.v014.i14*
25. **Plummer MA.** JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling; 2003.
26. **Stan Development Team.** Stan Modeling Language Users Guide and Reference Manual, v2.27; 2021. *https://mc-stan.org*
27. **Kellner K.** jagsUI: A Wrapper Around 'rjags' to Streamline 'JAGS' Analyses. R package version 1.5.0; 2018. *https://CRAN.R-project.org/package=jagsUI.*
28. **Knape J.** Decomposing trends in Swedish bird populations using generalized additive mixed models. *Journal of Applied Ecology*. 2016; 53: 1852–1861. DOI: *https://doi.org/10.1111/1365-2664.12720*
29. **Gelman A, Rubin DB.** Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*. 1992; 7(4): 457–472. DOI: *https://doi.org/10.1214/ss/1177011136*
30. **Wickham H.** ggplot2: Elegant Graphics for Data Analysis. 2016; New York: Springer-Verlag. DOI: *https://doi.org/10.1007/978-3-319-24277-4_9*

31. **Hafen R.** geofacet: 'ggplot2' Faceting Utilities for Geographical Data. R package version 0.1.10; 2019. *https://CRAN.R-project.org/package=geofacet*.
32. **Auguie B.** gridExtra: Miscellaneous Functions for "Grid" Graphics. R package version 2.3; 2017. *https://CRAN.R-project.org/package=gridExtra*.
33. **Garnier S.** viridis: Default Color Maps from 'matplotlib'. R package version 0.5.1; 2018. *https://CRAN.R-project.org/package=viridis*.
34. **Wickham H.** testthat: Get Started with Testing. *The R Journal.* 2011; 3(1): 5–10. DOI: *https://doi.org/10.32614/RJ-2011-002*.
35. **Csárdi G, FitzJohn R.** progress: Terminal Progress Bars. R package version 1.2.0; 2018. *https://CRAN.R-project.org/package=progress*.
36. **Slowikowski K.** ggrepel: Automatically Position Non-Overlapping Text Labels with 'ggplot2'. R package version 0.8.2; 2020. *https://CRAN.R-project.org/package=ggrepel*.
37. **Wickham H.** stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.3.1; 2018. *https://CRAN.R-project.org/package=stringr*.
38. **R Core Team.** R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing; 2018. URL *https://www.R-project.org/*.
39. **Bivand R, Keitt T, Rowlingson B.** rgdal: Bindings for the 'Geospatial' Data Abstraction Library. R package version 1.3-6; 2018. *https://CRAN.R-project.org/package=rgdal*.
40. **Wickham H, François R, Henry L, Müller K.** dplyr: A Grammar of Data Manipulation. R package version 0.8.3; 2019. *https://CRAN.R-project.org/package=dplyr*.
41. **Pebesma E.** Simple Features for R: Standardized Support for Spatial Vector Data. *The R Journal.* 2018; 10(1): 439–446. DOI: *https://doi.org/10.32614/RJ-2018-009*
42. **Sarkar D, Andrews F.** latticeExtra: Extra Graphical Utilities Based on Lattice. R package version 0.6-28; 2016. *https://CRAN.R-project.org/package=latticeExtra*.
43. **Ratnakumar S, Mick T, Davis T.** rappdirs: Application Directories: Determine Where to Save Data, Caches, and Logs. R package version 0.3.1; 2016. *https://CRAN.R-project.org/package=rappdirs*.

TO CITE THIS ARTICLE:
Edwards BPM, Smith AC 2021 bbsBayes: An R Package for Hierarchical Bayesian Analysis of North American Breeding Bird Survey Data. *Journal of Open Research Software,* 9: 19. DOI: *https://doi.org/10.5334/jors.329*

Submitted: 01 April 2020 Accepted: 07 July 2021 Published: 20 July 2021

Journal of Open Research Software is a peer-reviewed open access journal published by Ubiquity Press.