Simulating Effects of Imperfect Detectability in Bird Surveys

A Dissertation

SUBMITTED TO THE FACULTY OF THE

UNIVERSITY OF MINNESOTA

BY

ELIZABETH ANNE RIGBY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Dr. Douglas H. Johnson

November, 2016

Acknowledgments

I would like to thank my advisor, Doug Johnson, and my committee, David Andersen, Todd Arnold, Matthew Etterson, and James Nichols, for their invaluable guidance, advice, know-how, and good humor throughout this process. Thank you to the organizations and individuals who performed and supported the field work in Chapter 3: USFWS Morris Wetland Management District, The Belwin Conservancy, Three Rivers Park District, Lisa Harn Elliott, Larry Gillete, Tara Kelly, Chris Potter, and Sarah Thompson. Thank you to USFWS Region 3 Ecological Services, who supported me with an internship for my last 2 years of graduate school. Thank you to the staff at USGS Patuxent Wildlife Research Center and USGS Bird Banding Lab, whose input was immensely helpful for designing this project. Thank you to Marcus Beck, Peter Blancher, Bob Dunlap, Alex Egan, Lawrence Gray, Scott Loss, and Krishna Pacifici, who graciously provided technical advice, expertise, and data. Thank you to David Tyndall, J. Adam Rogers, Julie Rose, Robert Seybold, and Ginny Weckstein, whose advice about graduate school was instrumental in getting me to this point.

I would also like to thank my family, Julie and Wayne Rigby, Jane Rigby, Andrea Leistra, and Ayden Rigby-Leistra for their ceaseless support and encouragement. Most of all, my thanks and gratitude go to Shane Fjestad, who has seen me through the good days and the bad days and never doubted that I could do this.

Abstract

Counts obtained from point count bird surveys can be treated as an index to bird abundance, but imperfect detectability can complicate inferences about abundance. Adjustment analysis methods, including double-observer, replicated counts, removal, and distance sampling methods, have been developed to estimate detection in addition to abundance. These methods require additional information to estimate detection, which may entail added logistical costs or be additional sources of error. It is not clear when or if adjustment methods outperform index methods, or how the benefits of adjustment methods compare to their costs. I simulated point counts of birds, modeling birds spatially as moving within bivariate normal territories, modeling song production as an autocorrelated process, and modeling perceptibility as a logit function of distance to the observer. In Chapter 1, I simulated counts using a test scenario with parameters reflecting surveys of black-throated blue warblers (BTBW, *Setophaga caerulescens*), analyzed counts using index and adjustment analysis methods, then evaluated and compared the performance of analysis methods. Estimates from index methods underestimated true density of birds (D_p) for all survey types, but were highly correlated with true density. Adjusted estimates from distance sampling and removal analysis methods showed a reduction in bias as compared to index estimates, but had reduced correlation with true density. Adjusted estimates from double-observer analysis methods were nearly unchanged from index estimates. Adjusted estimates from replicated counts analysis methods were susceptible to highly inflated density estimates, resulting in extremely high bias and low correlation with true density. Index methods, while biased, were better correlated with true density and would provide better information about changes in abundance than an adjustment analysis method for the BTBW scenario. If detection is constant and relative abundance is sufficient to meet survey objectives, using an index method is often preferable. For systems with variable detection probability where inference about absolute abundance is necessary to meet objectives, practitioners should select adjustment methods suited to model the source of imperfect detection in their system. Ill-suited adjustment methods will not improve inference and are no more useful than an index.

In Chapter 2, I used the model to simulate counts for scenarios with high or low availability and high or low perceptibility. I also included scenarios where abundance was confounded with perceptibility, and scenarios where they were independent. I then analyzed count data using index methods and adjustment methods. Although index methods were biased and only had a strong correlation with true density when detectability was high, adjustment methods generally did not offer an improvement. As compared to index methods, adjustment method performance ranged from far worse (replicated counts), to no added value (double-observer) to moderate improvement (in bias only, for removal and distance sampling in specific scenarios). Practitioners should carefully consider the sources of variation in detection probability in their system. If detection components are unknown or known to be variable, I advise practitioners to perform a pilot study to estimate detection components. Additionally, practitioners should standardize their methods to increase availability and perceptibility in their surveys and to lower the variation in these detection components.

In Chapter 3, I conducted simulated bird surveys using recorded bird songs to assess factors affecting detection probability in grassland bird point counts. I used mixed effects logistic regression models to estimate factors affecting detection probability and to estimate and visualize the variation in the area around the observer where birds can be perceived (the perceptible area). I conducted simulated surveys with 8926 binary opportunities for detection in Minnesota grasslands in 2011 and 2012. Species, distance to the observer, wind speed and direction, observer, and density of vegetation all affected detection of recorded bird songs. Species had a strong effect; the size of the predicted perceptible area around the observer differed by more than 10-fold among species. Wind also had a strong effect on detection. As wind speed increased, probability of detection downwind of the observer was reduced and the perceptible area around the observer became smaller and more asymmetrical. The effective distance at which an observer is more likely to detect a bird than to not detect it may differ among species and angles to the wind, even within the same survey. I recommend using fixed-radius counts for bird surveys in grasslands and reducing the variation in detection probability by standardizing surveys across wind conditions.

Table of Contents

List of Tables

List of Figures

Chapter I

Effects of Imperfect Detection on Inferences from Bird Surveys: A Simulation Study

Summary: Counts obtained from point count bird surveys can be treated as an index to bird abundance, but imperfect detectability can complicate inferences about abundance. Adjustment analysis methods, including double-observer, replicated counts, removal, and distance sampling methods, have been developed to estimate detection in addition to abundance. These methods require additional information to estimate detection, which may entail added logistical costs or be additional sources of error. It is not clear when or if adjustment methods outperform index methods, or how the benefits of adjustment methods compare to their costs. I simulated point counts of birds, modeling birds spatially as moving within bivariate normal territories, modeling song production as an autocorrelated process, and modeling perceptibility as a logit function of distance to the observer. I simulated counts using a test scenario with parameters reflecting surveys of black-throated blue warblers (BTBW, *Setophaga caerulescens*), analyzed counts using index and adjustment analysis methods, then evaluated and compared the performance of analysis methods. Estimates from index methods underestimated true density of birds (D_p) for all survey types, but were highly correlated with true density. Adjusted estimates from distance sampling and removal analysis methods showed a reduction in bias as compared to index estimates, but had reduced correlation with true density. Adjusted estimates from double-observer analysis methods were nearly unchanged from index estimates. Adjusted estimates from replicated counts analysis methods were susceptible to highly inflated density estimates, resulting in extremely high bias and low correlation with true density. For replicated counts, using the maximum count among replicated counts (an index method) produced estimates with lower bias than N-mixture model estimates. Index methods, while biased, were better correlated with true density and would provide better information about changes in abundance than an adjustment analysis method for the BTBW scenario. If detection is constant and relative abundance is sufficient to meet survey objectives, using an index method is often preferable. For

systems with variable detection probability where inference about absolute abundance is necessary to meet objectives, practitioners should select adjustment methods suited to model the source of imperfect detection in their system. Ill-suited adjustment methods will not improve inference and are no more useful than an index.

Point count bird surveys are commonly used to address a number of objectives, including abundance estimation and population monitoring (Scott and Ralph 1981). Point counts are ubiquitous in avian monitoring, yet there is significant debate regarding how count information can best be used. Of particular importance is the role played by detection probability (p, see Glossary, Appendix A), which is described as

$$p = \frac{C}{N} \text{ (eq. 1.1),}$$

where C is the count obtained during a point count survey and N is the number of birds present.

If p is constant, or if variation in p is small compared to variation in C (Johnson 2008), C can serve as an index to N, which is the basis for index surveys (Dawson 1981, Conroy 1996). The relationship between N and C can become muddied or totally obscured if variation in p is great or associated with sites being compared (e.g., different detection probabilities for different habitats). If p and N are not independent, C could provide misleading information about N. Index methods do not directly provide information about p, and both p and N are unknown in field surveys. Any inferences about N from index methods must therefore rely on an assumed relationship between N and C.

To better discuss the factors affecting detection probability, p can be broken into parts, as by Nichols et al. (2009). They address detection of birds within a superpopulation (N^*), defined as all birds whose territories or home ranges at least partially overlap the area over which inferences will be made (the area of inference). Nichols et al. (2009) decomposed p from equation 1.1 into 4 parts: p_s, the probability that a bird's territory at least partially overlaps the surveyed area of a survey site; p_p, the probability that a bird is present in the surveyed area at the time of the survey, given that its territory at least partially overlaps the surveyed area of a survey site; p_a, availability, the probability that a bird is available (e.g., vocalizes) during a survey, given that it is present; and p_d, perceptibility, the probability that a bird is detected, given that it is present in the surveyed area, and available during the survey. The expected value for a count (E(C)) during a survey is thus

$$\mathrm{E}(C) = N^{*} p_s \, p_p \, p_a \, p_d, \text{ (eq. 1.2).}$$

Detection probability can be affected by a wide variety of factors (Verner 1985), including species (Diefenbach et al. 2003), survey elements (length of survey, survey type; Dawson et al. 1995, Bollinger et al. 1988, Cimprich 2009), behavioral factors (singing rate, volume, and motion of birds; Wilson and Bart 1985, McShea and Rappole 1997, Alldredge 2007b), environmental factors (precipitation, wind speed, ambient noise, time of day, time within the breeding season, even tides; Robbins 1981a, Wilson and Bart 1985, Zembal and Massey 1987, Rosenberg and Blancher 2005, Pacifici et al. 2008), and observer effects (hearing ability, skill, distance from the source; Sauer et al. 1994, Alldredge 2007b). Survey methods can be adjusted to reduce variability in detection probability, such as by using experienced observers (Robbins et al. 1986), training observers (Kepler and Scott 1981), or using a standardized survey length and survey radius (Ralph et al. 1993, Ralph et al. 1995, North American Breeding Bird Survey 1998, Matsuoka et al. 2014). Survey methods may also be adjusted to maximize components of detection probability, such as maximizing p_a by conducting surveys when birds are most likely to sing (Robbins 1981b) or maximizing p_d by constricting survey radius such that all available birds can be assumed to be detected (Ralph et al. 1995).

Two major groups of analysis methods are available for making inferences about abundance: “index methods”, which assume that counts are an index to abundance, and “adjustment methods”, which estimate both detection probability and abundance. The simplest form of an index is the simple count analysis method, where counts from a survey are used as an estimator of abundance. Other indices add information to reduce bias. A maximum count uses the largest count among repeat visits to a site as an index to abundance. A bounded count also requires repeat visits, using twice the maximum count, minus the second largest count (Robson and Whitlock 1964, Johnson et al. 2007). Common use of indices includes acknowledging their limitations; Seber (1982) recommends that users "simply recognize that the estimates are biased and treat them as relative rather than absolute measures of abundance" (pg. 458). I consider four adjustment methods. Double-observer (also referred to as multiple-observer) analysis methods (Cook and Jacobson 1979, Nichols et al. 2000) use the discrepancies in individual detections between two observers to estimate detection probability. Distance sampling analysis methods (Burnham et al. 1980, Buckland et al. 1993) estimate detection probability as a function of distance from the observer, assuming that detection probability at the observer's location is 1. Removal (Farnsworth et al. 2002) and time-of-detection (Alldredge et al. 2007a) analysis methods estimate detection probability by comparing initial detections that occur during different periods of the survey. The replicated counts analysis method (Royle 2004) uses N-mixture models to estimate detection probability across sites with temporally replicated counts. By estimating components of detection probability (eq. 1.2), adjustment methods theoretically reduce the bias for estimates of N, as compared to index analysis methods which do not estimate p.

Within an analysis method, one or more estimators may be used to estimate abundance, where an estimator is defined as a statistic (i.e., a function of the data) that is used to infer the value of an unknown parameter. For example, in the simple count analysis method, the sum of counts across all sites within a year forms the "simple index estimator" for the number of birds present at those sites. It is important to note that the meanings of the quantities estimated by detection and abundance estimators differ among adjustment analysis methods, and that no single adjustment method estimates all components of p. Recall that for index methods, variation in detection probability is not separated from abundance (equation 1.2), so counts estimate $N^{*} p_s\, p_p\, p_a\, p_d$. Distance sampling and double-observer analysis methods model detection probability using variation in the observation process, exhibited by heterogeneity of detection due to distance from the observer for distance sampling methods, and discrepancies in the detection of individual birds by observers for double-observer methods. The detection estimator for these methods therefore estimates p_d, and the abundance estimator estimates $N^* p_s\, p_p\, p_a$. Removal or time-of-detection analysis methods (hereafter called removal analysis methods) model detection probability using temporal variation within surveys, which incorporates variation both in the observation process and availability (e.g., song production). The detection estimator for removal methods therefore estimates $p_a\, p_d$ and the abundance estimator estimates $N^* p_s\, p_p$. For the replicated counts analysis method, detection probability is modeled using variation among temporally replicated visits to multiple sites, which includes variation in observation, availability, and movement of birds within their home ranges. The detection estimator for replicated counts methods therefore estimates $p_p\, p_a\, p_d$ and the abundance estimator estimates $N^* p_s$.

When comparing abundance estimates across analysis methods, it is often preferable to discuss the density of birds (D) within the surveyed area (A), where

$$D = \frac{N}{A} \quad \text{(eq. 1.3).}$$

Density estimates account for any differences across methods in area surveyed, while abundance estimates do not. For example, fixed-radius surveys set a maximum distance from the observer beyond which birds are not recorded (Ralph and Scott 1995), but distance sampling methods customarily determine a maximum distance for observations by truncating a percentage of the most-distant observations (Buckland et al. 2001). If comparing years of surveys with different numbers of survey sites, annual abundance across sites must be summarized as mean site-abundance or converted to density. To avoid confusion when comparing surveys with different radii or number of sites, I use density rather than abundance to discuss quantification of bird populations.

Although these analysis methods are conceptually attractive, there can be drawbacks to using them. Double-observer methods (Nichols et al. 2000) require data collected via multiple field observers making simultaneous observations and replicated count methods (Royle 2004) require data collected with repeat visits to sites. For both analysis methods, if human effort is kept constant, the number of overall sites visited is reduced compared to a simple count, resulting in smaller survey extent. Removal methods (Farnsworth et al. 2002, Alldredge et al. 2007a) typically require the observer to spend more time at each site compared to a simple count survey, at some logistical cost. Birds also have more opportunities to move during longer survey periods common in removal survey types, which could bias results (Scott and Ramsey 1981, Granholm 1983, Dawson et al. 1995). Johnson (2008) argued that adjustment methods are "an assumption or a consequence of an assumption" and that their use is not universally preferable to index methods. Adjustment methods may entail additional sources of error, such as when distance must be estimated by observers for distance sampling methods (Alldredge et al. 2007c) or consensus must be reached by multiple observers for double-observer methods (Alldredge et al. 2006). How birds are surveyed has the potential to affect inferences about bird density, but these effects have not been quantified, nor has consensus been reached on the best analysis methods to use in different situations.

Thus far, most comparisons among analysis methods have been field studies (e.g., Moore et al. 2004, Forcey et al. 2006, Thompson and La Sorte 2008). The drawback of modeling detection probability in field studies is that the true population is unknown, so the accuracy of estimators is also unknown. By simulating counts and comparing analysis methods, it is possible to compare estimates to a known population, but few such simulation studies have been attempted. Efford and Dawson (2009) assessed bias in abundance estimators by simulating counts and including heterogeneity in detection due to distance from the observer. Although that study provided valuable information on estimator performance, it only addressed one component of detection probability (p_d) and one source of detection variation (distance to the observer). Until now, no comprehensive simulation has included variation in all components of detection (p_s, p_p, p_a, and p_d).

Here, I present a model of bird surveys that incorporates variation in detection at multiple levels to produce counts that can be analyzed with a variety of estimators for index and adjustment analysis methods. I describe the model generally and evaluate the performance of density estimators using parameters reflecting surveys of black-throated blue warblers (BTBW, *Setophaga caerulescens*).

BTBW have several advantages for a preliminary analysis. First, nesting densities have been shown to be correlated with density of shrubs (Steele 1992, Steele 1993, Holmes et al. 2005), which is easily translatable into a model where a habitat variable affects abundance. BTBW are relatively well-studied, with ample spatial information available (Holmes et al. 2005), so parameters were based on empirical data. BTBW exhibit stable or increasing population trends across their range (Holmes et al. 1986, Sauer et al. 2014), making it a non-controversial choice for an example species. Finally, recordings of BTBW songs were used by Pacifici et al. (2008), Alldredge et al. (2007b), and Simons et al. (2007) in their estimations of perceptibility, allowing the parameterization of perceptibility to be based on more empirical data than would be possible for most species.

Methods

Model Structure

I coded model in Program R (R Development Core Team, Vienna, Austria, http://www.R-project.org). I simulated surveys for singing males of one species at a time. I refer to each research question to which I applied this model as a scenario. Input parameters (affecting, for example, song rate or movement rate) were customized for each scenario to reflect the biology of bird species and habitat of interest.

The conceptual foundation for this model was equation 1.2. Based on this deterministic conceptual model, I developed a stochastic simulation model of the detection process in bird surveys. I used stochastic processes to model variation in detection at three fundamental levels: spatial arrangement (represented conceptually by p_s and p_p), availability (p_a), and perceptibility (p_d).

In discussing bird surveys, I distinguish the “survey type”, meaning a specific survey scheme of temporal and spatial replication of survey sites and number of observers, from the analysis method. I consider five survey types: (1) simple counts, where counts are conducted without collection of accessory data, (2) double-observer, where counts are conducted with two simultaneous observers, (3) distance sampling, where the observer estimates the distance to each bird counted, (4) removal, where the survey period is split into 3 time periods and the observer records the time period in which each bird was first detected, and (5) replicated counts, where counts are conducted at sites visited 3 times within each season (Table 1.3). More than one analysis method can often be applied to data from a particular survey type, e.g., count data from a double-observer survey type can be analyzed using an index method or an adjustment method. Adjustment methods require data gathered in a specific survey type. For example, to use a replicated counts analysis method (Royle 2004), the data must be collected in a replicated counts survey type, where multiple visits to sites are conducted within a season.

A single run of the model represented one year or season of surveys. To make reference to the R code easier, I refer to variables by their R object names, in italics. The stochastic model structure included 6 hierarchical levels: (1) year (*y* = 1, 2, … , *NYears*), (2) site, defined as a single point visited and surveyed by the observer(s) (*i* = 1, 2, ... , *NSurveySites*,), (3) bird (*j* = 1, 2, … , *NBirds.yi*), (4) replication, defined as the within-season visit to a survey site (*r* = 1, 2, …, *NReps*), (5) interval, defined as a short period of time akin to the duration of one bird song, usually 2 or 3 seconds (*k* = 1, 2, … , *NIntervals*), and (6) observer (*o* = 1, 2, …, *NSimultaneousObservers*). Interval length was constant within a scenario. Survey types allot survey effort differently, which required running *NYears* simulations for each survey type within a scenario (e.g., for *NYears* = 30, 150 total simulations were necessary to generate data for 5 survey types). Variable *NYears* was scenario-specific, *NSurveySites*, *NReps*, *NIntervals*, and *NSimultaneousObservers* were survey type-specific, and *NBirds.yi* was site-specific.

Within the code for the model (Appendix B), R object names included an identifier, similar to a subscript, but instead connected with a “.”. For example, an observer-specific effect is coded with a “*.yijrko*”, such as a detection of bird *j* by observer *o* in year *y*, at site *i*, during interval *k* of replication *r* (e.g., *Detected.yijrko*). Scenario-specific and survey type-specific effects have no identifier, such as the number of sites surveyed annually, *NSurveySites*.

Abundance

I modeled surveys for each site *i* in a 2000 m x 2000 m Cartesian grid centered on the observer (total area = 400 ha). I chose this scale to be sufficiently large that birds could move into and out of the observational range of the observer during the survey.

I modeled abundance as a function of habitat available at sites. The proportion of site *i* covered by habitat (*PercentHabitat.yi*) was drawn from a scenario-specific distribution. The remainder of each site was covered by matrix, with a proportion equal to 1-*PercentHabitat.yi*. Mean density of birds in habitat (*HabitatDensity.y*) was greater than or equal to mean density of birds in matrix (*MatrixDensity.y*). Specialist species could be modeled with greater disparity between *HabitatDensity.y* and *MatrixDensity.y*; generalist species could be modeled with *HabitatDensity.y* ≈ *MatrixDensity.y*.

I modeled site-specific abundance as Poisson-distributed (*NBirds.yi*), modeled as the sum of 2 Poisson-distributed random variables, *BirdsInHabitat.yi* and *BirdsInMatrix.yi*, which were described respectively by parameters *LambdaHabitat.yi* and *LambdaMatrix.yi*. *LambdaHabitat.yi* and *LambdaMatrix.yi* were each a product of the size of the modeled area around the observer (*Area.yi* = 400 ha), the density of birds in habitat or matrix (*HabitatDensity.y* and *MatrixDensity.y*), and *PercentHabitat.yi*. Thus,

$$\begin{gathered} \mathit{NBirds.yi} \sim [\ \mathrm{Poisson}(\mathit{HabitatDensity.y} \times \mathit{Area.yi} \times \mathit{PercentHabitat.yi}) + \\ \mathrm{Poisson}(\mathit{MatrixDensity.y} \times \mathit{Area.yi} \times (1\text{-}\mathit{PercentHabitat.yi}))\ ]\ . \quad \text{(eq. 1.4)} \end{gathered}$$

Spatial Modeling

I modeled locations for each bird *j* using x- and y-coordinates based on a bivariate normal distribution, resulting in elliptical territories. Spatial parameters (*Spatial.yij*) were generated for each bird *j*, including the center of the territory (*CenterX.yij, CenterY.yij*), the area of a 95% elliptical density contour (*Area.yij*), the eccentricity of the ellipse (*Ecc.yij*), and an angle of rotation (*Theta.yij*; for distributions governing spatial parameters, see Appendix C).

I allowed territories to overlap only peripherally. I compared territory edges using a scenario-specific percentage for the elliptical density contour, and eliminated territories if any of the 4 axes or center overlapped the utilization ellipse of any other territory. Direct overlap comparison was computationally intensive, so I added a precursor step to reduce simulation run-time (Appendix C). Each bird *j* had an interval-specific location (*Location.yijrk*) and distance from the observer(s) (*Distance.yijrk*). For each interval $k > 1$, a Bernoulli-distributed random variable *DoesBirdMove.yijrk* was generated via the mean probability that the bird moved (*PrBirdMoves.yijrk*), parameterized with movement information for the species. If *DoesBirdMove.yijrk* = 1, a new *Location.yijrk* was generated from the bivariate normal distribution. If *DoesBirdMove.yijrk* = 0, then the location remained the same (*Location.yijrk* = *Location.yijr(k-1)*).

Availability

I modeled bird availability as an interval-specific event *Sings.yijrk* with 2 possible states (1 = song, 0 = no song). To produce temporal song patterns reflective of breeding males, I incorporated autocorrelation at 2 scales. Coarse-scale temporal autocorrelation referred to bird *j* being in or out of “singing mode,” a state in which vocalization is frequent and songs occur at relatively regular intervals. If the bird was not in singing mode (i.e., if *SingingMode.yijrk* = 0), then bird *j* necessarily did not vocalize during interval *k* and *Sings.yijrk* = 0. If the bird was in singing mode, (i.e., if *SingingMode.yijrk* = 1), then *Sings.yijrk* could be 0 or 1: these vocalizations were modeled with fine-scale temporal autocorrelation.

I modeled coarse-scale temporal autocorrelation with a positive recurrent Markov chain (Stroock 2005), where the state of bird *j* at interval *k* is related to its state at interval *k*-1. There were 2 possible states, being in singing mode (S) or not being in singing mode (NS). The coarse-scale transition matrix P_{coarse} described the probabilities of remaining in a state or switching states, given the previous state (Table 1.1). For example, for a bird that was in singing mode in the previous interval, P(S|S) is the probability of remaining in singing mode and P(NS|S) is the probability of switching to non-singing mode. Because the rows of any transition matrix sum to 1,

$$P(S|S) + P(NS|S) = 1 \ , \ \text{(eq. 1.5)}$$

and

$$P(S|NS) + P(NS|NS) = 1 \ . \ \text{(eq. 1.6)}$$

Empirical data describing singing mode is scarce. Indeed, if singing mode is a biological phenomenon, it may be impossible to accurately measure in the field; birds that are in singing mode but not currently singing are indistinguishable from birds not in singing mode. There are, however, some species-specific data describing the probability that a bird sings at least once in a several minute period (e.g., Emlen 1977). I refer to this type of song rate information as the "singing probability," and assume that the number of birds in singing mode that never sing is negligible. By making some assumptions about the transition matrix, I created a Markov chain process that produced a population of simulated birds with the desired singing probability.

The steady-state vector [q_1 q_2] (also called the limiting or stationary distribution) is the vector of the proportion of time spent in each state (singing mode, q_1, or non-singing mode, q_2) in the long run (i.e., after the initial state has been "forgotten"; Stroock 2005). It also represents the average proportion of the population in each state in any given interval. The probability that a bird *j* is in singing mode at least once in *NIntervals* intervals, or singing probability, is:

$$Z_{NIntervals} = q_1 + (1 - q_1) \times (1 - P(NS|NS)^{NIntervals}) \ , \ \text{(eq. 1.7)}$$

where q_1 is the probability that bird *j* begins in singing mode and $(1 - P(NS|NS)^{NIntervals})$ is the probability that the bird never switches into singing mode (given that it did not begin in singing mode; for additional information about modeling $Z_{NIntervals}$, see Appendix C). For birds in singing mode, I modeled fine-scale autocorrelation with a Markov chain to produce the desired pattern of songs and pauses. Interval length equaled the length of a song in the model, so an interval represented a binary opportunity for detection. The Markov chain had 4 states: singing (S) and 3 stages of not singing (NS1, NS2, NS3). Singing was always followed by 2 intervals without singing (P(NS1|S) = P(NS2|NS1) = 1). After the second interval of non-singing, the bird would either sing or not sing, according to the probabilities in the fine-scale transition matrix *TransitionMatrix.fine*. I used successive approximation to select transition probabilities that produced pauses in the song pattern with the desired pause length mean and standard deviation.

Perceptibility

Observer-specific perceptibility was modeled as a Bernoulli-distributed event with probability of detection *perceptibility.yijrko*. If bird *j* was detected by observer *o* during interval *k*, then *Detected.yijrko* = 1; if not, *Detected.yijrko* = 0. Modeling of *perceptibility.yijrko* was based on a logit link. Specifically,

$$\text{Logit}\ (perceptibility.yijrko) = \log\left(\frac{perceptibility.yijrko}{1-perceptibility.yijrko}\right) = \beta_0 + \beta_v X_v\ , \text{(eq. 1.8)}$$

for *v* covariates X with coefficients β.

Observers' estimation of the distance between the observer and birds may include observer error (Alldredge et al. 2007c). I simulated observer-estimated distances (*ObsEstimatedDistance.yijrko*) stochastically, adding an observer-specific error due to observation to the true distance *Distance.yijrk*.

Each bird had an observer-specific count status *Count.yijrko*, where *Count.yijrko* = 1 indicated that the bird was counted and *Count.yijrko* = 0 indicated that it was not counted. If a bird was not detected (*Detected.yijrko* = 0), then *Count.yijrko* = 0. If a bird was detected (*Detected.yijrko* = 1), it was counted only if the observer estimated distance to the bird (*ObsEstimatedDistance.yijrko*) was within the survey radius (*MaxSurveyDistance*).

Analysis Methods

For all adjustment analysis methods, I estimated abundance within each year by fitting models and comparing them using AIC (Burnham and Anderson 2002). For removal sampling analysis, I fit models using function multinomPois() in Program unmarked (Fiske and Chandler 2011), a package for Program R. To fit N-mixture models, I used function pcount() in Program unmarked. For removal and N-mixture analysis methods, I compared 3 models: a null model, and models with *PercentHabitat.yi* as a site-specific covariate affecting abundance or detection.

I used function multinomPois() in Program unmarked to obtain adjusted abundance estimates for the double-observer survey type, using the independent observer approach (Alldredge et al. 2006). I compared a null model, a model with observer effect, and models with an observer effect and with *PercentHabitat.yi* as a site-specific covariate affecting abundance or detection (4 models total). I also used the Nichols et al. (2000) estimator to obtain adjusted abundance estimates using the dependent observer approach.

I carried out conventional and covariate distance sampling analysis with Program Distance version 6.2 release 1 (Thomas et al. 2010) and hierarchical distance sampling with function distsamp() in Program unmarked. With Program Distance, I fitted 9 conventional distance sampling models (all combinations of 3 key functions, half-normal, hazard rate, and uniform, and 3 adjustment methods, cosine, polynomial, and hermite; Buckland et al. 2001). I also fitted two models with a site-level covariate for *PercentHabitat.yi*, using hazard-rate and half-normal key functions. With Program unmarked, I tested 3 null models using each of the key functions, and models for all key functions using *PercentHabitat.yi* as a covariate affecting abundance or detection (9 models total). All models used continuous distance (not binned). I fitted models and estimated parameters within year using Program unmarked (where one year was one model run). To fit models and estimate parameters in Program Distance, I used year as "region" and obtained estimates for each year.

For non-distance sampling survey types, I used 3 radii for surveys (*MaxSurveyDistance* = 50 m, 100 m, and 150 m). For the distance sampling survey type, I truncated the most distant 10% of observations to prevent model over-fitting, as suggested by Buckland et al. (2001), and used the year-specific truncation distance as *MaxSurveyDistance*. For all adjusted estimates, I estimated abundance and converted it to density by dividing by the survey area A, where $A = \pi \times MaxSurveyDistance^2$.

In evaluating estimators, it is critical to consider what "true" density is used for comparison. Invoking Nichols et al. (2009), I used equation 1.2 and conceptual equation

1.1 to define 4 abundances and densities that may be useful when providing comparison to estimators. Abundance $N_s = N^* p_s$ and corresponding density $D_s = N^* p_s / A$ refer to birds with territories that overlapped the survey radius. Because territories were defined in my model by a bivariate normal probability distribution, the exact boundaries of territories were uncertain. I defined site-specific N_s (*CloseTerrBirds.yi*) as the number of birds having territories with 95% utilization distributions overlapping the survey radius. Abundance $N_p = N^* p_s\, p_p$ and corresponding density $D_p = N^* p_s\, p_p / A$ refer to birds present within the survey radius at the beginning of the survey. Site-specific N_p was *CloseBirds.yi*. Abundance $N_a = N^* p_s\, p_p\, p_a$ and corresponding density $D_a = N^* p_s\, p_p\, p_a / A$ refer to birds available, given that they were present within the survey radius at the beginning of the survey. Site-specific N_a was *CloseSingers.yi*. Conceptually, the realized count $C = N^* p_s\, p_p\, p_a\, p_d$ refers to birds detected, given that they were available and present within the survey radius at the beginning of the survey. Site-specific C was *RawCount.yi*, and included birds that were detected and estimated by the observer to be within the survey radius at the time of first detection. Thus, *RawCount.yi* could include birds that moved into the survey radius during the survey or were falsely estimated to be within the survey radius and were not included in *CloseSingers.yi*.

Assuming a closed population, where birds do not move among sites and abundance is constant within surveys, the total abundance across sites is $\Sigma\, N_p$. For bird surveys where the objective is to make inference about abundance across sites, N_p is therefore the optimal “true” abundance to use for comparison when evaluating estimators. I therefore used D_p as the primary representative of true density for comparisons of estimators, with D_s and D_a reported for reference.

Within each survey type, I report true density, index method density estimates, and adjustment method density estimates (birds/ha) for the sake of comparison, even though indices are not usually assumed to directly estimate density (e.g., Seber 1982). I calculated true density as true abundance divided by the survey area A, where $A = \pi \times MaxSurveyDistance^2$. Mean and standard deviation for density estimators and true density were calculated across 30 simulated years.

The primary index I use is the simple index density estimator, or the sum of counts across all sites within a year, divided by the area surveyed. The simple index estimator with perfect distance estimation refers to the birds that would have been counted if the observer's estimation of distance were perfect (i.e., there were no errors in determining if birds were inside or outside the survey radius), and is provided as a comparison to the simple index estimator to illustrate the effect of distance estimation error on indices. I also report density estimates from 2 additional indices for the replicated count survey type. Maximum count density was the sum across sites of all site-specific maximum counts (among the 3 counts within a year), divided by the area surveyed. Bounded count density was the sum of the bounded counts (twice the maximum count, minus the second largest count; Johnson et al. 2007), divided by the area surveyed. For adjustment methods, I report model-averaged density estimates and estimates produced by the top model in each year. For all estimates, I removed outliers and reported the number removed, where outliers were defined as density estimates that were > 3 standard deviations away from the mean of remaining estimates.

To investigate the relationship between estimates and true density, I calculated year-specific bias as the difference between estimated density and true D_p (negative bias indicated underestimation of true density). I report median bias to reduce the effect of some density estimates that were inflated. I also calculated Pearson correlation coefficients between density estimators and D_s, D_p, and D_a. I calculated bias and correlation coefficients after removing outliers.

Parameters for BTBW Scenario

For the BTBW scenario, I ran 30 simulations representing 30 years of simulated data for each survey type (simple count, double-observer, removal, replicated count, and distance sampling). *NSurveySites*, *NReps*, *NSimultaneousObservers*, and *SurveyLength* were survey type-specific and resulted in the same amount of human effort for each survey type (Table 1.3).

Surveys began on ordinal date 150 (i.e., May 30). For removal surveys, 6 surveys occurred per day; for all other methods, 7 surveys occurred per day. Simulated survey time and travel time between sites varied by survey type (Table 1.3), with a combined time with length *LogisticalSurveyTime* minutes. Surveys were planned to begin immediately after the previous survey, starting at dawn (*TimeinIntervals.yirk* = 0 intervals) with start times *PlannedStartTimesAll*. To model variation in travel time, the actual start times for surveys (*ActualStartTimesAll*) were normally distributed, with mean = *PlannedStartTimesAll* and SD = 5 minutes. For the removal survey type, *PlannedStartTimesAll* had range 0-150 minutes after sunrise. For all other survey types, *PlannedStartTimesAll* had range 0-138 minutes after sunrise.

Abundance Parameters

I used an empirical estimate for density of breeding pairs in BTBW habitat (*HabitatDensity.y* = 0.534 birds/ha, Holmes et al. 1986) and derived the density of birds in matrix (*MatrixDensity.y* = 0.00305 birds/ha) from additional empirical parameters (Appendix C). I assumed a year-specific proportion of the study area was covered by habitat (*StudyHabitatProportion.y* ~ U(0.7,1)). The proportion of site *i* in year *y* covered by habitat (*PercentHabitat.yi*) was drawn from a beta distribution with μ = *StudyHabitatProportion.y* and $\theta = 8$ (alternative parameterization of the beta distribution from Link and Barker 2010: 319). As a result, *PercentHabitat.yi* had mean = 0.85 (SD=0.14). Site-specific abundance was given by equation 1.4. For an average site (*PercentHabitat.yi* = 0.85), mean density was 0.454 birds/ha.

Spatial Parameters

Territory overlap was evaluated using 56.5% elliptical density contours. This level of overlap was chosen to allow density of birds in habitat *HabitatDensity.y* = 0.534 birds/ha (Holmes et al. 1986) and mean territory size *MeanTerrArea* = 3.6 ha (Sherry and Holmes 1985). I used standard deviation of territory size *SDTerrArea* = 1.0 ha. Bird-specific territory size *Area.yij* was log-normally distributed, where *Area.yij* ~ lognormal(3.6 ha, 1.0 ha).

Interval-specific probability of movement (*PrBirdMoves.yijrk*) was normally distributed, *PrBirdMoves.yijrk* ~ N(0.005, 0.0005). Using these parameters, a 3-minute survey will have 36% of birds modeled that move at least once on average, with a mean number of movements = 0.45; a 10-minute survey will have 78% of birds modeled that move at least once on average, with a mean number of movements = 1.5. These movement parameters were chosen to reflect a frequently-moving species; for 3 passerine species, Granholm (1983) found the probability of movement within a 10-minute period was 36%, 64%, and 72%.

Availability Parameters

To estimate song length, I timed songs and pauses to the nearest second for the first minute of 4 BTBW recordings from the Macaulay Library (1992, 1994, 2000, 2010). Mean song length in recordings was 2.11 seconds (SD = 0.33 seconds), mean pause length was 6.64 seconds (SD = 1.47 seconds). Assuming an equal ratio of songs and pauses (songs having length *SongLength* and pauses having length *PauseLength*, both having units in intervals), the proportion of time spent singing in the recordings was *q1.fine*, where

$$q1.fine = SongLength/(SongLength+PauseLength) = 0.241. \quad \text{(eq. 1.9)}$$

I selected fine-scale availability parameters to produce a similar pattern to that observed in the Macaulay Library (1992, 1994, 2000, 2010) recordings. I set the length of a single interval, *IntervalLength* = 2 seconds, which also represented the length of a song. Through successive approximation, I determined values for transition matrix *TransitionMatrix.fine* (Table 1.2). Monte Carlo simulations using *TransitionMatrix.fine* produced song patterns with mean pause length = 6.3 seconds (SD = 1.25 seconds) and a realized proportion of time spent singing while in singing mode = 0.241 (akin to *q1.fine*, eq. 1.9).

For coarse-scale autocorrelation, the interval-specific probability of being in singing mode for at least 1 interval within 10 minutes was termed the singing probability, or *PrSing.yijrk*. *PrSing.yijrk* was modeled as a function of day within the breeding season (represented as a maximum daily song rate, *MaxDailySongRate.yir*) and time of day (represented as a weight, *SongWeight.yijrk*),

$$PrSing.yijrk = MaxDailySongRate.yir \times SongWeight.yijrk \text{ . (eq. 1.10)}$$

The replication-specific maximum daily song rate (*MaxDailySongRate.yir*) was based on arrival and departure dates of birds on breeding territories (Holmes et al. 2005, Figure 1.2). *MaxDailySongRate.yir* = 0.9 for the BTBW scenario, which is representative of warblers during periods of high availability (e.g., Stacier et al. 2006, Robbins et al. 2009). The result was *PrSing.yijrk* with range 0.72 – 0.90 for the period when most surveys occurred (from dawn until surveys completed, approximately 150 minutes later). *PrSing.yijrk* as low as 0.18 was possible for pre-dawn surveys, but occurred rarely (Figure 1.3). The function for *SongWeight.yijrk* (Figure 1.3) was selected to produce *PrSing.yijrk* values that reflected information from Breeding Bird Surveys (P. Blancher, Environment Canada, personal communication). Values for the transition matrix P_{coarse} (Table 1.1) were selected via optimization such that $Z_{NIntervals}$ (eq. 1.7) equaled the desired *PrSing.yijrk*, with constant P(S|S) = 0.98.

Perceptibility Parameters

Using equation 1.8, I chose parameters affecting *perceptibility.yijrko* (Figure 1.4) based on detection estimates for BTBW in Pacifici et al. (2008). For the BTBW scenario, I used coefficients (Table 1.4) from Pacifici et al. (2008) estimated for mixed pine-hardwood forest with leaves present, assuming those conditions were most applicable to BTBW breeding season surveys. I modeled ambient noise as a replication-specific binary condition, where presence of noise indicated an effect on the slope and intercept equal to that of an added 10 dB “brown noise” from speakers 5 m from observers seen in Pacifici et al. (2008). The probability of a replication having that level of ambient noise was *PrNoise* = 0.15. For the observer effect on the intercept, I calculated the standard deviation (SD = 0.820) among observer coefficients (K. Pacifici, personal communication) for BTBW observations in Pacifici et al. (2008). I assumed that Observer 1 was ½ SD better than average and Observer 2 was ½ SD worse than average (observer effects = ±0.410).

Error in distance estimation by observers was normally distributed, incorporating error estimates reported in Alldredge et al. (2007c). I used the overall mean error reported for trained observers (7.6 m, SD = 21.4 m; Alldredge et al. 2007c) for distances ≥ 62.3 m (the mean for distances investigated). For closer distances (< 62.3 m), I assumed the mean error and variation decreased as a linear function as distance approached 0 m (Figure 1.5).

Results

For one year of simulated counts for the double-observer survey type, the Program unmarked model with detection as a function of observer and percent habitat did not converge for the 150-m survey radius and was removed from the suite of competing models. For one year of simulated counts for the removal survey type, the Program unmarked model with detection as a function of percent habitat did not converge for the 50-m survey radius, and was removed from model selection. No more than 4 estimates (of 30) were removed as outliers per estimator (Table 1.5).

Simple index density estimates (Table 1.5) underestimated true density of birds (D_p) for all survey types and at all survey radii (Figures 1.6, 1.7), although bias for simple index estimates (Table 1.6) for the removal survey type was lower than for other survey types. Simple index estimates with perfect distance estimation showed similar bias to simple index estimates, indicating observer error in estimating distance (i.e., errors in determining if birds were inside or outside the survey radius) did not strongly affect simple index estimator performance.

For the double-observer survey type, no estimator was less biased than the simple index estimator (Table 1.6, Figures 1.6 b, 1.7 b) and the adjusted estimates using the dependent (Nichols et al. 2000) and independent (Alldredge et al. 2006) approaches were nearly unchanged from simple index estimates. Birds that were available during surveys often sang many times (mean = 9.06 songs, SD = 6.37), so even with the difference modeled in observer skill, a bird detected by one observer was typically detected by both observers (i.e., perceptibility was very high).

For the removal survey type, the simple index estimator was less biased than for other survey types (Table 1.6) and the difference between true D_p and true D_a was smaller than for other survey types (Table 1.5). Both results indicate a greater proportion of birds within the survey radius were available over the length of the survey as compared to other survey types, due to the longer survey period (10 minutes; all other survey types had 3-minute surveys). Adjusted density estimates for the removal survey type were less biased than simple index estimates for 50-m and 100-m surveys (Table 1.6, Figure 1.7 c).

For the replicated counts survey type, adjusted density estimates from N-mixture models were highly inflated for approximately half to two-thirds of simulated years (Figures 1.6 d, 1.7 d). These inflated estimates were common enough that they did not meet the criteria to be removed as outliers (i.e., > 3 standard deviations away from the mean of remaining estimates). Inflated density estimates occurred when estimated detection was approximately ≤ 0.06 (Figure 1.8). N-mixture model density estimates for 50-m radius surveys were the most inflated; inflation was somewhat less for surveys with larger radii (Table 1.6, Figure 1.8). Bounded count and maximum count density estimates generally had less bias than the simple index estimates for the replicated counts survey type, although the bounded count density estimator overestimated density for 50-m and 100-m radius surveys (Table 1.6, Figure 1.7 d).

For the distance sampling survey type, adjusted density estimates were slightly less biased than simple index estimates (Table 1.6 b, Figure 1.7 e). In Program Distance, the top model as determined by AIC comparison had a uniform key function with a simple polynomial adjustment (Table 1.7). The bulk of the AIC weight was spread across 9 models, but density estimates were similar across those models (Table 1.7).

Generally, index estimators were significantly positively correlated with true density. For 100-m and 150-m radius surveys, simple index estimators were significantly correlated with D_s, D_p, and D_a, and had the strongest correlation with D_a (Table 1.8). For 50-m radius surveys, simple index estimators for 4 of the 5 survey types were significantly correlated only with D_p and D_a, and had the strongest correlation with D_a. A notable exception was the bounded count estimator, which had lower correlation with true densities than the simple index estimator (Table 1.8, Figure 1.7 d).

Adjusted density estimators had less consistent correlation with true density across survey types than did the simple index estimator (Table 1.8). For the double-observer survey type, adjusted density estimates were as correlated with true density as simple index estimates (estimates were very similar). For the removal survey type, adjusted density estimates were more weakly correlated with true densities than simple index estimates. For the replicated count survey type, N-mixture model estimates were not significantly correlated with any true density (D_s, D_p, or D_a) for 50-m radius surveys and had a significant negative correlation with true density for 100-m and 150-m radius surveys (Table 1.8). This lack of correlation is due to highly inflated N-mixture model density estimates for many simulated years (Figures 1.6 d, 1.7d). For the distance sampling survey type, adjusted density estimates were more weakly correlated with true densities than simple index estimates (Table 1.8d).

Estimates using the smallest survey radius were generally less biased than estimates using larger radii, but for adjusted estimators only, they were also more likely to have inflated estimates. Surveys with larger radii generally had stronger correlation with true densities.

Discussion

Analysis methods used to estimate bird density from simulated BTBW counts varied widely in their performance, and adjustment analysis methods generally did not outperform index analysis methods. Simple index estimates were biased, but they were also highly correlated with true density (D_p), particularly for surveys with larger radii. The simple index estimator would therefore track population changes well, providing valuable information for management. Among survey types, the removal survey type showed the least bias, largely because the additional time spent surveying allowed a greater proportion of the population to be available to be detected. This advantage could be diminished in real-world removal surveys if birds were more likely to be doublecounted due to the longer survey period (double-counting was not included in my model). Compared to simple index estimates, adjusted estimates in the distance sampling and removal survey types showed a reduction in bias. The maximum count density estimator for the replicated counts survey type also showed a reduction in bias compared to the simple index estimator. Adjusted density estimates were less strongly correlated with D_p than were simple index estimates (Table 1.8), with the exception of adjusted estimates for the double-observer survey type, which did not differ. Adjusted density estimates using N-mixture models for the replicated counts survey type were prone to inflated estimates and high positive bias. Unadjusted counts, while biased, were better correlated with true abundance and would provide better information about changes in abundance than an adjustment analysis method for the BTBW scenario, where abundance and detection were independent of one another.

My bird survey counts model is extensive and included variation in detection due to spatial arrangement, availability, and perceptibility of birds. By incorporating all three components and using empirical data to inform model parameters, this model allows a more comprehensive investigation into detection probability than has previously been attempted with simulation. Complexity, however, is a double-edged sword because modeling requires making assumptions about the detection process. I modeled birds spatially as remaining in territories, I modeled song production as an autocorrelated process, and I modeled perceptibility as a logit function of distance to the observer. To the extent that these assumptions are violated, or that the parameters I selected do not represent a particular species of interest, this model will not accurately reflect counts from real bird surveys. Also, any simulation is a simplification. My model does not include double-counting, misidentification, false positives, swamping of observers, or effects of the observer (or other birds) on song production or movement. Also, the BTBW scenario is a situation where an index might be expected to perform well because abundance and detection probability are not confounded. Investigating additional scenarios with a wider range of parameters affecting detection probability would expand the utility of this study's conclusions.

Simulations can provide insight that field studies cannot. Nichols et al. (2009) recommended that analysis methods for estimating abundance be evaluated for situations where assumptions were likely violated, as capture-recapture models were evaluated in the 1970s and 1980s. My model represents one such evaluation. Because birds move throughout my simulated surveys, my model allows violations of the closure assumption (that there is no change in the population of birds within the sample area during a survey) which is assumed across all analysis methods I consider (Nichols et al. 2009). Simulated birds in the BTBW scenario moved often (on average, 78% of birds moved at least once within a 10-minute survey period for the removal survey type and 36% of birds moved at least once within a 3-minute survey period for all other survey types). With a mean (uncompressed) territory size of 3.6 ha (about half the area surveyed with 150-m radius point count), many birds moved into or out of the survey area, violating the closure assumption. The removal survey type was longer (10 minutes) than all other survey types (3 minutes), allowing more birds to become available, but also allowing more birds to move into or out of the survey radius. For surveys with the BTBW parameters, indices for removal surveys were less biased than all other survey types, an indication that the benefits of increased availability can outweigh the increase in violations of the closure assumption for longer surveys.

By including observer error and variation in estimation of distance to birds, my model allows violations of the assumption that birds are correctly recorded as inside or outside the survey radius, commonly assumed across analysis methods (Nichols et al. 2009), and the distance sampling analysis method's assumption that distances to birds are estimated accurately (Thomas et al. 2002). Error in observer-estimated distance had mean 7.6 m for distances > 62.3 m (Alldredge et al. 2007c). The mean error for observations was therefore ≤ 12% on average (i.e., 7.6/62.3), a relatively mild effect. For simple index estimators, there was little improvement in bias or correlation with D_p when observer error was omitted (i.e., distance estimation was perfect). At the effect size estimated by Alldredge et al. (2007c), observer error in estimation of distance is unlikely to substantially bias results.

For distance sampling and double-observer survey types, the Program unmarked adjustment estimators were relatively unbiased estimators of D_a (Figure 6b, 6e). The value of estimating D_a, however, may vary situationally. Having good inferences about the density of singing birds is only useful if that density can be related to the total density of birds (i.e, to have an estimate of p_a or to be able to make an assumption about p_a). Availability is highly variable across time of day, day within season, and mating status (Wilson and Bart 1985, McShea and Rappole 1997, Rosenberg and Blancher 2005), so p_a would best be estimated simultaneously with abundance (e.g., Diefenbach et al. 2007). Doing so may be time-consuming and expensive. Estimators that are better correlated with D_p, such as removal adjustment estimators (Figure 6c), provide clearer inference about the total density of birds.

I found no benefit to using the double-observer adjustment method in the BTBW scenario because perceptibility was ~1 or ~0 for most birds (Figure 1.4); only birds within a narrow range of radii had perceptibility such that discrepancies in observation were probable. Adjustment methods that are ill-suited to model the detection component responsible for imperfect detection will not improve inference and are no more useful than an index. The double-observer adjusted estimates had similar bias to the simple index estimates, yet an unaware practitioner might claim that the adjusted estimates were unbiased because he or she used an adjustment method and detection probability was “accounted for”. I recommend against assuming that any adjustment method “accounts for” detection probability, unless there is evidence or reasoning that the adjustment method correctly addressed the source(s) of imperfect detection in the system.

When selecting an analysis method to estimate abundance, a crucial first step is for practitioners to use their knowledge of the system to consider detection and to consider their objectives. If detection components can be reasonably assumed to be constant and relative abundance is sufficient to meet survey objectives, using an index method is preferable. If detection is low but constant and absolute abundance is of interest, a simple correction factor may be used. For any detection components that are unknown or known to be variable, I recommend that practitioners perform a pilot study to estimate the mean value and variability of each component. If a detection component is highly variable, the best adjustment method will include that detection component in its detection estimator. However, adjustment estimators risk over-correcting counts (such as the N-mixture models here) and should be used cautiously if estimated detection is low. An alternative method is to follow Skalski and Robson's (1992) recommendation to collect information to estimate detection and then to select among competing models that do and do not include adjustment. The downside of this suggestion is that logistical costs of collecting such data will be spent even if an index method is eventually selected. The difficulty of recording the distance to all birds detected, the survey interval in which they are first detected, and reconciling observations between observers may not be trivial, especially in multi-species surveys or surveys with many individuals. Still, if the data can be collected accurately without hindering the counting accuracy of observers, those costs may be worthwhile if detection in a system is not well understood.

Whenever possible, I recommend standardizing surveys to reduce variation in detection components. Using longer survey times in my model (as seen in the removal survey type) increased availability and increased the correlation of the simple index estimator with D_p. I recommend performing surveys during times of high availability (e.g., morning surveys during the height of breeding season singing) or using methods to increase availability such as callback surveys for secretive marsh birds (Conway and Gibbs 2005) to increase the correlation of counts with D_p. Callback methods, however, can cause birds to move towards the observer (e.g., Johnson et al. 2014) and should be carefully investigated before use. To the extent possible, I also recommend training and testing observers to increase p_d and reduce its variability (Kepler and Scott 1981).

To varying degrees, adjusted estimators for the replicated counts, removal, and distance sampling survey types were susceptible to inflation due to low estimates of detection (Figure 1.8), especially for surveys with the smallest radius. If using adjustment methods, I recommend removing potential outlier estimates, particularly if an unusually high abundance is estimated for a year with low estimated detection. For the removal and distance sampling survey types, this inflation happened rarely and removing outliers adequately corrected estimates. Inflation was the norm, however, for density estimates from N-mixture models for the replicated counts survey type, a problem also explored by

Dennis et al. (2015). Because the replicated counts survey type required repeated visits, the N-mixture model density estimator had a small sample size (20 survey sites) and frequently estimated very low *p* (Figure 8), resulting in inflated estimates of density. I suggest that future simulations should examine if this adjustment method performs better with a larger sample size, but it is noteworthy that no other analysis method suffered this drawback.

Table 1.1: Definitions for the transition matrix (P_{coarse}) used to modeling autocorrelation for coarse-scale bird availability. P(S|S) = 0.98 and P(NS|S) = 0.02 for all scenarios. P(NS|NS) and P(S|NS) varied to produce $Z_{NIntervals}$ (eq. 1.7) that equaled the desired interval-specific probability of a bird being in singing mode for at least 1 interval within 10 minutes (*PrSing.yijrk*).

		State at interval *k*	
		Bird *j* is in singing mode in interval *k*.	Bird *j* is not in singing mode in interval *k*.
State at interval *k*-1	Bird *j* was in singing mode in interval *k*-1.	P(S|S): Probability that bird *j* is in singing mode in interval *k*, given that it was in singing mode in interval *k*-1.	P(NS|S): Probability that bird *j* is not in singing mode in interval *k*, given that it was in singing mode in interval *k*-1.
	Bird *j* was not in singing mode in interval *k*-1.	P(S|NS): Probability that bird *j* is in singing mode in interval *k*, given that it was not in singing mode in interval *k*-1.	P(NS|NS): Probability that bird *j* is not in singing mode in interval *k*, given that was not in singing mode in interval *k*-1.

Table 1.2: Transition matrix (*TransitionMatrix.fine*) values used to model fine-scale autocorrelation for the black-throated blue warbler (BTBW) scenario. The pattern of fine-scale singing was produced using 1 state in which *Sings.yijrk* = 1 (S) and 3 states in which *Sings.yijrk* = 0 (NS1, NS2, NS3). Values represent the probability of transitioning to the given state at interval *k*, given the previous state at interval *k* – 1.

		State at Interval *k*			
		S	NS1	NS2	NS3
State at Interval *k*-1	S	0	1	0	0
	NS1	0	0	1	0
	NS2	0.08	0	0	0.92
	NS3	0.80	0	0	0.20

Table 1.3: Parameter values for survey types in the black-throated blue warbler (BTBW) scenario. Survey time (sites × survey length) and logistical time (20 minutes travel to each site per observer) were used to estimate human effort needed to accomplish surveys.

Survey Type	Sites	Replications	Survey Length (min)	Simultaneous Observers	Survey Time (min)	Logistical Time (min)	Total (min)
Simple counts	60	1	3	1	180	1200	1380
Double-observer	30	1	3	2	180	1200	1380
Removal	46	1	10	1	460	920	1380
Replicated counts	20	3	3	1	180	1200	1380
Distance sampling	60	1	3	1	180	1200	1380

Table 1.4: Perceptibility coefficients for the black-throated blue warbler (BTBW) scenario, based on values from Pacifici et al. (2008). Observer-specific perceptibility (*perceptibility.yijrko*) was modeled with equation 1.8.

Intercept	13.3
Slope	-0.109
Observer 1 Intercept Effect	0.410
Observer 2 Intercept Effect	-0.410
Noise Slope Effect	-0.023
Noise Intercept Effect	-0.849

Table 1.5 (a): Density estimates and true densities (birds/ha) for surveys with 50-m radius for the black-throated blue warbler (BTBW) scenario. Density was calculated for area A around the observer, where $A = \pi \times$ survey radius^2. True densities are D_s, the density of birds with territories that overlapped the survey area ($D_s = N^* p_s / A$); D_p, the density of birds present at the beginning of the survey ($D_p = N^* p_s p_p / A$); and D_a, the density of available birds present at the beginning of the survey ($D_a = N^* p_s p_p p_a / A$). Mean and standard deviation (SD) for density estimates and true density were calculated across 30 simulated years.

		Estimate			True Density					
					D_s		D_p		D_a	
Survey Type	Estimator	Mean	SD	Outliers Removed	Mean	SD	Mean	SD	Mean	SD
Simple Counts	Simple Index Density	0.174	0.057	0	1.68	0.232	0.392	0.080	0.163	0.049
	Simple Index Density with Perfect Distance	0.172	0.050	0						
Double-Observer	Simple Index Density	0.279	0.122	0	1.71	0.309	0.446	0.127	0.192	0.090
	Simple Index Density with Perfect Distance	0.199	0.083	0						
	Model-Averaged Adjusted Density	0.298	0.120	0						
	Top Model Adjusted Density	0.304	0.130	0						
	Nichols et al. (2000) Density	0.294	0.111	2						
Removal	Simple Index Density	0.321	0.099	0	1.81	0.263	0.402	0.116	0.312	0.088
	Simple Index Density with Perfect Distance	0.328	0.078	0						
	Model-Averaged Adjusted Density	0.414	0.139	3						
	Top Model Adjusted Density	0.441	0.170	2						

Table 1.5 (a) continued.

Survey Type	Estimator	Estimate Mean	Estimate SD	Outliers Removed	True Density D_s Mean	D_s SD	D_p Mean	D_p SD	D_a Mean	D_a SD
Replicated Counts	Simple Index Density	0.183	0.060	0	1.55	0.290	0.458	0.109	0.191	0.062
	Simple Index Density with Perfect Distance	0.199	0.062	0						
	Model-Averaged Adjusted Density	44.2	35.8	0						
	Top Model Adjusted Density	46.6	40.1	0						
	Maximum Count Density	0.465	0.141	0						
	Bounded Count Density	0.849	0.267	0						

Table 1.5 (b): Density estimates and true densities (birds/ha) for surveys with 100-m radius for the black-throated blue warbler (BTBW) scenario. Density was calculated for area A around the observer, where $A = \pi \times$ survey radius^2. True densities are D_s, the density of birds with territories that overlapped the survey area ($D_s = N^* p_s / A$); D_p, the density of birds present at the beginning of the survey ($D_p = N^* p_s p_p / A$); and D_a, the density of available birds present at the beginning of the survey ($D_a = N^* p_s p_p p_a / A$). Mean and standard deviation for density estimates and true density were calculated across 30 simulated years.

		Estimate			True Density					
					D_s		D_p		D_a	
Survey Type	Estimator	Mean	SD	Outliers Removed	Mean	SD	Mean	SD	Mean	SD
Simple Counts	Simple Index Density	0.155	0.026	0	1.39	0.130	0.410	0.055	0.176	0.028
	Simple Index Density with Perfect Distance	0.182	0.027	0						
Double-Observer	Simple Index Density	0.222	0.046	0	1.43	0.152	0.437	0.067	0.185	0.042
	Simple Index Density with Perfect Distance	0.199	0.039	0						
	Model-Averaged Adjusted Density	0.224	0.047	0						
	Top Model Adjusted Density	0.226	0.048	0						
	Nichols et al. (2000) Density	0.222	0.046	0						
Removal	Simple Index Density	0.314	0.041	0	1.45	0.135	0.427	0.057	0.325	0.043
	Simple Index Density with Perfect Distance	0.360	0.045	0						
	Model-Averaged Adjusted Density	0.385	0.069	0						
	Top Model Adjusted Density	0.385	0.069	0						

Table 1.5 (b) continued.

		Estimate			True Density					
					D_s		D_p		D_a	
Survey Type	Estimator	Mean	SD	Outliers Removed	Mean	SD	Mean	SD	Mean	SD
Replicated Counts	Simple Index Density	0.167	0.037	0	1.41	0.154	0.421	0.070	0.175	0.038
	Simple Index Density with Perfect Distance	0.180	0.038	0						
	Model-Averaged Adjusted Density	10.8	9.37	0						
	Top Model Adjusted Density	11.6	10.2	0						
	Maximum Count Density	0.333	0.053	0						
	Bounded Count Density	0.533	0.083	0						

Table 1.5 (c): Density estimates and true densities (birds/ha) for surveys with 150-m radius for the black-throated blue warbler (BTBW) scenario. Density was calculated for area A around the observer, where $A = \pi \times$ survey radius^2. True densities are D_s, the density of birds with territories that overlapped the survey area ($D_s = N^* p_s / A$), D_p, the density of birds present at the beginning of the survey ($D_p = N^* p_s p_p / A$), and D_a, the density of available birds present at the beginning of the survey ($D_a = N^* p_s p_p p_a / A$). Mean and standard deviation for density estimates and true density were calculated across 30 simulated years.

		Estimate			True Density					
					D_s		D_p		D_a	
Survey Type	Estimator	Mean	SD	Outliers Removed	Mean	SD	Mean	SD	Mean	SD
Simple Counts	Simple Index Density	0.138	0.020	0	1.03	0.099	0.413	0.049	0.175	0.025
	Simple Index Density with Perfect Distance	0.154	0.022	0						
Double-Observer	Simple Index Density	0.161	0.026	1	1.08	0.106	0.432	0.048	0.188	0.032
	Simple Index Density with Perfect Distance	0.169	0.029	1						
	Model-Averaged Adjusted Density	0.162	0.026	1						
	Top Model Adjusted Density	0.162	0.026	1						
	Nichols et al. (2000) Density	0.161	0.026	1						
Removal	Simple Index Density	0.285	0.033	0	1.10	0.106	0.435	0.053	0.330	0.039
	Simple Index Density with Perfect Distance	0.318	0.040	0						
	Model-Averaged Adjusted Density	0.355	0.058	1						
	Top Model Adjusted Density	0.354	0.055	1						

Table 1.5 (c) continued.

		Estimate			True Density					
					D_s		D_p		D_a	
Survey Type	Estimator	Mean	SD	Outliers Removed	Mean	SD	Mean	SD	Mean	SD
Replicated Counts	Simple Index Density	0.140	0.023	0	1.05	0.117	0.418	0.050	0.176	0.025
	Simple Index Density with Perfect Distance	0.153	0.023	0						
	Model-Averaged Adjusted Density	4.65	4.00	0						
	Top Model Adjusted Density	4.86	4.28	0						
	Maximum Count Density	0.243	0.028	0						
	Bounded Count Density	0.355	0.038	0						

Table 1.5 (d): Density estimates and true densities (birds/ha) for surveys with unlimited radius for the black-throated blue warbler (BTBW) scenario (distance sampling survey type only). Data were right-truncated, eliminating the most distant 10% of observations (Buckland et al. 2001). For index estimates and true densities, the area surveyed was considered to be $A = \pi \times$ survey radius^2, where the survey radius was the truncation distance. True densities are D_s, the density of birds with territories that overlapped the survey area ($D_s = N^* p_s / A$); D_p, the density of birds present at the beginning of the survey ($D_p = N^* p_s p_p / A$); and D_a, the density of available birds present at the beginning of the survey ($D_a = N^* p_s p_p p_a / A$). Mean and standard deviation for density estimates and true density were calculated across 30 simulated years.

		Estimate			True Density					
					D_s		D_p		D_a	
Survey Type	Estimator	Mean	SD	Outliers Removed	Mean	SD	Mean	SD	Mean	SD
Distance Sampling	Simple Index Density	0.144	0.022	0	1.05	0.113	0.431	0.053	0.189	0.026
	Simple Index Density with Perfect Distance	0.153	0.023	0						
	Model-Averaged Adjusted Density	0.213	0.051	4						
	Top Model Adjusted Density	0.217	0.050	2						
	Program Distance Top Model Density	0.260	0.027	0						
	Program Distance Model-Averaged Density	0.237	0.021	0						

Table 1.6 (a): Bias for density estimates for the black-throated blue warbler (BTBW) scenario, for non-distance survey types. Bias was calculated as the median difference between the density estimate and the density of birds present at the beginning of the survey, D_p. $D_p = N^* p_s p_p / A$, where $A = \pi \times \text{survey radius}^2$.

Survey Type	Estimator	Bias					
		50-m radius		100-m radius		150-m radius	
Simple Counts	Simple Index Density	-0.223	-59%	-0.249	-61%	-0.263	-67%
	Simple Index Density with Perfect Distance Estimation	-0.223	-55%	-0.223	-56%	-0.251	-63%
Double-Observer	Simple Index Density	-0.170	-37%	-0.202	-48%	-0.269	-62%
	Simple Index Density with Perfect Distance Estimation	-0.255	-59%	-0.233	-55%	-0.259	-61%
	Model-Averaged Adjusted Density	-0.144	-32%	-0.201	-48%	-0.267	-62%
	Top Model Adjusted Density	-0.127	-30%	-0.202	-48%	-0.267	-62%
	Nichols et al. (2000) Density	-0.170	-37%	-0.202	-48%	-0.267	-62%
Removal	Simple Index Density	-0.083	-23%	-0.111	-26%	-0.146	-34%
	Simple Index Density with Perfect Distance Estimation	-0.069	-16%	-0.066	-15%	-0.114	-26%
	Model-Averaged Adjusted Density	-0.011	-2%	-0.048	-12%	-0.098	-23%
	Top Model Adjusted Density	-0.007	-2%	-0.048	-11%	-0.097	-23%
Replicated Counts	Simple Index Density	-0.265	-60%	-0.255	-61%	-0.281	-67%
	Simple Index Density with Perfect Distance Estimation	-0.255	-56%	-0.244	-57%	-0.270	-63%
	Model-Averaged Adjusted Density	58.6	13113%	11.1	2295%	2.24	520%
	Top Model Adjusted Density	80.1	14553%	12.4	2527%	2.08	529%
	Maximum Count Density	-0.021	-4%	-0.088	-22%	-0.170	-41%
	Bounded Count Density	0.350	70%	0.106	26%	-0.050	-12%

Table 1.6 (b): Bias for density estimates for the black-throated blue warbler (BTBW) scenario, for surveys with unlimited radius (distance sampling survey type only). Bias was calculated as the median difference between the density estimate and the density of birds present at the beginning of the survey, D_p. $D_p = N^* p_s p_p / A$, where $A = \pi \times$ survey radius^2. Data were right-truncated, eliminating the most distant 10% of observations (Buckland et al. 2001). For index estimates and true densities, the survey radius was considered to be the truncation distance.

Survey Type	Estimator	Bias (Radius determined by 10% truncation)	
Distance Sampling	Simple Index Density	-0.293	-67%
	Simple Index Density with Perfect Distance Estimation	-0.283	-64%
	Model-Averaged Adjusted Density	-0.226	-53%
	Top Model Adjusted Density	-0.222	-53%
	Program Distance Top Model Density	-0.176	-39%
	Program Distance Model-Averaged Density	-0.205	-45%

Table 1.7: Models compared in Program Distance using AIC (Burnham and Anderson 2002) for the black-throated blue warbler (BTBW) scenario. Data were right-truncated, eliminating the most distant 10% of observations (Buckland et al. 2001).

Key Function	Adjustment	Covariate	Parameters	Δ AIC	AIC Weight	Effective Detection Radius (m)	Estimated Density (birds/ha)	Density 95% CI Lower	Density 95% CI Upper
Uniform	Polynomial	none	2	0	0.211	138	0.260	0.234	0.288
Hazard	Hermite	none	3	0.490	0.165	137	0.261	0.236	0.289
Hazard	Polynomial	none	3	0.891	0.135	140	0.253	0.236	0.271
Hazard	Cosine	none	2	0.980	0.129	141	0.247	0.234	0.260
Hazard	NA	Percent Habitat	3	1.01	0.127	142	0.246	0.237	0.255
Half-Normal	Polynomial	none	2	1.01	0.127	136	0.267	0.244	0.293
Uniform	Hermite	none	3	2.46	0.062	137	0.261	0.235	0.291
Half-Normal	Cosine	none	5	4.51	0.022	124	0.322	0.236	0.439
Uniform	Cosine	none	5	4.66	0.021	124	0.320	0.238	0.431
Half-Normal	Hermite	none	2	15.1	>0.01	134	0.277	0.241	0.318
Half-Normal	NA	Percent Habitat	2	30.6	>0.01	127	0.306	0.294	0.319

Table 1.8 (a): Pearson correlation coefficients (ρ) for correlation of density estimates with true density for surveys with 50-m radius for the black-throated blue warbler (BTBW) scenario. Density was calculated for area A around the observer, where $A = \pi \times$ survey radius^2. Each estimator was compared to D_s, the density of birds with territories that overlapped the survey radius ($D_s = N^* p_s / A$); D_p, the density of birds present at the beginning of the survey ($D_p = N^* p_s p_p / A$); and D_a, the density of available birds present at the beginning of the survey ($D_a = N^* p_s p_p p_a / A$). Significant correlation coefficients (according to a *t*-distribution with *NYears*-2 degrees of freedom) are noted: * $P<0.05$, **$P<0.01$, ***$P<0.001$.

Survey Type	Estimator	ρ D_s		ρ D_p		ρ D_a	
Simple Counts	Simple Index Density	0.143		0.337		0.689	***
	Simple Index Density: Perfect Distance	0.278		0.565	**	0.869	***
Double-Observer	Simple Index Density	0.223		0.530	**	0.693	***
	Simple Index Density: Perfect Distance	-0.021		0.580	***	0.894	***
	Model-Averaged Adjusted Density	0.098		0.565	**	0.700	***
	Top Model Adjusted Density	0.017		0.550	**	0.660	***
	Nichols et al. (2000) Density	0.236		0.426	*	0.651	***
Removal	Simple Index Density	0.378	*	0.635	***	0.615	***
	Simple Index Density: Perfect Distance	0.096		0.709	***	0.740	***
	Model-Averaged Adjusted Density	0.278		0.505	**	0.582	**
	Top Model Adjusted Density	0.307		0.343		0.399	*

Table 1.8 (a) continued.

Survey Type	Estimator	ρ D_s	ρ D_p		ρ D_a	
Replicated Counts	Simple Index Density	0.211	0.609	***	0.761	***
	Simple Index Density: Perfect Distance	0.270	0.690	***	0.869	***
	Model-Averaged Adjusted Density	-0.007	-0.204		-0.146	
	Top Model Adjusted Density	-0.075	-0.301		-0.280	
	Maximum Count Density	0.265	0.561	**	0.700	***
	Bounded Count Density	0.282	0.491	**	0.610	***

Table 1.8 (b): Pearson correlation coefficients (ρ) for correlation of density estimates with true density for surveys with 100-m radius for the black-throated blue warbler (BTBW) scenario. Density was calculated for area A around the observer, where $A = \pi \times$ survey radius^2. Each estimator was compared to D_s, the density of birds with territories that overlapped the survey radius ($D_s = N^* p_s / A$); D_p, the density of birds present at the beginning of the survey ($D_p = N^* p_s p_p / A$); and D_a, the density of available birds present at the beginning of the survey ($D_a = N^* p_s p_p p_a / A$). Significant correlation coefficients (according to a t-distribution with *NYears*-2 degrees of freedom) are noted: * $P<0.05$, **$P<0.01$, ***$P<0.001$.

Survey Type	Estimator	ρ D_s		ρ D_p		ρ D_a	
Simple Counts	Simple Index Density	0.534	**	0.648	***	0.816	***
	Simple Index Density: Perfect Distance	0.623	***	0.742	***	0.928	***
Double-Observer	Simple Index Density	0.515	**	0.388	*	0.762	***
	Simple Index Density: Perfect Distance	0.543	**	0.641	***	0.928	***
	Model-Averaged Adjusted Density	0.460	*	0.347		0.745	***
	Top Model Adjusted Density	0.411	*	0.299		0.710	***
	Nichols et al. (2000) Density	0.515	**	0.388	*	0.762	***
Removal	Simple Index Density	0.586	***	0.583	***	0.583	***
	Simple Index Density: Perfect Distance	0.647	***	0.764	***	0.802	***
	Model-Averaged Adjusted Density	0.426	*	0.198		0.090	
	Top Model Adjusted Density	0.442	*	0.200		0.086	

Table 1.8 (b) continued.

Survey Type	Estimator	ρ D_s		ρ D_p		ρ D_a	
Replicated Counts	Simple Index Density	0.696	***	0.643	***	0.833	***
	Simple Index Density: Perfect Distance	0.691	***	0.772	***	0.925	***
	Model-Averaged Adjusted Density	-0.337		-0.358		-0.509	**
	Top Model Adjusted Density	-0.335		-0.365	*	-0.503	**
	Maximum Count Density	0.656	***	0.583	***	0.733	***
	Bounded Count Density	0.427	*	0.345		0.432	*

Table 1.8 (c): Pearson correlation coefficients (ρ) for correlation of density estimates with true density for surveys with 150-m radius for the black-throated blue warbler (BTBW) scenario. Density was calculated for area A around the observer, where $A = \pi \times$ survey radius^2. Each estimator was compared to D_s, the density of birds with territories that overlapped the survey radius ($D_s = N^* p_s / A$); D_p, the density of birds present at the beginning of the survey ($D_p = N^* p_s p_p / A$); and D_a, the density of available birds present at the beginning of the survey ($D_a = N^* p_s p_p p_a / A$). Significant correlation coefficients (according to a *t*-distribution with *NYears*-2 degrees of freedom) are noted: * $P<0.05$, **$P<0.01$, ***$P<0.001$.

Survey Type	Estimator	ρ D_s		ρ D_p		ρ D_a	
Simple Counts	Simple Index Density	0.616	***	0.660	***	0.817	***
	Simple Index Density: Perfect Distance	0.702	***	0.690	***	0.877	***
Double-Observer	Simple Index Density	0.607	***	0.573	**	0.853	***
	Simple Index Density: Perfect Distance	0.574	**	0.587	***	0.918	***
	Model-Averaged Adjusted Density	0.605	***	0.571	**	0.851	***
	Top Model Adjusted Density	0.605	***	0.571	***	0.851	***
	Nichols et al. (2000) Density	0.601	***	0.567	**	0.849	***
Removal	Simple Index Density	0.816	***	0.809	***	0.851	***
	Simple Index Density: Perfect Distance	0.788	***	0.822	***	0.846	***
	Model-Averaged Adjusted Density	0.527	**	0.519	**	0.553	**
	Top Model Adjusted Density	0.568	***	0.553	**	0.579	***

Table 1.8 (c) continued.

Survey Type	Estimator	$\rho\ D_s$		$\rho\ D_p$		$\rho\ D_a$	
Replicated Counts	Simple Index Density	0.728	***	0.764	***	0.873	***
	Simple Index Density: Perfect Distance	0.749	***	0.787	***	0.876	***
	Model-Averaged Adjusted Density	-0.360		-0.386	*	-0.430	*
	Top Model Adjusted Density	-0.369	*	-0.379	*	-0.406	*
	Maximum Count Density	0.647	***	0.697	***	0.754	***
	Bounded Count Density	0.405	*	0.445	*	0.423	*

Table 1.8 (d): Pearson correlation coefficients (ρ) for correlation of density estimates with true density for surveys with unlimited radius for the black-throated blue warbler (BTBW) scenario (distance sampling survey type only). Data were right-truncated, eliminating the most distant 10% of observations (Buckland et al. 2001). For estimates and true densities, the area surveyed was considered to be $A = \pi \times$ survey radius^2, where the survey radius was the year-specific truncation distance (mean 158 m). True densities are D_s, the density of birds with territories that overlapped the survey area ($D_s = N^* p_s / A$); D_p, the density of birds present at the beginning of the survey ($D_p = N^* p_s p_p / A$); and D_a, the density of available birds present at the beginning of the survey ($D_a = N^* p_s p_p p_a / A$). Significant correlation coefficients (according to a *t*-distribution with *NYears*-2 degrees of freedom) are noted: * $P<0.05$, **$P<0.01$, ***$P<0.001$.

Survey Type	Estimator	Mean Radius	ρ D_s		ρ D_p		ρ D_a	
Distance Sampling	Simple Index Density	158	0.743	***	0.714	***	0.839	***
	Simple Index Density: Perfect Distance	158	0.777	***	0.765	***	0.866	***
	Model-Averaged Adjusted Density	158	0.384		0.555	**	0.665	***
	Top Model Adjusted Density	158	0.296		0.506	**	0.592	***
	Program Distance Top Model Density	158	0.492	**	0.593	***	0.717	***
	Program Distance Model-Averaged Density	158	0.497	**	0.587	***	0.709	***

Chapter II

Performance of Bird Density Estimators for Simulated Counts with Differential Availability and Perceptibility

Summary: Point count surveys of birds are commonly analyzed to make inferences about bird abundance or density. Counts are considered the product of true abundance (*N*) and *p*, the detection probability, and can be used as an index to abundance if *p* is constant. Adjustment analysis methods, such as double-observer, removal, replicated counts, and distance sampling methods, attempt to estimate both *N* and *p* to provide better inference. These adjustment methods have not been thoroughly investigated in a simulated setting, and it is uncertain how they perform in comparison to one another. In Chapter 1, I described a model to simulate bird survey counts, which included variation in spatial arrangement, availability (the probability that a bird is available (vocalizes) during a survey, given that it is present) and perceptibility (the probability that a bird is detected, given that it is available and present). Here, I used the model to simulate counts for scenarios with high or low availability and high or low perceptibility. I also included scenarios where abundance was confounded with perceptibility, and scenarios where they were independent. I then analyzed count data using index methods and adjustment methods. Although index methods were biased and only had a strong correlation with true density when detectability was high, adjustment methods generally did not offer an improvement. As compared to index methods, adjustment method performance ranged from far worse (replicated counts), to no added value (double-observer) to moderate improvement (in bias only, for removal and distance sampling in specific scenarios). Practitioners should carefully consider the sources of variation in detection probability in their system. If detection components are unknown or known to be variable, I advise practitioners to perform a pilot study to estimate detection components. Additionally, practitioners should standardize their methods to increase availability and perceptibility in their surveys and to lower the variation in these detection components.

Point count surveys of birds, consisting of the collection of count data, are the most common data source used to make inferences about bird occurrence, distribution, and population status and trends (Rosenstock et al. 2002). Recall from equation 1.1 that any count (C) obtained from a survey can be considered the product of N, the number of birds present, and p, the detection probability. Nichols et al. (2009) decomposed that detection probability into 4 parts: p_s, the probability that a bird's territory at least partially overlaps the surveyed area of a survey site; p_p, the probability that a bird is present in the surveyed area at the time of the survey; p_a or availability, the probability that a bird is available (vocalizes) during a survey, given that it is present; and p_d or perceptibility, the probability that a bird is detected, given that it is present in the surveyed area and available during the survey (equation 1.2).

Analysis methods used to analyze bird count data can be roughly categorized into two groups: index methods, which use counts as an index to abundance, and adjustment methods, which attempt to estimate both abundance and detectability (Rosenstock et al. 2002, Johnson 2008, Nichols et al. 2009). Many adjustment methods have been proposed, including double-observer methods (Nichols et al. 2000), distance sampling methods (Burnham et al. 1980, Buckland et al. 1993), removal methods (Farnsworth et al. 2002, Alldredge et al. 2007a), and replicated counts methods (Royle 2004). By parsing out detection probability, these methods attempt to directly estimate abundance, rather than relying on counts as an index (Rosenstock et al. 2002, Nichols et al. 2009).

Despite the wealth of analysis methods available, there is a lack of clarity regarding the relative performance of these methods, particularly the best method to use in specific field situations (Johnson 2008, Nichols et al. 2009). Chapter 1 describes the simulation I created to simulate bird survey counts, with the goal of comparing estimators among analysis methods. I define an estimator as a statistic that is used to infer the value of an unknown population parameter. For each scenario (defined as a set of parameters designed to reflect a particular biological situation), the simulation generated counts by modeling variability in N, p_s, p_p, p_a, and p_d as follows. Abundance modeling (N) included variation among survey sites due to an underlying habitat covariate. Spatial modeling (p_s and p_p) included variation in bird locations by using a bivariate normal distribution to simulate birds on territories. Availability modeling (p_a) included variation in song occurrence by using Markov chains to create autocorrelation and varying the probability of singing due to time of day and time within season. Perceptibility modeling (p_d) included variation in observation by using a logit-link function to describe the probability of perception as a function of distance from the observer and varying the shape of the logit function due to observer skill and presence of ambient noise. In Chapter 1, I selected scenario parameters to represent surveys of black-throated blue warblers (*Setophaga caerulescens*): a suitable test case, but limited in its scope.

To expand the range of inference possible regarding the performance of abundance estimators, I here apply the model to additional scenarios. To provide the most useful inference, I created scenarios that reflected the variation in detection probability likely to occur across species and habitats where point count surveys are commonly used. Availability is known to vary widely among species (Emlen 1977, Mayfield 1981), but also within species due to time of day (Hochachka et al. 2009) or reproductive status (Stacier et al. 2006, Robbins et al. 2009). Perceptibility can vary with species (Wolf et al. 1995, Schieck 1997, Alldredge et al. 2007b, Pacifici et al. 2008, also see Chapter 3), habitat conditions (Schieck 1997, Pacifici et al. 2008), and observer (Alldredge et al. 2007b, Pacifici et al. 2008, also see Chapter 3).

Using counts as indices to abundance is particularly suspect when detection is confounded with abundance. For example, Ruiz-Gutiérrez et al. (2010) found differences in detectability across habitat types and species' level of forest-dependence, concluding that their occupancy and colonization results would have been biased if they had used index methods. This confounding effect would be apparent in a species that favored dense vegetation where perceptibility was low. Using an index method to estimate abundance for such a species might not show increased abundance at sites with dense vegetation when compared to sites with sparser vegetation and higher detection. To assess the effect of such a confounding on estimator performance, I included scenarios where both perceptibility and site-specific abundance were a function of habitat, an underlying covariate.

My objective was to evaluate the effect of variation in availability, perceptibility, and the effect of habitat on perceptibility on the performance of density estimators for both index methods and adjustment methods. I applied the model described in Chapter 1 to scenarios with parameters reflecting the variation in availability, perceptibility, and effect of habitat on perceptibility that can be commonly expected across the wide range of species and habitats where point count bird surveys are conducted. I then evaluated the performance of density estimators, using simulated counts generated by that model.

Methods

I simulated bird survey counts (Chapter 1) to investigate the effect of availability, perceptibility, and a confounded effect of perceptibility with abundance on the performance of several commonly used density estimators. I used 3 binary options for scenario parameters: high vs. low availability, high vs. low perceptibility, and an effect of habitat on perceptibility vs. no effect. The result was 8 scenarios including all possible binary combinations (Table 2.1). All scenarios used the same spatial and abundance parameters (below).

Within each scenario, I conducted separate simulations for each survey type, defined as a specific survey scheme of temporal and spatial replication of survey sites and number of observers (see Chapter 1, Table 1.3). I considered 5 survey types: simple counts, double-observer, distance sampling, removal, and replicated counts (Table 1.3). Adjustment estimators are survey type-specific (e.g., double-observer adjustment estimators can only be applied to data from the double-observer survey type). Simple index estimators (counts, or counts converted to density) can be applied to any survey type.

Abundance

For each site, where a site was 400 ha modeled in a Cartesian grid with the observer at the center, I modeled site-specific abundance (*NBirds.yi*) as a Poisson-distributed random variable, described by parameter *LambdaHabitat.yi*. *LambdaHabitat.yi* was a product of the size of the modeled area around the observer (*Area.yi* = 400 ha), the density of birds in habitat (*HabitatDensity*), and a site-specific proportion of habitat (*PercentHabitat.yi*). Thus,

$$NBirds.yi \sim \text{Poisson}(HabitatDensity.y \times Area.yi \times PercentHabitat.yi) \quad \text{(eq. 2.1)}.$$

Within each survey type, *PercentHabitat.yi* was randomly selected for year 1, with half of sites having a high proportion of habitat (*PercentHabitat.yi* ~ N(0.8, 0.09)) and half of sites having a low proportion of habitat (*PercentHabitat.yi* ~ N(0.4, 0.09)). To reduce nuisance variation, *PercentHabitat.yi* in year 1 was the same across scenarios within survey types, as if the same sites were being surveyed under different biological situations. Variation in *PercentHabitat.yi* was normally-distributed [N(0,0.1)] and added annually, such that 95% of annual changes in *PercentHabitat.yi* would be $\leq$ 2%.

Across species and habitats, bird density and territory size are highly variable. I examined studies of forest bird density and territory size in variable habitats to guide my selection of parameters (Table 2.2). I assigned site abundance by assuming that all birds were found in habitat (as opposed to matrix). *HabitatDensity* was 1 bird/ha in year 1 and declined linearly to 0.7 birds/ha in year 30. A high *PercentHabitat.yi* site with *PercentHabitat.yi* = 0.8 (the mean for high *PercentHabitat.yi* sites) would have an abundance drawn from a Poisson distribution with $\lambda = 0.8 \times 400$ ha × 1 bird/ha = 320 birds in year 1. A similar site in year 30 would have an abundance drawn from a Poisson distribution with $\lambda = 0.8 \times 400$ ha × 0.7 bird/ha = 224 birds. A low *PercentHabitat.yi* site with *PercentHabitat.yi* = 0.4 (the mean for low *PercentHabitat.yi* sites) would have an abundance drawn from a Poisson distribution with $\lambda = 0.4 \times 400$ ha × 1 bird/ha = 160 birds in year 1 or $\lambda = 0.4 \times 400$ ha × 0.7 bird/ha = 112 birds in year 30.

Spatial

Movements for each simulated bird were within a bivariate normally distributed territory (Chapter 1), with a constant probability of movement within the territory = 0.005 between interval *k* and interval *k*+1. Interval length was 2 seconds.

Individual bivariate normal territories were prevented from overlapping in the model by eliminating a territory if any of the 4 axes or center of the territory overlapped the utilization ellipse of any existing territory (or vice versa). Such overlap comparisons were conducted on 60% utilization ellipses for the bivariate normal territories (56.5% utilization ellipses were used for Chapter 1, based on territory and density information available for black-throated blue warblers). For high density (~1 bird/ha) sites, I used 0.5 ha as the maximum allowable size for 60% utilization ellipses. This limitation prevented the modeled space from being more than 50% filled with territories (the point at which computation becomes unreasonably slow, as determined by trial and error, see Appendix C). From Jennrich and Turner (1969, equation 12), a 0.5-ha 60% utilization ellipse corresponds to a 95% utilization ellipse with area 1.64 ha. Using the ovenbird (*Seiurus aurocapilla*) as an example (Smith and Shugart 1987), I assumed territories could be compressed 4-fold. Thus, 95% utilization ellipses for uncompressed territories (at low-density sites) had a maximum size of 1.64 ha × 4 = 6.56 ha. Territory size was log-normally distributed, with mean size of the 95% utilization ellipse 1.64 ha – 6.56 ha (SD = 1 ha), in keeping with the range of territory sizes observed for passerines by Schoener (1968).

Availability

As in Chapter 1, I modeled availability as an interval-specific event *Sings.yijrk* with 2 possible states (1 = song, 0 = no song). To produce temporal song patterns reflective of breeding males, I incorporated autocorrelation at 2 scales. Coarse-scale autocorrelation referred to bird *j* being in or out of "singing mode," a state in which vocalization is frequent and songs occur at relatively regular intervals. If the bird was not in singing mode (i.e., if *SingingMode.yijrk* = 0), then bird *j* necessarily did not vocalize during interval *k* and *Sings.yijrk* = 0. If the bird was in singing mode, (i.e., if *SingingMode.yijrk*

= 1), then *Sings.yijrk* could be 0 or 1: these vocalizations were modeled with fine-scale autocorrelation.

Field estimates of availability (p_a) are highly variable. Availability is often reported as the probability that a bird sings at least once in X minutes (for brevity, PrSingX). Emlen (1977) found a wide range of p_a (0-0.80) among 16 species for 5 different weeks of the breeding season. Mayfield (1981) found nearly the same range (0-0.90) among 20 species for "census efficiency" over 10 replicate surveys (census efficiency in that study was roughly equivalent to PrSing10). Much intraspecific variation in availability is related to time of day (e.g., Hochachka et al. 2009) and reproductive status (e.g., Stacier et al. 2006, Robbins et al. 2009). To capture the range seen in field studies, I developed 2 availability scenarios: high p_a and low p_a.

I used estimates of high availability (including estimates for unpaired males) to parameterize the high p_a scenarios (Table 2.3). Using these estimates, I selected PrSing5 = 0.80 as the mean availability for birds in the high p_a scenarios. (Note: under the Markov chain availability modeling described in Chapter 1, PrSing5 = 0.80 corresponds to PrSing3 = 0.67 and PrSing10 = 0.94.)

I used estimates of availability from low-availability species (such as grassland birds and paired males) to parameterize the low p_a scenarios (Table 2.3). Additionally, Conway and Gibbs (2005) found probability of detection < 0.3 for 10 of 12 secretive marsh bird species passively surveyed (their estimates include surveys of varying lengths, and thus I did not include them in Table 2.3). Using these estimates, I selected PrSing5 = 0.45 as the mean availability for birds in the low p_a scenarios. (Note, PrSing5 = 0.45 corresponds to PrSing3 = 0.34 and PrSing10 = 0.65.)

For availability modeling, PrSing5 was equal to the product of a maximum daily song rate and a time-of-day weight (see Chapter 1). I held the maximum daily song rate at 0.99 for the high-availability scenario, because many high-availability species have periods when availability during a 3-4 minute survey is almost assured (e.g., Scott et al.

2005, Stacier et al. 2006, Robbins et al. 2009, Hochachka et al. 2009). I used a maximum daily song rate of 0.56 for low p_a scenarios.

For the time-of-day weight, I consulted data from P. Blancher (Environment Canada, personal communication, Figure 2.1). Based on these curves, I created a time-of-day weight function for high and low p_a (Figure 2.2). I estimated the mean time-of-day weight by averaging across the time needed to survey a Breeding Bird Survey (BBS) route (from 30 minutes before sunrise until 4 hours after sunrise; North American Breeding Bird Survey 1998). For high p_a, the product of this mean weight and the maximum daily song rate was 0.8, the desired PrSing5. Lacking evidence that time of day affects low-availability species differently than high-availability species, I used the same time-of-day weight function for both scenarios (Figure 2.2), achieving the desired mean PrSing5 = 0.45 for low p_a scenarios. I did not include variation due to ordinal date in maximum daily song rate (thus assuming that all surveys fell within a peak singing period for the species). To simulate variation inherent in field work, planned survey start times were spaced 23 minutes apart (30 minutes for the removal survey type) from sunrise to 2.5 hours after sunrise, and actual starting times were normally distributed based on planned times (SD = 5 minutes). Because these simulated surveys do not last as long as BBS surveys and availability peaks early in the day (Figure 2.2), the realized availability in my model may have exceeded the desired PrSing5 = 0.8 and PrSing5 = 0.45.

Perceptibility

I modeled Perceptibility as a function of distance to the observer (x). I used 2 perceptibility scenarios: high p_d [such as a brown thrasher (*Toxostoma rufum*) or white-throated sparrow (*Zonotrichia albicollis*); Figure 2.3] and low p_d [such as a black-and-white warbler (*Mniotilta varia*) or golden-crowned kinglet (*Regulus satrapa*); Figure 2.3]. All curves used a logit link, where $\text{logit}(p_d) = \beta_0 + \beta_1 x$. For the high p_d scenarios, I used parameters $\beta_0 = 6$, $\beta_1 = -0.02$ for the average observer in optimal observation conditions (Figure 2.3). This function was informed by p_d curves for a highly detectable species (brown thrasher) in Pacifici et al. (2008). For low p_d scenarios, I used the black-and-white warbler (BAWW) as a guide because many detection studies have included BAWW as a low-detectability species (e.g., Alldredge et al. 2007, Pacifici et al. 2008). For the average observer in optimal observation conditions, I used parameters $\beta_0 = 6.3$, and $\beta_1 = -0.07$ for low p_d scenarios (Figure 2.3).

I modeled observer skill as an effect on the intercept (β_0) for both the high p_d and low p_d scenarios. The observer skill effect was year-specific (as if new observers were selected every year, which differs from the model in Chapter 1) and drawn from a normal distribution [N(0,0.75)], meaning 95% of observer skill effects were within the range -1.5 to 1.5 (effect size = 3). This effect size was informed by the effect of observer on detection in Pacifici et al. (2008), where the effect sizes for 3 species were 2.3, 3.0, and 4.2. Observers (*n* = 6) in Alldredge et al. (2007b) had about a third that variability. That difference may be in part due to the larger sample of observers (*n* = 12) used in Pacifici et al. (2008). Observer skill was also estimated in Chapter 3; effect size for 4 observers varied among species, with range 1.0 to 3.4. The choice to use the same observer skill effect for high and low p_d scenarios is debatable, as the estimated observer effects in Pacifici et al. (2008) were larger for BAWW (observer coefficient range -3.1 to 1.1) than for other species (brown thrasher range -1.7 to 0.5; black-throated blue warbler range -2.0 to 1.0). With comparison only among 3 species in Pacifici et al. (2008), I did not see adequate evidence that observer variability is greater for species with lower detectability.

I modeled observation conditions as an effect on the slope (β_1) for both the high p_d and low p_d scenarios, with range -0.02 to 0 for high p_d scenarios and range -0.04 to 0 for low p_d scenarios. For scenarios where perceptibility was a function of habitat, the observation conditions effect was the product of the slope effect and the site-specific percent habitat (which also affects site abundance). For scenarios where perceptibility was not a function of habitat, the observation conditions effect was the product of the slope effect and a uniformly distributed, site-specific random variable between 0 and 1. The observation conditions effect was therefore similar in magnitude across all scenarios, but was only confounded with abundance in scenarios where perceptibility was a function of habitat. In keeping with the patterns in habitat effects on detection seen in Pacifici et al.

(2008), the detection functions I used were steeper for sites with poorer observation conditions, with detection dropping off at shorter distances (Figure 2.3).

The resulting high and low p_d functions I used were consistent with perceptibility estimated in field studies. For the most-detectable species, Pacifici et al. (2008) estimated p_d from 60% - 90% at 100 m for the 4 lowest-detectability habitats investigated (with ambient noise) and p_d from 90%-100% for the other 4 habitats. For the high p_d function I used, at 100 m observers had p_d with range 62% - 97% in the poorest observation conditions and p_d with range 92% - 99.5% for the best observation conditions. For the least-detectable species, Pacifici et al. (2008) estimated the range of distances at which p_d = 50% to be 50 m – 130 m for the 8 habitats investigated. For the low p_d function I used, p_d = 50% at 44 – 111 m. For the 4 least-detectable species studied, Wolf et al. (1995) estimated the distance at which p_d = 50% to be < 100 m. My low p_d function was consistent, with p_d = 50% for all observation conditions and observers at distances ≤ 112 m. For the 4 most detectable species studied, Wolf et al. (1995) estimated the distance at which p_d = 50% to be 170-272 m. My high p_d function had a similar range for average observers (p_d = 50% at 150-300 m across observation conditions) but included more observer variation (p_d = 50% at 112-375 m across all observers and observation conditions) than seen in Wolf et al. (1995). Schieck (1997) found detection of BAWW songs ranged from 67% to 100% at 50 m and from 0% to 67% at 100 m across habitat types; my low p_d function had range 33% to 99% at 50 m and 0.2% to 69% at 100 m across all observers and observation conditions. For the 3 most detectable species studied, Schieck (1997) found p_d = 1 at 100 m across a range of observation conditions; my high p_d function had range 88% to 98% at 100 m for average observers across all observation conditions. The range of maximum detection distance (63 m – 137 m) for BAWW songs in Simons et al. (2007) is consistent with the range of distances at which detection becomes negligible for my low p_d function (p_d < 5% occurs between 70 and 153 m across all observation conditions and observers). Note, however, that Simons et al. (2007) compared among wind and noise conditions, not habitats.

Estimators and True Density

I analyzed simulated counts generated by my model to produce abundance estimates. I then converted abundance estimates to density (birds/ha, equation 1.3) because density is a common currency across survey types and accounts for the difference in area surveyed across methods. For example, fixed-radius surveys set a maximum distance from the observer beyond which birds are not recorded (Ralph and Scott 1995), but distance sampling methods customarily determine a maximum distance for observations by truncating a percentage of the most-distant observations (Buckland et al. 2001). If comparing years of surveys with different numbers of survey sites, annual abundance across sites must be summarized as mean site-abundance or converted to density. To avoid confusion when comparing surveys with different radii or number of sites, I use density rather than abundance to discuss estimation of bird population parameters.

For each density estimator, I calculated correlation with true density, D_p (the density of simulated birds present within the survey radius at the beginning of the survey). Assuming a closed population (birds do not move among sites and abundance is constant within surveys), the total abundance across sites is ΣN_p. For bird surveys where the objective is to make inference about abundance across sites, N_p is therefore the optimal "true" abundance to use for comparison when evaluating estimators. I therefore used D_p as the primary representative of true density for comparisons of estimators. For distance sampling adjustment methods, observations were truncated such that the most distant 10% were excluded (Buckland et al. 2001). This truncation distance was used as the survey radius for calculating density for distance sampling adjustment estimators. For all other estimators, density was calculated for a 150-m survey radius.

I used 2 criteria for evaluating estimators: density estimator bias as compared to true density, and density estimator correlation with true density. I calculated year-specific bias as the difference between estimated density and true D_p (negative bias indicated underestimation of true density). I also calculated Pearson correlation coefficients between density estimators and true D_p. I calculated bias and correlation coefficients after removing outliers (defined as density estimates that were > 3 standard deviations away from the mean of remaining estimates) and report the number of outliers removed.

Simple Index Estimator Analysis

My first analysis investigated the performance of simple index estimators, defined as the sum of counts across all sites within a year, divided by the area surveyed. I compared bias of simple index estimators for surveys with 150-m radius across survey types using a linear model, performing an analysis of variance across survey types. Then, within each survey type, I compared bias of simple index estimators across the 3 binary parameter options for scenarios (high/low availability, high/low perceptibility, and perceptibility as a function of percent habitat or not), using a linear model with the binary parameter options as factors. I analyzed correlation of simple index estimators with D_p by calculating a 95% confidence interval for each correlation coefficient using a Fisher Z-transformation (Zou 2007). Overlap comparison for confidence intervals can be too conservative, resulting in the null hypothesis (that correlation coefficients do not differ) being rejected less often than it should (Schenker and Gentleman 2001). To compare correlation coefficients of interest, I calculated a 95% confidence interval for the difference between correlation coefficients, treating the correlations as independent (Zou 2007).

Adjustment Estimator Analysis

For comparison to simple index estimators, I used Program unmarked to calculate model-averaged density estimates as well as top model density estimates for the double-observer, distance sampling, removal, and replicated counts survey types (see Chapter 1 for estimator details). I also included estimators specific to survey type, namely the Nichols et al. (2000) estimator for the double-observer survey type (using the dependent observer approach), model-averaged and top model results from Program Distance for the distance sampling survey type, and bounded count (Johnson et al. 2007) and maximum count estimators for the replicated counts survey type. Bounded count density was the sum of the bounded counts (twice the maximum count minus the second largest count), divided by the area surveyed. Maximum count density was the sum of all maximum counts for each site (among the 3 counts within a year) divided by the area surveyed.

Again using bias and correlation with true density (D_p) as indicators of performance, I compared simple index estimators to other estimators. Within each survey type and pair of binary parameter options (high/low availability and high/low perceptibility), I used a linear model to compare estimator bias for the simple index estimator and adjusted density estimators from Program unmarked (model-averaged results and results from using the top model for each simulated year). All reports of significance for linear models of effects on estimator bias use $\alpha = 0.05$.

Within each scenario and survey type, I calculated the correlation coefficient for each estimator with true density (D_p) across 30 simulated years. I then calculated a 95% confidence interval for each correlation coefficient using a Fisher Z-transformation (Zou 2007). To avoid overly conservative comparisons (Schenker and Gentleman 2001), I calculated a 95% confidence interval for the difference between correlation coefficients of interest, treating the correlations as independent (Zou 2007). I compared estimators within each survey type, and for simple index estimators and adjusted density estimators from Program unmarked, I compared estimators across survey types.

Trend Analysis

I calculated the population trend for true abundance (N_p) and estimators of abundance as the geometric mean rate of change from year 1 to year 30 (Link and Sauer 1998). For each estimator with site-specific estimates, I calculated the mean trend (and 95% confidence interval) across all sites. For sites with a zero estimate in year 1 or 30, I added 1 to site abundance in both years to avoid inestimable trends due to the log transform in the Link and Sauer (1998) method.

Results

Simple Index Estimators

Simple index estimators were negatively biased for all survey types (Table 2.4, Appendix D). Bias differed among survey types (analysis of variance $P < 0.001$). Across all scenarios, simple index estimators were the least biased for the removal survey type (mean = -0.170 birds/ha, SE = 0.00706); the range of mean bias across other survey types was -0.295 birds/ha (SE = 0.00706) to -0.259 birds/ha (SE = 0.00706).

Bias of simple index estimators was significantly more negative for scenarios with low availability than for scenarios with high availability ($P < 0.001$ for all survey types), and significantly more negative for scenarios with low perceptibility than for high perceptibility ($P < 0.001$ for all survey types). Within each survey type, low availability had a greater effect on simple index estimator bias than did low perceptibility (coefficients for low availability had range -0.176 to -0.129 across survey types, coefficients for low perceptibility had range -0.141 to -0.115 across survey types). For the distance sampling and removal survey types, simple index estimators were less biased for scenarios with perceptibility as a function of percent habitat as compared to scenarios without ($P = 0.012$ and $P = 0.016$, respectively). For the simple count, double-observer, and replicated counts survey types, simple index estimator bias did not differ for scenarios with perceptibility as a function of percent habitat as compared to scenarios without (P range 0.14 - 0.45 across survey types).

For scenarios with high availability and perceptibility (Scenarios 1 and 5, Table 2.1), correlation of simple index estimators with D_p was uniformly high (> 0.75) across survey types (Table 2.5, Figure 2.4). Within survey types, simple index estimator correlation with D_p was more variable but did not significantly differ across other scenarios. The removal survey type had simple index estimators with the least variation in correlation with D_p (range 0.719 – 0.916 across scenarios) and the replicated counts survey type had the most variation (range 0.0149 – 0.904 across scenarios) (Figure 2.4).

Adjustment Estimators

Estimator performance differed by survey type (Appendix D). Across all estimators, a maximum of 4 (of 30) estimates were removed as outliers (Appendix D). For the double-observer survey type, the density estimates from Program unmarked and Nichols et al. (2000) estimates (Table 2.4, Appendix D: D6, D10, D14) were nearly identical to simple index estimates (Table 2.4, Appendix D: D2). Bias did not differ for that survey type between adjustment estimators and index estimators (Table 2.4). Within scenarios, the correlation of density estimators with D_p was similar across estimators for the double-observer survey type (Table 2.5, Figure 2.5).

For Program unmarked estimators in the distance sampling survey type, the mean truncation distance for observations had range 372 m – 410 m for the scenarios with high perceptibility, and range 130 m – 140 m for scenarios with low perceptibility. For the distance sampling survey type, all estimators were negatively biased ($P < 0.01$; Table 2.4, Appendix D: D5, D9, D13, D17). For scenarios with low perceptibility, adjusted estimators were significantly less negatively biased than simple index estimators ($P < 0.001$). For scenarios with high perceptibility and high availability, Program Distance estimators and simple index estimators had similar bias but Program unmarked estimators were significantly less negatively biased than simple index estimators ($P < 0.001$). For scenarios with high perceptibility and low availability, the Program unmarked top model estimator, Program Distance estimators, and simple index estimators had similar bias, but the Program unmarked model-averaged estimator was significantly less negatively biased ($P = 0.038$). For Scenario 1, the correlation of the simple index estimator with D_p was greater than for all adjustment estimators for the distance sampling survey type (Figure 2.6). For Scenarios 2, 5, and 6 (Table 2.1), the correlation of the simple index estimator with D_p was greater than for the Program unmarked estimators, but not the Program Distance estimators. For all other scenarios using the distance sampling survey type, there was no difference in the correlation with D_p among estimators.

For removal survey type scenarios with low availability (Scenarios 2, 4, 6, and 8; Table 2.1), adjusted estimators were significantly less negatively biased than the simple index estimator ($P < 0.001$, Appendix D: D4, D8, D12). For removal survey type scenarios with high availability and high perceptibility, adjusted estimators and the simple index estimator were similarly negatively biased. For removal survey type scenarios with high availability and low perceptibility, adjusted estimators were less negatively biased compared to the simple index estimator ($P < 0.001$). Simple index estimators had a higher correlation with D_p than did adjusted estimators for low availability scenarios in the removal survey type (Table 2.5, Figure 2.7). For high availability scenarios for the removal survey type, there was no difference among estimators in the correlation with D_p.

For the replicated counts survey type, all adjustment estimators showed significant positive bias compared to simple index estimators. Many adjusted estimates were highly inflated for replicated count surveys, despite removal of outliers (Appendix D: D7, D11). Bias for bounded count estimators and maximum count estimators did not significantly differ from bias for simple index estimators, but had positive coefficients. Simple index estimators for the replicated counts survey type (Appendix D: D3) had greater correlation with D_p than adjustment estimators for Scenarios 1, 2, 5, 6, and 7 (Table 2.1, Figure 2.8). For the replicated counts survey type, all estimators for had similar correlation with D_p for Scenarios 3, 4, and 8 (Table 2.1). Correlation with D_p did not differ between the maximum count estimator and the bounded count estimator within any scenario for the replicated counts survey type.

Trend

There was a negative trend in true abundance (N_p) over 30 simulated years, with range -0.92% to -2.0% annually across scenarios and survey types. Across all survey types, the simple index estimator detected a negative trend for abundance for all but one high-availability scenario (distance sampling survey type, Scenario 3; Table 2.1). For simple counts and replicated counts survey types, the simple index estimator did not detect a trend for any low availability scenarios. For the removal survey type, the simple index estimator detected a trend for low availability and high perceptibility scenarios, but not low availability and low perceptibility scenarios. For the double-observer survey type, the simple index estimator detected a trend only for Scenario 6 (Table 2.1) among low availability scenarios. For the distance sampling survey type, the simple index estimator detected a trend for all low availability scenarios except Scenario 2 (Table 2.1).

For the double-observer and distance sampling survey types, all adjusted estimators detected a negative trend for all scenarios, with the exception of the low availability, low perceptibility scenarios for the distance sampling survey type. Trend detection by adjusted estimators was more mixed for the replicated counts and removal survey types. For the replicated counts survey type, a negative trend was detected for scenarios 4, 6, and 8 by both adjusted estimators, for scenario 3 by the model-averaged adjusted estimator only, and for scenario 5 by the top model adjusted estimator only (for scenario numbers, see Table 2.1). For the removal survey type, adjusted estimators detected a negative trend for scenarios 3 and 4, as well as in scenario 6 for the top model estimators only.

Discussion

Although index methods were biased and only had a strong correlation with true density when detectability was high, adjustment methods generally did not offer an improvement. Compared to bias of index methods and correlation of index methods with true density, adjustment method performance ranged from far worse (replicated counts), to no added value (double-observer) to moderate improvement (in bias only, for removal and distance sampling in specific scenarios). Adjustment methods performed better at detecting a trend in abundance with 2 of the 4 adjustment survey types (distance sampling and double-observer), but index methods detected a negative trend when availability was high.

These results expand upon Chapter 1 by applying the model to a wider array of scenarios. By using parameters informed by empirical data, this analysis examines the performance of density estimators across the range of availability and perceptibility expected to occur in bird surveys. The model I used makes critical assumptions about bird behavior and human observations, most importantly that abundance is influenced by habitat availability, that birds move within bivariate normal territories, that song production is an autocorrelated process, and that detection as a function of distance takes the form of a logit function. If a bird's behavior or detection violates these assumptions (by moving linearly across the landscape, say, or singing randomly through time) these simulation results may not be applicable. Additionally, the natural world is complicated, and any simulation is a simplification. This model does not include double-counting, false positives, species misidentification, an influence of the observer on the behavior of the birds, or interactions among birds.

Still, simulation studies can provide an important evaluation of analysis methods. Efford and Dawson (2009) assessed bias in abundance estimates by simulating counts and including distance-related heterogeneity of individual detection probability. They concluded that no adjustment method effectively estimated population size from point counts. While adjustment methods (Nichols et al. 2000, Burnham et al. 1980, Buckland et al. 1993, Farnsworth et al. 2002, Alldredge et al. 2007a, Royle 2004) and combinations of adjustment methods (Riddle et al. 2010, Stanislav et al. 2010, Amundson et al. 2014) have proliferated, few attempts have been made to evaluate these methods via simulation. To my knowledge, my model is the most comprehensive modeling of the detection process in bird surveys to date. Ideally, bird survey practitioners can evaluate availability and perceptibility in their subject species and choose a survey type and analysis method informed by the scenario here that most closely matches their system.

Counts can provide valuable information about bird populations and, when considered carefully, their use as an index to abundance can be constructive. Indices performed best for scenarios with high availability and perceptibility. To the extent possible, practitioners should standardize their methods to increase availability and perceptibility in their surveys and to lower their variation. Banks-Leite et al. (2014) found that controlling for covariates of detection probability before data collection was just as effective as using adjustment methods, and recommended that analysis methods be tailored to research objectives. Adjusting survey timing (both regarding time of day and time within season) to the time when birds are most consistently vocal can increase availability. The standardized North American Marsh Bird Monitoring Protocol (Conway 2011) also uses broadcast recordings of calls as a mechanism for increasing availability. Surveys targeting other birds with known low availability (such as grassland birds) could potentially benefit from a similar playback procedure. Longer survey periods also increase availability. Surveyors seeking to maximize availability can perform 5- or 10-minute surveys (Ralph et al. 1993, 1995, Matsuoka et al. 2014) rather than commonly-used 3-minute surveys. Observer training can increase perceptibility and lower variability among observers (Kepler and Scott 1981, McLaren and Cadman 1999). Conducting surveys during preferable conditions (e.g., low wind) can increase perceptibility (Robbins 1981a, Chapter 3). My results suggest that these efforts have a greater potential benefit to assuring reliable inferences from count data than using adjustment estimators.

Collecting additional information necessary for adjustment methods comes at some cost. In my model, human effort was held constant, so that cost was represented by fewer surveyed sites for the double-observer, replicated counts, and removal survey types as compared to the simple counts and distance sampling survey types. There may also be an added cost to the observer's attention due to recording additional information, exhibited by a reduction in p_d. Collecting information on the distance to each bird for the distance sampling survey type, the interval in which they were observed for the removal survey type, and reconciling detections between observers for the double-observer survey type may take up a non-trivial portion of an observer's attention. The problem may be exacerbated for multi-species surveys or high-abundance sites. Incorporating the additional information for an adjustment method incorporates an addition potential source of error (e.g., Alldredge et al 2007c). If estimating absolute density is a desired objective, the cost of collecting removal data was arguably worthwhile, as adjusted removal estimates were less biased than other survey types. For the double-observer survey type, however, only half as many sites were surveyed as compared to the simple counts, but there was no added value in using the double-observer adjusted estimator. Replicated counts surveys had only a third of the sites as compared to simple counts surveys, producing highly unreliable adjustment estimators (e.g., Dennis et al. 2015). When additional data collected for adjustment methods do not produce better estimates, the best survey type is the one that maximizes the number of sites surveyed.

Most importantly, no adjustment method consistently provided better density estimates than index methods across the biological scenarios I compared. Because there is no "silver bullet" analysis method that can perform well across the range of bird availability and perceptibility, practitioners should carefully consider the sources of variation in detection probability in their system, and select an appropriate method. If detection components are unknown or known to be variable, I advise practitioners to perform a pilot study to estimate the mean value and variability of each component. Alternatively, practitioners may follow Skalski and Robson's (1992) recommendation to collect information to estimate detection and then to select among competing models that do and do not include adjustment. The downside of this suggestion is that logistical costs of collecting such data will be spent even if an index method is eventually selected. Still, those costs may be well spent if detection in a system is not well understood.

Table 2.1: Scenarios parameterized to reflect 3 binary options for scenario parameters: high vs. low availability, high vs. low perceptibility, and the presence or absence of an effect of habitat on perceptibility. The result was 8 scenarios making up all possible binary combinations. For high and low availability and perceptibility parameters, see Figures 2 and 3.

Scenario	Availability	Perceptibility	Perceptibility ~ Habitat
1	high	high	yes
2	low	high	yes
3	high	low	yes
4	low	low	yes
5	high	high	no
6	low	high	no
7	high	low	no
8	low	low	no

Table 2.2: Forest songbird densities and territory size reported in 3 studies across multiple species and habitats.

	Density (birds/ha)			Territory Size (ha)		
Study	Maximum	Mean	Median	Minimum	Maximum	Mean
Schoener 1968[a]				0.11	13.23	
Morse 1976	2.08	0.90	0.76	0.22[b]	1.1	
Smith and Shugart 1987				0.13	0.38	0.26
Thompson et al. 1992	1.13	0.19	0.12			
Wenny et al. 1993	0.21	0.11	0.09			

[a] Minimum and maximum territory sizes reported for passerines only.

[b] Approximate.

Table 2.3: Availability estimates for passerine species used to select parameters for high and low availability scenarios.

Minimum[a]	Maximum	Mean	Species	Source
		0.85	Kirtland's warbler (*Setophaga kirtlandii*)	Mayfield 1981
		0.55	prairie warbler (*Setophaga discolor*)	Mayfield 1981
0.43	0.78		house wren (*Troglodytes aedon*)	Wilson and Bart 1985
0	0.99	0.96	least Bell’s vireo (*Vireo belli pusillus*)	Scott et al. 2005
0.92	1	0.99	American redstart, unpaired (*Setophaga ruticilla*)	Stacier et al. 2006
0.16	0.74	0.49	American redstart, paired	Stacier et al. 2006
		0.44	Henslow’s sparrow (*Ammodramus henslowii*)	Diefenbach et al. 2007
		0.12	grasshopper sparrow (*Ammodramus savannarum*)	Diefenbach et al. 2007
0.09	> 1	0.51	multiple	Confer et al. 2008
0.92	0.97		cerulean warbler, unpaired (*Setophaga cerulean*)	Robbins et al. 2009
0.54	0.62		cerulean warbler, paired	Robbins et al. 2009
		0.62	cerulean warbler (all)	Robbins et al. 2009
0.75	1		Florida grasshopper sparrow (*Ammodramus savannarum floridanus*)	Hochachka et al. 2009

[a] Estimates of availability refer to the probability that a bird sings at least once within 5 minutes (PrSing5). Estimates published for different time periods were transformed, using the assumptions that song intervals were 2 seconds long and the probability of remaining in singing mode given that the bird was previously in singing mode was 0.98.

Table 2.4: Density estimates, true density, and bias of estimates, scenarios 1-8 (Table 2.1). Each density estimate was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$).

	Survey Type	Estimator	Density Estimate	True Density (D_p)	Median Bias	Percent Bias
Scenario 1	Simple Counts	Simple Index Density	0.346	0.477	-0.130	-27%
	Double-Observer	Simple Index Density	0.415	0.490	-0.068	-14%
		Model-Averaged Adjusted Density	0.417	0.490	-0.068	-14%
		Top Model Adjusted Density	0.417	0.490	-0.067	-14%
		Nichols et al. (2000) Density	0.415	0.490	-0.068	-14%
	Replicated Counts	Simple Index Density	0.363	0.497	-0.132	-27%
		Model-Averaged Adjusted Density	1.65	0.497	0.568	114%
		Top Model Adjusted Density	1.43	0.497	0.446	90%
		Bounded Count Density	0.631	0.497	0.133	27%
		Maximum Count Density	0.494	0.497	-0.002	-0.47%
	Removal	Simple Index Density	0.433	0.484	-0.054	-11%
		Model-Averaged Adjusted Density	0.435	0.484	-0.050	-10%
		Top Model Adjusted Density	0.435	0.484	-0.050	-10%
	Distance Sampling	Simple Index Density	0.314	0.501	-0.190	-38%
		Model-Averaged Adjusted Density	0.485	0.506	-0.004	-0.8%
		Top Model Adjusted Density	0.486	0.506	0.004	0.8%
		Program Distance Model-Averaged Density	0.383	0.501	-0.115	-23%
		Program Distance Top Model Density	0.383	0.501	-0.115	-23%

Table 2.4 continued: Density estimates, true density, and bias of estimates, scenarios 1-8 (Table 2.1). Each density estimator was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$).

	Survey Type	Estimator	Density Estimate	True Density (D_p)	Median Bias	Percent Bias
Scenario 2	Simple Counts	Simple Index Density	0.162	0.473	-0.312	-66%
	Double-Observer	Simple Index Density	0.193	0.503	-0.307	-61%
		Model-Averaged Adjusted Density	0.193	0.503	-0.307	-61%
		Top Model Adjusted Density	0.193	0.503	-0.307	-61%
		Nichols et al. (2000) Density	0.193	0.503	-0.307	-61%
	Replicated Counts	Simple Index Density	0.169	0.511	-0.347	-68%
		Model-Averaged Adjusted Density	5.71	0.511	7.289	1428%
		Top Model Adjusted Density	6.13	0.511	7.493	1468%
		Bounded Count Density	0.410	0.511	-0.105	-21%
		Maximum Count Density	0.284	0.511	-0.229	-45%
	Removal	Simple Index Density	0.304	0.484	-0.172	-36%
		Model-Averaged Adjusted Density	0.394	0.484	-0.122	-25%
		Top Model Adjusted Density	0.375	0.484	-0.126	-26%
	Distance Sampling	Simple Index Density	0.144	0.499	-0.348	-70%
		Model-Averaged Adjusted Density	0.171	0.498	-0.328	-66%
		Top Model Adjusted Density	0.170	0.498	-0.328	-66%
		Program Distance Model-Averaged Density	0.174	0.499	-0.325	-65%
		Program Distance Top Model Density	0.173	0.499	-0.326	-65%

Table 2.4 continued: Density estimates, true density, and bias of estimates, scenarios 1-8 (Table 2.1). Each density estimator was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$).

	Survey Type	Estimator	Density Estimate	True Density (D_p)	Median Bias	Percent Bias
Scenario 3	Simple Counts	Simple Index Density	0.206	0.477	-0.276	-58%
	Double-Observer	Simple Index Density	0.241	0.490	-0.248	-51%
		Model-Averaged Adjusted Density	0.246	0.490	-0.243	-50%
		Top Model Adjusted Density	0.246	0.490	-0.243	-50%
		Nichols et al. (2000) Density	0.245	0.490	-0.245	-50%
	Replicated Counts	Simple Index Density	0.224	0.497	-0.275	-55%
		Model-Averaged Adjusted Density	3.07	0.497	0.641	129%
		Top Model Adjusted Density	3.32	0.497	0.378	76%
		Bounded Count Density	0.442	0.497	-0.067	-14%
		Maximum Count Density	0.330	0.497	-0.166	-33%
	Removal	Simple Index Density	0.340	0.484	-0.132	-27%
		Model-Averaged Adjusted Density	0.388	0.484	-0.102	-21%
		Top Model Adjusted Density	0.379	0.484	-0.104	-22%
	Distance Sampling	Simple Index Density	0.232	0.497	-0.257	-52%
		Model-Averaged Adjusted Density	0.445	0.498	-0.057	-12%
		Top Model Adjusted Density	0.429	0.498	-0.058	-12%
		Program Distance Model-Averaged Density	0.467	0.497	-0.029	-5.9%
		Program Distance Top Model Density	0.467	0.497	-0.029	-5.9%

Table 2.4 continued: Density estimates, true density, and bias of estimates, scenarios 1-8 (Table 2.1). Each density estimator was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$).

	Survey Type	Estimator	Density Estimate	True Density (D_p)	Median Bias	Percent Bias
Scenario 4	Simple Counts	Simple Index Density	0.091	0.473	-0.390	-83%
	Double-Observer	Simple Index Density	0.110	0.503	-0.394	-78%
		Model-Averaged Adjusted Density	0.112	0.503	-0.393	-78%
		Top Model Adjusted Density	0.112	0.503	-0.393	-78%
		Nichols et al. (2000) Density	0.112	0.503	-0.393	-78%
	Replicated Counts	Simple Index Density	0.091	0.511	-0.422	-83%
		Model-Averaged Adjusted Density	4.43	0.511	4.577	896%
		Top Model Adjusted Density	4.48	0.511	2.430	476%
		Bounded Count Density	0.266	0.511	-0.246	-48%
		Maximum Count Density	0.172	0.511	-0.342	-67%
	Removal	Simple Index Density	0.198	0.481	-0.277	-58%
		Model-Averaged Adjusted Density	0.323	0.481	-0.181	-38%
		Top Model Adjusted Density	0.297	0.481	-0.200	-42%
	Distance Sampling	Simple Index Density	0.102	0.501	-0.389	-78%
		Model-Averaged Adjusted Density	9.67	0.502	-0.298	-59%
		Top Model Adjusted Density	15.8	0.502	-0.292	-58%
		Program Distance Model-Averaged Density	0.217	0.501	-0.272	-54%
		Program Distance Top Model Density	0.217	0.501	-0.272	-54%

Table 2.4 continued: Density estimates, true density, and bias of estimates, scenarios 1-8 (Table 2.1). Each density estimator was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$).

	Survey Type	Estimator	Density Estimate	True Density (D_p)	Median Bias	Percent Bias
Scenario 5	Simple Counts	Simple Index Density	0.352	0.479	-0.125	-26%
	Double-Observer	Simple Index Density	0.420	0.500	-0.083	-16%
		Model-Averaged Adjusted Density	0.420	0.500	-0.083	-16%
		Top Model Adjusted Density	0.420	0.500	-0.083	-16%
		Nichols et al. (2000) Density	0.420	0.500	-0.083	-16%
	Replicated Counts	Simple Index Density	0.357	0.497	-0.138	-28%
		Model-Averaged Adjusted Density	2.78	0.497	0.759	153%
		Top Model Adjusted Density	2.66	0.497	0.644	130%
		Bounded Count Density	0.629	0.497	0.129	26%
		Maximum Count Density	0.491	0.497	-0.011	-2.1%
	Removal	Simple Index Density	0.437	0.484	-0.049	-10%
		Model-Averaged Adjusted Density	0.440	0.484	-0.047	-10%
		Top Model Adjusted Density	0.440	0.484	-0.047	-10%
	Distance Sampling	Simple Index Density	0.297	0.498	-0.198	-40%
		Model-Averaged Adjusted Density	0.373	0.502	-0.134	-27%
		Top Model Adjusted Density	0.370	0.502	-0.137	-27%
		Program Distance Model-Averaged Density	0.363	0.498	-0.126	-25%
		Program Distance Top Model Density	0.363	0.498	-0.126	-25%

Table 2.4 continued: Density estimates, true density, and bias of estimates, scenarios 1-8 (Table 2.1). Each density estimator was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$).

	Survey Type	Estimator	Density Estimate	True Density (D_p)	Median Bias	Percent Bias
Scenario 6	Simple Counts	Simple Index Density	0.158	0.471	-0.317	-67%
	Double-Observer	Simple Index Density	0.196	0.486	-0.290	-60%
		Model-Averaged Adjusted Density	0.197	0.486	-0.290	-60%
		Top Model Adjusted Density	0.197	0.486	-0.290	-60%
		Nichols et al. (2000) Density	0.196	0.486	-0.290	-60%
	Replicated Counts	Simple Index Density	0.170	0.505	-0.328	-65%
		Model-Averaged Adjusted Density	4.67	0.505	3.340	661%
		Top Model Adjusted Density	5.15	0.505	3.247	642%
		Bounded Count Density	0.394	0.505	-0.117	-23%
		Maximum Count Density	0.278	0.505	-0.210	-42%
	Removal	Simple Index Density	0.311	0.481	-0.166	-35%
		Model-Averaged Adjusted Density	0.388	0.481	-0.100	-21%
		Top Model Adjusted Density	0.381	0.481	-0.110	-23%
	Distance Sampling	Simple Index Density	0.138	0.500	-0.366	-73%
		Model-Averaged Adjusted Density	0.200	0.501	-0.292	-58%
		Top Model Adjusted Density	0.197	0.501	-0.296	-59%
		Program Distance Model-Averaged Density	0.168	0.500	-0.341	-68%
		Program Distance Top Model Density	0.167	0.500	-0.341	-68%

Table 2.4 continued: Density estimates, true density, and bias of estimates, scenarios 1-8 (Table 2.1). Each density estimator was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$).

	Survey Type	Estimator	Density Estimate	True Density (D_p)	Median Bias	Percent Bias
Scenario 7	Simple Counts	Simple Index Density	0.182	0.477	-0.305	-64%
	Double-Observer	Simple Index Density	0.214	0.494	-0.269	-54%
		Model-Averaged Adjusted Density	0.218	0.494	-0.266	-54%
		Top Model Adjusted Density	0.218	0.494	-0.267	-54%
		Nichols et al. (2000) Density	0.218	0.494	-0.267	-54%
	Replicated Counts	Simple Index Density	0.200	0.504	-0.294	-58%
		Model-Averaged Adjusted Density	1.48	0.504	0.306	61%
		Top Model Adjusted Density	1.39	0.504	0.320	63%
		Bounded Count Density	0.423	0.504	-0.081	-16%
		Maximum Count Density	0.308	0.504	-0.192	-38%
	Removal	Simple Index Density	0.305	0.484	-0.171	-35%
		Model-Averaged Adjusted Density	0.353	0.484	-0.120	-25%
		Top Model Adjusted Density	0.353	0.484	-0.124	-26%
	Distance Sampling	Simple Index Density	0.215	0.506	-0.290	-57%
		Model-Averaged Adjusted Density	0.476	0.508	-0.056	-11%
		Top Model Adjusted Density	0.489	0.508	-0.035	-6.8%
		Program Distance Model-Averaged Density	0.478	0.506	-0.019	-3.8%
		Program Distance Top Model Density	0.481	0.506	-0.016	-3.1%

Table 2.4 continued: Density estimates, true density, and bias of estimates, scenarios 1-8 (Table 2.1). Each density estimator was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$).

	Survey Type	Estimator	Density Estimate	True Density (D_p)	Median Bias	Percent Bias
Scenario 8	Simple Counts	Simple Index Density	0.080	0.473	-0.396	-84%
	Double-Observer	Simple Index Density	0.097	0.494	-0.382	-77%
		Model-Averaged Adjusted Density	0.100	0.494	-0.377	-76%
		Top Model Adjusted Density	0.100	0.494	-0.376	-76%
		Nichols et al. (2000) Density	0.098	0.494	-0.378	-76%
	Replicated Counts	Simple Index Density	0.084	0.506	-0.417	-83%
		Model-Averaged Adjusted Density	3.92	0.506	1.048	207%
		Top Model Adjusted Density	4.19	0.506	0.687	136%
		Bounded Count Density	0.256	0.506	-0.238	-47%
		Maximum Count Density	0.162	0.506	-0.338	-67%
	Removal	Simple Index Density	0.175	0.481	-0.306	-64%
		Model-Averaged Adjusted Density	0.274	0.481	-0.233	-48%
		Top Model Adjusted Density	0.272	0.481	-0.235	-49%
	Distance Sampling	Simple Index Density	0.092	0.507	-0.413	-81%
		Model-Averaged Adjusted Density	0.309	0.505	-0.290	-57%
		Top Model Adjusted Density	0.251	0.505	-0.311	-62%
		Program Distance Model-Averaged Density	0.243	0.507	-0.272	-54%
		Program Distance Top Model Density	0.269	0.507	-0.246	-49%

Table 2.5: Pearson correlation coefficients (ρ) for density estimators, scenarios 1-8 (Table 2.1). Each density estimator was compared to true density D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$). Upper and lower limits of a 95% confidence interval (CI) were calculated with a Fisher Z-transformation (Zou 2007).

	Survey Type	Estimator	Survey Radius (m)	ρ D_p	Upper CI	Lower CI
Scenario 1	Simple Counts	Simple Index Density	150	0.890	0.947	0.780
	Double-Observer	Simple Index Density	150	0.878	0.941	0.757
		Model-Averaged Adjusted Density	150	0.874	0.939	0.751
		Top Model Adjusted Density	150	0.873	0.938	0.747
		Nichols et al. (2000) Density	150	0.878	0.940	0.756
	Replicated Counts	Simple Index Density	150	0.901	0.952	0.800
		Model-Averaged Adjusted Density	150	-0.074	0.294	-0.423
		Top Model Adjusted Density	150	0.205	0.527	-0.167
		Bounded Count Density	150	0.801	0.901	0.619
		Maximum Count Density	150	0.890	0.947	0.779
	Removal	Simple Index Density	150	0.916	0.960	0.829
		Model-Averaged Adjusted Density	150	0.909	0.956	0.817
		Top Model Adjusted Density	150	0.910	0.957	0.818
	Distance Sampling	Simple Index Density	150	0.919	0.961	0.836
		Simple Index Density	410	0.930	0.966	0.856
		Model-Averaged Adjusted Density	410	0.766	0.883	0.561
		Top Model Adjusted Density	410	0.704	0.849	0.460
		Program Distance Model-Averaged Density	410	0.618	0.800	0.331
		Program Distance Top Model Density	410	0.618	0.800	0.331

Table 2.5 continued: Pearson correlation coefficients (ρ) for density estimators, scenarios 1-8 (Table 2.1). Each density estimator was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$). Upper and lower limits of a 95% confidence interval (CI) were calculated with a Fisher Z-transformation (Zou 2007).

	Survey Type	Estimator	Survey Radius (m)	ρ D_p	Upper CI	Lower CI
Scenario 2	Simple Counts	Simple Index Density	150	0.785	0.893	0.592
	Double-Observer	Simple Index Density	150	0.463	0.706	0.124
		Model-Averaged Adjusted Density	150	0.490	0.723	0.158
		Top Model Adjusted Density	150	0.491	0.723	0.159
		Nichols et al. (2000) Density	150	0.463	0.706	0.124
	Replicated Counts	Simple Index Density	150	0.764	0.882	0.558
		Model-Averaged Adjusted Density	150	-0.239	0.132	-0.552
		Top Model Adjusted Density	150	-0.272	0.098	-0.576
		Bounded Count Density	150	0.623	0.803	0.338
		Maximum Count Density	150	0.741	0.869	0.520
	Removal	Simple Index Density	150	0.824	0.913	0.660
		Model-Averaged Adjusted Density	150	0.490	0.722	0.157
		Top Model Adjusted Density	150	0.447	0.695	0.103
	Distance Sampling	Simple Index Density	150	0.830	0.916	0.669
		Simple Index Density	406	0.881	0.942	0.762
		Model-Averaged Adjusted Density	406	0.177	0.505	-0.195
		Top Model Adjusted Density	406	0.212	0.532	-0.160
		Program Distance Model-Averaged Density	406	0.791	0.896	0.602
		Program Distance Top Model Density	406	0.791	0.896	0.602

Table 2.5 continued: Pearson correlation coefficients (ρ) for density estimators, scenarios 1-8 (Table 2.1). Each density estimator was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$). Upper and lower limits of a 95% confidence interval (CI) were calculated with a Fisher Z-transformation (Zou 2007).

	Survey Type	Estimator	Survey Radius (m)	ρ D_p	Upper CI	Lower CI
Scenario 3	Simple Counts	Simple Index Density	150	0.691	0.842	0.440
	Double-Observer	Simple Index Density	150	0.724	0.860	0.491
		Model-Averaged Adjusted Density	150	0.732	0.864	0.504
		Top Model Adjusted Density	150	0.707	0.850	0.465
		Nichols et al. (2000) Density	150	0.736	0.866	0.511
	Replicated Counts	Simple Index Density	150	0.557	0.764	0.247
		Model-Averaged Adjusted Density	150	0.180	0.507	-0.193
		Top Model Adjusted Density	150	0.168	0.498	-0.204
		Bounded Count Density	150	0.513	0.737	0.187
		Maximum Count Density	150	0.615	0.798	0.327
	Removal	Simple Index Density	150	0.741	0.869	0.519
		Model-Averaged Adjusted Density	150	0.505	0.732	0.176
		Top Model Adjusted Density	150	0.406	0.668	0.053
	Distance Sampling	Simple Index Density	140	0.674	0.832	0.414
		Simple Index Density	150	0.538	0.752	0.220
		Model-Averaged Adjusted Density	140	0.556	0.763	0.244
		Top Model Adjusted Density	140	0.584	0.780	0.283
		Program Distance Model-Averaged Density	140	0.627	0.805	0.345
		Program Distance Top Model Density	140	0.627	0.805	0.345

Table 2.5 continued: Pearson correlation coefficients (ρ) for density estimators, scenarios 1-8 (Table 2.1). Each density estimator was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$). Upper and lower limits of a 95% confidence interval (CI) were calculated with a Fisher Z-transformation (Zou 2007).

	Survey Type	Estimator	Survey Radius (m)	ρ D_p	Upper CI	Lower CI
Scenario 4	Simple Counts	Simple Index Density	150	0.609	0.795	0.319
	Double-Observer	Simple Index Density	150	0.169	0.499	-0.204
		Model-Averaged Adjusted Density	150	0.190	0.515	-0.183
		Top Model Adjusted Density	150	0.192	0.517	-0.180
		Nichols et al. (2000) Density	150	0.173	0.502	-0.199
	Replicated Counts	Simple Index Density	150	0.265	0.571	-0.105
		Model-Averaged Adjusted Density	150	0.070	0.419	-0.298
		Top Model Adjusted Density	150	0.019	0.377	-0.344
		Bounded Count Density	150	0.444	0.693	0.099
		Maximum Count Density	150	0.436	0.688	0.090
	Removal	Simple Index Density	150	0.719	0.857	0.484
		Model-Averaged Adjusted Density	150	0.208	0.529	-0.164
		Top Model Adjusted Density	150	0.193	0.518	-0.179
	Distance Sampling	Simple Index Density	138	0.306	0.600	-0.061
		Simple Index Density	150	0.401	0.665	0.048
		Model-Averaged Adjusted Density	138	0.013	0.371	-0.349
		Top Model Adjusted Density	138	0.108	0.451	-0.263
		Program Distance Model-Averaged Density	138	0.347	0.629	-0.015
		Program Distance Top Model Density	138	0.347	0.629	-0.015

Table 2.5 continued: Pearson correlation coefficients (ρ) for density estimators, scenarios 1-8 (Table 2.1). Each density estimator was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$). Upper and lower limits of a 95% confidence interval (CI) were calculated with a Fisher Z-transformation (Zou 2007).

	Survey Type	Estimator	Survey Radius (m)	ρ D_p	Upper CI	Lower CI
Scenario 5	Simple Counts	Simple Index Density	150	0.871	0.937	0.743
	Double-Observer	Simple Index Density	150	0.773	0.886	0.572
		Model-Averaged Adjusted Density	150	0.774	0.887	0.573
		Top Model Adjusted Density	150	0.773	0.886	0.572
		Nichols et al. (2000) Density	150	0.773	0.886	0.572
	Replicated Counts	Simple Index Density	150	0.904	0.954	0.807
		Model-Averaged Adjusted Density	150	0.045	0.398	-0.321
		Top Model Adjusted Density	150	0.073	0.422	-0.295
		Bounded Count Density	150	0.838	0.920	0.684
		Maximum Count Density	150	0.901	0.952	0.801
	Removal	Simple Index Density	150	0.912	0.957	0.821
		Model-Averaged Adjusted Density	150	0.904	0.954	0.806
		Top Model Adjusted Density	150	0.904	0.954	0.807
	Distance Sampling	Simple Index Density	150	0.922	0.962	0.840
		Simple Index Density	372	0.889	0.946	0.778
		Model-Averaged Adjusted Density	372	0.704	0.849	0.460
		Top Model Adjusted Density	372	0.713	0.854	0.475
		Program Distance Model-Averaged Density	372	0.817	0.910	0.648
		Program Distance Top Model Density	372	0.817	0.910	0.648

Table 2.5 continued: Pearson correlation coefficients (ρ) for density estimators, scenarios 1-8 (Table 2.1). Each density estimator was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$). Upper and lower limits of a 95% confidence interval (CI) were calculated with a Fisher Z-transformation (Zou 2007).

	Survey Type	Estimator	Survey Radius (m)	ρ D_p	Upper CI	Lower CI
Scenario 6	Simple Counts	Simple Index Density	150	0.403	0.667	0.050
	Double-Observer	Simple Index Density	150	0.670	0.830	0.409
		Model-Averaged Adjusted Density	150	0.662	0.825	0.396
		Top Model Adjusted Density	150	0.661	0.825	0.395
		Nichols et al. (2000) Density	150	0.670	0.830	0.409
	Replicated Counts	Simple Index Density	150	0.679	0.835	0.422
		Model-Averaged Adjusted Density	150	0.127	0.466	-0.245
		Top Model Adjusted Density	150	0.087	0.434	-0.282
		Bounded Count Density	150	0.347	0.629	-0.015
		Maximum Count Density	150	0.537	0.752	0.219
	Removal	Simple Index Density	150	0.827	0.915	0.664
		Model-Averaged Adjusted Density	150	0.506	0.733	0.178
		Top Model Adjusted Density	150	0.445	0.694	0.101
	Distance Sampling	Simple Index Density	150	0.864	0.934	0.731
		Simple Index Density	372	0.832	0.917	0.673
		Model-Averaged Adjusted Density	372	0.588	0.782	0.289
		Top Model Adjusted Density	372	0.535	0.751	0.217
		Program Distance Model-Averaged Density	372	0.765	0.882	0.559
		Program Distance Top Model Density	372	0.765	0.882	0.559

Table 2.5 continued: Pearson correlation coefficients (ρ) for density estimators, scenarios 1-8 (Table 2.1). Each density estimator was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$). Upper and lower limits of a 95% confidence interval (CI) were calculated with a Fisher Z-transformation (Zou 2007).

	Survey Type	Estimator	Survey Radius (m)	ρ D_p	Upper CI	Lower CI
Scenario 7	Simple Counts	Simple Index Density	150	0.608	0.794	0.318
	Double-Observer	Simple Index Density	150	0.614	0.798	0.325
		Model-Averaged Adjusted Density	150	0.614	0.798	0.325
		Top Model Adjusted Density	150	0.615	0.798	0.327
		Nichols et al. (2000) Density	150	0.611	0.796	0.321
	Replicated Counts	Simple Index Density	150	0.806	0.904	0.628
		Model-Averaged Adjusted Density	150	0.079	0.427	-0.290
		Top Model Adjusted Density	150	-0.028	0.336	-0.384
		Bounded Count Density	150	0.678	0.834	0.420
		Maximum Count Density	150	0.766	0.883	0.560
	Removal	Simple Index Density	150	0.758	0.878	0.547
		Model-Averaged Adjusted Density	150	0.584	0.780	0.284
		Top Model Adjusted Density	150	0.524	0.744	0.201
	Distance Sampling	Simple Index Density	130	0.507	0.733	0.180
		Simple Index Density	150	0.622	0.803	0.338
		Model-Averaged Adjusted Density	130	0.383	0.653	0.027
		Top Model Adjusted Density	130	0.381	0.652	0.024
		Program Distance Model-Averaged Density	130	0.621	0.802	0.336
		Program Distance Top Model Density	130	0.621	0.802	0.336

Table 2.5 continued: Pearson correlation coefficients (ρ) for density estimators, scenarios 1-8 (Table 2.1). Each density estimator was compared to D_p (the density of birds present at the beginning of the survey, $D_p = N^* p_s p_p / A$). Upper and lower limits of a 95% confidence interval (CI) were calculated with a Fisher Z-transformation (Zou 2007).

	Survey Type	Estimator	Survey Radius (m)	ρD_p	Upper CI	Lower CI
Scenario 8	Simple Counts	Simple Index Density	150	0.513	0.737	0.187
	Double-Observer	Simple Index Density	150	0.500	0.729	0.170
		Model-Averaged Adjusted Density	150	0.533	0.750	0.214
		Top Model Adjusted Density	150	0.541	0.754	0.224
		Nichols et al. (2000) Density	150	0.496	0.727	0.165
	Replicated Counts	Simple Index Density	150	0.015	0.373	-0.347
		Model-Averaged Adjusted Density	150	0.194	0.518	-0.179
		Top Model Adjusted Density	150	0.170	0.500	-0.203
		Bounded Count Density	150	0.067	0.417	-0.300
		Maximum Count Density	150	0.059	0.411	-0.308
	Removal	Simple Index Density	150	0.730	0.863	0.501
		Model-Averaged Adjusted Density	150	0.173	0.502	-0.200
		Top Model Adjusted Density	150	0.132	0.470	-0.240
	Distance Sampling	Simple Index Density	135	0.413	0.673	0.062
		Simple Index Density	150	0.426	0.681	0.077
		Model-Averaged Adjusted Density	135	-0.006	0.355	-0.365
		Top Model Adjusted Density	135	0.101	0.445	-0.269
		Program Distance Model-Averaged Density	135	0.417	0.676	0.066
		Program Distance Top Model Density	135	0.417	0.676	0.066

Chapter III

Factors Affecting Detection Probability and Effective Area Surveyed for Grassland Bird Point Counts

Summary: I conducted simulated bird surveys using recorded bird songs to assess factors affecting detection probability in grassland bird point counts. I used mixed effects logistic regression models to estimate factors affecting detection probability and to estimate and visualize the variation in the area around the observer where birds can be perceived (the perceptible area). I conducted simulated surveys with 8926 binary opportunities for detection in Minnesota grasslands in 2011 and 2012. Species, distance to the observer, wind speed and direction, observer, and density of vegetation all affected detection of recorded bird songs. Species had a strong effect; the size of the predicted perceptible area around the observer differed by more than 10-fold among species. Wind also had a strong effect on detection. As wind speed increased, probability of detection downwind of the observer was reduced and the perceptible area around the observer became smaller and more asymmetrical. The effective distance at which an observer is more likely to detect a bird than to not detect it may differ among species and angles to the wind, even within the same survey. I recommend using fixed-radius counts for bird surveys in grasslands and reducing the variation in detection probability by standardizing surveys across wind conditions.

Point counts are commonly used in bird surveys to achieve a number of objectives, including estimating abundance, population monitoring, and evaluating factors affecting bird populations (Ralph et al. 1995). Point counts are ubiquitous in biological monitoring, yet there is significant debate regarding how count information can best be used. Of particular importance is the role played by detection probability. The expected value of the count (E(C)) for any point is equal to the product of the number of birds present (*N*) and the probability of detection (*p*; Nichols et al. 2009). Detection probability can be affected by a wide variety of factors (Verner 1985), including species (Diefenbach et al. 2003), survey elements (length of survey, survey type; Dawson et al. 1995, Bollinger et al. 1988, Cimprich 2009), behavioral factors (singing rate, volume, and motion of birds; Wilson and Bart 1985, McShea and Rappole 1997, Alldredge 2007b), environmental factors (precipitation, wind speed, ambient noise, time of day, time within the breeding season, even tides; Robbins 1981a, Wilson and Bart 1985, Zembal and Massey 1987, Rosenberg and Blancher 2005, Pacifici et al. 2008), and observer effects (hearing ability, skill, distance from the source; Sauer et al. 1994, Alldredge 2007b). Over the last 20 years, great effort has gone into crafting methods to account for detection probability in bird surveys ("adjustment methods"), such as double observer surveys (Nichols et al. 2000), replicated samples (Royle 2004), distance sampling (Burnham et al. 1980, Buckland et al. 1993), and removal (Farnsworth et al. 2002) or time-of-detection (Alldredge et al. 2007a) methods.

A component of detection probability is perceptibility, the probability of detecting a bird, given that it is present at the site and makes itself available for detection during the survey (e.g, it sings or is visible). Perceptibility decreases as a function of increased distance between the observer and the bird, which is the basis for distance sampling (Burnham et al. 1980, Buckland et al. 1993). Variation in perceptibility may also affect the distance at which birds are detected. I define the effective radius as the distance at which an observer is more likely to detect a bird than to not detect it (i.e, probability of detection ≥ 0.5). For example, 2 observers with different hearing abilities may have different effective radii. Observers with different effective radii have different areas in which they can effectively detect birds, or perceptible areas.

How variation in the effective radius affects an analysis depends on the analysis method used. In unlimited-distance point counts, observers record all birds detected, but for fixed-radius point counts observers censor detections at a certain radius (or record birds in "bands" of fixed radii; Ralph et al. 1993). Survey radii selected for fixed-radius counts vary, although 50 m is often used (Ralph et al. 1993, Ralph et al. 1995, Matsuoka et al.

2014). It is unclear how an observer’s effective radius and perceptible area compares to the survey radius and area theoretically surveyed for fixed-radius counts. For some analysis methods, such as using counts as a index to abundance, standardizing the survey radius or distance bands across surveys has an implicit assumption that detection within those bands will be constant across surveys. Even less clear is how the area surveyed differs among unlimited-radius counts.

Despite the emphasis on the importance of detectability in bird surveys, there has been relatively little research to quantify factors affecting detection probability in the field due to the difficulty of measuring factors affecting detectability of wild birds. Calculating detectability is simple if true populations are known, but that is seldom, if ever, the case in the field. Simons et al. (2007) described a multi-speaker system to broadcast bird songs to multiple observers, thus simulating bird surveys with known populations. Alldredge et al. (2007b) used the system to investigate effects on detectability in mature bottomland hardwood and mixed pine-hardwood forest, estimating effects of singing rate, species, observer, and distance, and interactions among main effects. Pacifici et al. (2008) used a similar setup to estimate effects of habitat type (deciduous forest vs. mixed pine-hardwood), presence/absence of foliage, ambient noise, observer, and distance. These studies remain the most intensive investigations to date.

Grasslands have markedly different structure and species composition than the forests examined in previous studies of factors affecting detectability (Alldredge et al. 2007b, Pacifici et al. 2008). Grassland birds across North America have experienced widespread population declines for decades, increasing conservation concern for the group (Peterjohn and Sauer 1999). Making population goals or planning conservation actions to benefit grassland bird populations requires an accurate assessment of population status and trends. Inferences about populations can be affected by the type of analysis method used to estimate abundance (Chapters 1 and 2). Selecting an analysis method to provide good inferences about abundance requires an understanding of the sources of variation in detection in a system (Chapter 2). An understanding of factors affecting detectability in grassland bird surveys would guide and improve bird survey methodology for an increasingly imperiled group.

To address these information needs, my objectives in this study were to (1) assess factors affecting the detection probability of recorded bird songs in a grassland setting, and (2) estimate and visualize the variation in the area around the observer where birds can be perceived (the perceptible area) in relation to factors affecting detection probability. Information gained would improve grassland bird surveyors' ability to assess the effect of variation in detection probability in their system, as well as better understand the true size of the area surveyed.

Study Area

In 2011, I carried out field work on Waterfowl Production Areas (WPAs) in the Morris Wetland Management District near Morris, Minnesota. Land cover in this region is highly fragmented, with agriculture (primarily corn and soybeans) dominant. Historically, this region was part of the tallgrass prairie ecosystem that stretched from Canada to Texas, though modern prairie in Minnesota has declined an estimated 99.6% from historical area (Samson and Knopf 1994). WPAs are protected for the dual purpose of waterfowl production and public recreation opportunities, including hunting (Morris Wetland Management District 2011). WPAs usually include a wetland for adult waterfowl and broods in addition to upland habitat for nesting (Morris Wetland Management District 2011). This study focused on upland prairie, though wetlands were included incidentally. In 2012, we performed field work on land owned by the Belwin Conservancy near Afton, Minnesota, Carver Park Reserve near Victoria, Minnesota and Murphy-Hanrehan Park Reserve near Burnsville, Minnesota. These sites were managed as grasslands to provide habitat for wildlife and for recreation.

Methods

Field Work

In 2011, I selected survey sites in the Morris Wetland Management District within a 30-km radius of Morris, Minnesota in Pope, Grant, Stevens, and Swift counties, with supplemental sites up to 70 km from Morris to provide a wider diversity of habitat conditions. In 2012, I selected survey sites within a 30-mile radius of the St. Paul campus of the University of Minnesota. I chose sites using aerial photography (U.S. Department of Agriculture 2010). Eligible sites had a grassland area with at least a 150-m radius that did not intersect with roads, developed land, or other manmade structures. Sites included ≥ 50% grassland. Site perimeters were within 200 m of a road to ensure access, but were located ≥ 500 m from highways to reduce noise interference. Roads adjacent to sites were seldom traveled, with passing traffic rates under 10 automobiles per hour. I eliminated actively grazed sites, and sites with elevation changes > 10 m and sites with tree cover > 20%. I selected survey sites to provide a wide range of habitat conditions, from lightly vegetated, recently grazed or mown sites to heavily vegetated, shrubby sites.

I surveyed 35 sites alone in 2011. Four observers (including myself) performed surveys simultaneously at 12 sites in 2012 (Table 3.1). Two observers in 2012 had experience identifying grassland bird songs in the field and two observers did not, although they had previously identified songs of other species. I classified the former 2 observers as experienced observers and the latter 2 as inexperienced observers. We performed surveys under a variety of wind conditions, with winds recorded from 0-9 ms^{-1}. We performed all surveys in autumn to minimize the presence of real singing birds.

We performed surveys at locations within each survey site as follows. Surveys took place at "sampling points" located along each of 6 linear "sampling legs" (Figure 3.1). Sampling points were located at 30, 60, 90, 120, and 150 m from the central point (as measured with a GPS). Sampling legs radiated from the central point, spaced 60° apart (Figure 3.1). The first leg of each site was oriented directly into the wind, or due north when there was no detectable wind. In 2011, I conducted surveys at all 6 legs (30 sampling points were surveyed per site). In 2012, we conducted surveys at only 4 legs (legs at either 60º and 120º or 240º and 300º were eliminated as redundant, 20 sampling points were surveyed per site). I eliminated sampling points if obstacles (ditches, deep water, heavy forest) prohibited observer access, but such eliminations were rare. When suitable grassland habitat was adjacent to non-suitable habitat (forested or wetland habitat), we performed surveys at suitable legs and omitted surveys of non-suitable legs. We did not perform any surveys at a site if fewer than half of legs could be surveyed.

In 2011, I broadcast recorded bird songs from the central point of each survey site with an Apple iPod music player connected to Poly-Planar MA4055 marine speakers via a Poly-Planar ME-60 amplifier. I positioned the speakers at the beginning of surveys for each survey leg to broadcast toward the observer. In 2012, I replaced the speakers with a TIC GS 5P OmniSpeaker, which broadcasts 360º, to allow simultaneous listening by observers in multiple orientations. I mounted all speakers such that they stood at 1-m height during surveys (roughly representative of perch height in grasslands).

I used song recordings from the Macaulay Library at Cornell University for 10 species that inhabit grasslands in Minnesota: horned lark (*Eremophila alpestris*, HOLA, Macaulay Library 1988b), bobolink (*Dolichonyx oryzivorus*, BOBO, Macaulay Library 1988a), eastern meadowlark (*Sturnella magna*, EAME, Macaulay Library 1998), vesper sparrow (*Pooecetes gramineus*, VESP, Macaulay Library 1988e), savannah sparrow (*Passerculus sandwichensis*, SAVS, Macaulay Library 1988d), grasshopper sparrow (*Ammodramus savannarum*, GRSP, 1990a), Henslow's sparrow (*Ammodramus henslowii*, HESP, 1992b), Le Conte's sparrow (*Ammodramus leconteii*, LCSP, 1988c), dickcissel (*Spiza Americana*, DICK, 1988f), and sedge wren (*Cistothorus platensis*, SEWR, Macaulay Library 1990b). I trimmed song recordings in .WAV format to 6-second files using Program Audacity (open source, http://audacity.sourceforge.net). To ensure that all songs were played at the same volume regardless of species, I adjusted song volume using the function Amplify in Program Audacity. Selected songs produced a maximum volume of 90 dB (± 1.5 dB; Brackenbury 1979, Simons et al. 2007) when measured 1 m from the speaker system using a Martel Electronics model 325 mini sound level meter

(Derry, NH, http://www.martelcorp.com). Within years, once I achieved consistent volume, I used the same 10 songs for all surveys. I calibrated song volume at each survey site to 90 dB at 1 m, measured with the sound level meter positioned upwind of the speakers.

Each simulated survey at a sampling point consisted of a 1-minute listening period during which the speaker system played a playlist of 10, 6-second audio files. Each audio file contained either a bird song or silence, with 1 to 10 songs played in each sampling event. I randomly selected number of songs from a uniform distribution rounded to the nearest integer (1-10) and assigned species of songs randomly, with replacement, from a list of 10 species. I ordered songs and pauses randomly, but eliminated playlists with consecutive songs of the same species to avoid confusion. I constructed playlists in Program R (R Development Core Team, Vienna, Austria, http://www.R-project.org) and played with Program iTunes (Apple Inc., Cupertino, CA, http://www.apple.com/itunes).

Surveys began with all observers at the site's center point. I triggered playback of playlists for the first leg and observers walked from the central point to their first sampling points, 30 m away along a survey leg. I always began by surveying the leg directly upwind. Other observers (if present) simultaneously performed surveys along legs in different directions. Each observer performed one survey at each sampling point along each survey leg once per site.

Playlists played automatically throughout a leg, with walking time (94 seconds in 2011, 60 seconds in 2012) allowed before each 1-minute listening period. The beginning of the listening period was signaled by loud beeps sounded at 9 seconds before the listening period and immediately before the listening period. The observers wrote down all detected bird songs heard during the listening periods, also noting if extraneous noise interfered with their observation ability. During each listening period I measured average wind speed over a 10-second period using a Kestrel 2000 anemometer (Birmingham, MI). There was no audible warning at the end of the listening period, requiring the observers to time the 1-minute interval. When listening periods at all 5 sampling points of the first leg were complete, the observers returned to the central point to begin surveys for subsequent survey legs.

I measured habitat characteristics using 2-m profile boards divided into 0.33-m sections by alternating black and white paint (Nudds 1977). I staked boards into the ground 1 m to the left or right of each sampling point and 1 m from the central point. We took care not to trample vegetation in front of each board. I estimated horizontal visibility for each profile board stratum to the nearest 5% from 30 m. I intended horizontal visibility as an index of vegetation density that might affect sound attenuation between the speakers and the observer. The horizontal visibility estimates for the 30-m sampling point, for example, were taken with the observer standing at 30 m from the central point and estimating percent visibility of the profile board stationed at the central point. I calculated a habitat index (H) from the horizontal visibility estimates, where $H = \Sigma$ (% stratum visibility × midpoint of stratum (m)) across all 6 strata, with a range 0 to 6. When wind was too extreme for profile boards to be used, we recorded song detections during listening periods as usual and I recorded habitat conditions on a calmer day, no more than 1 week later.

Analysis

I analyzed factors affecting detection with mixed effects logistic regression models, using correct detection of a broadcast song as the binary response variable. To allow for an iterative model selection process, I divided the data into 3 groups. I performed initial analyses using 25 randomly selected sites from 2011 (Group 1), withholding data from 10 sites for validation purposes (Group 2). Data from 2012 (12 sites) constituted Group 3. I included survey site as a random effect and analyzed each species separately.

For Group 1 data analysis, I rescaled all input variables via mean centering and dividing by 2 standard deviations (Gelman 2007). Fixed effects included wind speed in ms^{-1} (S), wind speed squared (S^2), a wind index [W, where $W = \cos(180° - \text{wind direction}) \times S$, with range S to $-S$], distance from the sound source to the observer in meters (D), distance from the sound source to the observer squared (D^2), habitat index (H), and observer-indicated binary variables for human-made noise (*M*), bird noise from real birds (*B*), and other noise (*N*). I created a model set of 197 logistic regression models with 0 to 6 covariates each. I created the set of models by including all combinations of covariates that fit the following rules:

(1) A model could include *S*, S^2, both, or neither.
(2) A model could include *D* or D^2, or neither.
(3) A model could include *M* or *B* or *N* or no noise covariate.
(4) A model could include *H* or not.
(5) A model could include *W* or not.
(6) All models included the random effect of survey site.

I ranked models by Akaike Information Criterion (AIC) values and calculated species-specific variable importance weights (VIW_i) for each fixed effect, where $\text{VIW}_i = \Sigma$ AIC weights for models in which fixed effect *i* appeared (Burnham and Anderson 2002). Average VIW_i was calculated as the mean of species-specific VIW_i. I designated fixed effects with mean $\text{VIW}_i \geq 0.5$ as "important" predictors of detection and designated fixed effects with mean $\text{VIW}_i < 0.5$ as "unimportant" fixed effects. I did not include unimportant fixed effects in further model selection steps.

I estimated important fixed effects using Group 1 data, predicted detection for Group 2 data, and evaluated the performance of predictions. I estimated important fixed effects using 6 mixed effects models containing the important fixed effects (*D, W, S,* S^2, and *H*) and the random effect of survey site. The model set included all combinations of covariates that fit the following rules:

(1) All prediction models included *D* and *W*.
(2) A model could include *S*, *S* and S^2, or neither.
(3) A model could include *H* or not.
(4) All models included the random effect of survey site.

For each detection opportunity, I assigned a predicted detection = 1 if the predicted probability of detection was ≥ 0.5 and predicted detection = 0 if the predicted probability of detection was < 0.5. I then calculated the proportion of correct predictions by model and species. I also examined model-predicted probability of detection and actual detections as functions of distance and wind index to assess model fit.

My next objective was to determine the best way to model the effect of observer for Group 3 data. I created a model set of 18 models, using the 6 mixed effects models used in the previous step and adding either O, a factor with 4 levels (1 for each observer); E, a factor with 2 levels (1 indicating an experienced observer and 0 indicating an inexperienced observer); or neither observer effect. Within each species, I ranked the 18 models by AIC and calculated VIW_i for each observer effect (O, E, and no effect).

The final model set therefore contained models with the following rules:

(1) All prediction models included D and W.
(2) A model could include S, S and S^2, or neither.
(3) A model could include H or not.
(4) All models included O or E, depending on species.
(5) All models included the random effect of survey site.

I used the final model set to estimate species-specific fixed and random effects for factors affecting detection using all data (Groups 1, 2, and 3). I ranked models by AIC within each species and calculated model weights. For observer effects, I used a simple coding contrast matrix (UCLA Statistical Consulting Group 2011), which produces an intercept that represents the grand mean (mean of group means) across the observer categories. The reported intercepts therefore do not represent the intercept for the reference observer (as with a dummy variable contrast) but rather the intercept for the average observer. I rescaled H via mean centering and dividing by 2 standard deviations (Gelman 2007) so that using a covariate value $H = 0$ referred to average horizontal visibility conditions.

Using fixed effects estimates from the final model set, I predicted and graphed species-specific, model-averaged probability of detection across a 1 x 1 m grid of points surrounding a hypothetical observer (located at the origin). I report predicted probability of detection for 4 wind speeds: 0 ms^{-1}, 1 ms^{-1}, 2.5 ms^{-1}, and 4.5 ms^{-1}. These wind speeds correspond to Beaufort numbers 0, 1, 2, and 3, respectively. The instructions for the Breeding Bird Survey recommend performing bird surveys when conditions warrant a Beaufort Number ≤ 2 and prohibits surveys if conditions warrant a Beaufort Number > 3 (Patuxent Wildlife Research Center 1998). For each wind speed examined, I also calculated species-specific perceptible areas, which I defined as the area around the observer for which the predicted probability of detection was ≥ 0.5.

Results

We surveyed sites from 9 September to 11 October 2011 and from 7 September to 21 October 2012. There were 8926 binary opportunities for detection across the 2-year study (Table 3.1).

For the analysis of Group 1 data, best-supported models varied among species for the 197 logistic regression models tested. Covariates D, W, S, S^2, and H had mean $\mathrm{VIW}_i \geq 0.5$ and were considered important predictors of detection (Table 3.2). Covariates D and W had highest VIW_i (Table 3.2) and appeared in all models with $\Delta\mathrm{AIC} < 5$ for all species. I therefore included D and W in all prediction models for later model sets.

The 6 models using fixed effects estimates from Group 1 data to predict detection for Group 2 data all performed well; the proportion of correct predictions had range 0.680 – 0.900 across species and models (Table 3.3). Variation in prediction performance was greater among species than within models (Table 3.3). EAME was the most predictable species, with range 86.2% – 90.0% correctly predicted detections among the 6 models. HESP was the least predictable, with 68.2% - 73.4% correctly predicted detections. LCSP had the lowest proportion of predicted detections (range 0.096 – 0.125) and EAME had the highest proportion of predicted detections (range 0.862 – 0.900). Visual examination of the distribution of incorrect predictions in reference to covariates did not show clusters of failed predictions, thus indicating good model fit.

For the analysis of Group 3 data, models with the observer effect *O* had the majority of model weight (higher VIW_i) for species BOBO, GRSP, and SEWR (Table 3.4). For all other species, models with the observer effect *E* had the majority of model weight. For species SAVS, summed model weights were nearly equal for each observer effect, but models including *E* had lower ΔAIC than similar models including *O*. I therefore included observer effect *O* in prediction models for BOBO, GRSP, and SEWR and included observer effect *E* in prediction models for all other species.

I estimated fixed effects for models in the final model set using all data (Groups 1, 2, and 3; Table 3.5). All fixed effects estimates for *D* were negative, indicating detection decreased with increasing distance between the observer and the sound source (Table 3.5). All fixed effects estimates for *W* were positive, indicating that detection was higher for locations upwind of the observer and lower for downwind locations. All fixed effects estimates for *E* were positive, indicating that observer experience increased probability of detection (Table 3.5). Species-specific model selection for the final model set showed that model weight was not concentrated in a single model for most species (Table 3.6). I therefore used fixed effect estimates (Table 3.5) to estimate model-specific probability of detection a 1x1 m grid of locations around the observer, then model-averaged estimates.

The predicted perceptible area around the observer varied among species and modeled wind speeds (Appendix E). The size of the perceptible area had range 0.474 - 5.30 ha across species for wind speed = 0 ms^{-1}, 0.389 - 5.60 ha for wind speed = 1 ms^{-1}, 0.275 - 5.31 ha across species for wind speed = 2.5 ms^{-1}, and 0.162 ha - 3.84 ha for wind speed = 4.5 ms^{-1} (Table 3.7, Figure 3.2). Asymmetry of the predicted perceptible area increased with increased wind speed (Appendix E). The perceptible area was concentrated upwind of the observer, particularly for greater wind speeds.

Discussion

Species, distance to the observer, wind speed and direction, observer, and density of vegetation all affected detection of recorded bird songs in grasslands. Although ambient noise has been shown in other studies to affect detection (Alldredge et al. 2007b, Simons et al. 2007), observer-reported presence of ambient noise was not a good predictor of detection in this study. Species had a strong effect on detection probability. Due to species-specific variation in perceptibility, the size of the predicted perceptible area around the observer differed by more than 10-fold among species when wind speed was held constant (Table 3.7). Wind also had a strong effect on detection. Wind is a particularly important concern for breeding season surveys of grassland birds because high winds are common during spring and grasslands lack structures or trees to block wind. As wind speed increased, probability of detection downwind of the observer was reduced and the perceptible area around the observer became smaller and more asymmetrical.

The perceptible area effectively surveyed by the observer varied widely due to species and wind speed. Fixed-radius bird survey methods (Ralph and Scott 1981) require that observers record birds within a certain survey radius. Because the perceptible area varied across species and wind conditions, it may be very difficult to know the effective radius for a particular survey. Indeed, in the presence of wind, the effective radius may be different in different directions during the same survey.

I found significant variation in perceptible area across species and wind conditions. Comparing counts across species, even from the same multi-species survey, is ill-advised unless the area surveyed for the species is similar, either because the species have similar perceptible areas, or if counts are constrained within a survey radius that is smaller than both effective radii. I recommend against comparing counts across species for unlimited-radius point counts unless the species have similar perceptible areas because the area surveyed for each species may vastly differ. Such counts should be used to estimate intraspecific relative abundance only.

The simulated bird surveys I describe were well-suited for estimating factors affecting detection of single bird songs. By using a mobile speaker system, I was able to survey a variety of grasslands, broadening the applicability of the results. Also, this method could be easily and inexpensively reproduced to estimate factors affecting detection in other landscapes or for other species. Mobility was a benefit when compared to the more elaborate and less mobile speaker systems used by Simons et al. (2007), Alldredge et al. (2007b), and Pacifici et al. (2008), but both systems required the simplifying assumption that birds were detected by sound only. The necessary simplifications of my speaker system had some additional drawbacks. I used only 1 sound source, playing only 1 song at a time, which is much simplified compared to a real bird survey. I was therefore not able to investigate double-counting, observer confusion of individual birds, or swamping effects of numerous birds. Still, this study represents an important expansion of the work of Simons et al. (2007), Alldredge et al. (2007b), and Pacifici et al. (2008) to a grassland setting. Estimating the role of wind, particularly, was important for understanding detection probability in grasslands.

To reduce the variation in detection probability across surveys, bird surveyors should standardize grassland surveys across wind conditions. Surveys should be conducted on no-wind or low-wind days to the greatest extent possible. Surveyors also should engage in training to increase their detection rate; experienced observers had higher detection than inexperienced observers in this study. The speaker system used in this study can also be used as a training tool. Testing identification and detection skills and comparing results to a known playlist is a practical way to increase detection rate for observers.

I recommend using fixed-radius counts for bird surveys in grasslands, such as recording birds separately within and outside a 50-m band (Matsuoka et al. 2014). If a species has a varying perceptible area larger than the survey radius, using an unlimited-radius count will add variation to counts. I recommend against comparing unadjusted counts of birds for different species as estimates of absolute abundance unless there is evidence that their perceptible areas are of similar size across the range of conditions surveyed. For example, BOBO and DICK had perceptible areas of similar size for higher wind speeds

(Table 3.7) and could be compared without introducing much error due to variation in perceptible area size.

Table 3.1: Opportunities for detection of recorded bird songs in grasslands near Morris, Minnesota (2011) and near the Minneapolis - St. Paul, Minnesota metro area (2012).

Year	Sites	Observers	Sampling Legs	Sampling Points	Detection Opportunities
2011	35	1	192	942	4445
2012	12	4	48	240	4481

Table 3.2: Variable importance weights (VIW$_i$) for each covariate i, for 197 logistic regression models of detection of recorded bird songs in grasslands. I analyzed data separately for each bird species; species are referenced by their alpha code (Bird Banding Laboratory 2016). VIW$_i$ were calculated as the sum of the Akaike Information Criterion (AIC) weights for the models in which the covariate appeared (Burnham and Anderson 2002). Fixed effects included distance from the sound source to the observer in meters (D), distance from the sound source to the observer squared (D^2), wind speed in ms^{-1} (S), wind speed squared (S^2), a wind index (W, where $W = \cos(180° - \text{wind direction}) \times S$, which had range S to $-S$), habitat index (H), and binary observer-indicated noise variables for noise from real birds (B), manmade noise (M), bird noise and other noise (N). Survey site was included in all models as a random effect.

	Covariate								
Species	D	D^2	S	S^2	W	H	B	M	N
BOBO	0.987	0.006	0.582	0.492	0.849	0.493	0.216	0.229	0.331
DICK	0.989	0.005	0.658	0.494	0.684	0.524	0.294	0.244	0.257
EAME	0.964	0.023	0.544	0.567	0.838	0.487	0.256	0.247	0.247
GRSP	0.991	0.005	0.554	0.480	0.613	0.489	0.258	0.249	0.240
HESP	0.968	0.019	0.547	0.564	0.790	0.493	0.281	0.236	0.234
HOLA	0.925	0.040	0.611	0.600	0.963	0.514	0.252	0.231	0.268
LCSP	0.973	0.017	0.534	0.467	0.546	0.481	0.248	0.253	0.244
SAVS	0.995	0.003	0.639	0.489	0.682	0.485	0.261	0.242	0.241
SEWR	0.975	0.012	0.685	0.501	0.929	0.482	0.241	0.244	0.251
VESP	0.909	0.044	0.709	0.555	0.871	0.553	0.250	0.243	0.243
Mean	0.968	0.017	0.606	0.521	0.777	0.500	0.256	0.242	0.256

Table 3.3: Species-specific predicted detection of recorded bird songs for 6 logistic regression models. I estimated fixed effects using Group 1 data, then predicted detection for Group 2 data. Fixed effects included distance from the sound source to the observer in meters (*D*), wind speed in ms^{-1} (*S*), wind speed squared (S^2), a wind index [*W*, where *W* = cos(180º - wind direction) × *S*, which had range *S* to –*S*], and habitat index (*H*). Species are referenced by their alpha code (Bird Banding Laboratory 2016).

Species	Proportion Detected	Fixed Effects	Proportion with > 0.5 Predicted Probability of Detection	Proportion Correctly Predicted	
				Mean	SD
GRSP	0.397	D + W	0.345	0.828	0.379
		D + W + H	0.345	0.828	0.379
		D + W + S + H	0.319	0.802	0.400
		D + W + S	0.319	0.802	0.400
		D + W + H + S + S^2	0.319	0.802	0.400
		D + W + S + S^2	0.336	0.802	0.400
SEWR	0.677	D + W	0.605	0.750	0.435
		D + W + H	0.621	0.766	0.425
		D + W + S + H	0.661	0.742	0.439
		D + W + S	0.597	0.710	0.456
		D + W + H + S + S^2	0.661	0.742	0.439
		D + W + S + S^2	0.597	0.710	0.456
HESP	0.609	D + W	0.445	0.680	0.468
		D + W + H	0.445	0.711	0.455
		D + W + S + H	0.531	0.734	0.443
		D + W + S	0.484	0.703	0.459
		D + W + H + S + S^2	0.508	0.727	0.447
		D + W + S + S^2	0.477	0.695	0.462
EAME	0.885	D + W	0.862	0.869	0.338
		D + W + H	0.869	0.862	0.347
		D + W + S + H	0.892	0.900	0.301
		D + W + S	0.900	0.892	0.311
		D + W + H + S + S^2	0.885	0.892	0.311
		D + W + S + S^2	0.877	0.869	0.338

Table 3.3 continued.

Species	Proportion Detected	Fixed Effects	Proportion with > 0.5 Predicted Probability of Detection	Proportion Correctly Predicted	
				Mean	SD
HOLA	0.797	D + W	0.692	0.783	0.414
		D + W + H	0.706	0.797	0.403
		D + W + S + H	0.762	0.825	0.381
		D + W + S	0.762	0.797	0.403
		D + W + H + S + S^2	0.748	0.825	0.381
		D + W + S + S^2	0.727	0.804	0.398
VESP	0.806	D + W	0.769	0.843	0.365
		D + W + H	0.806	0.866	0.342
		D + W + S + H	0.866	0.821	0.385
		D + W + S	0.858	0.813	0.391
		D + W + H + S + S^2	0.866	0.836	0.372
		D + W + S + S^2	0.873	0.828	0.378
DICK	0.695	D + W	0.636	0.768	0.423
		D + W + H	0.649	0.781	0.415
		D + W + S + H	0.662	0.768	0.423
		D + W + S	0.649	0.755	0.432
		D + W + H + S + S^2	0.662	0.768	0.423
		D + W + S + S^2	0.656	0.748	0.435
BOBO	0.730	D + W	0.680	0.803	0.399
		D + W + H	0.672	0.779	0.417
		D + W + S + H	0.738	0.828	0.379
		D + W + S	0.697	0.787	0.411
		D + W + H + S + S^2	0.738	0.828	0.379
		D + W + S + S^2	0.697	0.787	0.411
SAVS	0.703	D + W	0.694	0.739	0.441
		D + W + H	0.694	0.739	0.441
		D + W + S + H	0.721	0.766	0.425
		D + W + S	0.712	0.739	0.441
		D + W + H + S + S^2	0.730	0.757	0.431
		D + W + S + S^2	0.712	0.721	0.451

Table 3.3 continued.

Species	Proportion Detected	Fixed Effects	Proportion with > 0.5 Predicted Probability of Detection	Proportion Correctly Predicted	
				Mean	SD
LCSP	0.199	D + W	0.103	0.875	0.332
		D + W + H	0.096	0.838	0.370
		D + W + S + H	0.125	0.824	0.383
		D + W + S	0.125	0.809	0.395
		D + W + H + S + S^2	0.125	0.824	0.383
		D + W + S + S^2	0.125	0.809	0.395

Table 3.4: Variable importance weights (VIW) for comparison of observer effects. Models for Group 3 data contained observer effect *O*, a factor with 4 levels (1 for each observer), observer effect *E*, a factor with 2 levels (1 indicating an experienced observer and 0 indicating an inexperienced observer), or neither observer effect.

Species	*O*	*E*	No Observer Effect
BOBO	0.982	0.018	<0.001
DICK	0.274	0.675	0.051
EAME	0.198	0.802	<0.001
GRSP	0.920	0.078	0.002
HESP	0.128	0.854	0.018
HOLA	0.135	0.865	<0.001
LCSP	0.221	0.779	<0.001
SAVS	0.457	0.543	<0.001
SEWR	0.964	0.036	<0.001
VESP	0.163	0.837	<0.001

Table 3.5: Species-specific fixed effects estimates for 6 logistic regression models of detection of recorded bird songs, estimated using all data. Fixed effects included distance from the sound source to the observer in meters (*D*), wind speed in ms^{-1} (S), wind speed squared (S^2), a wind index (W, where W = cos(180º - wind direction) × S, which had range S to -S), habitat index rescaled via mean centering and dividing by 2 standard deviations (Gelman 2007) (*H*), observer effect (*O*) for 4 observers, and experienced observer effect (*E*). Species are referenced by their alpha code (Bird Banding Laboratory 2016).

Species	Fixed Effects	AIC Weight	Intercept	D	W	S	S^2	H (rescaled)	$O2$	$O3$	$O4$	E
BOBO	D + W + H + S + O	0.537	5.39	-0.044	0.467	-0.536		-0.458	-2.66	-1.93	-1.12	
BOBO	D + W + H + S + S^2 + O	0.199	5.42	-0.044	0.466	-0.564	0.004	-0.458	-2.66	-1.92	-1.12	
BOBO	D + W + S + O	0.193	5.25	-0.044	0.457	-0.497			-2.62	-1.90	-1.10	
BOBO	D + W + S + S^2 + O	0.071	5.28	-0.044	0.456	-0.519	0.003		-2.62	-1.90	-1.10	
BOBO	D + W + O	0.000	4.07	-0.043	0.460				-2.64	-1.90	-1.08	
BOBO	D + W + H + O	0.000	4.11	-0.043	0.467			-0.297	-2.67	-1.92	-1.09	
DICK	D + W + H + S + S^2 + E	0.704	4.41	-0.041	0.461	-0.083	-0.063	-0.546				0.850
DICK	D + W + H + S + E	0.164	4.84	-0.040	0.424	-0.489		-0.547				0.812
DICK	D + W + S + S^2 + E	0.107	4.31	-0.041	0.452	-0.030	-0.064					0.830
DICK	D + W + S + E	0.024	4.75	-0.041	0.416	-0.443						0.793
DICK	D + W + H + E	0.000	3.74	-0.040	0.383			-0.342				0.721
DICK	D + W + E	0.000	3.74	-0.040	0.380							0.715

Table 3.5 continued.

Species	Fixed Effects	AIC Weight	Inter-cept	*D*	*W*	*S*	S^2	*H* (re-scaled)	*O*2	*O*3	*O*4	*E*
EAME	D + W + H + S + S^2 + E	0.850	6.01	-0.046	0.637	0.314	-0.133	-0.722				2.13
EAME	D + W + S + S^2 + E	0.097	5.99	-0.047	0.624	0.341	-0.131					2.08
EAME	D + W + H + S + E	0.046	6.72	-0.045	0.590	-0.466		-0.694				2.04
EAME	D + W + S + E	0.006	6.71	-0.046	0.582	-0.427						2.00
EAME	D + W + H + E	0.000	5.66	-0.044	0.595			-0.579				2.01
EAME	D + W + E	0.000	5.70	-0.045	0.585							1.97
GRSP	D + W + S + S^2 + O	0.517	3.06	-0.057	0.444	0.208	-0.086		-1.48	-1.24	-1.09	
GRSP	D + W + H + S + S^2 + O	0.196	3.08	-0.057	0.444	0.192	-0.085	-0.060	-1.49	-1.25	-1.10	
GRSP	D + W + S + O	0.184	3.55	-0.057	0.414	-0.293			-1.46	-1.22	-1.07	
GRSP	D + W + H + S + O	0.075	3.58	-0.057	0.413	-0.307		-0.109	-1.47	-1.23	-1.08	
GRSP	D + W + O	0.021	2.88	-0.056	0.392				-1.44	-1.18	-1.02	
GRSP	D + W + H + O	0.008	2.88	-0.056	0.392			0.030	-1.43	-1.18	-1.02	
HESP	D + W + H + S + S^2 + E	0.708	5.14	-0.054	0.580	-0.099	-0.061	-0.624				0.869
HESP	D + W + H + S + E	0.208	5.59	-0.054	0.553	-0.516		-0.674				0.853
HESP	D + W + S + S^2 + E	0.072	4.88	-0.053	0.572	0.017	-0.069					0.866
HESP	D + W + S + E	0.012	5.37	-0.053	0.541	-0.448						0.848
HESP	D + W + H + E	0.000	4.40	-0.053	0.522			-0.453				0.804
HESP	D + W + E	0.000	4.36	-0.052	0.516							0.804

Table 3.5 continued.

Species	Fixed Effects	AIC Weight	Inter-cept	*D*	*W*	*S*	S^2	*H* (re-scaled)	*O*2	*O*3	*O*4	*E*
LCSP	D + W + S + E	0.403	2.77	-0.067	0.260	-0.348						1.27
LCSP	D + W + H + S + E	0.216	2.75	-0.068	0.259	-0.321		0.259				1.27
LCSP	D + W + S + S^2 + E	0.197	2.52	-0.067	0.272	-0.124	-0.038					1.28
LCSP	D + W + H + S + S^2 + E	0.110	2.48	-0.068	0.271	-0.076	-0.042	0.272				1.28
LCSP	D + W + H + E	0.042	2.03	-0.068	0.232			0.454				1.28
LCSP	D + W + E	0.033	1.95	-0.066	0.228							1.28
SAVS	D + W + H + S + E	0.634	4.63	-0.044	0.400	-0.483		-0.567				2.71
SAVS	D + W + H + S + S^2 + E	0.235	4.65	-0.044	0.399	-0.503	0.003	-0.568				2.71
SAVS	D + W + S + E	0.096	4.50	-0.044	0.398	-0.431						2.68
SAVS	D + W + S + S^2 + E	0.035	4.49	-0.044	0.398	-0.427	-0.001					2.68
SAVS	D + W + H + E	0.000	3.45	-0.042	0.386			-0.327				2.62
SAVS	D + W + E	0.000	3.44	-0.042	0.385							2.61
SEWR	D + W + S + O	0.385	4.73	-0.041	0.518	-0.490			-1.14	-2.24	0.165	
SEWR	D + W + H + S + O	0.279	4.78	-0.041	0.519	-0.512		-0.250	-1.16	-2.25	0.148	
SEWR	D + W + S + S^2 + O	0.199	4.57	-0.041	0.528	-0.340	-0.023		-1.15	-2.24	0.166	
SEWR	D + W + H + S + S^2 + O	0.138	4.63	-0.041	0.528	-0.372	-0.022	-0.243	-1.17	-2.26	0.150	
SEWR	D + W + O	0.000	3.64	-0.040	0.491				-1.01	-2.07	0.313	
SEWR	D + W + H + O	0.000	3.64	-0.040	0.491			-0.060	-1.01	-2.07	0.312	

Table 3.5 continued.

Species	Fixed Effects	AIC Weight	Intercept	D	W	S	S^2	H rescaled	$O2$	$O3$	$O4$	E
HOLA	D + W + H + S + S^2 + E	0.341	4.98	-0.042	0.629	-0.065	-0.064	-0.394				1.54
HOLA	D + W + H + S + E	0.267	5.40	-0.043	0.622	-0.439		-0.391				1.53
HOLA	D + W + S + S^2 + E	0.216	4.87	-0.042	0.622	-0.028	-0.063					1.51
HOLA	D + W + S + E	0.175	5.29	-0.043	0.615	-0.400						1.50
HOLA	D + W + E	0.001	4.31	-0.041	0.618							1.46
HOLA	D + W + H + E	0.000	4.32	-0.041	0.622			-0.192				1.47
VESP	D + W + H + S + S^2 + E	0.702	5.10	-0.040	0.569	-0.087	-0.083	-0.515				2.38
VESP	D + W + H + S + E	0.147	5.65	-0.040	0.536	-0.615		-0.517				2.35
VESP	D + W + S + S^2 + E	0.125	4.98	-0.040	0.560	-0.048	-0.082					2.31
VESP	D + W + S + E	0.026	5.57	-0.040	0.532	-0.574						2.29
VESP	D + W + E	0.000	4.09	-0.038	0.526							2.20
VESP	D + W + H + E	0.000	4.09	-0.038	0.531			-0.304				2.23

Table 3.6: Model selection results for final model set of 6 logistic regression models of detection of recorded bird songs, estimated using all data. Fixed effects included distance from the sound source to the observer in meters (*D*), wind speed in ms^{-1} (*S*), wind speed squared (S^2), a wind index (*W*, where *W* = cos(180º - wind direction) × *S*, which had range *S* to -*S*), rescaled habitat index (*H*), observer effect (*O*), and experienced observer effect (*E*). Species are referenced by their alpha code (Bird Banding Laboratory 2016).

Species	Fixed Effects	AIC	Δ AIC	AIC Weight	Log Likelihood
BOBO	D + W + H + S + O	733.7	0.000	0.537	-357.9
BOBO	D + W + H + S + S^2 + O	735.7	1.983	0.199	-357.8
BOBO	D + W + S + O	735.8	2.046	0.193	-359.9
BOBO	D + W + S + S^2 + O	737.7	4.036	0.071	-359.9
BOBO	D + W + O	753.8	20.09	0.000	-369.9
BOBO	D + W + H + O	754.1	20.42	0.000	-369.1
DICK	D + W + H + S + S^2 + E	745.4	0.000	0.704	-364.7
DICK	D + W + H + S + E	748.3	2.911	0.164	-367.2
DICK	D + W + S + S^2 + E	749.2	3.768	0.107	-367.6
DICK	D + W + S + E	752.2	6.752	0.024	-370.1
DICK	D + W + H + E	768.8	23.33	0.000	-378.4
DICK	D + W + E	769.1	23.64	0.000	-379.5
EAME	D + W + H + S + S^2 + E	566.9	0.000	0.850	-275.4
EAME	D + W + S + S^2 + E	571.2	4.331	0.097	-278.6
EAME	D + W + H + S + E	572.7	5.850	0.046	-279.4
EAME	D + W + S + E	576.6	9.769	0.006	-282.3
EAME	D + W + H + E	583.6	16.76	0.000	-285.8
EAME	D + W + E	585.7	18.84	0.000	-287.8
GRSP	D + W + S + S^2 + O	692.8	0.000	0.517	-337.4
GRSP	D + W + H + S + S^2 + O	694.7	1.939	0.196	-337.4
GRSP	D + W + S + O	694.8	2.067	0.184	-339.4
GRSP	D + W + H + S + O	696.6	3.861	0.075	-339.3
GRSP	D + W + O	699.2	6.441	0.021	-342.6
GRSP	D + W + H + O	701.2	8.425	0.008	-342.6

Table 3.6 continued.

Species	Fixed Effects	AIC	Δ AIC	AIC Weight	Log Likelihood
HOLA	$D + W + H + S + S^2 + E$	749.6	0.000	0.341	-366.8
HOLA	D + W + H + S + E	750.1	0.489	0.267	-368.1
HOLA	$D + W + S + S^2 + E$	750.6	0.914	0.216	-368.3
HOLA	D + W + S + E	751.0	1.337	0.175	-369.5
HOLA	D + W + E	762.2	12.58	0.001	-376.1
HOLA	D + W + H + E	763.5	13.87	0.000	-375.8
LCSP	D + W + S + E	522.7	0.000	0.403	-255.3
LCSP	D + W + H + S + E	523.9	1.248	0.216	-255.0
LCSP	$D + W + S + S^2 + E$	524.1	1.432	0.197	-255.1
LCSP	$D + W + H + S + S^2 + E$	525.3	2.598	0.110	-254.6
LCSP	D + W + H + E	527.2	4.526	0.042	-257.6
LCSP	D + W + E	527.7	5.009	0.033	-258.8
SAVS	D + W + H + S + E	744.0	0.000	0.634	-365.0
SAVS	$D + W + H + S + S^2 + E$	746.0	1.989	0.235	-365.0
SAVS	D + W + S + E	747.8	3.772	0.096	-367.9
SAVS	$D + W + S + S^2 + E$	749.8	5.772	0.035	-367.9
SAVS	D + W + H + E	763.0	19.02	0.000	-375.5
SAVS	D + W + E	763.0	19.05	0.000	-376.5
SEWR	D + W + S + O	757.2	0.000	0.385	-370.6
SEWR	D + W + H + S + O	757.8	0.645	0.279	-369.9
SEWR	$D + W + S + S^2 + O$	758.5	1.320	0.199	-370.3
SEWR	$D + W + H + S + S^2 + O$	759.2	2.051	0.138	-369.6
SEWR	D + W + O	778.0	20.78	0.000	-382.0
SEWR	D + W + H + O	779.9	22.70	0.000	-381.9
HESP	$D + W + H + S + S^2 + E$	719.5	0.000	0.708	-351.7
HESP	D + W + H + S + E	721.9	2.452	0.208	-354.0
HESP	$D + W + S + S^2 + E$	724.1	4.571	0.072	-355.0
HESP	D + W + S + E	727.7	8.211	0.012	-357.8
HESP	D + W + H + E	739.5	20.02	0.000	-363.8
HESP	D + W + E	741.2	21.69	0.000	-365.6

Table 3.6 continued.

Species	Fixed Effects	AIC	Δ AIC	AIC Weight	Log Likelihood
VESP	D + W + H + S + S^2 + E	698.7	0.000	0.702	-341.3
VESP	D + W + H + S + E	701.8	3.125	0.147	-343.9
VESP	D + W + S + S^2 + E	702.1	3.453	0.125	-344.1
VESP	D + W + S + E	705.2	6.576	0.026	-346.6
VESP	D + W + E	732.3	33.64	0.000	-361.1
VESP	D + W + H + E	732.3	33.66	0.000	-360.2

Table 3.7: The perceptible area around the observer, predicted using fixed effects estimated using all data. Wind speeds 0, 1, 2.5, and 4.5 are associated with Beaufort numbers 0, 1, 2, and 3, respectively (Patuxent Wildlife Research Center 1998).

Species	Wind Speed (ms^{-1})	Wind Speed (mph)	Area (m^2)	Area (ha)
EAME	0	0	52986	5.30
SAVS	0	0	34780	3.48
GRSP	0	0	9683	0.97
LCSP	0	0	4739	0.47
VESP	0	0	51371	5.14
HOLA	0	0	45361	4.54
HESP	0	0	29417	2.94
SEWR	0	0	40197	4.02
BOBO	0	0	46331	4.63
DICK	0	0	37954	3.80
EAME	1	2.2	55972	5.60
SAVS	1	2.2	28043	2.80
GRSP	1	2.2	9841	0.98
LCSP	1	2.2	3891	0.39
VESP	1	2.2	47017	4.70
HOLA	1	2.2	41363	4.14
HESP	1	2.2	27097	2.71
SEWR	1	2.2	32990	3.30
BOBO	1	2.2	37766	3.78
DICK	1	2.2	34768	3.48
EAME	2.5	5.6	53111	5.31
SAVS	2.5	5.6	19881	1.99
GRSP	2.5	5.6	9021	0.90
LCSP	2.5	5.6	2753	0.28
VESP	2.5	5.6	37599	3.76
HOLA	2.5	5.6	35004	3.50
HESP	2.5	5.6	22646	2.26
SEWR	2.5	5.6	24199	2.42
BOBO	2.5	5.6	27348	2.73
DICK	2.5	5.6	28191	2.82

Table 3.7 continued.

Species	Wind Speed (ms^{-1})	Wind Speed (mph)	Area (m^2)	Area (ha)
EAME	4.5	10.1	38377	3.84
SAVS	4.5	10.1	12577	1.26
GRSP	4.5	10.1	6686	0.67
LCSP	4.5	10.1	1617	0.16
VESP	4.5	10.1	23492	2.35
HOLA	4.5	10.1	27224	2.72
HESP	4.5	10.1	16282	1.63
SEWR	4.5	10.1	16606	1.66
BOBO	4.5	10.1	17881	1.79
DICK	4.5	10.1	18277	1.83

Figure 1.1: Singing probability (the probability that a bird sings at least once in a given number of minutes) for different values of P(S|S), the probability that a bird sings given that it sang in the previous interval. Singing probability is shown as a function of P(S|S) and q_1, the average proportion of total birds singing at any given interval. I used P(S|S) = 0.98 for all scenarios; P(S|S) = 0.92 and P(S|S) = 0.99 are shown for comparison. For P(S|S) = 0.92, birds transition too frequently, resulting in singing probability ≈ 1 if > 10% of birds sing in any given interval. For P(S|S) = 0.99, too few birds transition between modes.

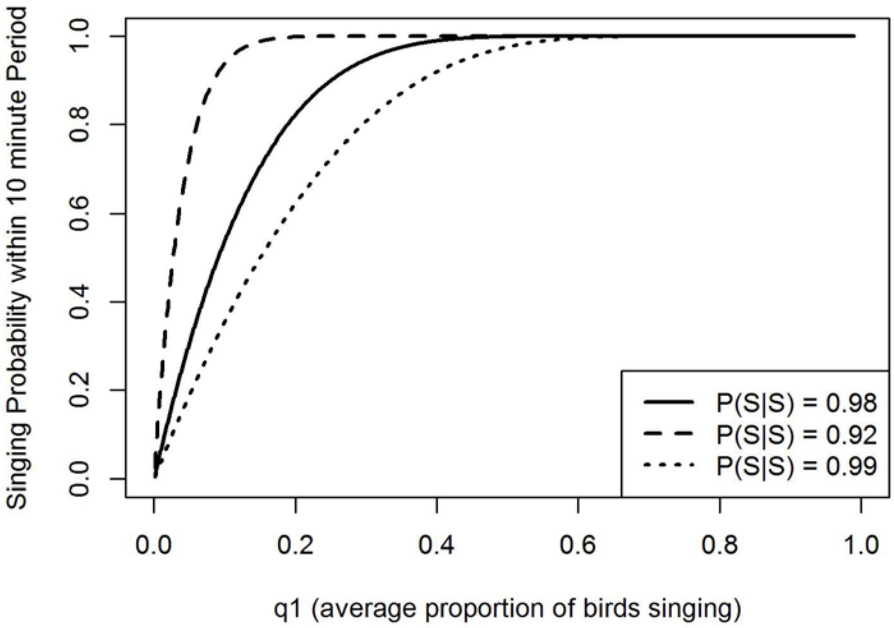


Figure 1.2: Maximum daily song rate (*MaxDailySongRate.yir*) for the black-throated blue warbler (BTBW) scenario, simulated as a function of ordinal date. Song rate was highest when birds arrived on breeding grounds (day 122) and declined later in the season (Holmes et al. 2005). The BTBW scenario used the ordinal date range 150-160, so *MaxDailySongRate.yir* = 0.9.

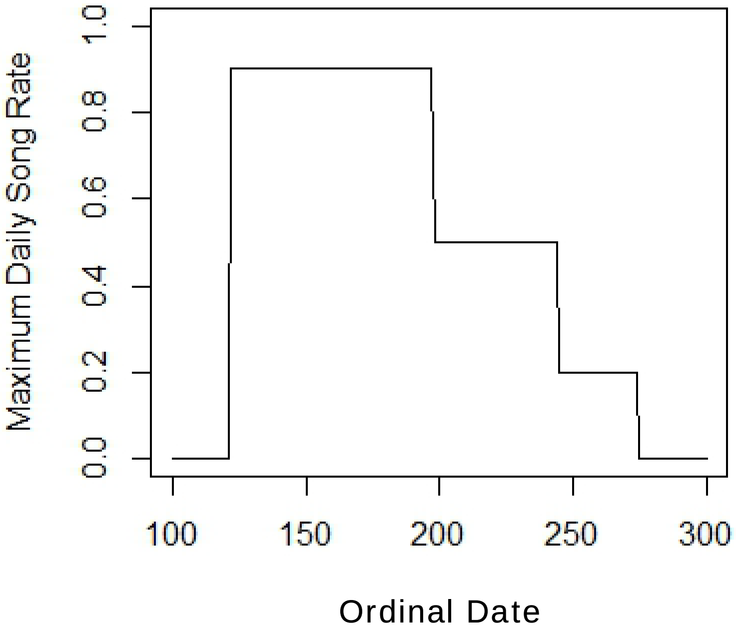


Figure 1.3: The time of day weight (*SongWeight.yijrk*) as a function of time (minutes after sunrise) for the black-throated blue warbler (BTBW) scenario. Interval-specific singing probability (*PrSing.yijrk*), the product of the maximum daily song rate (*MaxDailySongRate.yir*) and a weight due to the time of day (*SongWeight.yijrk*). The time of day weight was based on information from Breeding Bird Surveys (P. Blancher, Environment Canada, personal communication). The vertical grey lines represent the planned beginning and end for surveys within a day (actual starting times included variation that could exceed those limits).

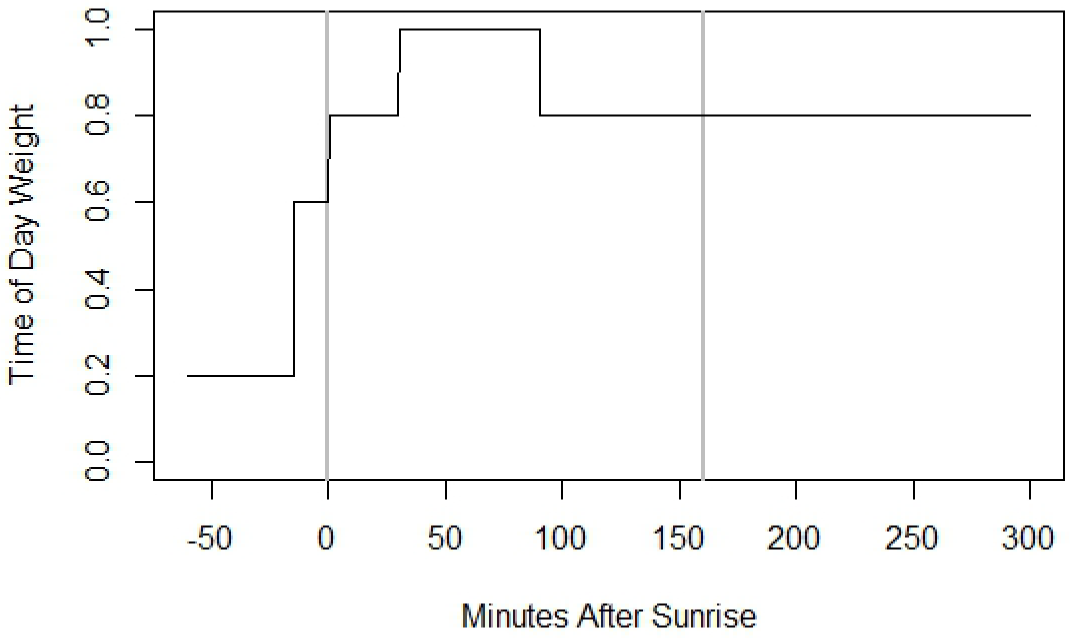


Figure 1.4: Observer-specific perceptibility of a single song (a) and perceptibility given 9 opportunities for detection (b), as a function of distance to observer for the black-throated blue warbler (BTBW) scenario. Mean number of songs produced by birds that sang was 9.06 for double-observer surveys (SD = 6.37 songs). The dashed lines represent perceptibility for the Observers 1 and 2 when ambient noise was present during surveys; the solid lines represent perceptibility during surveys without noise. For the BTBW scenario, presence of noise was replication-specific, with probability of presence *PrNoise* = 0.15.

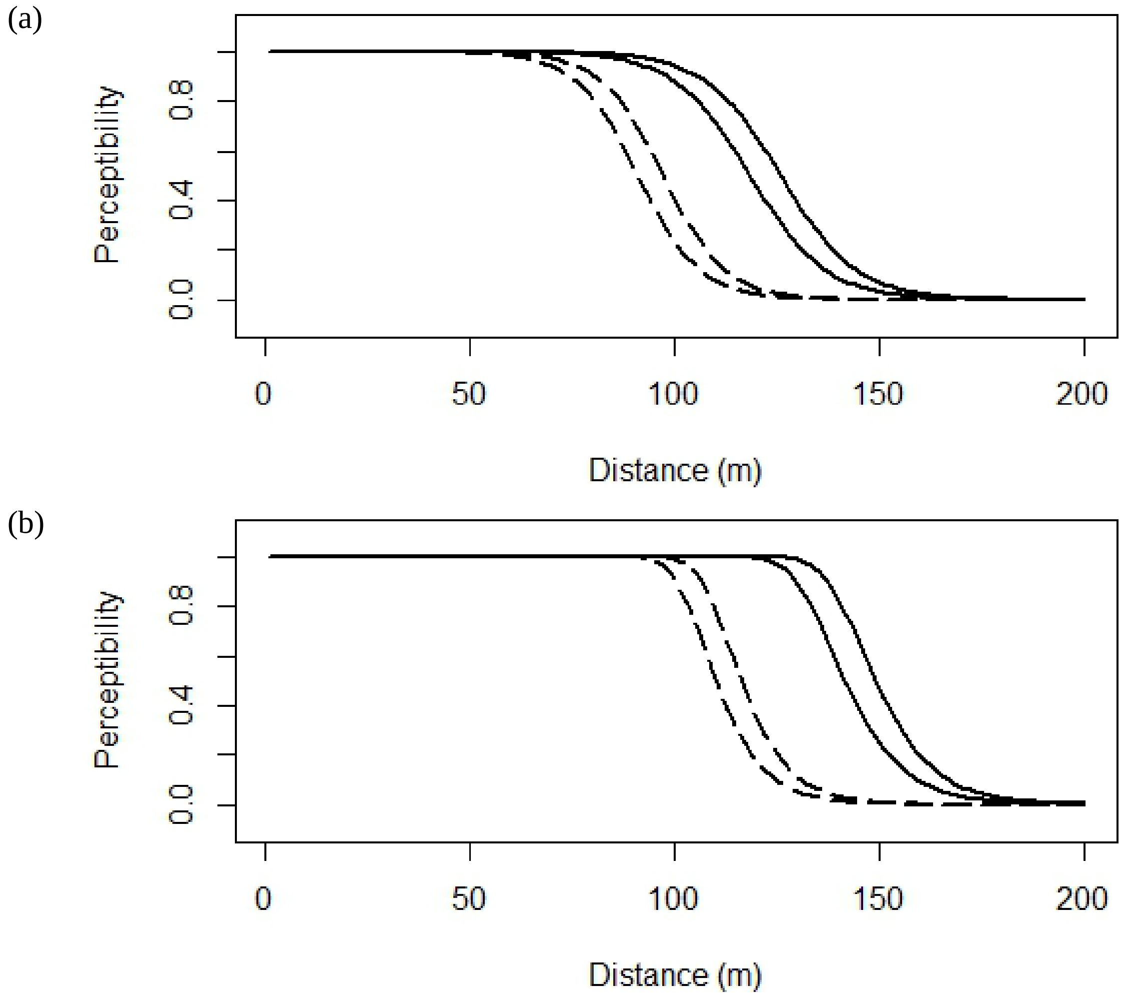


Figure 1.5: Observer-estimated distance (*ObsEstimatedDistance.yijrko*) for the black-throated blue warbler (BTBW) scenario, simulated stochastically as the sum of the true distance (*Distance.yijrk*) and a value for observer error. I used error estimates (mean error = 7.6 m, SD = 21.4 m) for observer-estimated distance from Alldredge et al. (2007c), assumed error was 0 at 0 m, and assumed error changed as a linear function between 0 m and the mean of distances Alldredge et al. investigated (62.3 m). Mean observer-estimated distance (solid line) ± standard deviation (dashed lines) are shown.

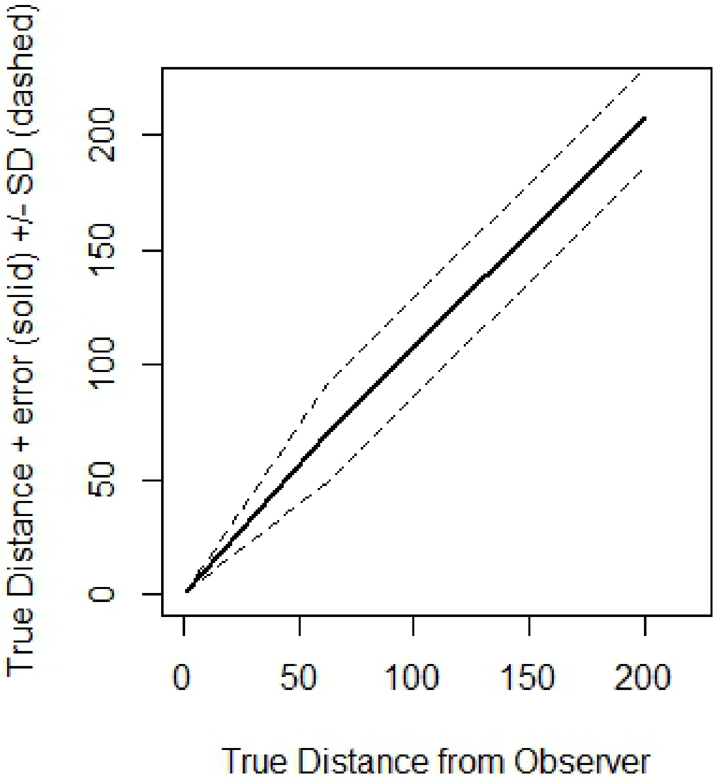


Figure 1.6 (a): Comparison of density estimates to true densities for the simple counts survey type for the black-throated blue warbler (BTBW) scenario. Estimates are from 30 simulated years. Density was calculated for area A around the observer, where $A = \pi \times$ survey radius^2. Estimated density is shown compared to true density values D_s, the density of birds with territories that overlapped the survey radius ($D_s = N^* p_s / A$); D_p, the density of birds present at the beginning of the survey ($D_p = N^* p_s p_p / A$); and D_a, the density of available birds present at the beginning of the survey ($D_a = N^* p_s p_p p_a / A$). Perfect estimates would fall on the y=x line (shown for reference).

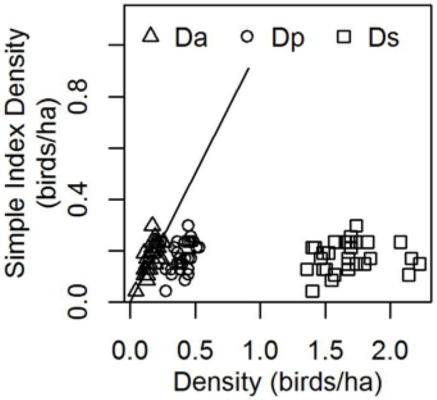
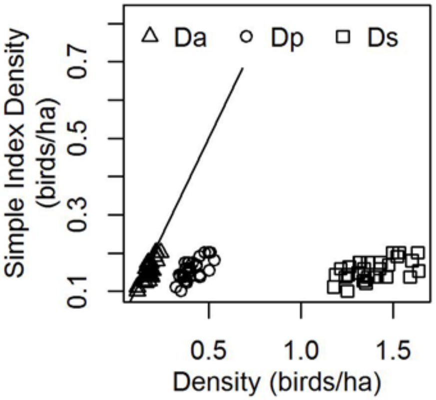
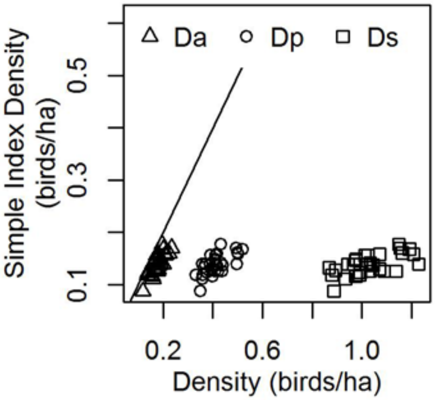
	50-m Survey Radius	100-m Survey Radius	150-m Survey Radius
Simple Index Density Estimator			

Figure 1.6 (b): Comparison of density estimates to true densities for the double-observer survey type for the black-throated blue warbler (BTBW) scenario. Estimates are from 30 simulated years. Density was calculated for area A around the observer, where $A = \pi \times \text{survey radius}^2$. True density values are D_s, the density of birds with territories that overlapped the survey radius ($D_s = N^* p_s / A$); D_p, the density of birds present at the beginning of the survey ($D_p = N^* p_s p_p / A$); and D_a, the density of available birds present at the beginning of the survey ($D_a = N^* p_s p_p p_a / A$). Perfect estimates would fall on the y = x line (shown for reference).

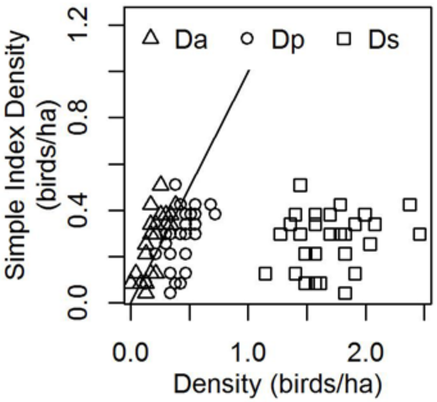
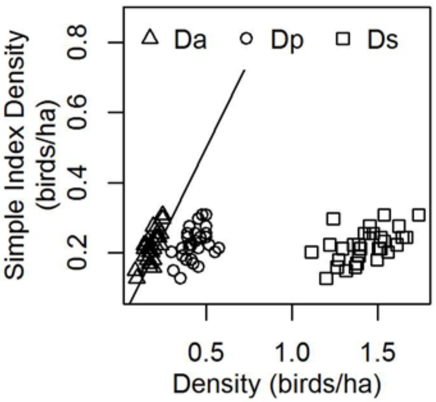
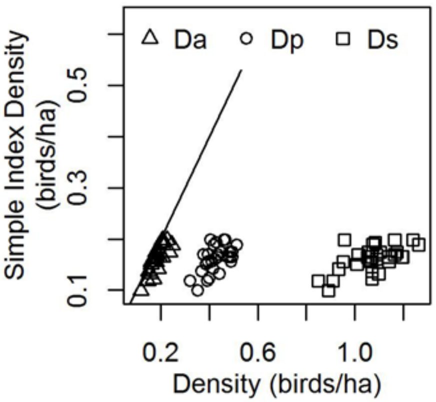
	50-m Survey Radius	100-m Survey Radius	150-m Survey Radius
Simple Index Density Estimator			

Figure 1.6 (b) continued.

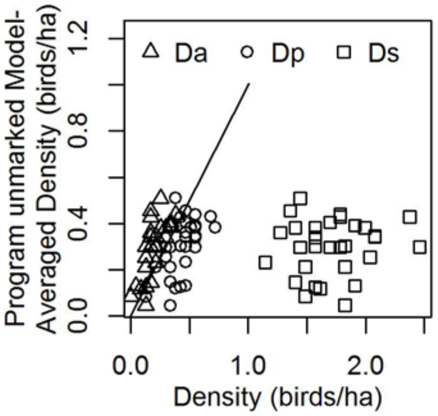
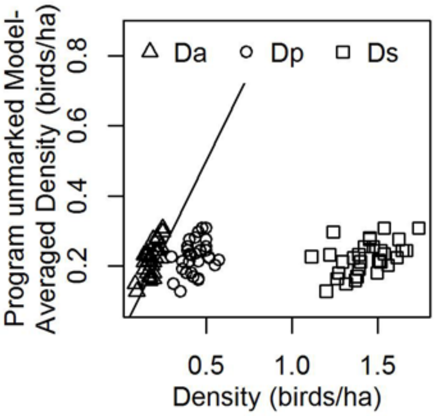
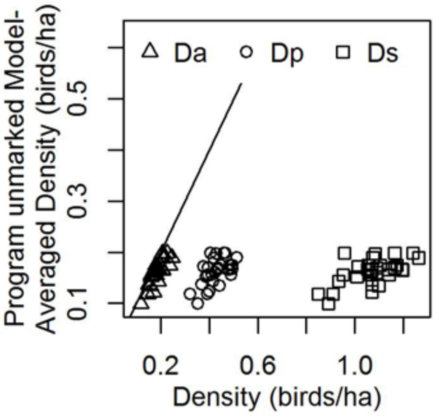
	50-m Survey Radius	100-m Survey Radius	150-m Survey Radius
Model-Averaged Adjusted Density			

Figure 1.6 (c): Comparison of density estimates to true densities for the removal survey type for the black-throated blue warbler (BTBW) scenario. Estimates are from 30 simulated years. Density was calculated for area A around the observer, where $A = \pi \times$ survey radius^2. True density values are D_s, the density of birds with territories that overlapped the survey radius ($D_s = N^* p_s / A$); D_p, the density of birds present at the beginning of the survey ($D_p = N^* p_s p_p / A$); and D_a, the density of available birds present at the beginning of the survey ($D_a = N^* p_s p_p p_a / A$). Perfect estimates would fall on the y = x line (shown for reference).

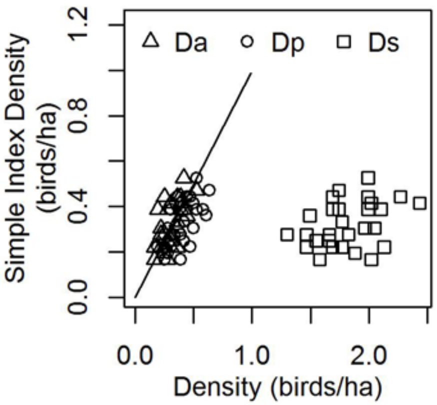
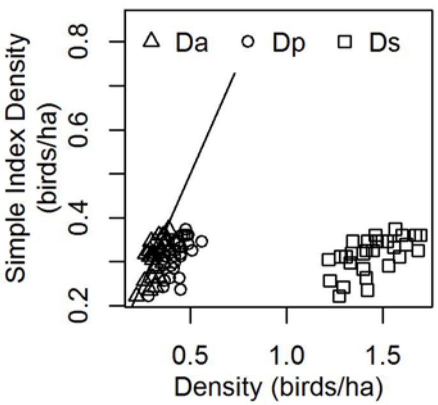
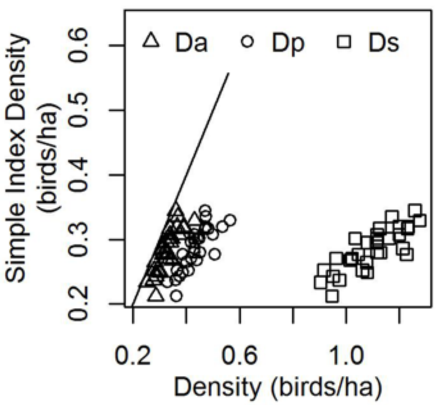
	50-m Survey Radius	100-m Survey Radius	150-m Survey Radius
Simple Index Density Estimator			

Figure 1.6 (c) continued.

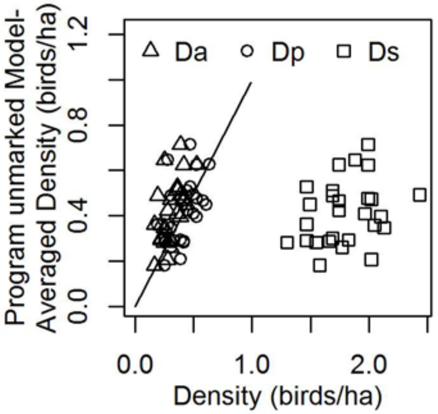
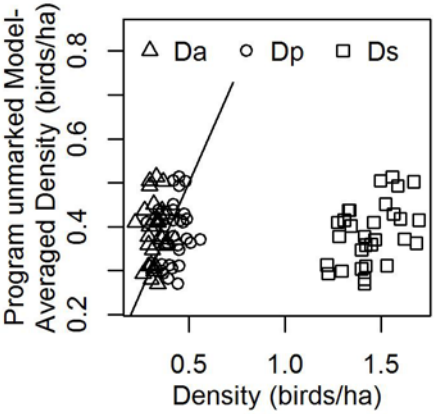
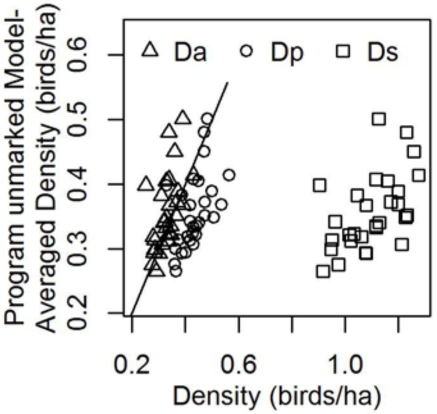
	50-m Survey Radius	100-m Survey Radius	150-m Survey Radius
Model-Averaged Adjusted Density			

Figure 1.6 (d): Comparison of density estimates to true densities for the replicated counts survey type for the black-throated blue warbler (BTBW) scenario. Estimates are from 30 simulated years. Density was calculated for area *A* around the observer, where $A = \pi \times$ survey radius^2. True density values are D_s, the density of birds with territories that overlapped the survey radius ($D_s = N^* p_s / A$); D_p, the density of birds present at the beginning of the survey ($D_p = N^* p_s p_p / A$;, and D_a, the density of available birds present at the beginning of the survey ($D_a = N^* p_s p_p p_a / A$). Perfect estimates would fall on the y = x line (shown for reference).

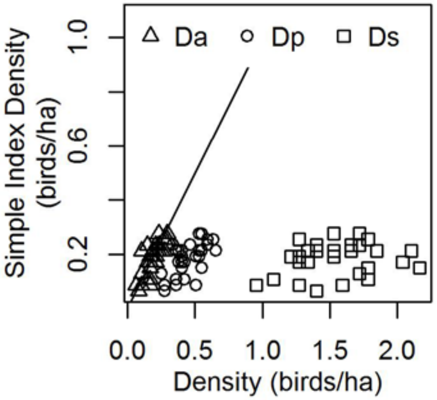
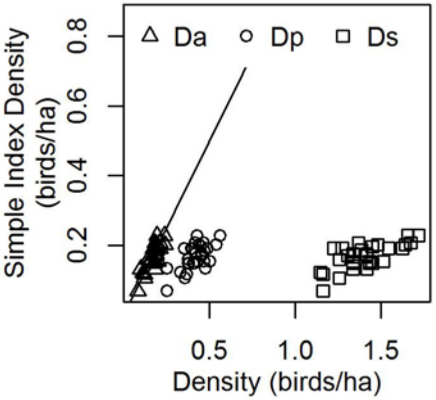
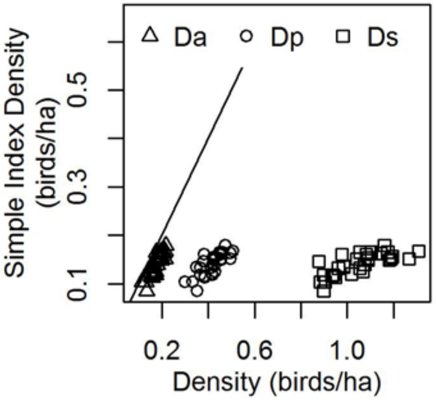
	50 -m Survey Radius	100-m Survey Radius	150-m Survey Radius
Simple Index Density Estimator			

Figure 1.6 (d) continued.

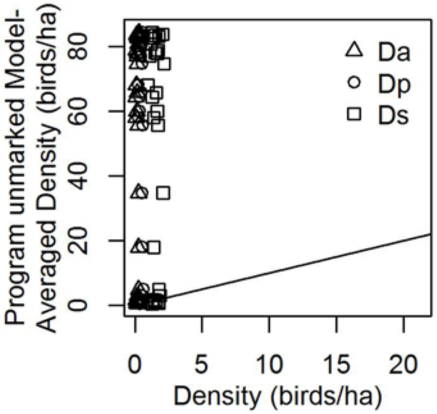
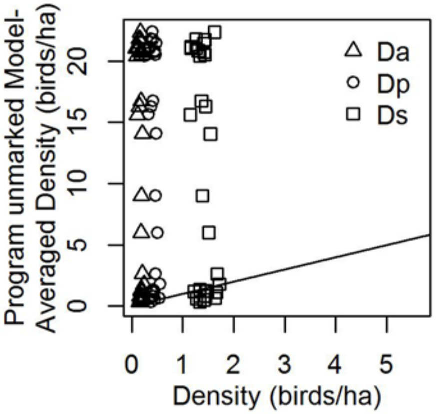
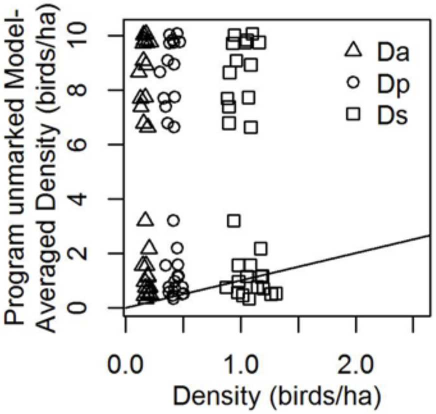
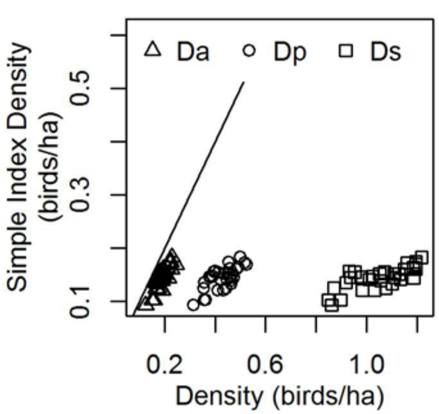
	50 -m Survey Radius	100-m Survey Radius	150-m Survey Radius
Model-Averaged Adjusted Density			

Figure 1.6 (e): Comparison of density estimates to true densities for the distance sampling survey type for the black-throated blue warbler (BTBW) scenario. Data were right-truncated, eliminating the most distant 10% of observations (Buckland et al. 2001). Estimates are from 30 simulated years. Density was calculated for area A around the observer, where $A = \pi \times$ truncation distance^2 (mean truncation distance 158 m). True density values are D_s, the density of birds with territories that overlapped the survey radius ($D_s = N^* p_s / A$); D_p, the density of birds present at the beginning of the survey ($D_p = N^* p_s p_p / A$); and D_a, the density of available birds present at the beginning of the survey ($D_a = N^* p_s p_p p_a / A$). Estimators were evaluated for surveys with radius 50 m, 100 m, and 150 m. Perfect estimates would fall on the y = x line (shown for reference).

Simple Index Density Estimator



Model-Averaged Adjusted Density

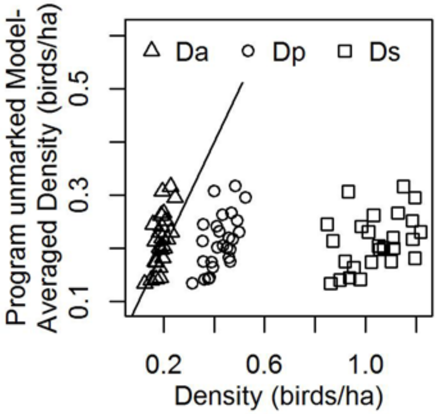


Figure 1.7 (a): Comparisons of density estimates (birds/ha) from index methods to the true density of birds present during the survey (D_p) for the simple counts survey type for the black-throated blue warbler (BTBW) scenario. The simple index estimator used observer-estimated distance to determine if birds were within the survey radius. The simple index estimator with perfect observer estimation of distance assumed observers knew distance to birds exactly.

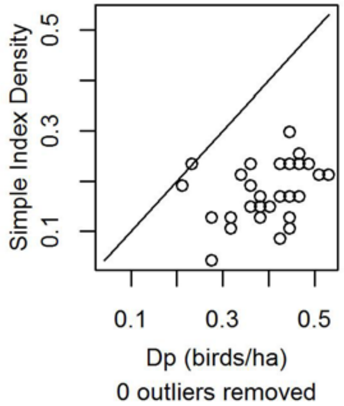
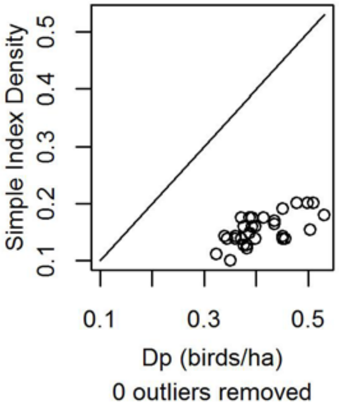
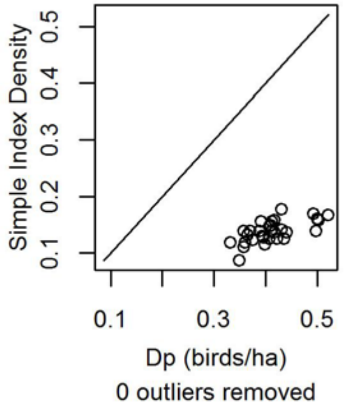
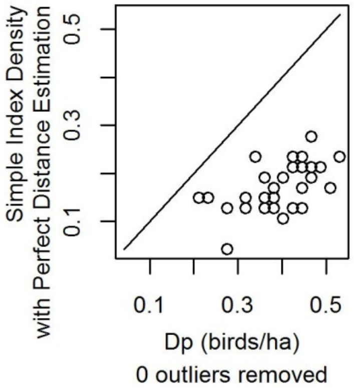
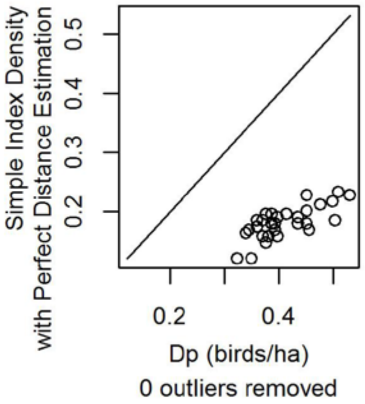
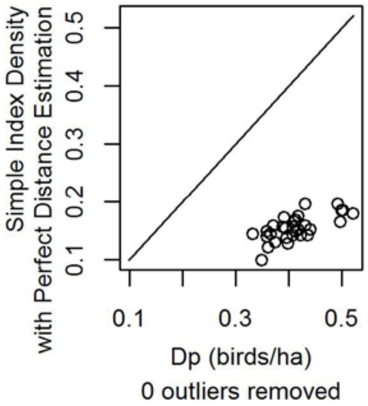
	50-m Survey Radius	100-m Survey Radius	150-m Survey Radius
Simple Index Density Estimator			
Simple Index Density Estimator with Perfect Observer Estimation of Distance			

Figure 1.7 (b): Comparisons of estimates (birds/ha) from index and adjustment methods to the true density of birds present during the survey (D_p) for the double-observer survey type for the black-throated blue warbler (BTBW) scenario. Estimates using the Nichols et al. (2000) double-observer analysis method were similar for independent and dependent observer methods; only results from the independent method are shown for simplicity.

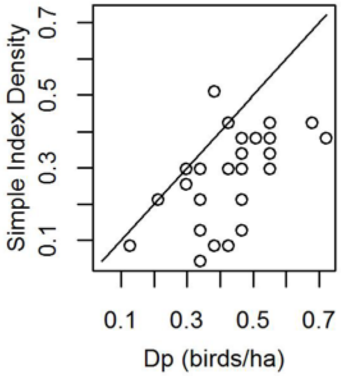
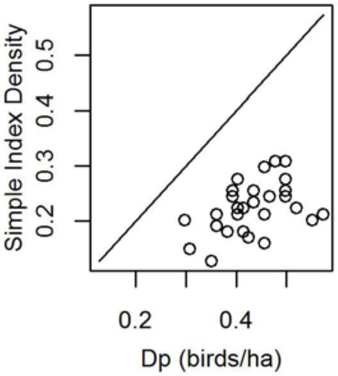
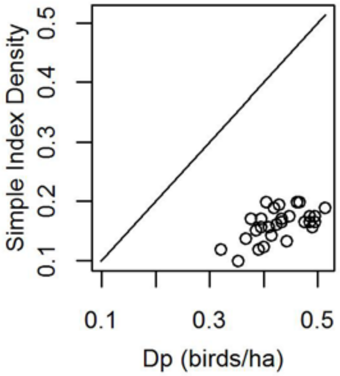
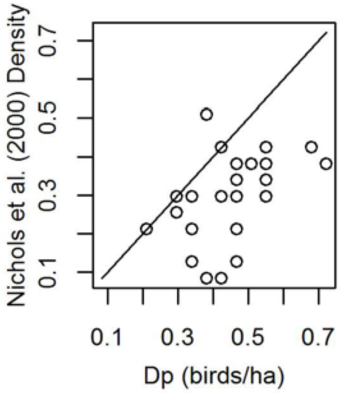
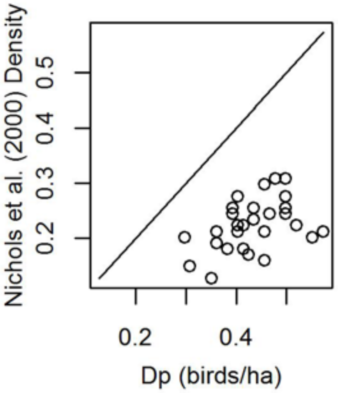
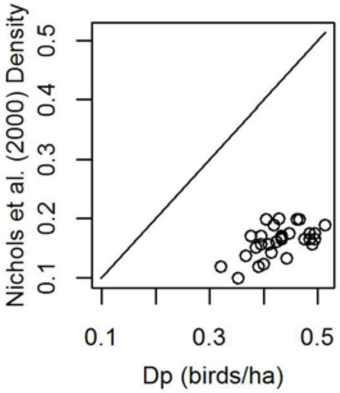
	50-m Survey Radius	100-m Survey Radius	150-m Survey Radius
Simple Index Density Estimator	 0 outliers removed	 0 outliers removed	 1 outlier removed
Adjusted Density Estimator (Dependent Observers)	 2 outliers removed	 0 outliers removed	 1 outlier removed

Figure 1.7 (b) continued.

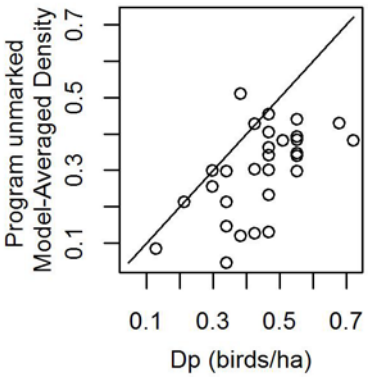
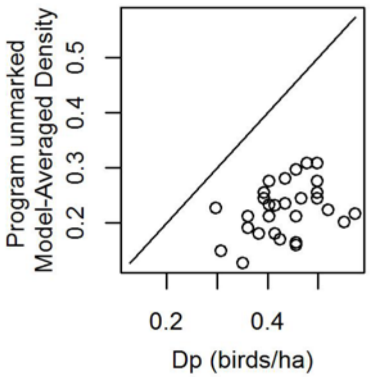
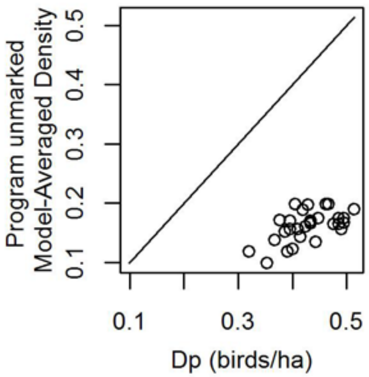
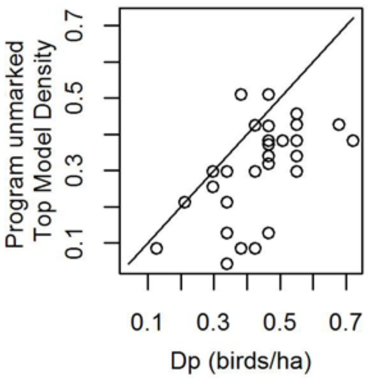
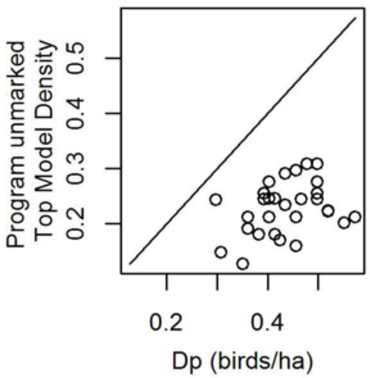
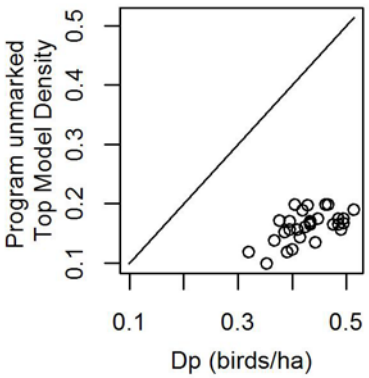
	50-m Survey Radius	100-m Survey Radius	150-m Survey Radius
Model-Averaged Adjusted Density (Independent Observers)	 0 outliers removed	 0 outliers removed	 1 outlier removed
Top Model Adjusted Density (Independent Observers)	 0 outliers removed	 0 outliers removed	 1 outlier removed

Figure 1.7 (c): Comparisons of estimates (birds/ha) from index and adjustment methods to the true density of birds present during the survey (D_p) for the removal survey type for the black-throated blue warbler (BTBW) scenario.

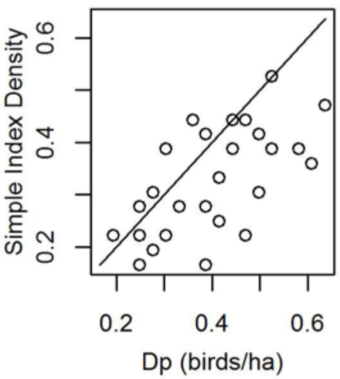
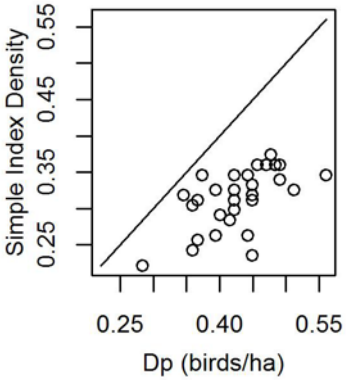
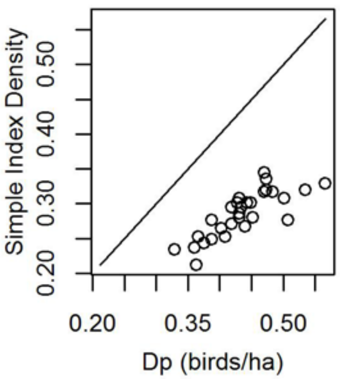
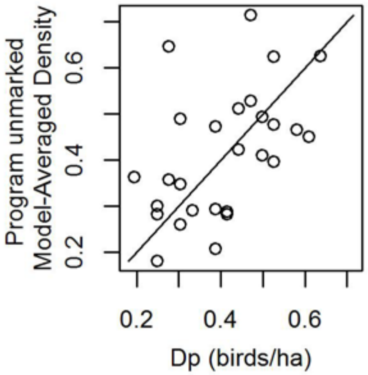
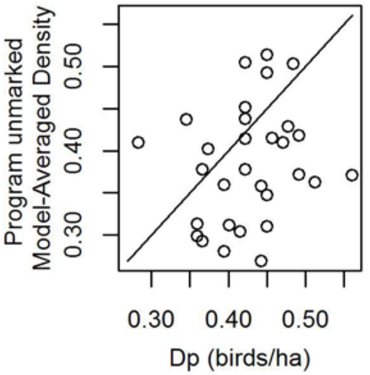
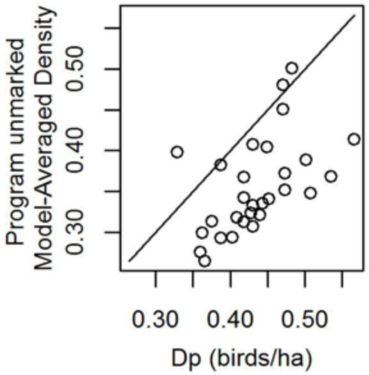
	50-m Survey Radius	100-m Survey Radius	150-m Survey Radius
Simple Index Density Estimator	 0 outliers removed	 0 outliers removed	 0 outliers removed
Model-Averaged Adjusted Density	 3 outliers removed	 0 outliers removed	 1 outlier removed

Figure 1.7 (c): continued.

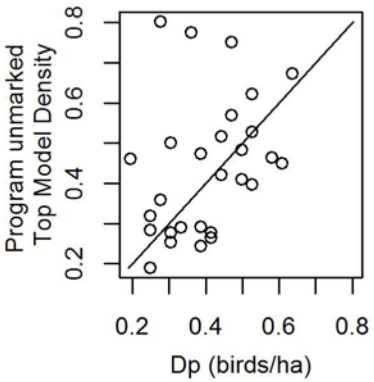
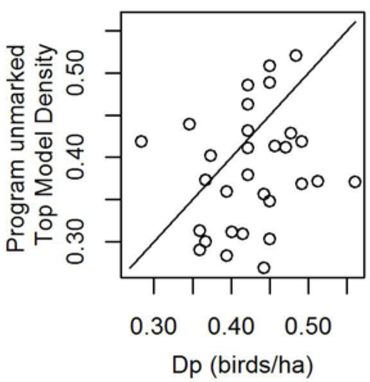
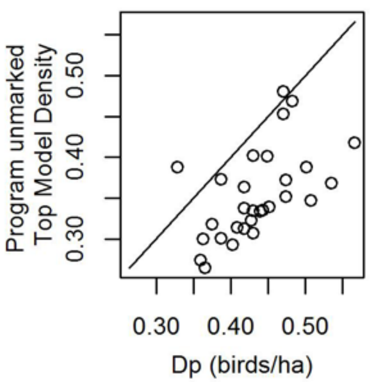
	50-m Survey Radius	100-m Survey Radius	150-m Survey Radius
Top Model Adjusted Density	 2 outliers removed	 0 outliers removed	 1 outlier removed

Figure 1.7 (d): Comparisons of estimates (birds/ha) from index and adjustment methods to the true density of birds present during the survey (D_p) for the replicated counts survey type for the black-throated blue warbler (BTBW) scenario. Maximum count density was the sum of all maximum counts for each site (among the 3 counts within a year), divided by the area surveyed. Bounded count density was the sum of the bounded counts (twice the maximum count, minus the second largest count, Johnson et al. 2007), divided by the area surveyed.

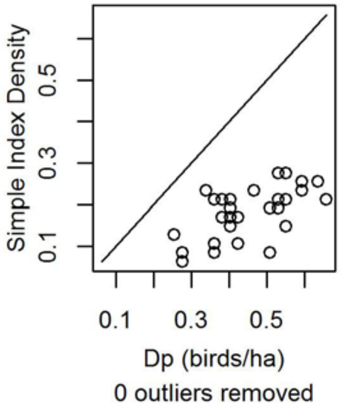
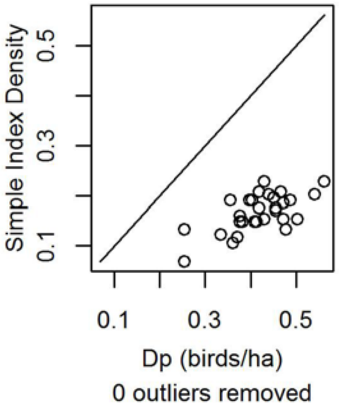
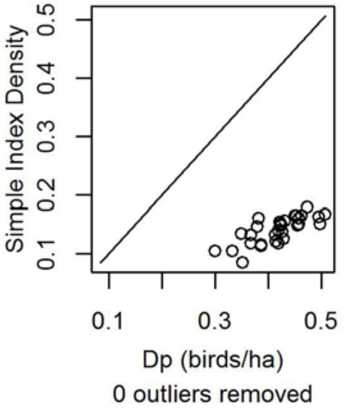
	50-m Survey Radius	100-m Survey Radius	150-m Survey Radius
Simple Index Density Estimator			

Figure 1.7 (d) continued.

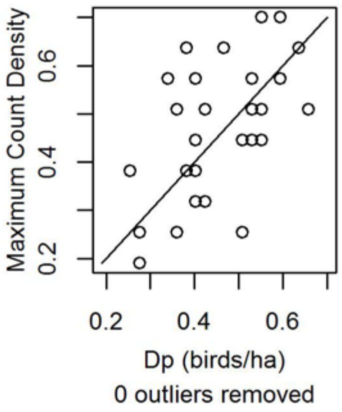
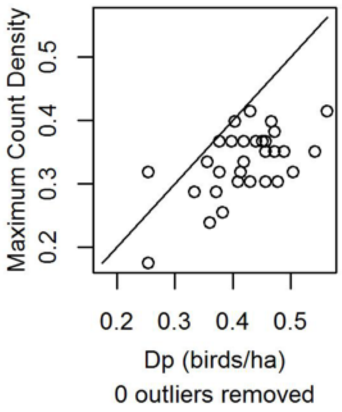
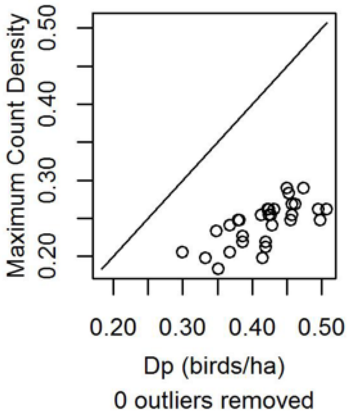
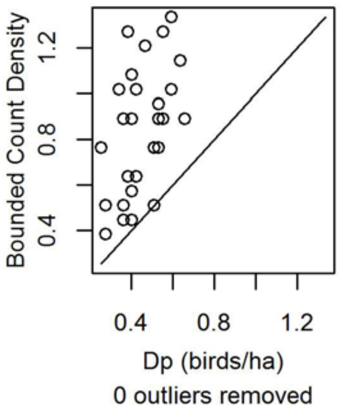
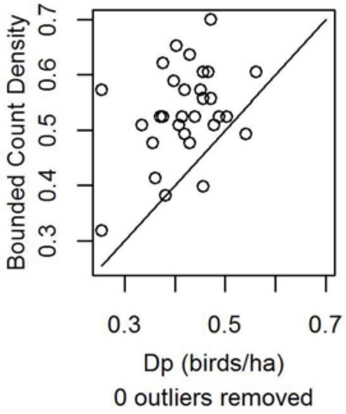
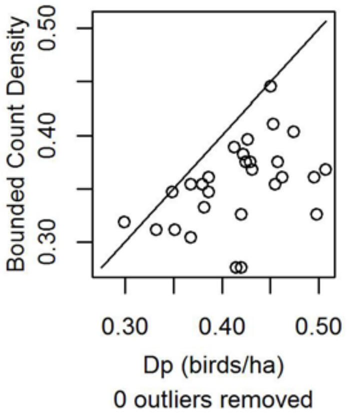
	50-m Survey Radius	100-m Survey Radius	150-m Survey Radius
Maximum Count Density Estimator			
Bounded Count Density Estimator			

Figure 1.7 (d) continued.

50-m Survey Radius

100-m Survey Radius

150-m Survey Radius

Model-Averaged Adjusted Density

Top Model Adjusted Density

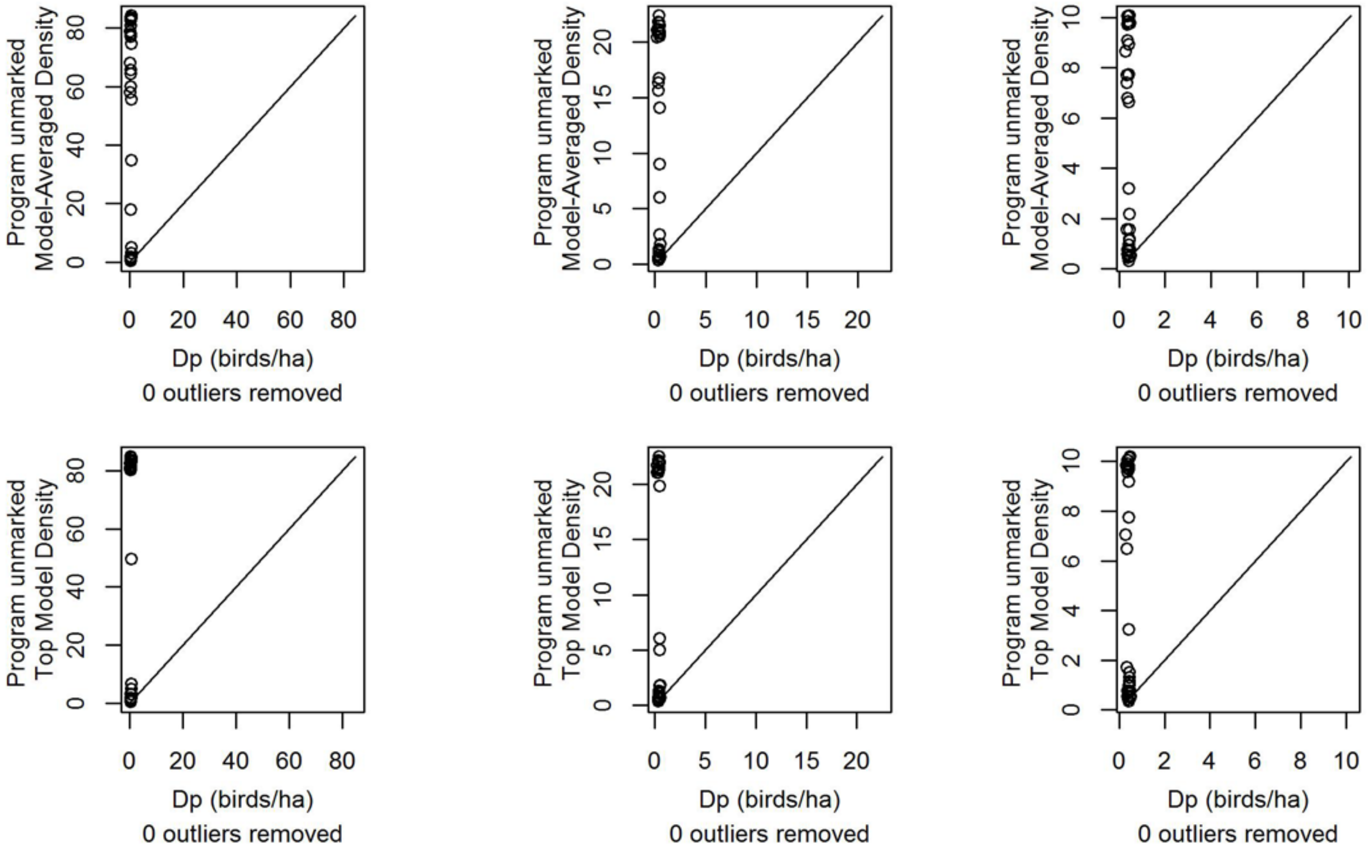
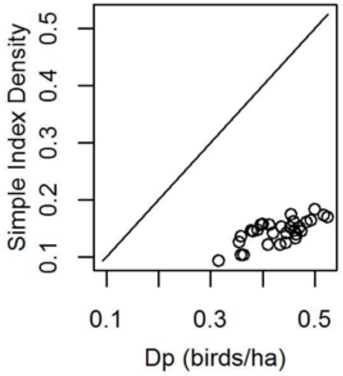


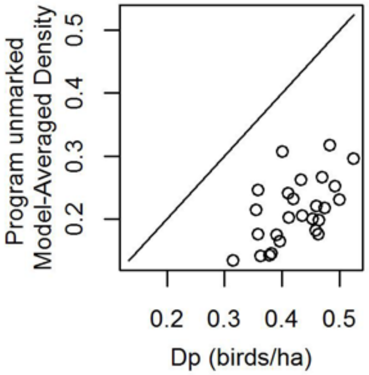
Figure 1.7 (e): Comparisons of estimates (birds/ha) from index and adjustment methods to the true density of birds present during the survey (D_p) for the distance sampling survey type for the black-throated blue warbler (BTBW) scenario. Data were right-truncated, eliminating the most distant 10% of observations (Buckland et al. 2001).

Simple Index Density Estimate



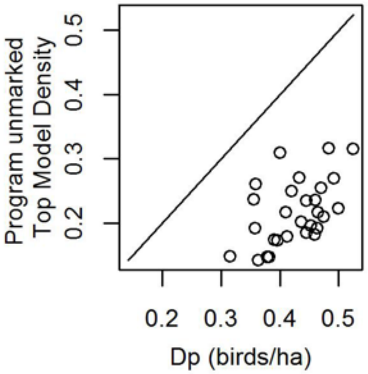
0 outliers removed

Model-Averaged Adjusted Density (Program unmarked)



4 outliers removed

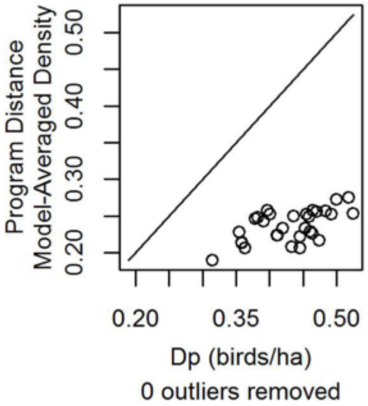
Top Model Adjusted Density (Program unmarked)



2 outliers removed

Figure 1.7 (e): continued.

Model-Averaged
Adjusted Density
(Program Distance)



Top Model
Adjusted Density
(Program Distance)

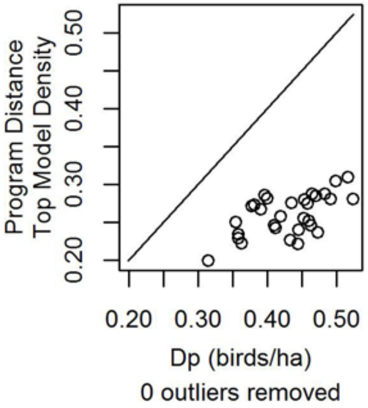


Figure 1.8: Adjusted density estimates from N-mixture models for the replicated counts survey type, as a function of estimated detection (*p*), for the black-throated blue warbler (BTBW) scenario. Density estimates were inflated when detection was estimated < 0.06.

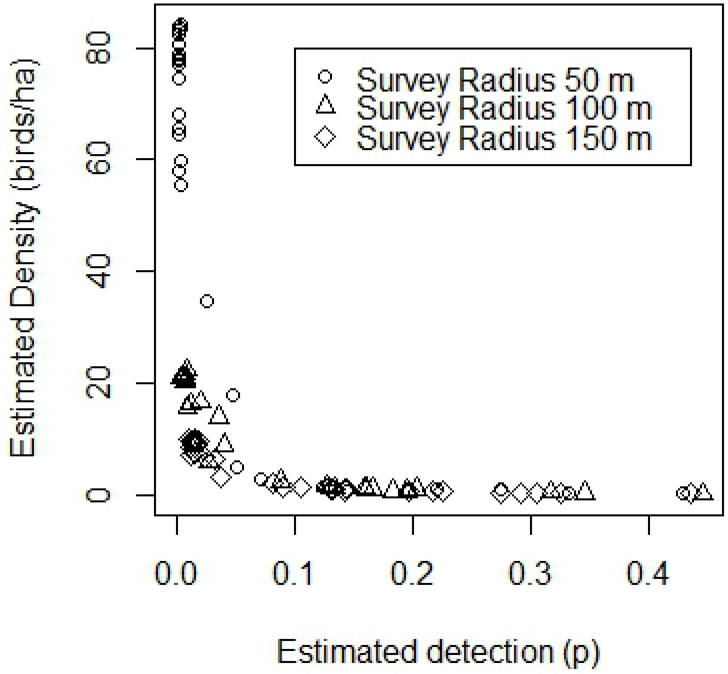


Figure 2.1:Bird detections by time of day (as reflected by stop number), as seen on the Breeding Bird Survey (BBS; P. Blancher, Environment Canada, personal communication) for 4 species: American crow (*Corvus brachyrhynchos*), mourning dove (*Zenaida macroura*), ovenbird (*Seiurus aurocapilla*), and yellow-bellied sapsucker (*Sphyrapicus varius*). Species were chosen because they displayed a similar pattern of detection. Because detections are summed across many BBS routes, years, and observers, I assumed that differences in detection are largely due to changes in availability. Lines are 6th order polynomial regressions fit to the data (Rosenberg and Blancher 2005).

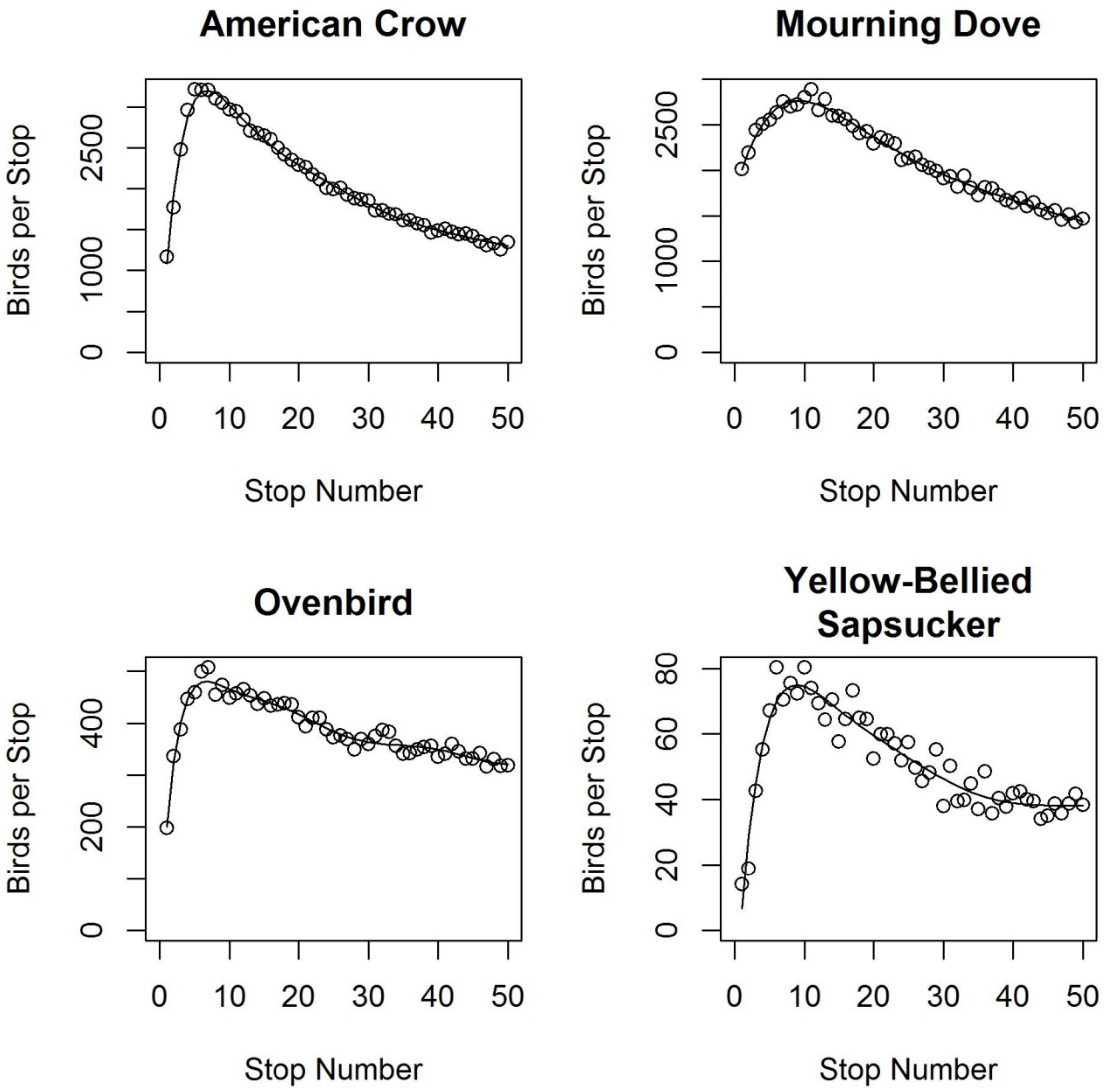


Figure 2.2: Availability for high and low availability scenarios. Availability was modeled as PrSing5 (probability that a bird sings at least once in a 5-minute period) and was the product of the maximum daily song rate and the time-of-day weight. Solid line represents high-availability species, maximum daily song rate = 0.99, mean PrSing5 = 0.80. Dashed line represents low-availability species, maximum daily song rate = 0.56, mean PrSing5 = 0.45. The shaded area represents approximate survey period in a simulated survey (actual simulations included variation in survey period). Shape of the curves was based on 4 species provided by P. Blancher (Environment Canada, personal communication), see Figure 2.1.

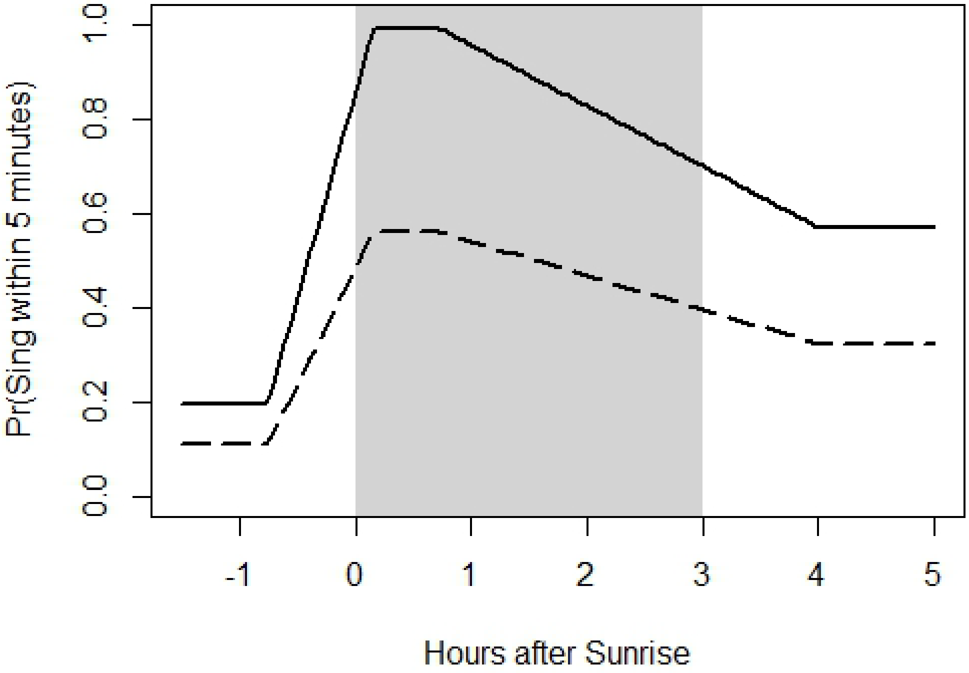


Figure 2.3 – Perceptibility (p_d) for high perceptibility (top) and low perceptibility (bottom) scenarios, as a function of distance from the observer. The range of perceptibility for an average observer (thick lines) varied continuously between optimal observation conditions (thick dashed line) and the worst observation conditions (thick solid line). Thin lines indicate the range of observer ability between best and worst observers. For high perceptibility scenarios (top), curves were based on information from easy-to-detect species, such as brown thrashers (*Toxostoma rufum*). For low perceptibility scenarios (bottom), curves were based on hard-to-detect species such as black-and-white warblers (*Mniotilta varia*).

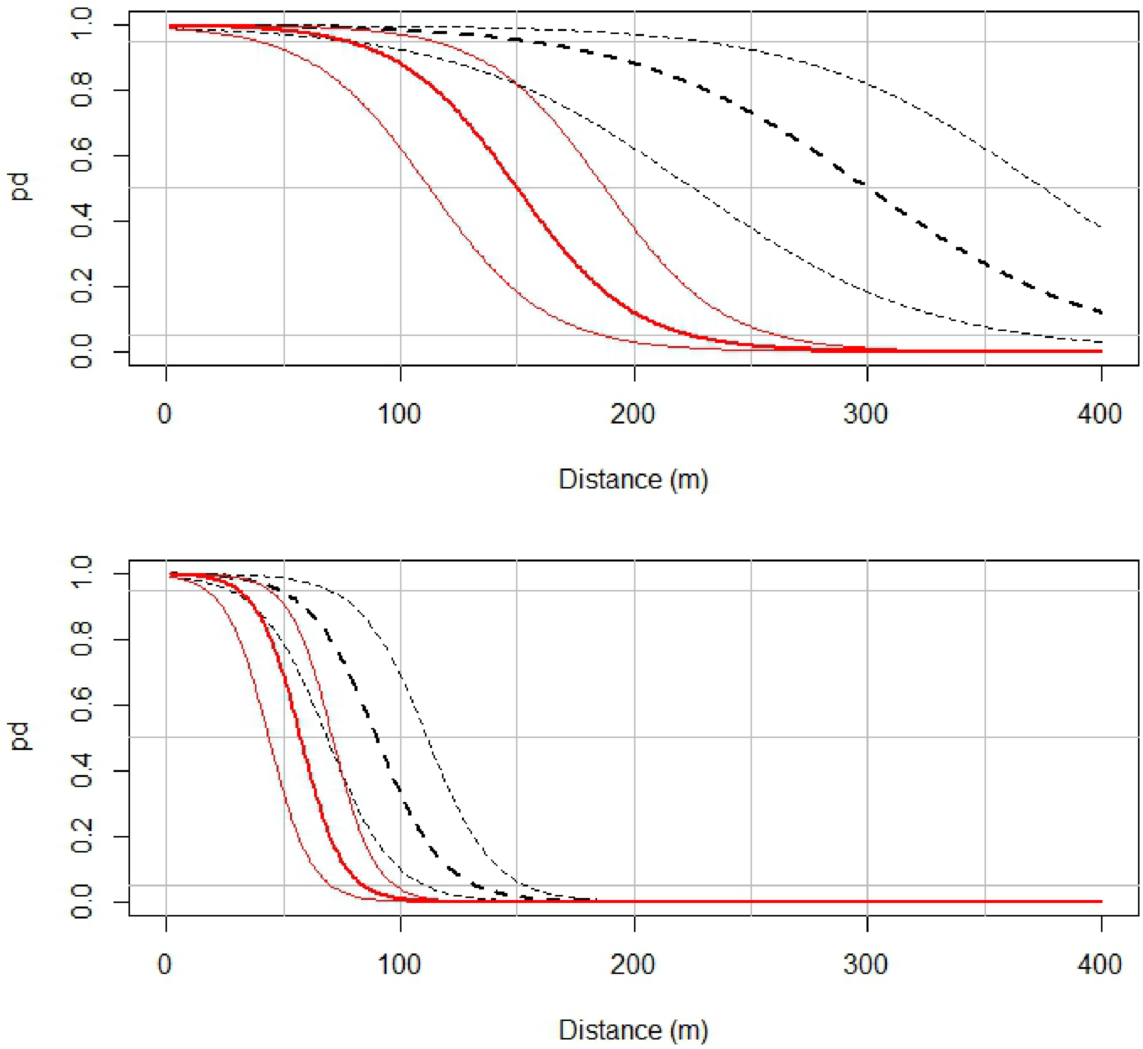


Figure 2.4: Correlations of simple index estimators with true density (D_p) was calculated across survey types and scenarios (Table 2.1). Correlation coefficients were highest and most uniform across survey types for scenarios with high availability and perceptibility (Scenarios 1 and 5) and for the removal survey type.

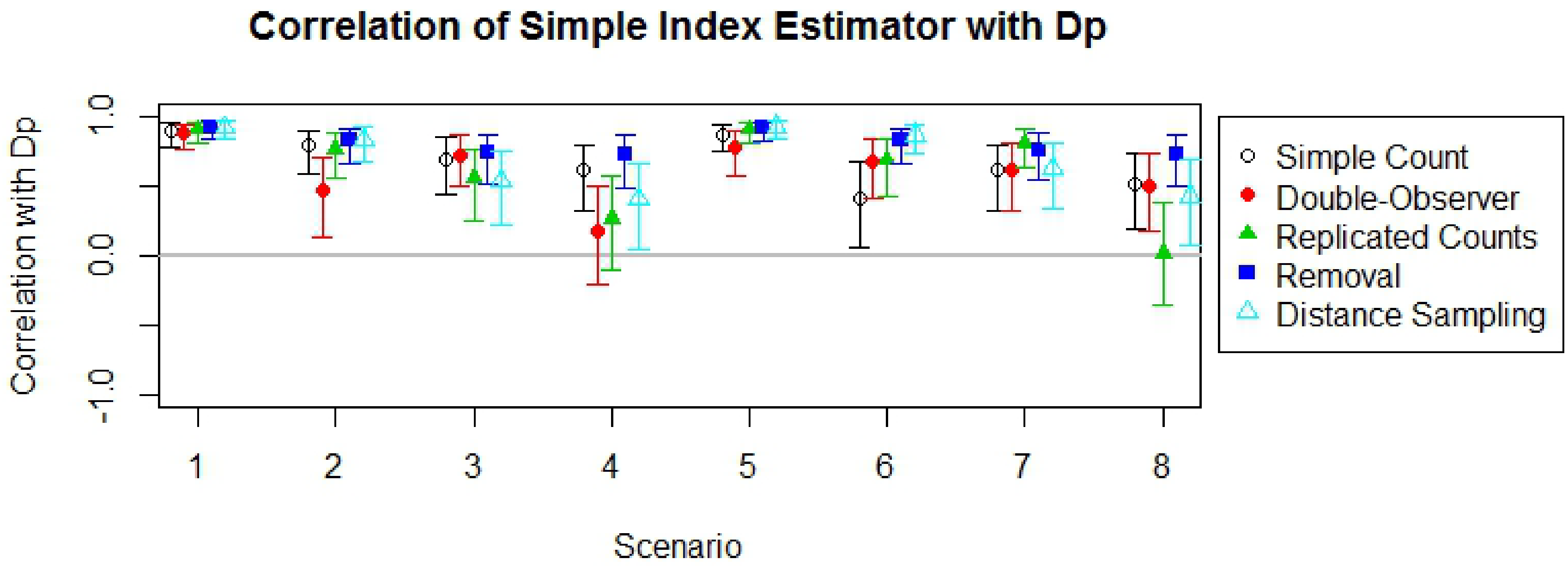


Figure 2.5: Correlation of estimators with true density (D_p) was calculated across scenarios (Table 2.1) for the double-observer survey type. Within each scenario, correlation with D_p did not differ among estimators for the double-observer survey type.

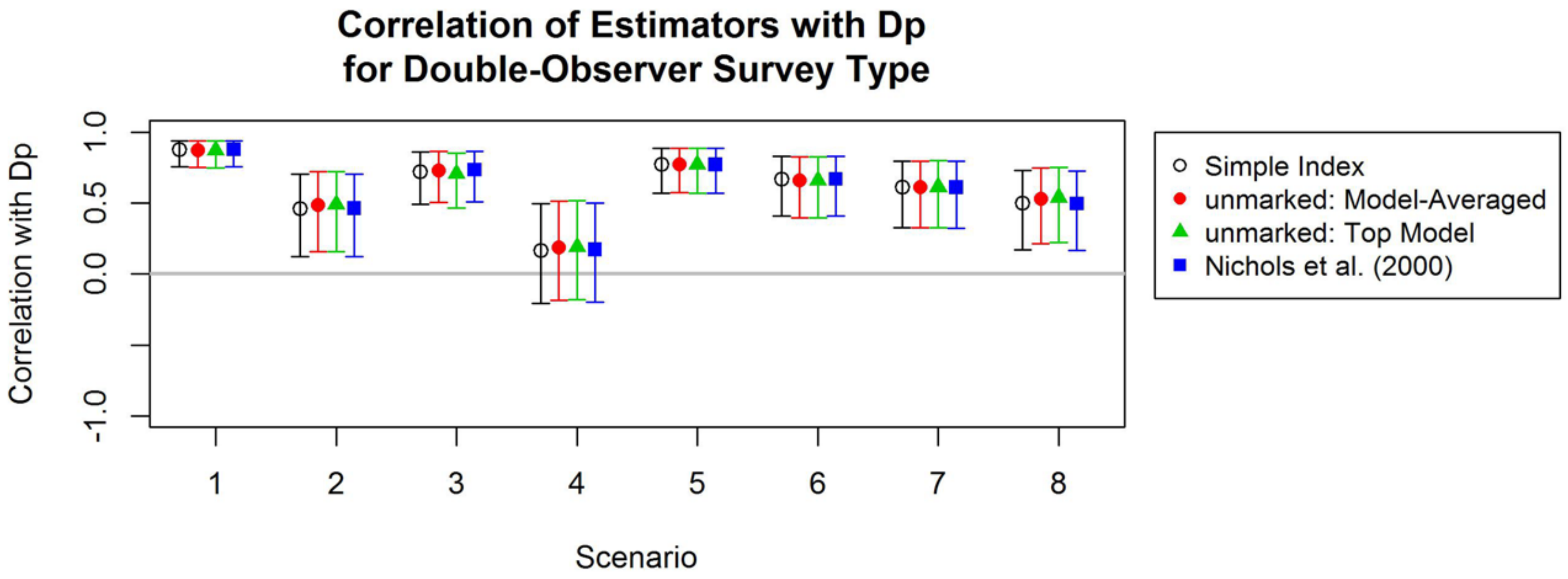


Figure 2.6: Correlation of estimators with true density (D_p) was calculated across scenarios (Table 2.1) for the distance sampling survey type. For Scenario 2, the correlation of the Program unmarked estimators with D_p was lower than the simple index estimator and Program Distance estimators, which was also true for the top model Program unmarked estimator only in Scenarios 3 and 7, and the model-averaged Program unmarked estimator only in Scenario 8.

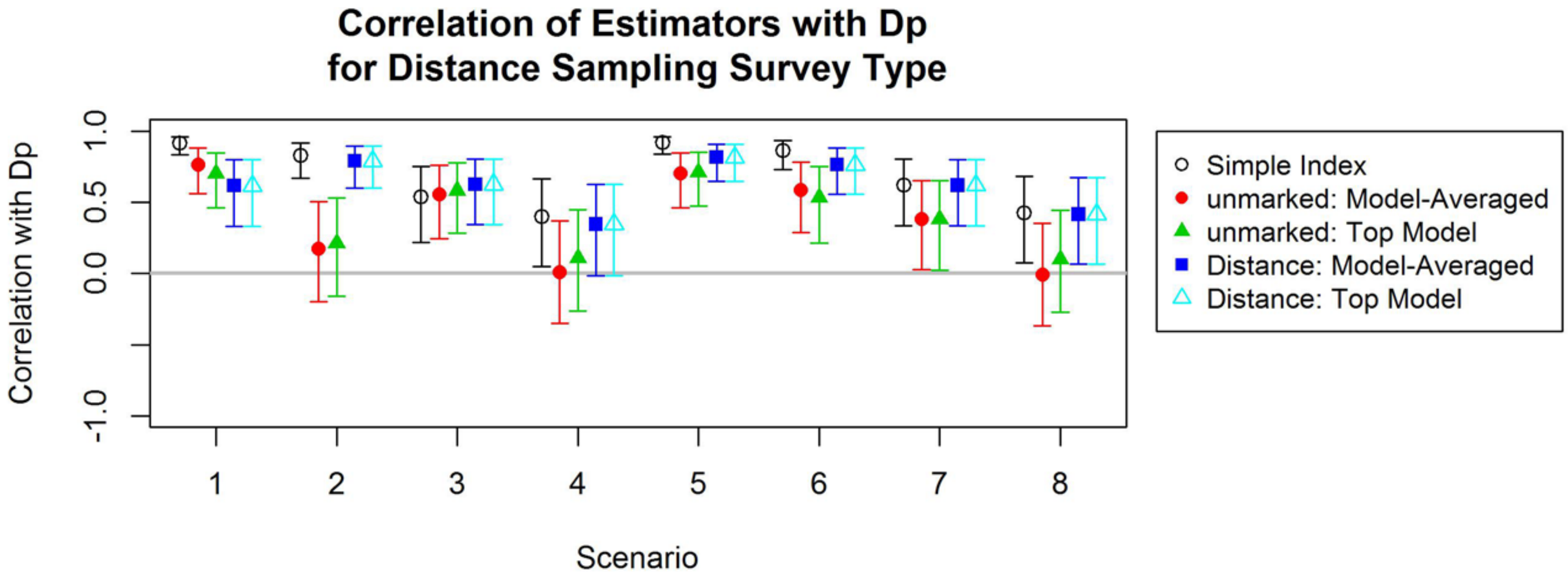


Figure 2.7: Correlation of estimators with true density (D_p) was calculated across scenarios (Table 2.1) for the removal survey type. Simple index estimators had a significantly greater correlation with D_p than adjusted estimators did for removal Scenarios 1,2,4,6, and 8. For Scenarios 5 and 7, there was no difference among estimators in the correlation with D_p.

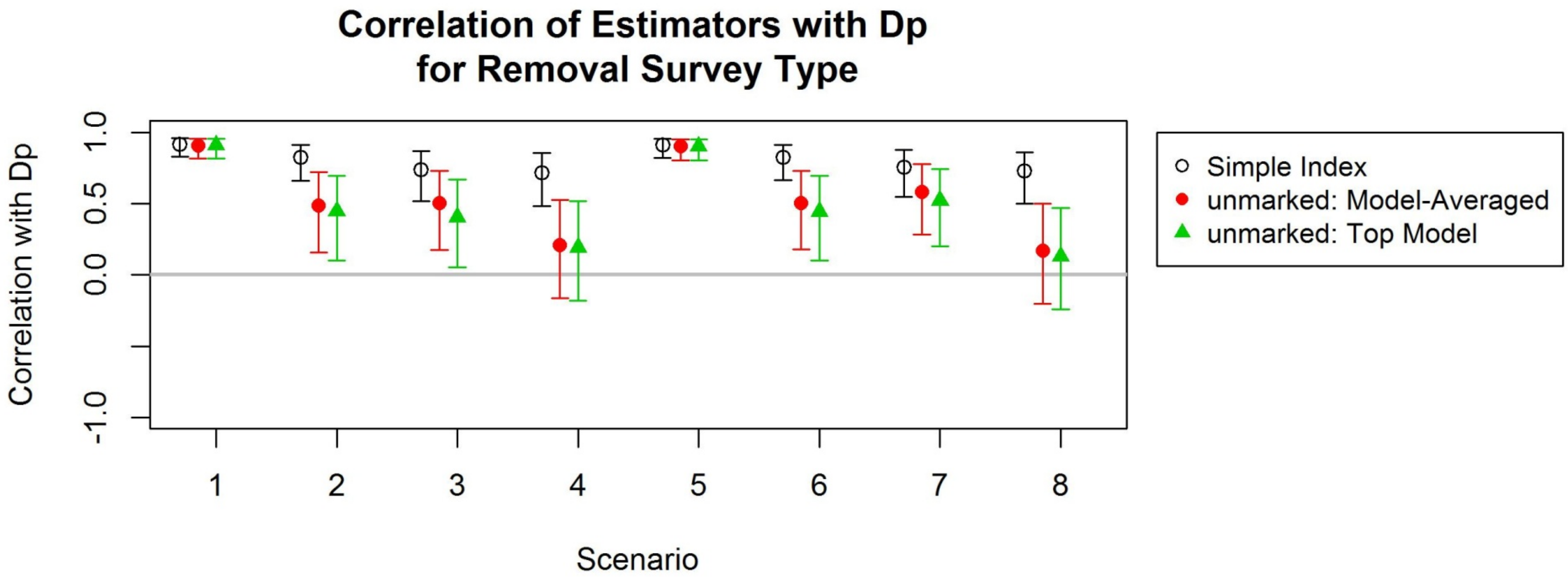


Figure 2.8: Correlation of estimators with true density (D_p) was calculated across scenarios (Table 2.1) for the replicated counts survey type. Simple index estimators had greater correlation with D_p than adjusted estimators for Scenarios 1, 2, 5, 6, and 7. For scenarios 3, 4, and 8, all estimators for had similar correlation with D_p. Correlation with D_p did not differ between the maximum count estimator and the bounded count estimator within any scenario.

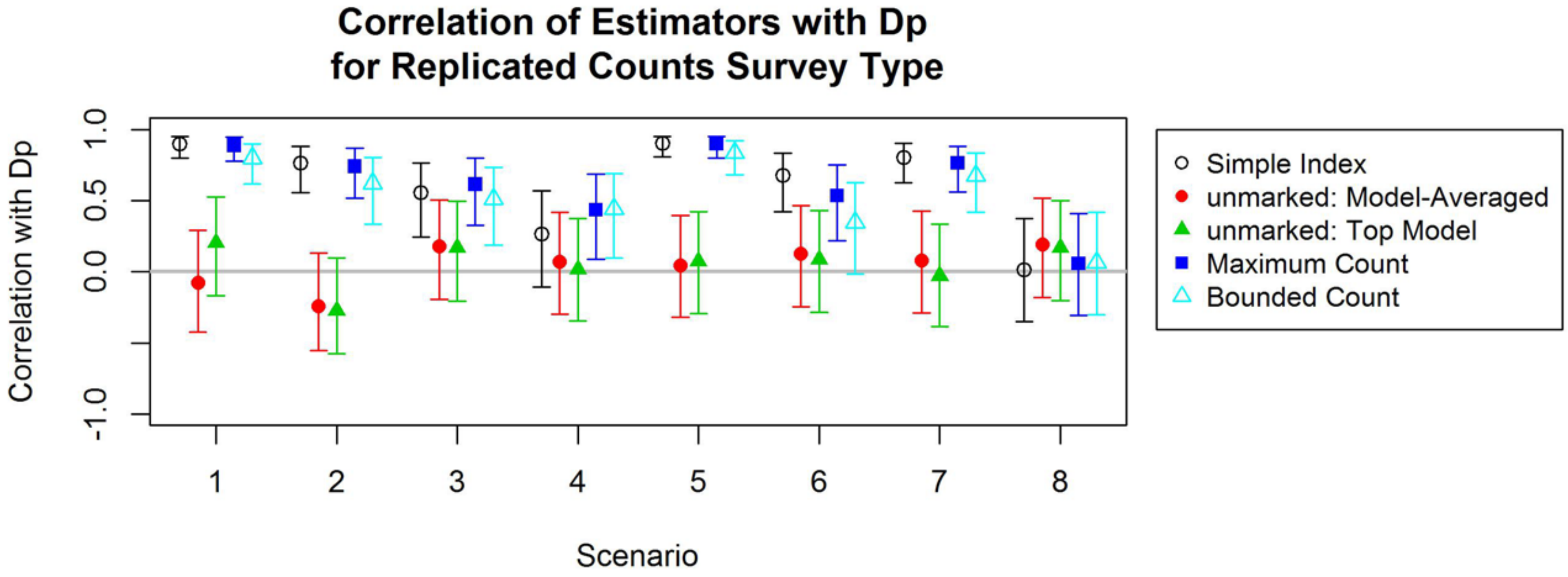


Figure 3.1: Survey site layout. The central point of each survey site (filled dot) was surrounded by 6 survey legs in a radial arrangement. Surveys occurred at sampling points along the legs, spaced 30 m apart. Speakers broadcast bird songs from the central point and detections were recorded at each sampling point. Orientation of the survey legs was determined by wind direction (first survey extended directly into the wind).

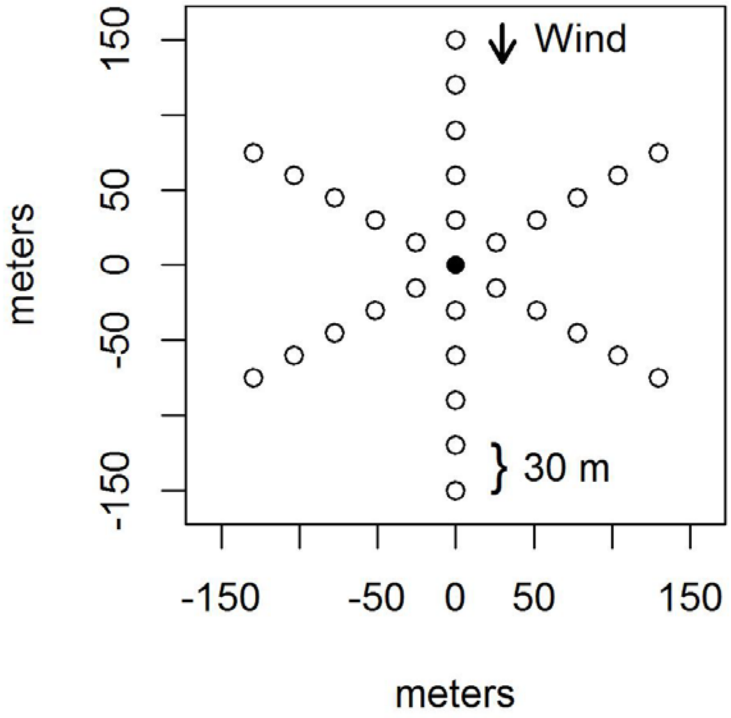
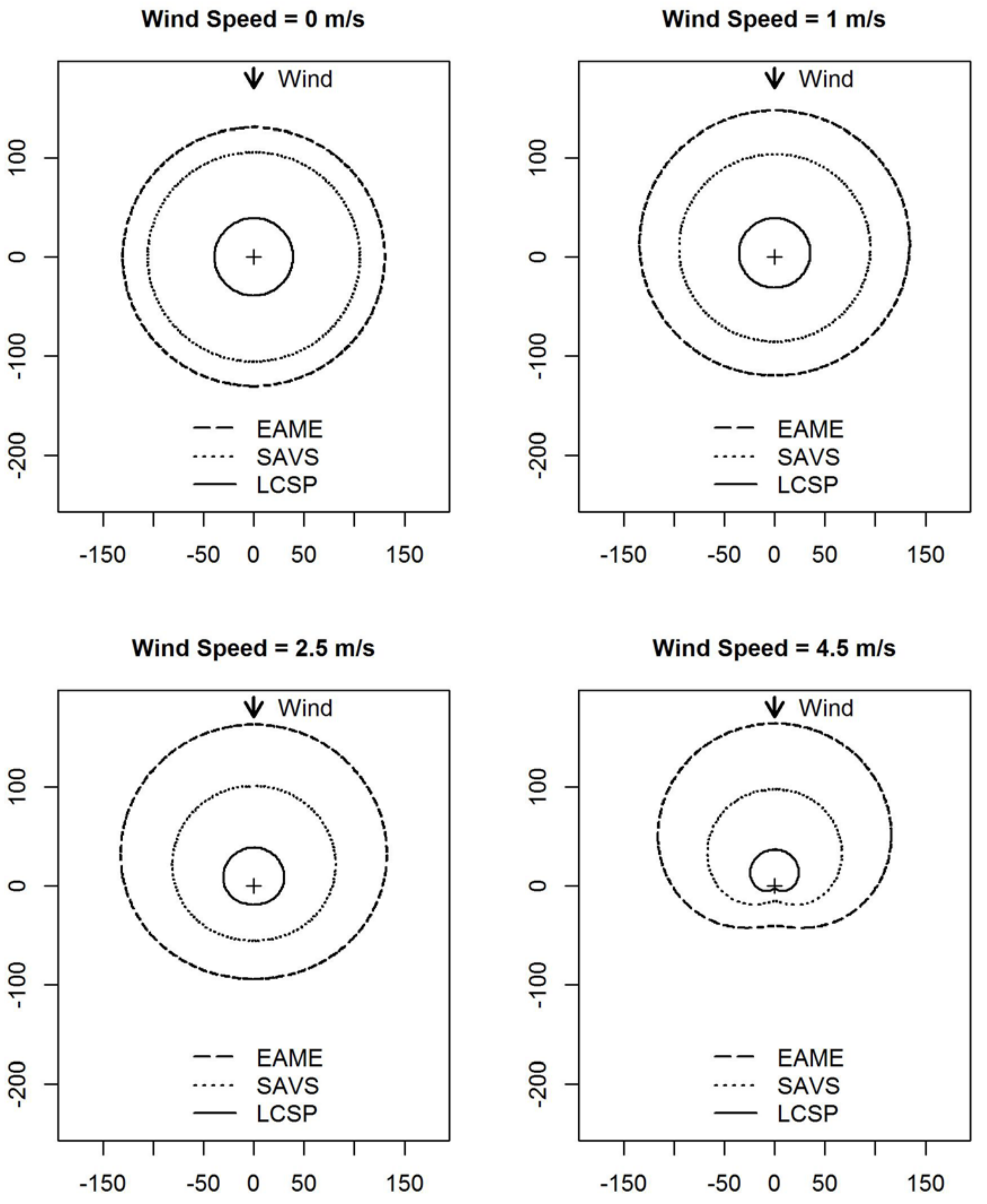


Figure 3.2: Outlines of the perceptible area around the observer (for which the predicted probability of detection was ≥ 0.5), for 3 species. EAME and LCSP had the largest and smallest perceptible areas, respectively. Wind direction was modeled as coming from due north. The position of the observer at (0,0) is marked with a +. Distances are in meters.



Bibliography

Alldredge, M. W, K. H. Pollock, and T. R. Simons (2006). Estimating detection probabilities from multiple-observer point counts. The Auk 123:1172-1182.

Alldredge, M. W., K. H. Pollock, T. R. Simons, J. A. Collazo, and S. A. Shriner (2007a). Time-of-detection method for estimating abundance from point-count surveys. The Auk 124:653–664.

Alldredge, M. W., T. R. Simons, and K. H. Pollock (2007b). Factors affecting aural detections of songbirds. Ecological Applications 17:948-955.

Alldredge, M. W., T. R. Simons, and K. H. Pollock (2007c). A field evaluation of distance measurement error in auditory avian point count surveys. Journal of Wildlife Management 71:2759-2766.

Amundson, C. L., J. A. Royle, and C. M. Handel (2014). A hierarchical model combining distance sampling and time removal to estimate detection probability during avian point counts. The Auk 131:476-494.

Banks-Leite, C., R. Pardini, D. Boscolo, C. R. Cassano, T. Püttker, C. S. Barros, and J. Barlow (2014). Assessing the utility of statistical adjustments for imperfect detection in tropical conservation science. Journal of Applied Ecology 51:849-859.

Bird Banding Laboratory (2016). Species table and recommended band sizes. https://www.pwrc.usgs.gov/BBL/MANUAL/speclist.cfm, accessed 15 October 2016.

BirdLife International (2016). Species factsheet: *Dendroica caerulescens*. http://www.birdlife. org/datazone/species/factsheet/22721673.

Bollinger, E. K., T. A. Gavin, and D. C. McIntyre (1988). Comparison of transects and circular-plots for estimating Bobolink densities. Journal of Wildlife Management 52:777-786.

Brackenbury, J. H. (1979). Power capabilities of the avian sound-producing system. Journal of Experimental Biology 78:163-166.

Buckland, S. T., D. R. Anderson, K. P. Burnham, J. L. Laake, D. L. Borchers, and L. Thomas (2001). Introduction to Distance Sampling. Oxford University Press, United Kingdom.

Buckland, S. T., K. P. Burnham, D. R. Anderson, and J. L. Laake (1993). Distance Sampling: Estimation of Abundance of Biological Populations. Chapman and Hall, London, United Kingdom.

Burnham, K. P., and D. R. Anderson (2002). Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach. Springer, New York, USA.

Burnham, K. P., D. R. Anderson, and J. L. Laake (1980). Estimation of density from line transect sampling of biological populations. Wildlife Monographs 72:3-202.

Cimprich, D. A. (2009). Effect of count duration on abundance estimates of Black-capped Vireos. Journal of Field Ornithology 80:94-100.

Cook, R. D., and J. O. Jacobson (1979). A design for estimating visibility bias in aerial surveys. Biometrics 35:735-742.

Confer, J. L., R. E. Serrell, M. Hager, and E. Lahr (2008). Field tests of the Rosenberg-Blancher method for converting point counts to abundance estimates. The Auk 125:932-938.

Conroy, M. J. (1996). Abundance indices. In Measuring and Monitoring Biological Diversity: Standard Methods for Mammals (D. E. Wilson, F. R. Cole, J. D. Nichols, R. Rudran, and M. S. Foster, Editors.). Smithsonian Institution Press, Washington, D. C., USA.

Conway, C.J. (2011). Standardized North American marsh bird monitoring protocol. Waterbirds 34:319-346.

Conway, C. J., and J. P. Gibbs (2005). Summary of intrinsic and extrinsic factors affecting detection probability of marsh birds. Wetlands 31:403-411.

Dawson, D. G. (1981). Counting birds for a relative measure (index) of density. In Estimating Numbers of Terrestrial Birds. (C. J. Ralph and J. M. Scott, Editors). Studies in Avian Biology No. 6.

Dawson, D. K., D. R. Smith, and C. S. Robbins (1995). Point count length and detection of forest neotropical migrant birds. USDA Forest Service General Technical Report PSW-GTR-149.

Dennis, E. B., B. J. T. Morgan, and M. S. Ridout (2015). Computational aspects of N-mixture models. Biometrics 71:237-246.

Diefenbach, D. R., D. W. Brauning, and J. A. Mattice (2003). Variability in grassland bird counts related to observer differences and species detection rates. The Auk 120:1168-1179.

Diefenbach, D. R., M. R. Marshall, J. A. Mattice, and D. W. Brauning (2007). Incorporating availability for detection in estimates of bird abundance. The Auk 124:96-106.

Efford, M. G., and D. K. Dawson (2009). Effect of distance-related heterogeneity on population size estimates from point counts. The Auk 126:100-111.

Emlen, J. T. (1977). Estimating breeding season bird densities from transect counts. The Auk 94:455-468.

Farnsworth, G. L., K. H. Pollock, J. D. Nichols, T. R. Simons, J. E. Hines, and J. R. Sauer (2002). A removal method for estimating detection probabilities from point-count surveys. The Auk 119:414–425.

Fiske, I. J., and R. B. Chandler (2011). unmarked: An R package for fitting hierarchical models of wildlife occurrence and abundance. Journal of Statistical Software, 43:1-23.

Forcey, G. M., J. T. Anderson, F. K. Ammer, and R. C. Whitmore (2006). Comparison of two double-observer point-count approaches for estimating breeding bird abundance. Journal of Wildlife Management 70:1674-1681.

Granholm, S. L. (1983). Bias in density estimated due to movement of birds. The Condor 85:243-248.

Hochachka, W. M., M. Winter, and R. A. Charif (2009). Sources of variation in singing probability of Florida grasshopper sparrows, and implications for design and analysis of auditory surveys. The Condor 111:349-360.

Holmes, R. T., P. P. Marra, and T. W. Sherry (1996). Habitat-specific demography of breeding black-throated blue warblers (*Dendroica caerulescens*): implications for population dynamics. Journal of Animal Ecology 65:183-195.

Holmes, R. T., N. L. Rodenhouse, and T. S. Sillett (2005). Black-throated blue warbler. In The Birds of North America Online (A. Poole, Editor). Cornell Lab of Ornithology, Ithaca, NY, USA. doi:10.2173/bna.87

Holmes, R. T., T. W. Sherry, and F. W. Sturges (1986). Bird community dynamics in a temperate deciduous forest: long-term trends at Hubbard Brook. Ecological Monographs 56:201-220.

Jennrich, R. I., and F. B. Turner (1969). Measurement of non-circular home range. Journal of Theoretical Biology 22:227-237.

Johnson, D. H. (2008). In defense of indices: the case of bird surveys. Journal of Wildlife Management 72:857-868.

Johnson, D. H., C. E. Braun, and M. A. Schroeder (2007). The bounded-count method for analysis of lek counts. In Monitoring populations of sage grouse (K. P. Reese and R. T. Bowyer, Editors.). College of Natural Resources Experiment Station Bulletin 88, University of Idaho, Moscow, USA.

Johnson, F. A., R. M. Dorazio, T. D. Castellón, J. Martin, J. O. Garcia, and J. D. Nichols (2014). Tailoring point counts for inference about avian density: dealing with nondetection and availability. Natural Resource Modeling 27:163-177.

Kepler, C. B., and J. M. Scott (1981). Reducing bird count variability by training observers. In Estimating Numbers of Terrestrial Birds. (C. J. Ralph and J. M. Scott, Editors). Studies in Avian Biology No. 6.

Link, W. A., and R. J. Barker (2010). Bayesian Inference with Ecological Applications. Academic Press, New York, USA.

Link, W. A., and J. R. Sauer (1998). Estimating population change from count data: Application to the North American breeding bird survey. Ecological Applications 8:258-268.

Macaulay Library (1988a). Bobolink: *Dolichonyx oryzivorus*. http://macaulaylibrary.org/audio/42237.

Macaulay Library (1988b). Horned Lark: *Eremophila alpestris*. http://macaulaylibrary.org/audio/41134.

Macaulay Library (1988c). Le Conte's Sparrow: *Ammodramus leconteii*. http://macaulaylibrary.org/audio/42207.

Macaulay Library (1988d). Savannah Sparrow: *Passerculus sandwichensis*. http://macaulaylibrary.org/audio/42219.

Macaulay Library (1988e). Vesper Sparrow: *Pooecetes gramineus*. http://macaulaylibrary.org/audio/42228.

Macaulay Library (1990f). Dickcissel: *Spiza Americana*. http://macaulaylibrary.org/audio/50234.

Macaulay Library (1990a). Grasshopper Sparrow: *Ammodramus savannarum*. http://macaulaylibrary.org/audio/50283.

Macaulay Library (1990b). Sedge Wren: *Cistothorus platensis stellaris*. http://macaulaylibrary.org/audio/50277.

Macaulay Library (1992a). Black-throated Blue Warbler: *Setophaga caerulescens*. http://macaulaylibrary.org/audio/73987.

Macaulay Library (1992b). Henslow's Sparrow: *Ammodramus henslowii*. http://macaulaylibrary.org/audio/73945.

Macaulay Library (1994). Black-throated Blue Warbler: *Setophaga caerulescens*. http://macaulaylibrary.org/audio/76520.

Macaulay Library (1998). Eastern Meadowlark: *Sturnella magna*. http://macaulaylibrary.org/audio/94367.

Macaulay Library (2000). Black-throated Blue Warbler: *Setophaga caerulescens*. http://macaulaylibrary.org/audio/107391.

Macaulay Library (2010). Black-throated Blue Warbler: *Setophaga caerulescens*. http://macaulaylibrary.org/audio/140091.

Matsuoka, S. M., C. L. Mahon, C. M. Handel, P. Sólymos, E. M. Bayne, P. C. Fontaine, and C. J. Ralph (2014). Reviving common standards in point-count surveys for broad inference across studies. The Condor 116:599-608.

Mayfield, H. F. (1981). Problems in estimating population size through counts of singing males. In Estimating Numbers of Terrestrial Birds. (C. J. Ralph and J. M. Scott, Editors). Studies in Avian Biology No. 6.

McLaren, M. A., and M. D. Cadman (1999). Can novice volunteers provide credible data for bird surveys requiring song identification?. Journal of Field Ornithology 70:481-490.

McShea, W. J., and J. H. Rappole (1997). Variable song rates in three species of passerines and implications for estimating bird populations. Journal of Field Ornithology 68:367-375.

Moore, J. E., D. M. Scheiman, and R. K. Swihart (2004). Field comparison of removal and modified double-observer modeling for estimating detectability and abundance of birds. The Auk 121:865-876.

Morris Wetland Management District (2011). Waterfowl Production Areas. http://www.fws.gov/midwest/morris/wpa.html, accessed 5 January 2012.

Morse, D. H. (1976). Variables affecting the density and territory size of breeding spruce-woods warblers. Ecology 57:290-301.

Nichols, J. D., J. E. Hines, J. R. Sauer, F. W. Fallon, J. E. Fallon, and P. J. Heglund (2000). A double-observer approach for estimating detection probability and abundance from point counts. The Auk 117:393-408.

Nichols, J. D., L. Thomas, and P. B. Conn (2009). Inferences about landbird abundance from count data: recent advances and future directions. In Modeling Demographic Processes in Marked Populations. (D. L. Thomson, E. G. Cooch, and M. J. Conroy, Editors). Springer-Verlag, New York, USA.

North American Breeding Bird Survey (1998). Instructions For Conducting the North American Breeding Bird Survey. U.S. Geological Survey Patuxent Wildlife Research Center, Laurel, MD. https://www. pwrc. usgs. gov/bbs/participate/instructions. html,.

Nudds, T.D. (1977). Quantifying the vegetative structure of wildlife cover. Wildlife Society Bulletin 5: 113-117.

Pacifici, K., T. R. Simons, K. H. Pollock (2008). Effects of vegetation and background noise on the detection process in auditory avian point-count surveys. The Auk 125:600-607.

Partners in Flight (2015). Population Estimates Database. http://rmbo.org/pifpopestimates/Database.aspx.

Patuxent Wildlife Research Center (1998). Instructions for Conducting the North American Breeding Bird Survey. https://www.pwrc.usgs.gov/bbs/participate/instructions.html, accessed 18 October 2016.

Peterjohn, B.G., and J.R. Sauer (1999). Population status of North American grassland birds from the North American breeding bird survey, 1966-1996. Studies in Avian Ecology 19:27-44.

Ralph, C. J., S. Droege, and J. R. Sauer (1995). Managing and monitoring birds using point counts: standards and applications. In Monitoring Bird Populations by Point Count. (C. J. Ralph, J. R. Sauer, and S. Droege, Editors). USDA Forest Service General Technical Report PSQ-GTR-149.

Ralph, C. J., G. R. Geupel, P. Pyle, T. E. Martin, and D. F. DeSante (1993). Handbook of field methods for monitoring landbirds. USDA Forest Service General Technical Report PSW-GTR-144.

Riddle, J. D., S. J. Stanislav, K. H. Pollock, C. E. Moorman, and F. S. Perkins (2010). Separating components of the detection process with combined methods: an example with northern bobwhite. Journal of Wildlife Management 74:1319-1325.

Robbins, C. S. (1981a). Bird activity levels related to weather. In Estimating Numbers of Terrestrial Birds. (C. J. Ralph and J. M. Scott, Editors). Studies in Avian Biology No. 6.

Robbins, C. S. (1981b). Effect of time of day on bird activity. In Estimating Numbers of Terrestrial Birds. (C. J. Ralph and J. M. Scott, Editors). Studies in Avian Biology No. 6.

Robbins, C. S., D. Bystrak, and P. H. Geissler (1986). The breeding bird survey: its first fifteen years, 1965-1979. U.S. Department of the Interior, U.S. Fish and Wildlife Service Resource Publication 157. http://pubs. usgs. gov/unnumbered/5230189/report. pdf.

Robbins, M. B., A. S. Nyari, M. Papes, and B. W. Benz (2009). Song rates, mating status, and territory size of cerulean warblers in Missouri Ozark riparian forest. Wilson Journal of Ornithology 121:283-289.

Robson, D. S., and J. H. Whitlock (1964). Estimation of a truncation point. Biometrika 51:33-39.

Rosenberg, K. V., and P. J. Blancher (2005). Setting numerical population objectives for priority landbird species. USDA Forest Service General Technical Report PSW-GTR-191.

Rosenstock, S. S., D. R. Anderson, K. M. Giesen, T. Leukering, and M. F. Carter (2002). Landbird counting techniques: current practices and an alternative. The Auk 119:46-53.

Royle, A. J. (2004). N-mixture models for estimating population size from spatially replicated counts. Biometrics 60:108-115.

Ruiz-Gutiérrez, V., E. F. Zipkin, and A. A. Dhondt (2010). Occupancy dynamics in a tropical bird community: unexpectedly high forest use by birds classified as non-forest species. Journal of Applied Ecology 47:621-630.

Samson, F. B., and F. L. Knopf (1994). Prairie conservation in North America. Bioscience 44: 418–421.

Sauer, J. R., J. E. Hines, J. E. Fallon, K. L. Pardieck, D. J. Ziolkowski, Jr., and W. A. Link (2014). The North American Breeding Bird Survey, Results and Analysis 1966 - 2012. Version 02.19.2014. USGS Patuxent Wildlife Research Center, Laurel, MD, USA. http://www. mbr-pwrc. usgs. gov/bbs/.

Sauer, J. R., B. G. Peterjohn, and W. A. Link (1994). Observer differences in the North American Breeding Bird Survey. The Auk 111:50-62.

Schenker, N., and J. F. Gentleman (2001). On judging the significance of differences by examining the overlap between confidence intervals. American Statistician 55:182-186.

Schieck, J. (1997). Biased detection of bird vocalizations affects comparisons of bird abundance among forested habitats. The Condor 99:179-190.

Schoener, T. W. (1968). Sizes of territories among birds. Ecology 49:123-141.

Scott, T. A., P. Lee, G. C. Greene, and D. A. McCallum (2005). Singing rate and detection probability: An example from the least Bell's vireo *(Vireo belli pusillus).* USDA Forest Service General Technical Report PSW-GTR-191.

Scott, J. M., and C. J. Ralph (1981). Introduction. In Estimating Numbers of Terrestrial Birds. (C. J. Ralph and J. M. Scott, Editors). Studies in Avian Biology No. 6.

Scott, J. M., and F. L. Ramsey. (1981). Length of count period as a possible source of bias in estimating bird densities. In Estimating Numbers of Terrestrial Birds. (C. J. Ralph and J. M. Scott, Editors). Studies in Avian Biology No. 6.

Seber, G. A. F. (1982). The Estimation of Animal Abundance and Related Parameters, 2nd edition. Charles Griffin & Company Ltd., London, UK.

Sherry, T. W., and R. T. Holmes *(*1985). Dispersion patterns and habitat responses of birds in northern hardwoods forests. Habitat Selection in Birds. (M. L. Cody, Editor). Academic Press, FL, USA.

Simons, T. R., M. W. Alldredge, K. H. Pollock, J. M. Wettroth (2007). Experimental analysis of the auditory detection process on avian point counts. The Auk 124:986-999.

Skalski, J. R., and D. S. Robson (1992). Techniques for Wildlife Investigators: Design and Analysis of Capture Data. Academic Press, San Diego, CA, USA.

Smith, T. M., and H. H. Shugart (1987). Territory size variation in the ovenbird: the role of habitat structure. Ecology 68:695-704.

Stacier, C. A., V. Ingalls, and T. W. Sherry (2006). Singing behavior varies with breeding status of American redstarts (*Setophaga ruticilla*). Wilson Journal of Ornithology 118:439-451.

Stanislav, S. J., K. H. Pollock, T. R. Simons, and M. W. Alldredge (2010). Separation of availability and perception processes for aural detection in avian point counts: a combined multiple-observer and time-of-detection approach. Avian Conservation and Ecology 5:3.

Steele, B. B. (1992). Habitat selection by breeding black-throated blue warblers at two spatial scales. Ornis Scandinavica 23:33-42.

Steele, B. B. (1993). Selection of foraging and nesting sites by black-throated blue warblers: their relative influence on habitat choice. The Condor 95:568-579.

Stroock, D. W. (2005). An introduction to Markov processes. Graduate Texts in Mathematics, Vol. 230. Springer, Berlin, Germany.

Thomas, L., S. T. Buckland, K. Burnham, D. Anderson, J. Laake, D. Borchers, and S. Strindberg (2002). Distance sampling. Encyclopedia of Environmetrics. Wiley, West Sussex, UK.

Thomas, L., S. T. Buckland, E. A. Rexstad, J. L. Laake, S. Strindberg, S. L. Hedley, J. R. B. Bishop, T. A. Marques, and K. P. Burnham (2010). Distance software: design and analysis of distance sampling surveys for estimating population size. Journal of Applied Ecology 47:5-14.

Thompson, F. R., III, W. D. Dijak, T. G. Kulowiec, and D. A. Hamilton (1992). Breeding bird populations in Missouri Ozark forests with and without clearcutting. Journal of Wildlife Management 56:23-30.

Thompson, F. R., III, and F. A. La Sorte (2008). Comparison of methods for estimating bird abundance and trends from historical count data. Journal of Wildlife Management 72:1674-1682.

UCLA Statistical Consulting Group (2011). R Library: Contrast Coding Systems for categorical variables. http://www.ats.ucla.edu/stat/r/library/contrast_coding.htm, accessed October 18, 2016.

U.S. Department of Agriculture (2010). National Agricultural Imagery Program Mosaic, Geospatial Data Gateway. http://datagateway.nrcs.usda.gov, accessed 23 August 2011.

Verner, J. (1985). Assessment of counting techniques. Current Ornithology 2:247-302.

Wenny, D. G., R. L. Clawson, J. Faaborg, and S. L. Sheriff (1993). Population density, habitat selection, and minimum area requirements of three forest-interior warblers in central Missouri. The Condor 95:968-979.

Wilson, D. M., and J. Bart (1985). Reliability of singing bird surveys: effects of song phenology during the breeding season. The Condor 87:69-73.

Wolf, A. T., R. W. Howe, and G. J. Davis (1995). Detectability of forest birds from stationary points in northern Wisconsin. In Monitoring Bird Populations by Point Count. (C. J. Ralph, J. R. Sauer, and S. Droege, Editors). USDA Forest Service General Technical Report PSQ-GTR-149.

Zembal, R., and B. W. Massey (1987). Seasonality of vocalizations by light-footed clapper rails. Journal of Field Ornithology 58:41–48.

Zou, G. Y. (2007). Toward using confidence intervals to compare correlations. Psychological Methods 12:399-413.

APPENDIX A

Glossary

A

The area surveyed during a bird survey.

Abundance
N, the number of birds that could potentially be counted during a survey. More precise definitions of abundance exist, see N^*, N_p, N_s, and N_a.

ActualStartTimesAll
The actual start time for surveys in the simulation. *ActualStartTimesAll* is normally-distributed
with mean *PlannedStartTimesAll* to include variation in travel time to sites or other logistical variation.

Adjustment Methods or Adjustment Analysis Methods
Analysis methods which attempt to estimate both detectability and abundance (or density). See also "Analysis Method".

Analysis Method
The method used to analyze data obtained from a survey. More than one analysis method can often be applied to data from a particular survey type, and adjustment analysis methods often require data gathered in a specific survey type.

Area.yij
The bird-specific parameter that describes the area of a 95% elliptical density contour of a bird's bivariate normally-distributed territory. *Area.yij* was log-normally distributed with mean *MeanTerrArea* and standard deviation *SDTerrArea*. See also *Spatial.yij*.

Availability
p_a: the probability that a bird is available (vocalizes) during a survey, given that it is present.

BAWW
Black-and-white warbler (*Mniotilta vari*), the species upon which parameters for the low p_d parameters in Chapter 2 were based.

BirdsInHabitat.yi
The site and year-specific number of birds simulated occurring in habitat in the 400 ha around the observer. *BirdsInHabitat.yi* is modeled as a Poisson-distributed random variable with parameter *LambdaHabitat.yi*.

BirdsInMatrix.yi
The site and year-specific number of birds simulated occurring in matrix in the 400 ha around the observer. *BirdsInMatrix.yi* is modeled as a Poisson-distributed random variable with parameter *LambdaMatrix.yi*.

Bounded Count Analysis Method
An analysis method that requires replicated visits to sites, using twice the maximum count, minus the second largest count as an index to abundance.

BTBW
Black-throated blue warbler (*Setophaga caerulescens*), the species upon which parameters for Chapter 1 were based.

C
see "Count".

CenterX.yij, *CenterY.yij*
The bird-specific parameters that describe the x and y coordinates of the center of a bird's bivariate normally-distributed territory. See also *Spatial.yij*.

CloseBirds.yi
Site-specific N_p, the abundance of birds present within the survey radius at the beginning of the survey.

CloseSingers.yi.
Site-specific N_a, the abundance of birds available and present within the survey radius at the beginning of the survey.

CloseTerrBirds.yi
Site-specific N_s: the number of simulated birds having territories with 95% utilization distributions overlapping the survey radius.

Count
C, The sum of birds detected during a survey. Theoretically, the count $C = N^{*} \times p_s p_p \times p_a \times p_d$. In R code, site-specific count was *RawCount.yi*, and included birds that were detected and estimated by the observer to be within the survey radius at the time of first detection. Thus, *RawCount.yi* could include birds that moved into the survey radius during the survey or were falsely estimated to be within the survey radius and were not included in *CloseSingers.yi* (see "N_a").

Count.yijrko
Observer-specific count status where *Count.yijrko* = 1 indicated that bird *j* was counted during interval *k* and *Count.yijrko* = 0 indicated that it was not counted.

D
Density of birds within the surveyed area (*A*). More precise definitions of density exist, see D_p, D_s, and D_a.

D_a

$N^{*} \times p_s \times p_p \times p_a$ / A: Density of birds available, given that they were present within the survey radius at the beginning of the survey.

Detected.yijrko
Bernoulli-distributed event with probability of detection *perceptibility.yijrko* describing whether bird *j* was detected by observer *o* during interval *k*. If bird *j* was detected, then *Detected.yijrko* = 1; if not, *Detected.yijrko* = 0.

Detection probability
p: the overall probability that a bird is detected during a survey

Distance Sampling Analysis Method
An analysis method that estimates detection probability as a function of distance from the observer, assuming that detection probability at the observer's location is 1 (Burnham et al. 1980, Buckland et al. 1993).

Distance Sampling Survey Type
A survey type where the observer estimates and records the distance to each bird counted. See also "Survey Type".

Distance.yijrk
The interval-specific distance from the observer(s) of bird *j*.

DoesBirdMove.yijrk
A Bernoulli-distributed random variable with parameter *PrBirdMoves.yijrk* that describes whether a bird moved between interval *k*-1 and interval *k*.

Double-Observer Analysis Method
An analysis method that uses the discrepancies in individual bird detections between two observers to estimate detection probability (Nichols et al. 2000). Also referred to as multiple-observer analysis method.

Double-Observer Survey Type
A survey type where counts are conducted with two simultaneous observers. See also "Survey Type".

D_p
$N^{*} \times p_s \times p_p$ / A: Density of birds present within the survey radius at the beginning of the survey.

D_s
$N^{*} \times p_s$ / A: Density of birds with territories that overlapped the survey radius. Because territories were defined in my simulation by a bivariate normal probability distribution, the exact boundaries of territories are uncertain. Site-specific D_s was the density of birds having territories with 95% utilization distributions overlapping the survey radius.

Ecc.yij
The bird-specific parameter that describes the eccentricity of the ellipse of a bird's bivariate normally-distributed territory. See also *Spatial.yij*.

Estimator
A statistic (i.e., a function of the data) that is used to infer the value of an unknown parameter, either abundance or density.

HabitatDensity.y
Year-specific mean density of birds in habitat. *HabitatDensity.y* is greater than or equal to mean density of birds in matrix (*MatrixDensity.y*).

Index Methods or Index Analysis Methods
Analysis methods which use counts as an index to abundance or density. See also "Analysis Method".

Interval
A short period of time with length *IntervalLength*, akin to the duration of one bird song, usually 2 or 3 seconds.

IntervalLength
The length of a single interval.

LambdaHabitat.yi
The parameter describing the Poisson-distributed random variable *BirdsInHabitat.yi*.

LambdaMatrix.yi
The parameter describing the Poisson-distributed random variable *BirdsInMatrix.yi*.

Location.yijrk
The interval-specific location of bird *j*.

LogisticalSurveyTime
Combined time needed to perform one bird survey, including survey time and travel time between sites.

MatrixDensity.y
Year-specific mean density of birds in matrix. *MatrixDensity.y* is less than or equal to mean density of birds in habitat (*HabitatDensity.y*).

MaxDailySongRate.yir
The replication-specific maximum daily song rate. See also *PrSing.yijrk*.

Maximum Count Analysis Method
An analysis method that uses the largest count among repeat visits to a site as an index to abundance.

MaxSurveyDistance
The radius of a survey.

MeanTerrArea
Mean territory size for simulated birds, used to model log-normally distributed *Area.yij*.

N
Abundance, the number of birds that could be potentially counted during a survey. More precise definitions of abundance exist, see N^*, N_p, N_s, and N_a.

N^*
Superpopulation: all birds whose territories or home ranges at least partially overlap the area over which inferences will be made (the area of inference).

N_a
$N^* \times p_s \times p_p \times p_a$: Abundance of birds available, given that they were present within the survey radius at the beginning of the survey. In R code, site-specific N_a was *CloseSingers.yi*.

NBirds.yi
Year and site-specific number of birds simulated within the 400 ha around the observer.

NIntervals
Survey type-specific number of intervals that occur during a survey. See also “Interval”.

N_p
$N^* \times p_s \times p_p$: Abundance of birds present within the survey radius at the beginning of the survey. In R code, site-specific N_p was *CloseBirds.yi*.

NReps
Survey type-specific number of replications. For the replicated counts survey type, *NReps* = 3, for all other survey types, *NReps* = 1. See also “Replications”.

N_s
$N^* \times p_s$: Abundance of birds with territories that overlapped the survey radius. Because territories were defined in my simulation by a bivariate normal probability distribution, the exact boundaries of territories are uncertain. In R code, site-specific N_s (*CloseTerrBirds.yi*) was the number of birds having territories with 95% utilization distributions overlapping the survey radius.

NSimultaneousObservers
Survey type-specific number of simultaneous observers performing surveys. For the double-observer survey type, *NSimultaneousObservers* = 2, for all other survey types, *NSimultaneousObservers* = 1.

NSurveySites
Survey type-specific number of sites simulated. See Table 3 for parameters.

NYears
Scenario-specific number of years simulated. For all scenarios considered, *NYears* = 30.

ObsEstimatedDistance.yijrko
Observer-specific estimated distance to bird j during interval k. Modeled by adding observer-specific error due to observation to the true distance *Distance.yijrk*.

p
Detection probability: the overall probability that a bird is detected during a survey.

p_a
Availability: the probability that a bird is available (vocalizes) during a survey, given that it is present.

PauseLength
Mean length of pauses between songs in recordings used to estimate parameters for modeling availability. See also *SongLength*.

P_{coarse}
The coarse-scale transition matrix describing the probabilities of remaining in a state or switching states, given the previous state (Table 1).

p_d
Perceptibility: the probability that a bird is detected, given that it is present in the surveyable area and available during the survey.

PercentHabitat.yi
The proportion of site *i* covered by habitat in year *y*.

Perceptibility
p_d: the probability that a bird is detected, given that it is present in the surveyable area and available during the survey.

perceptibility.yijrko
Observer-specific probability of detection for bird *j* during interval *k*.

PlannedStartTimesAll
Planned start times for bird surveys within the simulation. See also *ActualStartTimesAll*.

p_p
The probability that a bird is present in the surveyable area at the time of the survey, given that the bird's territory at least partially overlaps the surveyable area of a survey site.

PrBirdMoves.yijrk
The parameter describing Bernoulli-distributed random variable *DoesBirdMove.yijrk*.

PrNoise
The scenario-specific probability of a replication having an effect of ambient noise. *PrNoise* was used only for simulations in chapter 1.

PrSingX
The probability that a bird sings at least 1 time in X minutes. For example, PrSing5 refers to the probability that a bird sings at least 1 time in 5 minutes.

PrSing.yijrk
Interval-specific probability of being in singing mode for at least one interval within a designated number of minutes, also termed “singing probability”. For chapter 1 *PrSing.yijrk* referred to singing probability within 10 minutes, for chapter 2 *PrSing.yijrk* referred to singing probability within 5 minutes. *PrSing.yijrk* is the product of *MaxDailySongRate.yir* and *SongWeight.yijrk*.

p_s
The probability that a bird’s territory at least partially overlaps the surveyable area of a survey site.

q_1
For the 2-state Markov process modeling coarse-scale autocorrelation in singing, the proportion of time spent in singing mode in the long run (i.e., after the initial state has been “forgotten”). Also the average proportion of the population in singing mode in any given interval.

q1.fine
The proportion of time spent singing in recordings used to estimate parameters for modeling availability.

q_2
For the 2-state Markov process modeling coarse-scale autocorrelation in singing, the proportion of time spent in non-singing mode in the long run (i.e., after the initial state has been “forgotten”). Also the average proportion of the population in non-singing mode in any given interval.

RawCount.yi
Site-specific *C*, the count of birds detected and estimated by the observer to be within the survey radius at the time of first detection.

Removal Analysis Method

An analysis method that estimates detection probability by comparing first detections during different periods of the survey (Farnsworth et al. 2002). Also called time of detection analysis method (Alldredge et al. 2007a).

Removal Survey Type
A survey type where the survey period is split into time periods and the observer records the time period in which each bird was first detected.

Replications
For a survey type, the number of times within a season a site was surveyed.

Replicated Counts Analysis Method
An analysis method that uses N-mixture models to estimate detection probability across spatially and temporally replicated sites (Royle 2004).

Replicated Counts Survey Type
A survey type where counts are conducted at sites visited multiple times within each season.

Scenario
A set of model parameters designed to reflect a particular biological situation.

SDTerrArea
Standard deviation for territory size for simulated birds, used to model log-normally distributed *Area.yij*.

Simple Count Analysis Method
An analysis method wherein counts from a survey are used as an estimator of abundance. See also “Analysis Methods”.

Simple Count Survey Type
A survey type where counts are conducted without collection of accessory data. See also “Survey Type”.

Simple Index Estimator
An estimator in the simple count analysis method, equal to the sum of counts across all sites within a year.

SingingMode.yijrk
A binary, interval-specific event signifying if bird *j* was in signing mode during interval *k*. If *SingingMode.yijrk* = 1, bird *j* was in singing mode, if *SingingMode.yijrk* = 0, bird *j* was not in singing mode.

Singing Probability
The probability that a bird sings at least once within a given number of minutes.

Sings.yijrk
A binary, interval-specific event signifying if bird *j* sang during interval *k*. If *Sings.yijrk* = 1, then bird *j* sang, if *Sings.yijrk* = 0, then bird *j* did not sing.

Site
Within the simulation, a single location where observer(s) performed a bird survey.

SongLength
Mean length of songs in recordings used to estimate parameters for modeling availability. See also *PauseLength*.

SongWeight.yijrk
An interval-specific weight representing the effect of time of day on availability. See also *PrSing.yijrk*.

Spatial.yij
The bird-specific set of spatial parameters that describe a bird's bivariate normally-distributed territory.

Steady State Vector
For the transition matrix of a 2-state Markov chain, the vector $[q_1\ q_2]$ of the proportion of time spent in each state in the long run (i.e., after the initial state has been "forgotten"). Also called the limiting or stationary distribution.

Study Area
The region about which one wants to make inferences about bird abundance.

StudyHabitatProportion.
The year-specific proportion of the study area was covered by habitat.

Superpopulation
N^*: all birds whose territories or home ranges at least partially overlap the area over which inferences will be made (the area of inference).

Survey Type
A specific survey scheme of temporal and spatial replication of survey sites and number of observers (Chapter 1, Table 3). This dissertation considers 5 survey types: simple counts, double-observer, distance sampling, removal, and replicated counts.

Theta.yij
The bird-specific parameter that describes the angle of rotation of a bird's bivariate normally-distributed territory. See also *Spatial.yij*.

Time of Detection Analysis Method
See "Removal Analysis Method".

TimeinIntervals.yirk
The time of day for interval *k* during replication *r*. Dawn is *TimeinIntervals.yirk* = 0 and has units intervals (e.g., if *TimeinIntervals.yirk* = 50, the time is 50 intervals × *IntervalLength* seconds past dawn).

TransitionMatrix.fine
The fine-scale transition matrix describing the probabilities of remaining in a state or switching states, given the previous state (Table 2).

Trend
The geometric mean rate of change for a population from year 1 to the last year of surveys.

Year
Within the simulation, a complete set of sites surveyed.

$Z_{NIntervals}$
The probability that a bird *j* is in singing mode at least once in *NIntervals* intervals.

APPENDIX B

R Code for Point Count Simulation

```
##########################

##FUNCTIONS
##All functions for dissertation project

##Alternate parameterization of Beta Distribution
rbetaAlt <- function(n, MEAN, theta)
{
  a<- MEAN*theta        ## parameters = mean (mu) and concentration parameter (theta)
     instead of a,b
  b<- theta-MEAN*theta  ## See Link's Bayesian Analysis book, p. 319 bottom
  rbeta(n, a, b)
}

##Alternate parameterization of Beta Distribution for Vectorized inputs
rbetaAlt2 <- function(MEAN, theta)
{
  if(length(MEAN)==length(theta)){
    a<- MEAN*theta        ## parameters = mean (mu) and concentration parameter (theta)
     instead of a,b
    b<- theta-MEAN*theta  ## See Link's Bayesian Analysis book, p. 319 bottom
    return(rbeta(length(MEAN), a, b))
  }
  if(length(MEAN)!=length(theta)){return("rbetaAlt2 error: Mean and theta vectors are of
     unequal length")}
}

##Alternate parameterization of LogNormal distribution: input N, arithmetic mean and
     arithmetic SD of desired final distribution
rlnormAlt<-function(N, DesiredMean, DesiredSD)
```

```
{mu<-log(DesiredMean)-0.5*log(1+(DesiredSD^2)/(DesiredMean^2))
 sigma<-sqrt(log(1+(DesiredSD^2)/(DesiredMean^2)))
 rlnorm(N,mu,sigma)
}

#########################
##Spatial functions
#########################

##Add 2 vectors: a distance and angle from origin + a movement (distance and angle from
     1st position)
PolarSolve<- function(r1, theta1, rmove,thetamove)
{  ##PolarSolve is a function to add 2 (r,theta) vectors
  ##      Vector 1 = Distance and Angle (degrees!) of bird from observer at time i
  ##      Vector 2 = Distance and Angle (degrees!) of bird's position at time i+1 from
     position at time i
  radtheta1<- theta1*(2*pi/360)
  radthetamove<- thetamove*(2*pi/360)
  x1<- r1*cos(radtheta1)
  y1<- r1*sin(radtheta1)
  xmove<- rmove*cos(radthetamove)
  ymove<- rmove*sin(radthetamove)
  x2<- x1+xmove
  y2<- y1+ymove
  r2<- sqrt(x2^2+y2^2)
  radtheta2<- atan(y2/x2) ##Note: atan alone moves everything to right half of unit
     circle
  if(x2<0) {theta2<- (radtheta2+pi)*(360/(2*pi))}
  if(x2>=0) {theta2<- (radtheta2)*(360/(2*pi))}
  PolarSolveResults<- c(r2, theta2)
```

```
  return(PolarSolveResults)
  ##PolarSolve(r1, theta1, rmove,thetamove)[1] = New Distance of bird from Obs
  ##PolarSolve(r1, theta1, rmove,thetamove)[2] = New Angle of bird from Obs
}

##Calculate the distance between an observer at the origin and a bird with location (x,y)
ObsDistance<-function(x,y)
{Distance.yijrk<- sqrt(x^2+y^2)  ##radius
 BirdAnglePolar.yijrk<-atan2(y,x)    ##polar coordinates angle
 if(BirdAnglePolar.yijrk<0) BirdAnglePolar.yijrk<-BirdAnglePolar.yijrk+2*pi
 BirdAngleCompass.yijrk<- (-1*BirdAnglePolar.yijrk + (pi/2))  ##Compass rose angle of
     bird in relation to obs
 if(BirdAngleCompass.yijrk<0) BirdAngleCompass.yijrk<-BirdAngleCompass.yijrk+2*pi
 return(list("Distance.yijrk"=Distance.yijrk,
             "BirdAnglePolar.yijrk"=BirdAnglePolar.yijrk,
             "BirdAngleCompass.yijrk"=BirdAngleCompass.yijrk))
}

## Calculate the distance between the center of a single GenSpatialParameters
## object and the centers of a list of GenSpatialParameters objects.
DistanceCenters<-function(SpatialCandidate, SpatialList)
{ xc<-SpatialCandidate$CenterX.yij
  yc<-SpatialCandidate$CenterY.yij

  xs<-unname(unlist(SpatialList)[names(unlist(SpatialList))=="CenterX.yij"])
  ys<-unname(unlist(SpatialList)[names(unlist(SpatialList))=="CenterY.yij"])

  Distances<-sqrt((xc-xs)^2 + (yc-ys)^2)
  return(Distances)
}
```

```
##Rotate X and Y coordinates around THE ORIGIN by an angle, theta
RotatePoints<- function(X1, Y1, theta)
{ ##Input angle as degrees
  radtheta<- theta*(2*pi/360)
  Xnew<- X1*cos(radtheta) - Y1*sin(radtheta)
  Ynew<- X1*sin(radtheta) + Y1*cos(radtheta)
  RotateResults<- cbind(Xnew, Ynew)
  return(RotateResults)
  #RotatePoints(X2, Y2, Theta)[,1] #will return column of x coordinates
  #RotatePoints(X2, Y2, Theta)[,2] #will return column of y coordinates
}

##Generate parameters needed to generate Bivariate normal territories
##Input: Area (mean & SD) of territory and PercentUD.yij (the percent of locations
     contained by a HR that large)
##Output: CenterXyij, CenterY.yij, Theta.yij, SDy.yij, SDx.yij, Area.yij, a.yij, b.yij,
     Ecc.yij
GenSpatialParameters<- function(HRAreamean, ## mean territory/HR size
                                HRAreaSD,   ## SD territory/HR size
                                PercentUD.yij=0.95,  ## What % of location are contained
     by a territory of mean size?
                                ylim=c(-1000,1000),  ##How many meters wide is area
     modeled?
                                xlim=c(-1000,1000),  ##How many meters tall is area
     modeled?
                                OverlapUD=0.8)  #What % Utilization ellipse of used for
     overlap comparisons?
                                { CenterX.yij<- runif(1,xlim[1], xlim[2])  ##x-coordinate
     for center of ellipse
```

```
CenterY.yij<- runif(1,ylim[1], ylim[2])  ##y-coordinate for center of ellipse
Theta.yij<- runif(1,0,360)   ##Angle at which ellipse will be rotated

##Area of a 95% Home Range Ellipse (Jennrich and Turner 1969, eq. 13):
## Area = 6*pi*sqrt(determinent of var/covar matrix)
##      = 6*pi*SDx*SDy)  (assumes covariance = 0)
Area.yij <- rlnormAlt(1,HRAreamean, HRAreaSD) ##Area of territory/HR for bird j in
   survey site i
##Area.yij generated from a lognormal distribution to prevent negative areas
   (nonsensical)

#rhoij<-runif(1,-0.95, 0.95)  ##rhoij==0 !! covariance=0
#Ecc.yij<- rbetaAlt(1,.8,20)  ##Determine desired eccentricity of HR ij with beta
   distribution
Ecc.yij<-runif(1,0,0.95) ##Determine desired eccentricity of HR ij with uniform
   distribution
a.yij<-((Area.yij^2)/((1-Ecc.yij^2)*pi^2))^(1/4) ##intercept on major axis of similar
   ellipse centered at origin
b.yij<-Area.yij/(pi*a.yij)                     ##intercept on minor axis of similar
   ellipse centered at origin
C<- sqrt(log((1-PercentUD.yij)^-2))  ##(C^2)*pi*sqrt(det(Sigma))=Area of ellipse

SDx.yij<-sqrt((a.yij*Area.yij)/(C^2*pi*b.yij)) ##for 95% UD, C^2=6
SDy.yij<-b.yij*SDx.yij/a.yij

radtheta.yij<- Theta.yij*(2*pi/360)
Xmeanrotated<- CenterX.yij*cos(radtheta.yij) - CenterY.yij*sin(radtheta.yij)
Ymeanrotated<- CenterX.yij*sin(radtheta.yij) + CenterY.yij*cos(radtheta.yij)
Xdif.yij <- CenterX.yij - Xmeanrotated
Ydif.yij <- CenterY.yij - Ymeanrotated
```

```
Sigma<-matrix(c(SDx.yij^2, 0,0,SDy.yij^2),2,2)
eSigma<-eigen(Sigma)
U<-c(CenterX.yij, CenterY.yij)

##Unrotated Axes
UnRMajorAxis1<- U+C*(sqrt(eSigma$values[1])*eSigma$vectors[,1])
UnRMajorAxis2<- U-C*(sqrt(eSigma$values[1])*eSigma$vectors[,1])
UnRMinorAxis1<- U+C*(sqrt(eSigma$values[2])*eSigma$vectors[,2])
UnRMinorAxis2<- U-C*(sqrt(eSigma$values[2])*eSigma$vectors[,2])

##Rotated, unshifted Axes
#RotatePoints(UnRMajorAxis1[1], UnRMajorAxis1[2], Theta.yij)

RMajorAxis1<-RotatePoints(UnRMajorAxis1[1], UnRMajorAxis1[2], Theta.yij)
RMajorAxis2<-RotatePoints(UnRMajorAxis2[1], UnRMajorAxis2[2], Theta.yij)
RMinorAxis1<-RotatePoints(UnRMinorAxis1[1], UnRMinorAxis1[2], Theta.yij)
RMinorAxis2<-RotatePoints(UnRMinorAxis2[1], UnRMinorAxis2[2], Theta.yij)

##Rotated, shifted Axes
RSMajorAxis1.yij<-RMajorAxis1+ c(Xdif.yij,Ydif.yij)
RSMajorAxis2.yij<-RMajorAxis2+ c(Xdif.yij,Ydif.yij)
RSMinorAxis1.yij<-RMinorAxis1+ c(Xdif.yij,Ydif.yij)
RSMinorAxis2.yij<-RMinorAxis2+ c(Xdif.yij,Ydif.yij)

#####
##parameters for OverlapEllipse% UD ellipse, all other parameters same - useful for
   overlap calculation
## NOTE: OE stands for "Overlap Ellipse" in all variable names
AreaOE.yij<-Area.yij*((log((1-OverlapUD)^-2))/(log((1-PercentUD.yij)^-2))) ##Area of an
   OverlapEllipse% UD, other parameters same
```

```
aOE.yij<-((AreaOE.yij^2)/((1-Ecc.yij^2)*pi^2))^(1/4)  ##intercept on major axis of
   OverlapEllipse% UD ellipse centered at origin
bOE.yij<-AreaOE.yij/(pi*aOE.yij) ##intercept on minor axis of OE% UD ellipse centered
   at origin
COE<- sqrt(log((1-OverlapUD)^-2)) ## constant C for an OverlapEllipse% UD ellipse

SDOEx.yij<-sqrt((aOE.yij*AreaOE.yij)/(COE^2*pi*bOE.yij)) ##for 95% UD, C^2=6
SDOEy.yij<-bOE.yij*SDOEx.yij/aOE.yij
SigmaOE<-matrix(c(SDOEx.yij^2, 0,0,SDOEy.yij^2),2,2)
eSigmaOE<-eigen(SigmaOE)
##Unrotated Axes - OverlapEllipse% UD ellipse
UnROEMajorAxis1<- U+COE*(sqrt(eSigmaOE$values[1])*eSigmaOE$vectors[,1])
UnROEMajorAxis2<- U-COE*(sqrt(eSigmaOE$values[1])*eSigmaOE$vectors[,1])
UnROEMinorAxis1<- U+COE*(sqrt(eSigmaOE$values[2])*eSigmaOE$vectors[,2])
UnROEMinorAxis2<- U-COE*(sqrt(eSigmaOE$values[2])*eSigmaOE$vectors[,2])

##Rotated, unshifted Axes - OverlapEllipse% UD ellipse
#RotatePoints(UnROverlapEllipseMajorAxis1[1], UnROverlapEllipseMajorAxis1[2],
   Theta.yij)

ROEMajorAxis1<-RotatePoints(UnROEMajorAxis1[1], UnROEMajorAxis1[2], Theta.yij)
ROEMajorAxis2<-RotatePoints(UnROEMajorAxis2[1], UnROEMajorAxis2[2], Theta.yij)
ROEMinorAxis1<-RotatePoints(UnROEMinorAxis1[1], UnROEMinorAxis1[2], Theta.yij)
ROEMinorAxis2<-RotatePoints(UnROEMinorAxis2[1], UnROEMinorAxis2[2], Theta.yij)

##Rotated, shifted Axes - OverlapEllipse% UD ellipse
RSOEMajorAxis1.yij<-ROEMajorAxis1+ c(Xdif.yij,Ydif.yij)
RSOEMajorAxis2.yij<-ROEMajorAxis2+ c(Xdif.yij,Ydif.yij)
RSOEMinorAxis1.yij<-ROEMinorAxis1+ c(Xdif.yij,Ydif.yij)
RSOEMinorAxis2.yij<-ROEMinorAxis2+ c(Xdif.yij,Ydif.yij)
```

```
  ##Verify with plot that the rotation is correct
#   par(mfrow=c(1,3))
#
   plot(c(CenterXij,UnRMajorAxis1[1],UnRMajorAxis2[1],UnRMinorAxis1[1],UnRMinorAxis2[1]
   ),
#
   c(CenterYij,UnRMajorAxis1[2],UnRMajorAxis2[2],UnRMinorAxis1[2],UnRMinorAxis2[2]),
   pch=16, xlim=c(-15,15), ylim=c(-10,20))
#
#   plot(c(CenterXij,RMajorAxis1[1],RMajorAxis2[1],RMinorAxis1[1],RMinorAxis2[1]),
#        c(CenterYij,RMajorAxis1[2],RMajorAxis2[2],RMinorAxis1[2],RMinorAxis2[2]),
   pch=16, xlim=c(-15,15), ylim=c(-10,20))
#
#   plot(c(CenterXij,RSMajorAxis1[1],RSMajorAxis2[1],RSMinorAxis1[1],RSMinorAxis2[1]),
#        c(CenterYij,RSMajorAxis1[2],RSMajorAxis2[2],RSMinorAxis1[2],RSMinorAxis2[2]),
   pch=16, xlim=c(-15,15), ylim=c(-10,20))
#
return(list("CenterX.yij"=CenterX.yij,
            "CenterY.yij"=CenterY.yij,
            "Area.yij"=Area.yij,
            "Theta.yij"=Theta.yij,
            "a.yij"=a.yij,
            "b.yij"=b.yij,
            "SDx.yij"=SDx.yij,
            "SDy.yij"=SDy.yij,
            "Ecc.yij"=Ecc.yij,
            "PercentUD.yij"=PercentUD.yij,
            "Xdif.yij"=Xdif.yij,
            "Ydif.yij"=Ydif.yij,
```

```
              "RSMajorAxis1.yij"=RSMajorAxis1.yij,
              "RSMajorAxis2.yij"=RSMajorAxis2.yij,
              "RSMinorAxis1.yij"=RSMinorAxis1.yij,
              "RSMinorAxis2.yij"=RSMinorAxis2.yij,
              "RSOEMajorAxis1.yij"=RSOEMajorAxis1.yij,
              "RSOEMajorAxis2.yij"=RSOEMajorAxis2.yij,
              "RSOEMinorAxis1.yij"=RSOEMinorAxis1.yij,
              "RSOEMinorAxis2.yij"=RSOEMinorAxis2.yij,
              "AreaOE.yij"=AreaOE.yij,
              "aOE.yij"=aOE.yij,
              "bOE.yij"=bOE.yij,
              "SDOEx.yij"=SDOEx.yij,
              "SDOEy.yij"=SDOEy.yij
              ))
}

##Generate locations for a bird with territory parameters produced by
     GenSpatialParameters()
GenLocations<-function(NPoints, EllipseParm) ##EllipseParm includes
     CenterX.yij,CenterY.yij,Theta.yij,SDx.yij,SDy.yij)
{ require(MASS)
  EllipseParm<-unlist(EllipseParm)

  CenterX.yij<-EllipseParm["CenterX.yij"]
  CenterY.yij<-EllipseParm["CenterY.yij"]
  Theta.yij<-EllipseParm["Theta.yij"]
  SDx.yij<-EllipseParm["SDx.yij"]
  SDy.yij<-EllipseParm["SDy.yij"]

  ##Generate Bivariate normal points for ellipse
```

```
Sigma.yij<-matrix(c(SDx.yij^2,0,0,SDy.yij^2),2,2) #Covariance matrix
UnrotatedLocations.yijrk<- mvrnorm(NPoints, c(CenterX.yij, CenterY.yij),Sigma.yij)
if(NPoints==1) UnrotatedLocations.yijrk<-matrix(UnrotatedLocations.yijrk,1,2)
dimnames(UnrotatedLocations.yijrk)<-list(NULL,c("X.yijrk", "Y.yijrk"))
#return(UnrotatedLocations.yijrk)

##Rotate Points
RotatedLocations.yijrk<-
   RotatePoints(UnrotatedLocations.yijrk[,"X.yijrk"],UnrotatedLocations.yijrk[,"Y.yijrk
   "], Theta.yij)
#return(RotatedLocations.yijrk)

##Shift points back in place
radtheta<- Theta.yij*(2*pi/360)
Xmeanrotated<- CenterX.yij*cos(radtheta) - CenterY.yij*sin(radtheta)
Ymeanrotated<- CenterX.yij*sin(radtheta) + CenterY.yij*cos(radtheta)
Xdif.yij <- CenterX.yij - Xmeanrotated
Ydif.yij <- CenterY.yij - Ymeanrotated
RotatedAndShiftedLocations.yijrk<-
   cbind("XRS"=(RotatedLocations.yijrk[,"Xnew"]+Xdif.yij),"YRS"=(RotatedLocations.yijrk
   [,"Ynew"]+Ydif.yij))

##Plot unrotated, rotated, and rotated&shifted points
# par(mfrow=c(1,3))
# plot(UnrotatedLocationsijk, ylim=c(-20,20), xlim=c(-20,20))
# lines(CenterXij, CenterYij, col="red", pch=16, type="p")
# plot(RotatedLocationsijk, ylim=c(-20,20), xlim=c(-20,20))
# lines(CenterXij, CenterYij, col="red", pch=16, type="p")
# plot(RotatedAndShiftedLocationsijk, ylim=c(-20,20), xlim=c(-20,20))
# lines(CenterXij, CenterYij, col="red", pch=16, type="p")
```

```
  ##revise dimnames for NPoints=1 to avoid irritating false row name

  if(NPoints==1) {
    RSLdimnames<-dimnames(RotatedAndShiftedLocations.yijrk)
    dimnames(RotatedAndShiftedLocations.yijrk)<-list(NULL,RSLdimnames[[2]])}
  return(RotatedAndShiftedLocations.yijrk)
}

##Generate locations of ellipses for 2 stages of GenLocations() (Before,Rotated,
     Rotates&Shifted)
GenEllipseAll<-function(EllipseParm) #EllipseParm includes CenterXij, CenterYij, aij,
     bij, Thetaij)
{
  EllipseParm<-unlist(EllipseParm)
  CenterX.yij<-EllipseParm["CenterX.yij"]
  CenterY.yij<-EllipseParm["CenterY.yij"]
  a.yij<-EllipseParm["a.yij"]
  b.yij<-EllipseParm["b.yij"]
  Theta.yij<-EllipseParm["Theta.yij"]

  EllipseXs<-(CenterX.yij-a.yij):(CenterX.yij+a.yij) ##Range of x-coordinates for
     plotting Ellipse
  OriginXs<- -a.yij:a.yij                        ##Range of x-coordinates if ellipse were
     plotted at origin instead of centerpoint
  OriginYs<- sqrt((b.yij^2)*abs(1-(OriginXs^2/a.yij^2)))  ##y-coordinates if ellipse were
     plotted at origin instead of centerpoint

  radtheta<- Theta.yij*(2*pi/360)
  Xmeanrotated<- CenterX.yij*cos(radtheta) - CenterY.yij*sin(radtheta)
```

```
  Ymeanrotated<- CenterX.yij*sin(radtheta) + CenterY.yij*cos(radtheta)
  Xdif.yij <- CenterX.yij - Xmeanrotated
  Ydif.yij <- CenterY.yij - Ymeanrotated

  ##Ellipse before rotation
  EllipseBefore1<- cbind("EllipseBefore1X"=EllipseXs,
     "EllipseBefore1Y"=CenterY.yij+OriginYs)
  EllipseBefore2<- cbind("EllipseBefore2X"=EllipseXs[order(-EllipseXs)],
     "EllipseBefore2Y"=CenterY.yij-OriginYs)

  ##Ellipse rotated (around origin)
  EllipseR1<-RotatePoints(EllipseBefore1[,1], EllipseBefore1[,2], Theta.yij)
  EllipseR2<-RotatePoints(EllipseBefore2[,1], EllipseBefore2[,2], Theta.yij)

  ##Ellipse rotated and shifted
  EllipseRS1<-cbind("EllipseRS1X"=EllipseR1[,1]+Xdif.yij,
     "EllipseRS1Y"=EllipseR1[,2]+Ydif.yij)
  EllipseRS2<-cbind("EllipseRS2X"=EllipseR2[,1]+Xdif.yij,
     "EllipseRS2Y"=EllipseR2[,2]+Ydif.yij)

  HH<-
     data.frame(cbind(EllipseBefore1,EllipseBefore2,EllipseR1,EllipseR2,EllipseRS1,Ellips
     eRS2))
  return(HH)
}

##Generate locations of ellipses for plotting ONLY rotated and shifted Ellipses)
GenEllipsePlot<-function(EllipseParm)  ## EllipseParm includes CenterXij, CenterYij, aij,
     bij, Thetaij)
{
```

```
EllipseParm<-unlist(EllipseParm)
CenterX.yij<-EllipseParm["CenterX.yij"]
CenterY.yij<-EllipseParm["CenterY.yij"]
a.yij<-EllipseParm["a.yij"]
b.yij<-EllipseParm["b.yij"]
Theta.yij<-EllipseParm["Theta.yij"]

EllipseXs<-(CenterX.yij-a.yij):(CenterX.yij+a.yij) ##Range of x-coordinates for
   plotting Ellipse
OriginXs<- -a.yij:a.yij                        ##Range of x-coordinates if ellipse were
   plotted at origin instead of centerpoint
OriginYs<- sqrt((b.yij^2)*abs(1-(OriginXs^2/a.yij^2)))  ##y-coordinates if ellipse were
   plotted at origin instead of centerpoint

radtheta<- Theta.yij*(2*pi/360)
Xmeanrotated<- CenterX.yij*cos(radtheta) - CenterY.yij*sin(radtheta)
Ymeanrotated<- CenterX.yij*sin(radtheta) + CenterY.yij*cos(radtheta)
Xdif.yij <- CenterX.yij - Xmeanrotated
Ydif.yij <- CenterY.yij - Ymeanrotated

##Ellipse before rotation
EllipseBefore1<- cbind("EllipseBefore1X"=EllipseXs,
   "EllipseBefore1Y"=CenterY.yij+OriginYs)
EllipseBefore2<- cbind("EllipseBefore2X"=EllipseXs[order(-EllipseXs)],
   "EllipseBefore2Y"=CenterY.yij-OriginYs)

##Ellipse rotated (around origin)
EllipseR1<-RotatePoints(EllipseBefore1[,1], EllipseBefore1[,2], Theta.yij)
EllipseR2<-RotatePoints(EllipseBefore2[,1], EllipseBefore2[,2], Theta.yij)
```

```
  ##Ellipse rotated and shifted
  EllipseRS1<-cbind("EllipseRS1X"=EllipseR1[,1]+Xdif.yij,
     "EllipseRS1Y"=EllipseR1[,2]+Ydif.yij)
  EllipseRS2<-cbind("EllipseRS2X"=EllipseR2[,1]+Xdif.yij,
     "EllipseRS2Y"=EllipseR2[,2]+Ydif.yij)

  EllipseRSAll<- rbind(EllipseRS1, EllipseRS2)

  #HH<-
     data.frame(cbind(EllipseBefore1,EllipseBefore2,EllipseR1,EllipseR2,EllipseRS1,Ellips
     eRS2))
  return(EllipseRSAll)
}

##Generate locations of Overlap Ellipse % UD ellipses for plotting ONLY rotated and
     shifted Ellipses)
GenOEEllipsePlot<-function(EllipseParm)  ## EllipseParm includes CenterXij, CenterYij,
     aij, bij, Thetaij)
{
  EllipseParm<-unlist(EllipseParm)
  CenterX.yij<-EllipseParm["CenterX.yij"]
  CenterY.yij<-EllipseParm["CenterY.yij"]
  aOE.yij<-EllipseParm["aOE.yij"]
  bOE.yij<-EllipseParm["bOE.yij"]
  Theta.yij<-EllipseParm["Theta.yij"]

  EllipseXs<-(CenterX.yij-aOE.yij):(CenterX.yij+aOE.yij) ##Range of x-coordinates for
     plotting Ellipse
  OriginXs<- -aOE.yij:aOE.yij                        ##Range of x-coordinates if ellipse
     were plotted at origin instead of centerpoint
```

```
OriginYs<- sqrt((bOE.yij^2)*abs(1-(OriginXs^2/aOE.yij^2)))  ##y-coordinates if ellipse
   were plotted at origin instead of centerpoint

radtheta<- Theta.yij*(2*pi/360)
Xmeanrotated<- CenterX.yij*cos(radtheta) - CenterY.yij*sin(radtheta)
Ymeanrotated<- CenterX.yij*sin(radtheta) + CenterY.yij*cos(radtheta)
Xdif.yij <- CenterX.yij - Xmeanrotated
Ydif.yij <- CenterY.yij - Ymeanrotated

##Ellipse before rotation
EllipseBefore1<- cbind("EllipseBefore1X"=EllipseXs,
   "EllipseBefore1Y"=CenterY.yij+OriginYs)
EllipseBefore2<- cbind("EllipseBefore2X"=EllipseXs[order(-EllipseXs)],
   "EllipseBefore2Y"=CenterY.yij-OriginYs)

##Ellipse rotated (around origin)
EllipseR1<-RotatePoints(EllipseBefore1[,1], EllipseBefore1[,2], Theta.yij)
EllipseR2<-RotatePoints(EllipseBefore2[,1], EllipseBefore2[,2], Theta.yij)

##Ellipse rotated and shifted
EllipseRS1<-cbind("EllipseRS1X"=EllipseR1[,1]+Xdif.yij,
   "EllipseRS1Y"=EllipseR1[,2]+Ydif.yij)
EllipseRS2<-cbind("EllipseRS2X"=EllipseR2[,1]+Xdif.yij,
   "EllipseRS2Y"=EllipseR2[,2]+Ydif.yij)

EllipseRSAll<- rbind(EllipseRS1, EllipseRS2)

#HH<-
   data.frame(cbind(EllipseBefore1,EllipseBefore2,EllipseR1,EllipseR2,EllipseRS1,Ellips
   eRS2))
```

```
  return(EllipseRSAll)
}



##Check if point (x,y) is inside ellipse.  Returns TRUE / FALSE
XYCheckInsideEllipse<-function(x,y,ExistingEllipseParm)
{
  ExistingEllipseParm<-unlist(ExistingEllipseParm)
  CenterX.yij<-ExistingEllipseParm["CenterX.yij"]
  CenterY.yij<-ExistingEllipseParm["CenterY.yij"]
  a.yij<-ExistingEllipseParm["a.yij"]
  b.yij<-ExistingEllipseParm["b.yij"]
  Theta.yij<-ExistingEllipseParm["Theta.yij"]

  radtheta<- Theta.yij*(2*pi/360)
  A<- (cos(radtheta)^2)/a.yij^2 + (sin(radtheta)^2)/b.yij^2
  B<- 2*cos(radtheta)*sin(radtheta)*(1/a.yij^2 - 1/b.yij^2)
  C<- (sin(radtheta)^2)/a.yij^2 + (cos(radtheta)^2)/b.yij^2
  Verify<- A*x^2+ B*x*y+ C*y^2- (2*A*CenterX.yij + CenterY.yij*B)*x-
     (2*C*CenterY.yij+B*CenterX.yij)*y+
     (A*CenterX.yij^2+B*CenterX.yij*CenterY.yij+C*CenterY.yij^2)
  ifelse(Verify<=1, return(TRUE), return(FALSE))
}


##Check if any of 4 axes points OR center of NewEllipse are inside 1 Existing Ellipse.
##Returns TRUE / FALSE
AxesCheckInsideEllipse<-function(NewEllipseParm,ExistingEllipseParm)
{
```

```
Major1<-unlist(NewEllipseParm[["RSMajorAxis1.yij"]])
Major2<-unlist(NewEllipseParm[["RSMajorAxis2.yij"]])
Minor1<-unlist(NewEllipseParm[["RSMinorAxis1.yij"]])
Minor2<-unlist(NewEllipseParm[["RSMinorAxis2.yij"]])
CenterA<-c(NewEllipseParm[["CenterX.yij"]], NewEllipseParm[["CenterY.yij"]])

Axes<-(rbind(Major1,Major2, Minor1, Minor2, CenterA))

ExistingEllipseParm<-unlist(ExistingEllipseParm)
CenterX.yij<-ExistingEllipseParm["CenterX.yij"]
CenterY.yij<-ExistingEllipseParm["CenterY.yij"]
a.yij<-ExistingEllipseParm["a.yij"]
b.yij<-ExistingEllipseParm["b.yij"]
Theta.yij<-ExistingEllipseParm["Theta.yij"]
Inside<-rep(NA,nrow(Axes))

for(i in 1:nrow(Axes)){
  x<-Axes[i,1]
  y<-Axes[i,2]

  radtheta<- Theta.yij*(2*pi/360)
  A<- (cos(radtheta)^2)/a.yij^2 + (sin(radtheta)^2)/b.yij^2
  B<- 2*cos(radtheta)*sin(radtheta)*(1/a.yij^2 - 1/b.yij^2)
  C<- (sin(radtheta)^2)/a.yij^2 + (cos(radtheta)^2)/b.yij^2
  Verify<- A*x^2+ B*x*y+ C*y^2- (2*A*CenterX.yij + CenterY.yij*B)*x-
   (2*C*CenterY.yij+B*CenterX.yij)*y+
   (A*CenterX.yij^2+B*CenterX.yij*CenterY.yij+C*CenterY.yij^2)
  ifelse(Verify<=1, Inside[i]<-1, Inside[i]<-0)
  ##Above equations from
   http://www.maa.org/external_archive/joma/Volume8/Kalman/General.html
```

```
    ##Also saves as .htm in Detection/REFS/Ellipse Geomoetry & Bivariate Distributions
  }
  ifelse(sum(Inside)==0, return(FALSE), return(TRUE))
}

##Check if any of 4 axes points OR center of NewEllipse are inside multiple Existing
     Ellipses.
##Returns TRUE / FALSE
AxesCheckInsideEllipsesVectorized<-function(NewEllipseParm,ExistingEllipseParm)
{
  ##NewEllipseParm is a GenSpatialParameters generated list
  ##ExistingEllipseParm is a list of GenSpatialParameters lists

  ##Create matrix of Existing Ellipse parameters
  ExistingEllipses<-rbind(unlist(ExistingEllipseParm[[1]]))
  if(length(ExistingEllipseParm)>=2){
    for(ii in 2:length(ExistingEllipseParm)){
      ExistingEllipses<-rbind(ExistingEllipses,
                              rbind(unlist(ExistingEllipseParm[[ii]])))
    }
  }
  Major1<-unlist(NewEllipseParm[["RSMajorAxis1.yij"]])
  Major2<-unlist(NewEllipseParm[["RSMajorAxis2.yij"]])
  Minor1<-unlist(NewEllipseParm[["RSMinorAxis1.yij"]])
  Minor2<-unlist(NewEllipseParm[["RSMinorAxis2.yij"]])
  CenterA<-c(NewEllipseParm[["CenterX.yij"]], NewEllipseParm[["CenterY.yij"]])

  Axes<-(rbind(Major1,Major2, Minor1, Minor2, CenterA))

  CenterX.yij<-ExistingEllipses[,"CenterX.yij"]
```

```
  CenterY.yij<-ExistingEllipses[,"CenterY.yij"]
  a.yij<-ExistingEllipses[,"a.yij"]
  b.yij<-ExistingEllipses[,"b.yij"]
  Theta.yij<-ExistingEllipses[,"Theta.yij"]
  Inside<-list()

  for(i in 1:nrow(Axes)){
    x<-Axes[i,1]
    y<-Axes[i,2]

    radtheta<- Theta.yij*(2*pi/360)
    A<- (cos(radtheta)^2)/a.yij^2 + (sin(radtheta)^2)/b.yij^2
    B<- 2*cos(radtheta)*sin(radtheta)*(1/a.yij^2 - 1/b.yij^2)
    C<- (sin(radtheta)^2)/a.yij^2 + (cos(radtheta)^2)/b.yij^2
    Verify<- A*x^2+ B*x*y+ C*y^2- (2*A*CenterX.yij + CenterY.yij*B)*x-
     (2*C*CenterY.yij+B*CenterX.yij)*y+
     (A*CenterX.yij^2+B*CenterX.yij*CenterY.yij+C*CenterY.yij^2)
    ##if Verify<=1, an axis point from the new ellipse is inside an existing ellipse
    ##Above equations from
     http://www.maa.org/external_archive/joma/Volume8/Kalman/General.html
    ##Also saved as .htm in Detection/REFS/Ellipse Geomoetry & Bivariate Distributions
    Inside[[i]]<-sum(Verify<=1)
  }
  ifelse(sum(unlist(Inside))==0, return(FALSE), return(TRUE))
}


####

##Check if any of 4 axes points OR center of Overlap Ellipse % UD NewEllipse are
```

```
## inside 1 Existing Ellipse (Overlap Ellipse % UD too).
##Returns TRUE / FALSE
AxesCheckInsideEllipseOE<-function(NewEllipseParm,ExistingEllipseParm)
{
  Major1<-unlist(NewEllipseParm[["RSOEMajorAxis1.yij"]])
  Major2<-unlist(NewEllipseParm[["RSOEMajorAxis2.yij"]])
  Minor1<-unlist(NewEllipseParm[["RSOEMinorAxis1.yij"]])
  Minor2<-unlist(NewEllipseParm[["RSOEMinorAxis2.yij"]])
  CenterA<-c(NewEllipseParm[["CenterX.yij"]], NewEllipseParm[["CenterY.yij"]])

  Axes<-(rbind(Major1, Major2, Minor1, Minor2, CenterA))

  ExistingEllipseParm<-unlist(ExistingEllipseParm)
  CenterX.yij<-ExistingEllipseParm["CenterX.yij"]
  CenterY.yij<-ExistingEllipseParm["CenterY.yij"]
  a.yij<-ExistingEllipseParm["aOE.yij"]
  b.yij<-ExistingEllipseParm["bOE.yij"]
  Theta.yij<-ExistingEllipseParm["Theta.yij"]
  Inside<-rep(NA,nrow(Axes))

  for(i in 1:nrow(Axes)){
    x<-Axes[i,1]
    y<-Axes[i,2]

    radtheta<- Theta.yij*(2*pi/360)
    A<- (cos(radtheta)^2)/a.yij^2 + (sin(radtheta)^2)/b.yij^2
    B<- 2*cos(radtheta)*sin(radtheta)*(1/a.yij^2 - 1/b.yij^2)
    C<- (sin(radtheta)^2)/a.yij^2 + (cos(radtheta)^2)/b.yij^2
```

```
    Verify<- A*x^2+ B*x*y+ C*y^2- (2*A*CenterX.yij + CenterY.yij*B)*x-
     (2*C*CenterY.yij+B*CenterX.yij)*y+
     (A*CenterX.yij^2+B*CenterX.yij*CenterY.yij+C*CenterY.yij^2)
    ifelse(Verify<=1, Inside[i]<-1, Inside[i]<-0)
    ##Above equations from
     http://www.maa.org/external_archive/joma/Volume8/Kalman/General.html
    ##Also saves as .htm in Detection/REFS/Ellipse Geomoetry & Bivariate Distributions
  }
  ifelse(sum(Inside)==0, return(FALSE), return(TRUE))
}

##Check if any of 4 axes points OR center of Overlap Ellipse % UD NewEllipse are inside
##multiple Existing Overlap Ellipse % UD Ellipses.
##Returns TRUE / FALSE
AxesCheckInsideEllipsesVectorizedOE<-function(NewEllipseParm,ExistingEllipseParm)
{
  ##NewEllipseParm is a GenSpatialParameters generated list
  ##ExistingEllipseParm is a list of GenSpatialParameters lists

  ##Create matrix of Existing Ellipse parameters
  ExistingEllipses<-rbind(unlist(ExistingEllipseParm[[1]]))
  if(length(ExistingEllipseParm)>=2){
    for(ii in 2:length(ExistingEllipseParm)){
      ExistingEllipses<-rbind(ExistingEllipses,
                              rbind(unlist(ExistingEllipseParm[[ii]])))
    }
  }
  Major1<-unlist(NewEllipseParm[["RSOEMajorAxis1.yij"]])
  Major2<-unlist(NewEllipseParm[["RSOEMajorAxis2.yij"]])
  Minor1<-unlist(NewEllipseParm[["RSOEMinorAxis1.yij"]])
```

```
Minor2<-unlist(NewEllipseParm[["RSOEMinorAxis2.yij"]])
CenterA<-c(NewEllipseParm[["CenterX.yij"]], NewEllipseParm[["CenterY.yij"]])

Axes<-(rbind(Major1,Major2, Minor1, Minor2, CenterA))

CenterX.yij<-ExistingEllipses[,"CenterX.yij"]
CenterY.yij<-ExistingEllipses[,"CenterY.yij"]
a.yij<-ExistingEllipses[,"aOE.yij"]
b.yij<-ExistingEllipses[,"bOE.yij"]
Theta.yij<-ExistingEllipses[,"Theta.yij"]
Inside<-list()

for(i in 1:nrow(Axes)){
  x<-Axes[i,1]
  y<-Axes[i,2]

  radtheta<- Theta.yij*(2*pi/360)
  A<- (cos(radtheta)^2)/a.yij^2 + (sin(radtheta)^2)/b.yij^2
  B<- 2*cos(radtheta)*sin(radtheta)*(1/a.yij^2 - 1/b.yij^2)
  C<- (sin(radtheta)^2)/a.yij^2 + (cos(radtheta)^2)/b.yij^2
  Verify<- A*x^2+ B*x*y+ C*y^2- (2*A*CenterX.yij + CenterY.yij*B)*x-
   (2*C*CenterY.yij+B*CenterX.yij)*y+
   (A*CenterX.yij^2+B*CenterX.yij*CenterY.yij+C*CenterY.yij^2)
  ##if Verify<=1, an axis point from the new ellipse is inside an existing ellipse
  ##Above equations from
   http://www.maa.org/external_archive/joma/Volume8/Kalman/General.html
  ##Also saved as .htm in Detection/REFS/Ellipse Geomoetry & Bivariate Distributions
  Inside[[i]]<-sum(Verify<=1)
}
ifelse(sum(unlist(Inside))==0, return(FALSE), return(TRUE))
```

```
}


## Determine closest axis point to the observer
## Useful for determining if territory overlaps survey radius
## Note: will use axes for PercentUD.yij used to create territory ExistingEllipseParm
## (Default = 95% UD)
AxesClosestObs<-function(ExistingEllipseParm){
  AxesDistances<-
     c(unname(unlist(ObsDistance(ExistingEllipseParm[["RSMajorAxis1.yij"]][1],ExistingEll
     ipseParm[["RSMajorAxis1.yij"]][2])["Distance.yijrk"])),

     unname(unlist(ObsDistance(ExistingEllipseParm[["RSMajorAxis2.yij"]][1],ExistingEllip
     seParm[["RSMajorAxis2.yij"]][2])["Distance.yijrk"])),

     unname(unlist(ObsDistance(ExistingEllipseParm[["RSMinorAxis1.yij"]][1],ExistingEllip
     seParm[["RSMinorAxis1.yij"]][2])["Distance.yijrk"])),

     unname(unlist(ObsDistance(ExistingEllipseParm[["RSMinorAxis2.yij"]][1],ExistingEllip
     seParm[["RSMinorAxis2.yij"]][2])["Distance.yijrk"])))
  return(min(AxesDistances))
}


#########################
###Availability Functions
#########################

##Generate DidBirdSing for a single interval with a Markov Process
AutoCInstant<- function(DidBirdSingLastTime, PSS, PSNS){
```

```
  ifelse(DidBirdSingLastTime==1,
         X<-rbinom(1,1,PSS),
         X<-rbinom(1,1,PSNS))
  return(X)
}

##Determine steady state vector [q1 q2] for Markov process
##As t->infinity, Pr(Singing Mode=1)=q1, Pr(Singing Mode=0)=q2
##Also, q1 = proportion of birds in Singing Mode at any given time
MarkovSS<- function(PSS,PSNS){
  ##PSS= Pr(Singing Mode=1, given that it was 1 at t-1)
  ##PSNS= Pr(Singing Mode=1, given that it was 0 at t-1)

  pmatrix<- rbind(c(PSS, 1-PSS),
                  c(PSNS, 1-PSNS))  ##transition matrix

  q1<- (-1*pmatrix[2,1])/(pmatrix[1,1]-pmatrix[2,1]-1)
  q2<- (-1*pmatrix[1,2])/(pmatrix[2,2]-pmatrix[1,2]-1)

  return(c(q1,q2))
}

##Determine transition matrix value P(S|NS), given desired steady state vector & P(S|S)
MarkovTM<- function(q1, PSS){
  q2<- 1-q1
  PSNS<- (-1*q1*(PSS-1))/q2
  return(PSNS)
}

##Function for finding PSNS via optimization
```

```
##NOT a standalone function
SolvePSNS<-function(Par,PSNS,PrSing, PSS, DataIntervals){
  PSNS<-Par[1]
  qq<- {(PSNS/(1-PSS+PSNS) + (1-(PSNS/(1-PSS+PSNS)))*(1-((1-PSNS)^DataIntervals))) -
     PrSing}
  return(abs(qq))
}
##Example:
#optimize(SolvePSNS, interval=c(0,0.05), PrSing=0.6, PSS=0.92, DataIntervals=200,
     maximum=F)
##NOTE: **Starting values** are VERY important here
##For PSS=0.99 and PSS=0.92, all optimal values of P(S|NS) < 0.02
##Suggest using interval=c(0,0.05) for all
##DataIntervals refers to number of intervals in X minutes in Pr(bird sings at least once
     in X minutes), NOT NIntervals


##Determine the Pr(sing) within a survey,
##given NIntervals, P(S|S), and P(S|NS).
## 2 methods used, both should return equal results.
ZNIntervals1<-function(PSS1=0.98, PSNS1=0.00264495, NIntervals=90){
  -1*PSNS1/(PSS1-PSNS1-1) + (1-(-1*PSNS1/(PSS1-PSNS1-1)))*(1-(1-PSNS1)^NIntervals)
}

ZNIntervals2<-function(q1=0.1168009,PSNS1=0.00264495, NIntervals=90){
  ## q1 = steady-state proportion of birds in Singing Mode at any given time
  q1+ (1-q1)*(1-(1-PSNS1)^NIntervals)
}
# ZNIntervals1()
# ZNIntervals2()
```

```
##Convert Probability a bird sings at least once in A min (PrSingA) to
##  the Probability the bird sings at least once in B min (PrSingB).
##NOTE: Requires function SolvePSNS, MarkovSS, ZNIntervals1, and ZNIntervals2

ConvertPRSing<-function(PrSing.A=0.75, Minutes.A=10,Minutes.B=5,
                        PSS.yijrA=0.98,IntervalLengthA=2){

  ##Use the same optimization routine from Simulation to determine PSNS

  ##Here, PrSing.yijrk refers to Pr(bird sings at least 1x in ReportedPrSingMin min
     (usually 5 or 10))
  ##Therefore, use DataIntervals = ReportedPrSingMin (min) *60
     (sec/min)*(1/IntervalLength) (Intervals/sec)
  ##units for DataIntervals: (min) * (60sec/min) * (Intervals/sec) = Intervals
  ##PrSing calc is AFTER spatial to allow Pa ~ Distance to obs

  OptimizePSNS.yijrkA<- optimize(SolvePSNS,
                                 interval=c(0,0.05),   ##NOTE: **Starting values** are
     VERY important here, suggest using interval=c(0,0.05) for all
                                 PrSing=PrSing.A,
                                 PSS=PSS.yijrA,
                                 DataIntervals=Minutes.A*60*(1/IntervalLengthA),
                                 maximum=F)
  if(OptimizePSNS.yijrkA$objective>0.05) {stop("Failure to optimize PSNS")}
  PSNS.yijrkA<- OptimizePSNS.yijrkA$minimum
  q1.yijrkA<-MarkovSS(PSS.yijrA,PSNS.yijrkA)[1]
```

```
  ## Using the parameters you just estimated,
  ## estimate PrSingB 2 ways (these should be equal!)
  PrSing.B1<-ZNIntervals1(PSS1=PSS.yijrA, PSNS1=PSNS.yijrkA,
     NIntervals=Minutes.B*60*(1/IntervalLengthA))
  PrSing.B2<-ZNIntervals2(q1=q1.yijrkA,PSNS1=PSNS.yijrkA,
     NIntervals=Minutes.B*60*(1/IntervalLengthA))
  return(c(PrSing.B1, PrSing.B2))
}


#########################
###Perceptability Functions
#########################


## Produce an observer-estimated distance to bird j, given a known
## true distance between the observer and bird j.
## To output mean distance estimate, use Output="meanonly" (probabilistic answer).
## To output SD for error estimate, use Output="sdonly" (for plotting).
## To stochastically generate distance, use Output="stochastic".
##NOTE: More Complex version - NOT CURRENTLY USED
## DHJ felt the odd error curve in Alldredge could be situational
## Simplified version (above) used intead

ObserverEstDistance<-function(TrueDistance,
                              DistanceCategories,
                              Output="stochastic"){
  if(TrueDistance<0) stop("ObserverEstDistance() ERROR: Distance input cannot be
     negative")
  if(TrueDistance<max(DistanceCategories$Distance)){
```

```
  Row<-sum(DistanceCategories$Distance<=TrueDistance)

  ## Determine mean error for x=TrueDistance assuming a straight line
  ## between points (x1,y1) & (x2,y2)
  x1<-DistanceCategories[Row,"Distance"]
  x2<-DistanceCategories[Row+1,"Distance"]
  y1<-DistanceCategories[Row,"meanerror"]
  y2<-DistanceCategories[Row+1,"meanerror"]

  ## Equation for a straight line between points (x1,y1) & (x2,y2), for point
   x=TrueDistance
  MeanError<-y1+((y2-y1)/(x2-x1))*(TrueDistance-x1)

  ## Determine SD error for x=TrueDistance assuming a straight line
  ## between points (x1,y1) & (x2,y2)
  x3<-DistanceCategories[Row,"Distance"]
  x4<-DistanceCategories[Row+1,"Distance"]
  y3<-DistanceCategories[Row,"sderror"]
  y4<-DistanceCategories[Row+1,"sderror"]

  ## Equation for a straight line between points (x1,y1) & (x2,y2), for point
   x=TrueDistance
  SDError<-y3+((y4-y3)/(x4-x3))*(TrueDistance-x3)
}

## to extrapolate beyond existing distance data, use values for greatest data point
if(TrueDistance>=max(DistanceCategories$Distance)){
  MeanError<-
   DistanceCategories[DistanceCategories$Distance==max(DistanceCategories$Distance),"me
   anerror"]
```

```
    SDError<-
     DistanceCategories[DistanceCategories$Distance==max(DistanceCategories$Distance),"sd
     error"]

  }
  ## Mean observer-estimated distance for TrueDistance (probabilistic estimate)
  MeanEstDistance<-TrueDistance+MeanError

  ## Stochastic observer-estimated distance for TrueDistance
  StoEstDistance<-TrueDistance+rnorm(1,MeanError,SDError)

  ## Output the mean distance estimated for point x=TrueDistance
  if(Output=="meanonly") return(MeanEstDistance) ##Output the mean distance estimated for
     point x=TrueDistance
  if(Output=="stochastic") return(StoEstDistance)   ##Output a stochastic value for
     distance estimated for point x=TrueDistance
  if(Output=="sdonly") return(SDError)

}


#########################
### Estimators & Likelihoods
#########################

##Likelihood estimators
## The following functions are not standalone functions
## They are designed to estimate paramters via optim() or optimx()
```

```
#####################################
## MLEstimator for Multiple observer survey (Nichols 2000)
## This is a closed-form estimator:
##  it produces estimates without optim() .

MultObs.Nichols<-function(x11, x21, x22, x12){
  ## x11 = seen by obs 1 on stops when obs 1 was primary
  ## x21 = seen by obs 2 on stops when obs 1 was primary
  ## x22 = seen by obs 2 on stops when obs 2 was primary
  ## x12 = seen by obs 1 on stops when obs 2 was primary
  x11<-sum(x11)
  x22<-sum(x22)
  x12<-sum(x12)
  x21<-sum(x21)

  if(x11==0 | x22==0)print("N & p not calculable: One or more observers had Count=0 when primary observer.")

  p1hat<-(x11*x22-x12*x21)/(x11*x22+x22*x21)
  p2hat<-(x11*x22-x12*x21)/(x11*x22+x11*x12)
  phat<- 1-(x12*x21)/(x22*x11)
  xdotdot<-x11+x21+x22+x12
  Nhat<-xdotdot/phat
  return(list("p1hat"=p1hat,
              "p2hat"=p2hat,
              "phat"=phat,
              "xdotdot"=xdotdot,
              "Nhat"=Nhat))
}
```

```
####################################
##Optimizable estimator for Farnsworth Removal method
## 3 intervals: 3 min,2 min,5 min (10 min total)

FarnsworthRemoval10min.negLL<-
  function(Param,xes){
    #function(cc,qq,xes){#(Param,xes){ ##for mle2
    cc=Param[1]
    qq=Param[2]

#     cc<-0.3
#     qq<-0.7

    x1<-xes[1]
    x2<-xes[2]
    x3<-xes[3]

#     xes<-II.data[,c("FarnsRemovalPeriod1.yi",
#                "FarnsRemovalPeriod2.yi",
#                "FarnsRemovalPeriod3.yi")]

    NLL.A<- log(1-cc*qq^3)-log(1-cc*qq^10)
    NLL.B<- log(cc*qq^3)+log(1-qq^2)-log(1-cc*qq^10)
    NLL.C<- log(cc*qq^5)+log(1-qq^5)-log(1-cc*qq^10)
    NLL.All<- -1*(x1*NLL.A + x2*NLL.B + x3*NLL.C)

    return(sum(NLL.All))#NegLL)#negLL)
  }
```

```
#######################################################
##Function to calculate geometric mean rate of change
##From Link and Sauer 1998 "Estimating population change from count data..."
##Geometric mean rate of change = Bi(ta,tb)-1 (top R of pg 261)
##From eq.3, Bi(ta,tb)=exp[(hi(tb)-h(ta))/(tb-ta)]
##Note: assumes that time interval is from Year 1 of survey to year B (year B must be >1)
GeoRateChange<-function(PopulationYeara,PopulationYearb,Yearb){
  ##Because zero values screw up the trend estimates,
  ## add 1 to each abundance if only one is a zero.
  ##If both=0, it will produce an NaN (no information about trend).
  ##NOTE: this works with single values but did not work for vectors
  ## Moved to HabSimA6d script instead.
  # if(PopulationYeara==0 & PopulationYearb>0 |
  #    PopulationYeara>0 & PopulationYearb==0){
  #   PopulationYeara<-PopulationYeara+1
  #   PopulationYearb<-PopulationYearb+1
  # }
  TimeInterval<-Yearb-1  ##(tb-ta)
  hi.ta<-0  ##by definition of trajectory
  hi.tb<-log(PopulationYearb)-log(PopulationYeara) ##from eq. 1
  B.ta.tb<-exp((hi.tb-hi.ta)/TimeInterval)  ##eq. 3
  GeoMeanRateChange<-B.ta.tb-1 ##geo mean annual rate of change
  return(GeoMeanRateChange)
}

#####END#FUNCTIONS#############################################

#############################
##PARAMETERS FOR BTBW SCENARIO (CH.1)
```

```
## Load necessary packages
library(arm)
library(reshape)

#############################
##Scenario Parameters
{
  ##Survey Type
  #SurveyType<-    "multiple" # #"removal" #"nmixture" # "distance" #
  #NSurveySites<- 1  ## No. of survey sites surveyed within one iteration (year, etc.)

  ##Basic Modeling parameters
  NYears<-1           ## No. of Years or Seasons across which surveys take place
  NReps<- ifelse(SurveyType=="nmixture",3,1)          ## No. of replications (No. of
     times each survey site is surveyed WITHIN season)
  IntervalLength <- 2 ## No. of SECONDS or one unit of time during which bird may
     vocalize (or not)
  IntervalsPerHour<-(60*60)/IntervalLength  ##No. of intervals in 1 hour
  IntervalsPerMinute<-IntervalsPerHour/60  ##No. of intervals in 1 MINUTE
  SurveyLength<- ifelse(SurveyType=="removal",10,3)   ## No. of MINUTES that 1 bird
     survey lasts
  NIntervals<- (SurveyLength*60)/IntervalLength ##No. of intervals of IntervalLength that
     make up each survey
  if(as.integer(NIntervals)!=NIntervals){stop("ERROR - Number of intervals must be an
     integer.  Adjust parameter SurveyLength or IntervalLength.")}

  if(SurveyType=="removal"){
    RemovalPeriods<-3
    RemovalPeriod1Length<-2
    RemovalPeriod2Length<-3
```

```
  RemovalPeriod3Length<-5

  FarnsRemovalPeriod1Length<-3
  FarnsRemovalPeriod2Length<-2
  FarnsRemovalPeriod3Length<-5

  RemovalPeriod1Intervals<-(RemovalPeriod1Length/SurveyLength)*NIntervals
  RemovalPeriod2Intervals<-(RemovalPeriod2Length/SurveyLength)*NIntervals
  RemovalPeriod3Intervals<-(RemovalPeriod3Length/SurveyLength)*NIntervals

  FarnsRemovalPeriod1Intervals<-(FarnsRemovalPeriod1Length/SurveyLength)*NIntervals
  FarnsRemovalPeriod2Intervals<-(FarnsRemovalPeriod2Length/SurveyLength)*NIntervals
  FarnsRemovalPeriod3Intervals<-(FarnsRemovalPeriod3Length/SurveyLength)*NIntervals


   if(NIntervals!=sum(RemovalPeriod1Intervals,RemovalPeriod2Intervals,RemovalPeriod3Intervals) |

   NIntervals!=sum(FarnsRemovalPeriod1Intervals,FarnsRemovalPeriod2Intervals,FarnsRemovalPeriod3Intervals)){
    stop("Intervals do not evenly divide among Removal periods")
  }
}

Xlim<- c(-1000,1000)  #c(-500,500)     ##Min and max x coordinate for xy grid generated around observer
Ylim<- c(-1000,1000) #c(-500,500)   ##Min and max y coordinate for xy grid generated around observer
#EntireAreaRadius <-  2000 ## Radius (m) at which birds are modeled (so they can enter/leave survey site)
```

```
  #   SurveyAreaRadius<- 100 ##Fixed-distance cutoff (m) for observation during survey

#############################
  ##Population parameters
  ContinentalPopulation<- 2100000/2 #rep(2100000/2, NYears) ##Total continental
     Population of Species
  ## estimate from Partners in Flight - divide by 2 to get number of males
  ContinentalPopulationCoef<-rep(1,NYears)#c(0.1, 0.5, 1, 2, 5, 0.1, 0.5, 1, 2, 5)
  ContinentalPopulationAll.vector<-sort(ContinentalPopulationCoef*ContinentalPopulation)

  HabitatPreference.y<-0.9 ## % of total population found in habitat (as opposed to
     matrix) across entire range
  RangeArea<-3.62*10^11  ##(m^2)  Size of entire species' range (7*10^11 m^2 is the
     approx. size of Texas)
  ##RangeArea from http://www.birdlife.org/datazone/species/factsheet/22721673
  StudyArea<-3000*10000  ##(m^2) Size of study area (ps ~ StudyArea/RangeArea)
  ##NOTE: StudyArea is 10x size of Hubbard Brook
  ##http://www.hubbardbrook.org/overview/overview.shtml
    HabitatDensity.y<- 0.534/10000#0.62/10000 ##Mean density (birds/m^2) for habitat
     within species range
  ## from Holmes 1986, adjusted from individuals/10HA to males/m^2
  #   DensityCoef<-c(0.1, 0.5, 1, 2, 5, 0.1, 0.5, 1, 2, 5)
  #   HabitatDensity<-HabitatDensityMean*DensityCoef
  #

  RangeHabitatProportion.vector<-
     (HabitatPreference.y*ContinentalPopulationAll.vector)/(HabitatDensity.y*RangeArea)
     ## % of species' entire range that is habitat (as opposed to matrix)
  #   StudyHabitatProportion.vector<-rep(0.95,NYears) #RangeHabitatProportion.vector ## %
     of study area that is habitat (as opposed to matrix)
```

```
StudyHabitatProportion<-function(N=1, min, max){  ##Distribution for mean %Habitat for
   entire study area
  Value<-runif(N, min, max)
  return(Value)
}
StudyHabitatProportionmin<-0.7  ##Study area is between 70% and 100% habitat (mean for
   each Fauxyear)
StudyHabitatProportionmax<-1
##If study area is range-wide, use HabitatProportionSurveyedArea.y<-
   RangewideHabitatProportion.y
##If study area is a subset of the range, use expected mean % habitat among sites
   (e.g., we picked sites with ~60% habitat)

#PercentHabitatStudy<-  ## % of total habitat within species' range that is contained
   by study area
#  (StudyHabitatProportion.vector*StudyArea)/(RangeHabitatProportion.vector*RangeArea)


##BTBW pop. estimate=2,100,000 PAIRS, from Partners in Flight:
   http://rmbo.org/pifpopestimates/Database.aspx
#Ps <- 1             ## Proportion of ContinentalPopulationAll.vector that can be found
   in Area of Inference (AOI)
#AreaofInference<- 3.62*10^11  ##(m^2)  Size of Area of Inference (AOI),  (7*10^11 m^2
   is the approx. size of Texas)
#HabitatProportionAll<- rep(0.5,NYears)    ##Avg. % of AOI occupied by habitat
##HabitatProportionAll is the vector of %AOI occupied by habitat, length=NYears
HabitatProportionThetaAll<- rep(8,NYears)   ##Concentration parameter for Beta
   distribution - see rbetaAlt function
```

```
##########################
##Environmental / Temporal paramenters


  ##Survey timing set up
  StartingJulianDate<-150  ##First day of surveys
  SurveysPerDay<-ifelse(SurveyType=="removal",6,7)
  LogisticalSurveyTime <-ifelse(SurveyType=="removal",30,23)  ## mean MINUTES taken to
     conduct 1 survey, including travel time

  ##NOTE: DailySurveyStartTime not currently used - see PlannedStartTimesAll in "A3 -
     Simulation Script.R"
  DailySurveyStartTime <-0 ## mean minutes past sunrise when surveys begin daily
  # StartingTimes<- round(runif(10000,min=5,max=11),2) ##Hours, Military time

  EndingJulianDate<-220  ##Last possible day of surveys

  ##Biological Parameters
  ##Currently, Leaves are PRESENT for all surveys
  FirstLeavesDay<-StartingJulianDate #+ 10  ##First day when Leaves=1
  LeavesByDate<-c(rep(0,FirstLeavesDay-1), ##before Leaf Out
                  rep(1,(EndingJulianDate-FirstLeavesDay+1)))
  ##LeavesByDate is an index, where LeavesByDate[Day] indicates if there were leaves
     (0/1) on Julian Date "Day"

  #SpecialistIndex<- 0.6        ##(from 0-1) Index of specialist 0=Generalist, 1=
     Specialist
  BackgroundMovementRate<- 0.005  ## Species-specific Lowest reasonable probability of
     bird moving between interval t=0 and t=1
  ## 0.005 = medium mobility value from Granholm 1983
```

```
  MeanTerrArea<- 3.60*10000 ##3.6HA, from Sherry and Holmes 1985, pg 288 in "Habitat
     Selection in birds", Martin Cody ed. (m^2)
  SDTerrArea<-   10000 #1 HA            ##SD of territory area for this species (m^2)
  OverlapUDTerr<- 0.565  ## %Utilization Distribution at which overlap of territories is
     accessed
  UDTerr<-0.95  ##% Utilization Distribution modeled for all final bird territories

##########################################
  ##Perceptibility parameters
  PrGrassland.y<- rep(0,NYears) ##Probability that site i is grassland (as opposed to
     forest)
  PrDeciduous.y<- rep(0,NYears) ##If site i is forest, Probability that site i is
     Deciduous (as opposed to Mixed Pine-Deciduous)

  ##  NoiseLevelDistrib<-round(runif(10000,0,10),2) # runif(1,0,10)
  #   NoiseLevelMin<-0
  #   NoiseLevelMax<-10

  ##NOTE: Ambient Noise is currently BINARY at the amplitude Alldredge tested
  PrNoise<-0.15

  DailyMeanWindSpeedMin<-0.01  ##using true zero can create errors
  DailyMeanWindSpeedMax<-10
  RepMeanWindSpeedSD<-0.5

  ##MeanWindSpeedDistrib<-round(runif(10000,0,10),2) ##WindSpeed distribution
     (note:WindSpeed>0)
  WindVariability<- 0.94  ##Medium variability, from Justus 1977
  ##1.05 ##Low variability, from Justus 1977
  ##0.83  ##High variability, from Justus 1977
```

```
SpeciesIntercept<- 11.816 ## Logit(Pd) intercept for species in scenario of interest

WindIndexIEffect<- 0
GrasslandIEffect<- 0 ##effect of grassland categorical variable on intercept of
   Logit(Pd)
DeciduousIEffect<- 0  ##effect of deciduous categorical variable on intercept of
   Logit(Pd)
LeavesIEffect<- 1.47 ##effect of leaves categorical variable on intercept of Logit(Pd)
NoiseIEffect<- 0.1035 ##effect of ambient noise on intercept of Logit(Pd)
DistanceEffect.yijr <- -0.0644 ## mean slope of Logit(Pd)
WindIndexSEffect<- 0
GrasslandSEffect<- 0 ##effect of grassland categorical variable on slope of Logit(Pd)
   (interaction of Distance and grassland)
DeciduousSEffect<- 0 ##effect of deciduous categorical variable on slope of Logit(Pd)
   (interaction of Distance and Deciduous)
LeavesSEffect<- -0.0444 ##effect of leaves categorical variable on slope of Logit(Pd)
   (interaction of Distance and Leaves)
NoiseSEffect<-  -0.0233 ##effect of ambient noise on slope of Logit(Pd)
NoiseXLeavesEffect<- -0.9528
NTotalObservers<-2
NSimultaneousObservers<-ifelse(SurveyType=="multiple",2,1)
ObserverVariation<-0.4097  ##0.4097=0.5*SD of observers for BTBW (Pacifici 2008)

ObserverIDs<-LETTERS[1:NTotalObservers]
ObserverIEffects<-data.frame("A"=0.002425+0.4097, "B"=0.002425-0.4097, "C"=0.002425,
   "D"=0.002425, "E"=0.002425, "F"=0.002425, "G"=0.002425)  ##All Observers have
   average skill
ObserverSEffects<-data.frame("A"=0, "B"=0, "C"=0, "D"=0, "E"=0, "F"=0, "G"=0)
```

```
  PrCorrectID<- 1  ##Pr(Bird is correctly ID'd)
  PrDoubleCount<-0 ##Pr(1 bird is counted as 2)

#######################################
  ##Observer estimation of distance
  ObserverDistanceCategories<-data.frame(
    ## From Alldrege et al. 2007 "A field evaluation of distance measurement error..."

    ##Alldredge distance-dependent error curve:
    #     "Distance"=c(0,23,37,52,65,75,86,98),
    #     "meanerror"=c(0,18.5,-1.6,8.8,14.8,5.5,-2.1,-7.9),
    #     "sderror"=c(0,18.9,11,22.3,22.1,19.2,17.4,16.8))

    ##Alldredge error curve based on overall mean & SD error:
    "Distance"=c(0,62.286),
    "meanerror"=c(0,7.6),
    "sderror"=c(0,21.4))

  ##Distance above uses mean of measured distances in Alldredge (2007)= 62.286
  ##mean(c(23,37,52,65,75,86,98)) == 62.28571
  #
  #   #Plot the mean & SD error for Obs estimated Distance using above values
  #   plot(NULL, NULL, xlim=c(0,210), ylim=c(0,210))
  #   for(ii in 1:200){
  #     MEAN<-ObserverEstDistance(ii,ObserverDistanceCategories,Output="meanonly")
  #     SDD<-ObserverEstDistance(ii,ObserverDistanceCategories,Output="sdonly")
  #     points(ii,MEAN)
  #     points(ii,MEAN+SDD, col="blue")
  #     points(ii,MEAN-SDD,col="blue")
  #   }
```

```
  #   lines(c(0,200), c(0,200))
  #
  ##PLACEHOLDER FOR PrSing DISTRIBUTION
  ##Currently using *functions* JulDate & TimeOfDay to approximate PrSing

  ##Coarse-scale Markov song Parameters

  PSS.yijr<-0.98    ##Pr(Bird sings during next interval, given that it did sing before)

  ##Fine-scale Markov song parameters (Species/Scenario-specific)
  SongLength<-2.1/IntervalLength ##Number of intervals for avg. song length (interval=2
     sec)
  PauseLength<-6.6/IntervalLength ##Number of intervals for avg. pause between songs
  q1.fine<- SongLength/(SongLength+PauseLength)
  #   PSS.fine<- 0.001
  #   PSNS.fine<- MarkovTM(q1.fine, PSS.fine)
  SingingStates.fine<- c("S", "NS1", "NS2", "NS3")
  TransitionMatrix.fine<-matrix(c(0,0,0.08,0.8,1,0,0,0,0,1,0,0,0,0,0.92,0.2), nrow=4,
     ncol=4)  ##Transition matrix for fine-scale autocorrelation of Singing
  ## above TransitionMatrix.fine produces Pauses w/mean length 6.3 seconds (SD=1.25 sec,
     q1=0.241) (BTBW values)
  ## Values determined via simulation (file="Fine-scale autocorrelation - Simulation to
     determine values of 4-stage transition matrix.R")

  ReportedPrSingMin<-10  ## x minutes in reported song rates: Pr(Bird j sings w/in x
     minutes)
  ##PrSing.yijrk refers to Pr(bird sings at least 1x in ReportedPrSingMin min (usually 5
     or 10))

}
```

```
################################################################
## Scenario Distributions
################################################

##Generate a k-specific probabiliy of movement
PrBirdMoves<-function(N=1,MeanMovementRate){
  PrBirdMoves.yijrk<-MeanMovementRate+rnorm(N,0,0.0005)
  if(sum(PrBirdMoves.yijrk<0)>0) stop("Negative values generated for PrBirdMoves -
     reevaluate variation!")
  return(abs(PrBirdMoves.yijrk))
}
#hist(PrBirdMoves(10000,BackgroundMovementRate))

#######
##Generate probability that a bird flushes due to observer
PrBirdMoveObs<-function(Intercept=1,Slope=-0.1, Distance){
  PrMoveObs<- invlogit(Intercept+Slope*Distance)
  return(PrMoveObs)
}
##Plot Pr(Flush)~Distance
#plot(1:200,PrBirdMoveObs(Distance=1:200))


#######################################
##Maximum daily song rate (Pr bird j sings w/in ReportedPrSingMin min), based on Julian
     Date
MaxDailySongRate<-function(Day){
  if(Day<122) return(0)  ##before May 1, birds not on breeding grounds
```

```
  if(Day>=122 & Day<=197) return(0.9)  ## High PRSing, 1 May - 15 July
  if(Day>197 & Day<=244) return(0.5)   ## Med PRSing, 16 July - 31 August
  if(Day>244 & Day<=274) return(0.2)   ## Low PRSing, 1-30 September
  if(Day>274) return(0)   ##Oct 1 and after, birds not on breeding grounds

  ##MODERATE SINGER VERSION
  #   if(Day<122 ) return(0)  ##before , birds not on breeding grounds
  #   if(Day>=122 & Day<=197) return(0.5)  ## High PRSing, 1 May - 15 July
  #   if(Day>197 & Day<=244) return(0.25)   ## Med PRSing, 16 July - 31 August
  #   if(Day>244 & Day<=274) return(0.1)   ## Low PRSing, 1-30 September
  #   if(Day>274) return(0)   ##Oct 1 and after, birds not on breeding grounds

  ##WEAK SINGER VERSION
  #   if(Day<122 ) return(0)  ##before , birds not on breeding grounds
  #   if(Day>=122 & Day<=197) return(0.3)  ## High PRSing, 1 May - 15 July
  #   if(Day>197 & Day<=244) return(0.1)   ## Med PRSing, 16 July - 31 August
  #   if(Day>244 & Day<=274) return(0.1)   ## Low PRSing, 1-30 September
  #   if(Day>274) return(0)   ##Oct 1 and after, birds not on breeding grounds

}

# plot(NULL,NULL, ylim=c(0,1), xlim=c(0,370))
# for(ii in 1:365){
#   points(ii,MaxDailySongRate(ii))
# }

#######
##Song rate weight, based on Time of Day
## Units of Time.yijrk = No. intervals since sunrise (0=sunrise exactly)
## To visualize, assume sun rises 6 AM daily.
```

```
##Parameterized with BTBW data from Blancher (personal comm)
TimeOfDayWeight<-function(TimeinIntervals, IntervalsPerHour){
  if(TimeinIntervals<=-0.25*IntervalsPerHour) return(0.2)  ## before 15 min before
     sunrise
  if(TimeinIntervals>-0.25*IntervalsPerHour & TimeinIntervals<=0*IntervalsPerHour)
     return(0.6)  ##between 15 min before sunrise and sunrise
  if(TimeinIntervals>0*IntervalsPerHour & TimeinIntervals<=0.5*IntervalsPerHour)
     return(0.8)  ##between sunrise and 0.5 hours after sunrise
  if(TimeinIntervals>0.5*IntervalsPerHour & TimeinIntervals<=1.5*IntervalsPerHour)
     return(1)  ##between 0.5 hours after sunrise and 1.5 hours after sunrise
  if(TimeinIntervals>1.5*IntervalsPerHour & TimeinIntervals<=6*IntervalsPerHour)
     return(0.8)  ##between 1.5 hours after sunrise and 6 hours after sunrise
  if(TimeinIntervals>6*IntervalsPerHour) return(0.2)   ##after 6 hours after sunrise
     (~noon)
}

# plot(NULL,NULL, ylim=c(0,1), xlim=c(-6*IntervalsPerHour, 18*IntervalsPerHour))
# for(ii in seq(-6*IntervalsPerHour, 18*IntervalsPerHour,100)){
#   points(ii,TimeOfDayWeight(ii,IntervalsPerHour))
# }
# lines(c(0,0),c(0,1))

#######
## Mean Noise Level for rep r
##NOTE: Noise is currently BINARY at the amplitude Alldredge tested
##Based on NoiseLevelMin & NoiseLevelMax parameters, above
# NoiseLevelDistrib<-function(Number, Min, Max){
#   round(runif(Number,Min,Max),2) # runif(1,0,10)
# }
```

```
NoiseLevelDistrib<-function(NumberTrials){
  rbinom(NumberTrials,1,PrNoise) # runif(1,0,10)
}

#######
## Mean WindSpeed for Day.yr
##Based on DailyMeanWindSpeedMin & DailyMeanWindSpeedMax parameters, above
DailyMeanWindSpeedDistrib<-function(Number, Min, Max){
  round(runif(Number,Min,Max),2) # runif(1,0,10)
}

#######
## Mean WindSpeed for rep r
##Based on mean wind speed produced by above function & RepMeanWindSpeedSD
RepMeanWindSpeedDistrib<-function(Number, Mean, SD){
  round(rlnormAlt(Number,Mean,SD),2) # runif(1,0,10)
}


####################################################
##PARAMETERS FOR HABITAT SIMULATION SCENARIOS (Ch. 2)

##Scenario-Specific Parameters

########################
##pd scenarios (high/low)
if(is.pd.high==T){
  SpeciesIntercept<- 6 ## Logit(Pd) intercept for species in scenario of interest
  DistanceEffect.yijr <- -0.02 ## mean slope of Logit(Pd)
}
```

```
if(is.pd.high==F){
  SpeciesIntercept<- 6.3 ## Logit(Pd) intercept for species in scenario of interest
  DistanceEffect.yijr <- -0.07 ## mean slope of Logit(Pd)
}

########################
##For pd~Hab scenarios, use HabitatSEffect to add variability to site-specific pd
##For pd not ~Hab, use SiteSEffect to add variability to site-specific pd

##For low pd, effect on slope = -0.04
##For high pd, effect on slope = -0.02

if(is.PdHab==T){
  if(is.pd.high==T){
    HabitatSEffect<- -0.02 ##Coefficient on slope (of habitat with lowest detection)
    SiteSEffect<- 0
  }
  if(is.pd.high==F){
    HabitatSEffect<- -0.04 ##Coefficient on slope (of habitat with lowest detection)
    SiteSEffect<- 0
  }
}

if(is.PdHab==F){  ##site-specific variation for scenarios without pd~hab
  if(is.pd.high==T){
    HabitatSEffect<- 0
    SiteSEffect<- -0.02
    }
  if(is.pd.high==F){
```

```
    HabitatSEffect<- 0
    SiteSEffect<- -0.04
  }
}

########################
## Pa functions - ONLY Max Daily Song rate affected by pa high/low
## (TOD curves are the same for both scenarios)

if(is.pa.high==T){  ##High pa
  ##Maximum daily song rate (Pr bird j sings w/in ReportedPrSingMin min), based on Julian
     Date
  ##No seasonal effects
  MaxDailySongRate<-function(Day){return(0.99)}
}

if(is.pa.high==F){  ##Low pa
  ##Maximum daily song rate (Pr bird j sings w/in ReportedPrSingMin min), based on Julian
     Date
  ##No seasonal effects
  MaxDailySongRate<-function(Day){return(0.56)}
}


## Load necessary packages
library(arm)
library(reshape)
########################
##Scenario Parameters
{
```

```
##Survey Type
#SurveyType<-    "multiple" # #"removal" #"nmixture" # "distance" #
#NSurveySites<- 1  ## No. of survey sites surveyed within one iteration (year, etc.)

##Basic Modeling parameters
NYears<-1           ## No. of Years or Seasons across which surveys take place
NReps<- ifelse(SurveyType=="nmixture",3,1)          ## No. of replications (No. of
   times each survey site is surveyed WITHIN season)
IntervalLength <- 2 ## No. of SECONDS or one unit of time during which bird may
   vocalize (or not)
IntervalsPerHour<-(60*60)/IntervalLength  ##No. of intervals in 1 hour
IntervalsPerMinute<-IntervalsPerHour/60  ##No. of intervals in 1 MINUTE
SurveyLength<- ifelse(SurveyType=="removal",10,3)   ## No. of MINUTES that 1 bird
   survey lasts
NIntervals<- (SurveyLength*60)/IntervalLength ##No. of intervals of IntervalLength that
   make up each survey
if(as.integer(NIntervals)!=NIntervals){stop("ERROR - Number of intervals must be an
   integer.  Adjust parameter SurveyLength or IntervalLength.")}

if(SurveyType=="removal"){
  RemovalPeriods<-3
  RemovalPeriod1Length<-2
  RemovalPeriod2Length<-3
  RemovalPeriod3Length<-5

  FarnsRemovalPeriod1Length<-3
  FarnsRemovalPeriod2Length<-2
  FarnsRemovalPeriod3Length<-5

  RemovalPeriod1Intervals<-(RemovalPeriod1Length/SurveyLength)*NIntervals
```

```
  RemovalPeriod2Intervals<-(RemovalPeriod2Length/SurveyLength)*NIntervals
  RemovalPeriod3Intervals<-(RemovalPeriod3Length/SurveyLength)*NIntervals

  FarnsRemovalPeriod1Intervals<-(FarnsRemovalPeriod1Length/SurveyLength)*NIntervals
  FarnsRemovalPeriod2Intervals<-(FarnsRemovalPeriod2Length/SurveyLength)*NIntervals
  FarnsRemovalPeriod3Intervals<-(FarnsRemovalPeriod3Length/SurveyLength)*NIntervals


   if(NIntervals!=sum(RemovalPeriod1Intervals,RemovalPeriod2Intervals,RemovalPeriod3Int
   ervals) |

   NIntervals!=sum(FarnsRemovalPeriod1Intervals,FarnsRemovalPeriod2Intervals,FarnsRemov
   alPeriod3Intervals)){
    stop("Intervals do not evenly divide among Removal periods")
  }
}

Xlim<- c(-1000,1000)  #c(-500,500)     ##Min and max x coordinate for xy grid generated
   around observer
Ylim<- c(-1000,1000) #c(-500,500)   ##Min and max y coordinate for xy grid generated
   around observer
#EntireAreaRadius <-  2000 ## Radius (m) at which birds are modeled (so they can
   enter/leave survey site)
#   SurveyAreaRadius<- 100 ##Fixed-distance cutoff (m) for observation during survey

########################
##Population parameters
ContinentalPopulation<- 2100000/2 #rep(2100000/2, NYears) ##Total continental
   Population of Species
## estimate from Partners in Flight - divide by 2 to get number of males
```

```
ContinentalPopulationCoef<-rep(1,NYears)#c(0.1, 0.5, 1, 2, 5, 0.1, 0.5, 1, 2, 5)
ContinentalPopulationAll.vector<-sort(ContinentalPopulationCoef*ContinentalPopulation)

HabitatPreference.y<-1 ## % of total population found in habitat (as opposed to matrix) across entire range
RangeArea<-3.62*10^11  ##(m^2)  Size of entire species' range (7*10^11 m^2 is the approx. size of Texas)
StudyArea<-3000*10000  ##(m^2) Size of study area (ps ~ StudyArea/RangeArea)
##StudyArea from http://www.birdlife.org/datazone/species/factsheet/22721673
HabitatDensity.y<- 1/10000  ##Mean density (birds/m^2) for habitat within species range
##Assume 1 bird/ha in 100% habitat

HabitatDensity.y.List<-list()
HabitatDensity.y.List[[1]]<-1/10000  ##Mean density (birds/m^2) for habitat within species range
AnnualDensitySlope<- -0.01034/10000  ##Annual decline in Density (birds/m^2) - NOT a percentage
##Assume 1 bird/ha in year 1 in 100% habitat
if(SimReps>1){
  for(ii in 2:SimReps){
    HabitatDensity.y.List[[ii]]<-HabitatDensity.y.List[[1]]+AnnualDensitySlope*(ii-1)
  }
}
##30% reduction in density across 30 years (from 1 bird/ha to 0.7)

RangeHabitatProportion.vector<-(HabitatPreference.y*ContinentalPopulationAll.vector)/(HabitatDensity.y*RangeArea) ## % of species' entire range that is habitat (as opposed to matrix)
#   StudyHabitatProportion.vector<-rep(0.95,NYears) #RangeHabitatProportion.vector ## % of study area that is habitat (as opposed to matrix)
```

```
StudyHabitatProportion<-function(N=1, min, max){  ##Distribution for mean %Habitat for
   entire study area
  Value<-runif(N, min, max)
  return(Value)
}
StudyHabitatProportionmin<-0.7  ##Study area is between 70% and 100% habitat (mean for
   each Fauxyear)
StudyHabitatProportionmax<-1
##If study area is range-wide, use HabitatProportionSurveyedArea.y<-
   RangewideHabitatProportion.y
##If study area is a subset of the range, use expected mean % habitat among sites
   (e.g., we picked sites with ~60% habitat)

##Parameters to describe normal distributions for PercentHabitat.1i
LowPercentHabMean<-0.4
LowPercentHabSD<-0.09
HighPercentHabMean<-0.8
HighPercentHabSD<-0.09

##Parameters for variation added to PercentHabitat.yi annually
PercentHabitatVarMean<-0
PercentHabitatVarSD<-0.01  ##95% of annual changes will be <2%

#PercentHabitatStudy<-  ## % of total habitat within species' range that is contained
   by study area
#  (StudyHabitatProportion.vector*StudyArea)/(RangeHabitatProportion.vector*RangeArea)
```

```
##BTBW pop. estimate=2,100,000 PAIRS, from Partners in Flight:
   http://rmbo.org/pifpopestimates/Database.aspx
#Ps <- 1             ## Proportion of ContinentalPopulationAll.vector that can be found
   in Area of Inference (AOI)
#AreaofInference<- 3.62*10^11  ##(m^2)  Size of Area of Inference (AOI),  (7*10^11 m^2
   is the approx. size of Texas)
#HabitatProportionAll<- rep(0.5,NYears)    ##Avg. % of AOI occupied by habitat
##HabitatProportionAll is the vector of %AOI occupied by habitat, length=NYears
HabitatProportionThetaAll<- rep(8,NYears)   ##Concentration parameter for Beta
   distribution - see rbetaAlt function

##########################
##Environmental / Temporal parameters

##Survey timing set up
StartingJulianDate<-150  ##First day of surveys
SurveysPerDay<-ifelse(SurveyType=="removal",6,7)
LogisticalSurveyTime <-ifelse(SurveyType=="removal",30,23)  ## mean MINUTES taken to
   conduct 1 survey, including travel time

##NOTE: DailySurveyStartTime not currently used - see PlannedStartTimesAll in "A3 -
   Simulation Script.R"
DailySurveyStartTime <-0 ## mean minutes past sunrise when surveys begin daily
# StartingTimes<- round(runif(10000,min=5,max=11),2) ##Hours, Military time

EndingJulianDate<-220  ##Last possible day of surveys

##########################
##Biological Parameters
##Currently, Leaves are PRESENT for all surveys
```

```
FirstLeavesDay<-StartingJulianDate #+ 10  ##First day when Leaves=1
LeavesByDate<-c(rep(0,FirstLeavesDay-1), ##before Leaf Out
                rep(1,(EndingJulianDate-FirstLeavesDay+1)))
##LeavesByDate is an index, where LeavesByDate[Day] indicates if there were leaves
   (0/1) on Julian Date "Day"

#SpecialistIndex<- 0.6        ##(from 0-1) Index of specialist 0=Generalist, 1=
   Specialist
BackgroundMovementRate<- 0.005  ## Species-specific Lowest reasonable probability of
   bird moving between interval t=0 and t=1
## 0.005 = medium mobility value from Granholm 1983
MeanTerrArea<- 6.56*10000 ##6.56HA, in m^2
SDTerrArea<-   10000 #1 HA            ##SE of territory area for this species (m^2)
OverlapUDTerr<- 0.6  ## %Utilization Distribution at which overlap of territories is
   accessed
UDTerr<-0.95  ##% Utilization Distribution modeled for all final bird territories

##########################
##Perceptibility parameters
PrGrassland.y<- rep(0,NYears) ##Probability that site i is grassland (as opposed to
   forest)
PrDeciduous.y<- rep(0,NYears) ##If site i is forest, Probability that site i is
   Deciduous (as opposed to Mixed Pine-Deciduous)

##  NoiseLevelDistrib<-round(runif(10000,0,10),2) # runif(1,0,10)
#   NoiseLevelMin<-0
#   NoiseLevelMax<-10

##NOTE: Ambient Noise is currently BINARY at the amplitude Alldredge tested
PrNoise<-0
```

```
DailyMeanWindSpeedMin<-0.01  ##using true zero can create errors
DailyMeanWindSpeedMax<-10
RepMeanWindSpeedSD<-0.5

##MeanWindSpeedDistrib<-round(runif(10000,0,10),2) ##WindSpeed distribution
   (note:WindSpeed>0)
WindVariability<- 0.94  ##Medium variability, from Justus 1977
##1.05 ##Low variability, from Justus 1977
##0.83  ##High variability, from Justus 1977

##SCENARIO SPECIFIC - moved to separate script
#   ##For pd~Hab scenarios, use HabitatSEffect to add variability to site-specific pd
#   ##For pd not ~Hab, use SiteSEffect to add variability to site-specific pd
#   if(is.PdHab==T){
#      HabitatSEffect<- -0.02 ##Coefficient on slope (of habitat with lowest detection)
#      SiteSEffect<- 0}
#
#     if(is.PdHab==F){
#        HabitatSEffect<- 0
#        SiteSEffect<- -0.02}  ##added site-specific variation for scenarios without
   pd~hab
#
#   ##pd scenarios (high/low)
#   if(is.pd.high==T){
#     SpeciesIntercept<- 6 ## Logit(Pd) intercept for species in scenario of interest
#     DistanceEffect.yijr <- -0.02 ## mean slope of Logit(Pd)
#   }
#
#   if(is.pd.high==F){
```

```
#     SpeciesIntercept<- 10 ## Logit(Pd) intercept for species in scenario of interest
#     DistanceEffect.yijr <- -0.09 ## mean slope of Logit(Pd)
#   }

WindIndexIEffect<- 0
GrasslandIEffect<- 0 ##effect of grassland categorical variable on intercept of
   Logit(Pd)
DeciduousIEffect<- 0  ##effect of deciduous categorical variable on intercept of
   Logit(Pd)
LeavesIEffect<- 0 ##effect of leaves categorical variable on intercept of Logit(Pd)
NoiseIEffect<- 0 ##effect of ambient noise on intercept of Logit(Pd)
HabitatIEffect<-0  ##effect of habitat on intercept of Logit(Pd)
WindIndexSEffect<- 0
GrasslandSEffect<- 0 ##effect of grassland categorical variable on slope of Logit(Pd)
   (interaction of Distance and grassland)
DeciduousSEffect<- 0 ##effect of deciduous categorical variable on slope of Logit(Pd)
   (interaction of Distance and Deciduous)
LeavesSEffect<- 0 ##effect of leaves categorical variable on slope of Logit(Pd)
   (interaction of Distance and Leaves)
NoiseSEffect<-  0 ##effect of ambient noise on slope of Logit(Pd)
NoiseXLeavesEffect<- 0
NTotalObservers<-2
NSimultaneousObservers<-ifelse(SurveyType=="multiple",2,1)

ObserverIDs<-LETTERS[1:NTotalObservers]
ObserverSEffect<-0  ## year-specific obs effect - Slope
ObserverIMean<-0      ##rnorm mean for year-specific obs effect - Intercept
ObserverISD<-0.75     ##rnorm SD for year-specific obs effect - Intercept

PrCorrectID<- 1  ##Pr(Bird is correctly ID'd)
```

```
PrDoubleCount<-0 ##Pr(1 bird is counted as 2)

##Observer estimation of distance
ObserverDistanceCategories<-data.frame(
  ## From Alldrege et al. 2007 "A field evaluation of distance measurement error..."

  ##Alldredge distance-dependent error curve:
  #     "Distance"=c(0,23,37,52,65,75,86,98),
  #     "meanerror"=c(0,18.5,-1.6,8.8,14.8,5.5,-2.1,-7.9),
  #     "sderror"=c(0,18.9,11,22.3,22.1,19.2,17.4,16.8))

  ##Alldredge error curve based on overall mean & SD error:
  "Distance"=c(0,62.286),
  "meanerror"=c(0,7.6),
  "sderror"=c(0,21.4))



#############################################
##Coarse-scale Markov song Parameters
##NOTE: Same for both high & low pa
PSS.yijr<-0.98    ##Pr(Bird sings during next interval, given that it did sing before)

##Fine-scale Markov song parameters (Species/Scenario-specific)
SongLength<-2.1/IntervalLength ##Number of intervals for avg. song length (interval=2
   sec)
PauseLength<-6.6/IntervalLength ##Number of intervals for avg. pause between songs
q1.fine<- SongLength/(SongLength+PauseLength)
#   PSS.fine<- 0.001
#   PSNS.fine<- MarkovTM(q1.fine, PSS.fine)
```

```
  SingingStates.fine<- c("S", "NS1", "NS2", "NS3")
  TransitionMatrix.fine<-matrix(c(0,0,0.08,0.8,1,0,0,0,0,1,0,0,0,0,0.92,0.2), nrow=4,
     ncol=4)  ##Transition matrix for fine-scale autocorrelation of Singing
  ## above TransitionMatrix.fine produces Pauses w/mean length 6.3 seconds (SD=1.25 sec,
     q1=0.241) (BTBW values)
  ## Values determined via simulation (file="Fine-scale autocorrelation - Simulation to
     determine values of 4-stage transition matrix.R")

  ReportedPrSingMin<-5  ## x minutes in reported song rates: Pr(Bird j sings w/in x
     minutes)
  ##PrSing.yijrk refers to Pr(bird sings at least 1x in ReportedPrSingMin min (usually 5
     or 10))

}  ## end Scenario Parameters


################################################################
## Scenario Distributions
################################################

##Generate a k-specific probability of movement
PrBirdMoves<-function(N=1,MeanMovementRate){
  PrBirdMoves.yijrk<-MeanMovementRate+rnorm(N,0,0.0005)
  if(sum(PrBirdMoves.yijrk<0)>0) stop("Negative values generated for PrBirdMoves -
     reevaluate variation!")
  return(abs(PrBirdMoves.yijrk))
}
#hist(PrBirdMoves(10000,BackgroundMovementRate))

#######
```

```
# ##Generate probability that a bird flushes due to observer
# PrBirdMoveObs<-function(Intercept=1,Slope=-0.1, Distance){
#   PrMoveObs<- invlogit(Intercept+Slope*Distance)
#   return(PrMoveObs)
# }
# ##Plot Pr(Flush)~Distance
# #plot(1:200,PrBirdMoveObs(Distance=1:200))

########################
## Pa functions - ONLY Max Daily Song rate affected by pa high/low
## (TOD curves are the same for both scenarios)

##SCENARIO SPECIFIC - moved to separate script
# if(is.pa.high==T){  ##High pa
#   ##Maximum daily song rate (Pr bird j sings w/in ReportedPrSingMin min), based on
     Julian Date
#   ##No seasonal effects
#   MaxDailySongRate<-function(Day){return(0.99)}
# }
#
# if(is.pa.high==F){  ##Low pa
#   ##Maximum daily song rate (Pr bird j sings w/in ReportedPrSingMin min), based on
     Julian Date
#   ##No seasonal effects
#   MaxDailySongRate<-function(Day){return(0.56)}
# }

#######
##Song rate weight, based on Time of Day
## Units of Time.yijrk = No. intervals since sunrise (0=sunrise exactly)
```

```
## To visualize, assume sun rises 6 AM daily.
##Parameterized with AMCR,MODO,OVEN,YBSA, data from Blancher (personal comm)
##SAME CURVE used for high and low pa scenarios
##Distinguish the 2 by Max Daily Song Rate (0.99 vs. 0.56)
TimeOfDayWeight<-function(TimeinIntervals, IntervalsPerMin){
  Early.Slope.Minutes<-0.01477  ##Slope for line from -45 to 9 minutes (0=sunrise)
  Early.Int.Minutes<-0.8671  ##Intercept for line from -45 to 9 minutes

  ##Pa for time=TimeinIntervals, -45 to 9 minutes (point on early line described above)
  Early.Point<-(TimeinIntervals/IntervalsPerMin)*Early.Slope.Minutes +Early.Int.Minutes

  Late.Slope.Minutes<--0.002159  ##Slope for line from 44-240 min after sunrise
  Late.Int.Minutes<-1.095   ##Intercept for line from 44-240 min after sunrise

  ##Pa for time=TimeinIntervals, 44-240 minutes (point on late line described above)
  Late.Point<-(TimeinIntervals/IntervalsPerMin)*Late.Slope.Minutes +Late.Int.Minutes

  if(TimeinIntervals<(-45)*IntervalsPerMin) return(0.2)  ## before 45 min before sunrise
     (extrapolation beyond data, therefore assume =0.2)
  if(TimeinIntervals>=-45*IntervalsPerMin & TimeinIntervals<=9*IntervalsPerMin)
     return(Early.Point)  ##between 30 min before sunrise and 9 min after sunrise
  if(TimeinIntervals>9*IntervalsPerMin & TimeinIntervals<=44*IntervalsPerMin) return(1)
     ##9-42 min after sunrise
  if(TimeinIntervals>44*IntervalsPerMin & TimeinIntervals<=240*IntervalsPerMin)
     return(Late.Point)  ##after 42 Minutes after sunrise until end of BBS route
  if(TimeinIntervals>240*IntervalsPerMin) return(0.5768)   ##after BBS route is over
     (extrapolation beyone data, therefore assume=last value)
}

# ##Plot 4AM to noon
```

```
# plot(NULL,NULL, ylim=c(0,1), xlim=c(-2*IntervalsPerHour, 6*IntervalsPerHour))
# for(ii in seq(-6*IntervalsPerHour, 18*IntervalsPerHour,100)){
#   points(ii,TimeOfDayWeight(ii, IntervalsPerMinute))
# }
# lines(c(0,0),c(0,1))




#######
## Mean Noise Level for rep r
##NOTE: Noise is currently BINARY at the amplitude Alldredge tested
##Based on NoiseLevelMin & NoiseLevelMax parameters, above
# NoiseLevelDistrib<-function(Number, Min, Max){
#   round(runif(Number,Min,Max),2) # runif(1,0,10)
# }

NoiseLevelDistrib<-function(NumberTrials){
  rbinom(NumberTrials,1,PrNoise) # runif(1,0,10)
}

#######
## Mean WindSpeed for Day.yr
##Based on DailyMeanWindSpeedMin & DailyMeanWindSpeedMax parameters, above
DailyMeanWindSpeedDistrib<-function(Number, Min, Max){
  round(runif(Number,Min,Max),2) # runif(1,0,10)
}

#######
## Mean WindSpeed for rep r
```

```
##Based on mean wind speed produced by above function & RepMeanWindSpeedSD
RepMeanWindSpeedDistrib<-function(Number, Mean, SD){
  round(rlnormAlt(Number,Mean,SD),2) # runif(1,0,10)
}


########################
###END PARAMETERS################################

########################
##SIMULATION

##NOTE: model was originally designed to loop through years (1:NYears) via y
##Currently, years are looped via FauxYear in "A7 - Generic routine to run simulation via
     source"
## y==1 throughout this script for all FauxYear

##Generate data frames to track rare phenomena:
MisID<-data.frame(y=NA,i=NA,j=NA,r=NA,k=NA,o=NA)
DoubleCounted<-data.frame(y=NA,i=NA,j=NA,r=NA,k=NA,o=NA)

##Generate .yir-level & .yirk-level parameters (Independent of individual birds)
##Necessary here because reps are nested within birds in yijrk loops

##SurveyDates - assume no. of surveys/day = SurveysPerDay
## No weekends, survey order is i=1, then i=2, etc.
##CURRENTLY ASSUMES ALL YEARS ARE THE SAME
SurveyDatesAll<-StartingJulianDate+rep(0:100,each =
     SurveysPerDay)[1:(NSurveySites*NReps)]
```

```
SurveyDate.yir.List<-replicate(NYears, matrix(SurveyDatesAll ,NSurveySites, NReps),
     simplify=F)
##Access via SurveyDate.yir.List[[y]][i,r]

##Start time - how many MINUTES after sunrise does survey yir begin?
StartTimeinMinutes.yir.List<-replicate(NYears, matrix(NA ,NSurveySites, NReps),
     simplify=F)
for(yy in 1:NYears){
  PlannedStartTimesAll<-rep(0:(SurveysPerDay-
     1)*LogisticalSurveyTime,100)[1:(NSurveySites*NReps)]
  ActualStartTimesAll<-round(rnorm(n=length(PlannedStartTimesAll),
     mean=PlannedStartTimesAll,sd=5),2)

  StartTimeinMinutes.yir.List[[yy]]<-matrix(ActualStartTimesAll, NSurveySites, NReps)
     ##units = MINUTES
}
##Access via StartTimeinMinutes.yir.List[[y]][i,r]

##WindSpeed: r-specific mean (.yir)
##Must take into account surveys on the SAME DAY
## rep-specific mean generated from day-specific mean
# MeanWindSpeed.yir.List<-replicate(NYears,
#                                    matrix(rep(NA,NReps*NSurveySites),
#                                           #(MeanWindSpeedDistrib(NReps*NSurveySites,
     MeanWindSpeedMin, MeanWindSpeedMax),
#                                           NSurveySites, NReps), simplify=F)
# for(yy in 1:NYears){
#   UniqueDates<-unique(unlist(SurveyDate.yir.List[[yy]]))
#   for(UD in UniqueDates){
```

```
#     DailyMeanWindSpeed.yr<-DailyMeanWindSpeedDistrib(1, DailyMeanWindSpeedMin,
     DailyMeanWindSpeedMax)
#     for(ii in 1:NSurveySites){
#       for(rr in 1:NReps){
#         if(SurveyDate.yir.List[[yy]][ii,rr]==UD){
#           MeanWindSpeed.yir.List[[yy]][ii,rr]<-
     RepMeanWindSpeedDistrib(1,DailyMeanWindSpeed.yr, RepMeanWindSpeedSD)
#         }
#       }
#     }
#   }
# }

##Access via MeanWindSpeed.yir.List[[y]][i,r]


##WindSpeed: k-specific windspeed(.yirk)
##Windspeeds follow Weibull distribution, parameters from Justus 1977
##Because scale parameter c includes term (1/k), Weibull becomes unstable for small
     values of k
##Therefore, for low wind speeds, use k=0.5
# WindSpeed.yirk.List<-vector("list", NYears)
# for(yy in 1:NYears){
#   WindSpeed.yirk.List[[yy]]<-vector("list", NSurveySites)
#   for(ii in 1:NSurveySites){
#     WindSpeed.yirk.List[[yy]][[ii]]<-vector("list", NReps)
#     for(rr in 1:NReps){
#       ifelse(WindVariability*sqrt(MeanWindSpeed.yir.List[[yy]][ii,rr])>1,
#              kparameter<-WindVariability*sqrt(MeanWindSpeed.yir.List[[yy]][ii,rr]),
#              kparameter<-1)
```

```
#       cparameter<-MeanWindSpeed.yir.List[[yy]][ii,rr]/(gamma(1+1/kparameter))
#       WindSpeed.yirk.List[[yy]][[ii]][[rr]]<-rweibull(NIntervals, shape=kparameter,
     scale=cparameter)
#     }
#   }
# }
##Access via WindSpeed.yirk.List[[y]][[i]][[r]][k]

##Ambient Noise: r-specific 1/0 value (.yir)
# MeanNoiseLevel.yir.List<-replicate(NYears,
#                                    matrix(NoiseLevelDistrib(NReps*NSurveySites),#,
     NoiseLevelMin, NoiseLevelMax
#                                           NSurveySites, NReps), simplify=F)

##NOTE: if you want sites to have certain Noise Levels, need to generate this at the .yi
     level instead of the .yir (currently .yir)
##Access via MeanNoiseLevel.yir.List[[y]][i,r]

##AmbientNoise: k-specific noise (.yirk) (currently constant across rep)
# Noise.yirk.List<-vector("list", NYears)
# for(yy in 1:NYears){
#   Noise.yirk.List[[yy]]<-vector("list", NSurveySites)
#   for(ii in 1:NSurveySites){
#     Noise.yirk.List[[yy]][[ii]]<-vector("list", NReps)
#     for(rr in 1:NReps){
#       Noise.yirk.List[[yy]][[ii]][[rr]]<-
     rep(MeanNoiseLevel.yir.List[[yy]][ii,rr],NIntervals)
#       ##Noise Level is currently r-specific - to make it k-specific, add distribution
     above
#     }
```

```
#   }
# }
##Access via Noise.yirk.List[[y]][[i]][[r]][k]

##Year-specific Observer Effects
ObserverIEffects<-data.frame("A"=rnorm(1,ObserverIMean,ObserverISD),
                             "B"=rnorm(1,ObserverIMean,ObserverISD))
ObserverSEffects<-data.frame("A"=ObserverSEffect, "B"=ObserverSEffect)

ifelse(SurveyType=="multiple",
       PrimaryObsNumber.yir.List<-replicate(NYears,
     matrix(sample(1:NSimultaneousObservers,NReps*NSurveySites, replace=T),NSurveySites,
     NReps), simplify=F),
       PrimaryObsNumber.yir.List<-replicate(NYears,
     matrix(rep(NA,NReps*NSurveySites),NSurveySites, NReps), simplify=F))
##Access via PrimaryObsNumber.yir.List[[y]][i,r]


if(SurveyType!="multiple"){
  ##For all surveys except multiple observer, Observer identity is a replication-level
     variable
  ObserverIDNumber.yir.List<-replicate(NYears,
     matrix(sample(1:NTotalObservers,NReps*NSurveySites, replace=T),NSurveySites, NReps),
     simplify=F)
  ##Access via ObserverIDNumber.yir.List[[y]][i,r]
  ## for name, use ObserverIDs[ObserverIDNumber.yir.List[[y]][i,r]]
}

# Leaves.yir.List<-list()
# for(LeavesYears in 1:NYears){ #length(SurveyDate.yir.List)){
```

```
#   Leaves.yir.List[[LeavesYears]]<-
     matrix(LeavesByDate[SurveyDate.yir.List[[LeavesYears]]],NSurveySites, NReps)
# }
##Access via Leaves.yir.List[[y]][i,r]

#######################################
##Generate starting values & level-specific lists for storage
ListYOutcomes.y<-vector("list", NYears)
#ListYSummaryOutcomes.y<-vector("list", NYears)
ListIOutcomes.yi<-vector("list", NYears)
ListIRSummaryOutcomes.yi<-vector("list", NYears)
ListROutcomes.yijr<-vector("list", NYears)
# ListKOutcomes.yijrk<-vector("list", NYears)
# ListOOutcomes.yijrko<-vector("list", NYears)


for(y in 1:NYears){

  ##Generate year-specific parameters
  ##Areal Parameters
  StudyHabitatProportion.y<-StudyHabitatProportion(1,StudyHabitatProportionmin,
     StudyHabitatProportionmax)
  RangeHabitatArea.y<-RangeArea*RangeHabitatProportion.vector[y] ## Size (m^2) of habitat
     in species' range
  RangeMatrixArea.y<-RangeArea-RangeHabitatArea.y  ## Size (m^2) of matrix in species'
     range
  StudyHabitatArea.y<-StudyArea*StudyHabitatProportion.y ##Size (m^2) of habitat in Study
     Area
  StudyMatrixArea.y<-StudyArea-StudyHabitatArea.y  ##Size (m^2) of matrix in Study Area
  ##StudyHabitatArea.y+StudyMatrixArea.y==StudyArea
```

```
##Abundance & Density Parameters
ContinentalPopulation.y<-ContinentalPopulationAll.vector[y] ##Continental population in
   year y
RangeHabitatAbundance.y<-round(ContinentalPopulation.y*HabitatPreference.y,0) ## No.
   birds found in habitat in species' range
RangeMatrixAbundance.y<-round(ContinentalPopulation.y*(1-HabitatPreference.y),0) ## No.
   birds found in matrix in species' range

MatrixDensity.y<-RangeMatrixAbundance.y/RangeMatrixArea.y  ##Density of birds in matrix
   in species' range (birds/m^2)
HabitatDensity.y<-HabitatDensity.y.List[[FauxYear]]
StudyHabitatAbundance.y<- round(HabitatDensity.y*StudyHabitatArea.y,0)
StudyMatrixAbundance.y<-round(MatrixDensity.y*StudyMatrixArea.y,0) ##No. of birds in
   matrix in study area
Ns.y<-StudyHabitatAbundance.y + StudyMatrixAbundance.y ##Total No. birds in Study Area

HabitatProportionTheta.y<-HabitatProportionThetaAll[y] ##Concentration parameter for
   beta distrib. of HabitatProportion.y

##Generate starting values & level-specific lists for storage
ListIOutcomes.yi[[y]]<-vector("list", NSurveySites)
ListIRSummaryOutcomes.yi[[y]]<-vector("list", NSurveySites)
ListROutcomes.yijr[[y]]<-vector("list", NSurveySites)
#   ListKOutcomes.yijrk[[y]]<-vector("list", NSurveySites)
#   ListOOutcomes.yijrko[[y]]<-vector("list", NSurveySites)
TempAll.i.Outcomes.y<-vector("list", NSurveySites)

for(i in 1:NSurveySites){
```

```
##Time the simulation
TIMER<-proc.time()

##Generate survey site-specific parameters

##PercentHab is the same for each site i in Year 1, across all Scenarios
if(FauxYear==1){
  PercentHabitat.yi<- as.numeric(PercentHabitat.i.List[[y]][[i]][1])
  HabitatGroup.yi<- PercentHabitat.i.List[[y]][[i]][2]
}
##Add annual variation to PercentHab: random walk based on last year's value
if(FauxYear>1){
  PercentHabitat.yi<-as.numeric(PercentHabitat.i.List[[FauxYear-1]][[i]][1])
  PercentHabitat.yi<-
 PercentHabitat.yi+rnorm(1,PercentHabitatVarMean,PercentHabitatVarSD)
  HabitatGroup.yi<-PercentHabitat.i.List[[FauxYear-1]][[i]][2]
  if(i==1){PercentHabitat.i.List[[FauxYear]]<-vector("list", NSurveySites)}
  PercentHabitat.i.List[[FauxYear]][[i]][1]<-PercentHabitat.yi
  PercentHabitat.i.List[[FauxYear]][[i]][2]<-HabitatGroup.yi
}

SiteVariationS.yi<-runif(1,0,1)  ##Non-Habitat variation: i-specific
##if pd~Hab, no added variation

Area.yi<-abs(Xlim[2]-Xlim[1])*abs(Ylim[2]-Ylim[1])  ##Vicinity of survey site where
 birds are modeled
LambdaHabitat.yi<-PercentHabitat.yi*Area.yi*HabitatDensity.y
LambdaMatrix.yi<-(1-PercentHabitat.yi)*Area.yi*MatrixDensity.y
BirdsInHabitat.yi<-rpois(1,LambdaHabitat.yi)
BirdsInMatrix.yi<- rpois(1,LambdaMatrix.yi)
```

```
NBirds.yi<- BirdsInMatrix.yi+ BirdsInHabitat.yi
##Number of birds in habitat =Poisson(%Habitat*area*Density of birds in habitat)
##Number of birds in matrix = Poisson(%Matrix* area*Density of birds in matrix)
#     IsSiteGrassland.yi<-rbinom(1,1,PrGrassland.y)
#     if(IsSiteGrassland.yi==1) {IsSiteDeciduous.yi<-0}
#     if(IsSiteGrassland.yi==0) {IsSiteDeciduous.yi<-rbinom(1,1,PrDeciduous.y)}

if(NBirds.yi==0){

  ListROutcomes.yijr[[y]][[i]]<-vector("list", 1)
  #       ListKOutcomes.yijrk[[y]][[i]]<-vector("list", 1)
  #       ListOOutcomes.yijrko[[y]][[i]]<-vector("list", 1)

  ListROutcomes.yijr[[y]][[i]][[1]]<-vector("list", NReps)
  #       ListKOutcomes.yijrk[[y]][[i]][[1]]<-vector("list", NReps)
  #       ListOOutcomes.yijrko[[y]][[i]][[1]]<-vector("list", NReps)

  ##For NBirds.yi==0
  for(r in 1:NReps){
    StartTimeinIntervals.yir<-
 round(StartTimeinMinutes.yir.List[[y]][i,r]*IntervalsPerMinute,0)
 ##Minutes*(intervals/Minute)
    #         Leaves.yir<-Leaves.yir.List[[y]][i,r]

    ##Generate starting values & level-specific lists for storage
    #         ListKOutcomes.yijrk[[y]][[i]][[1]][[r]]<-vector("list", NIntervals)
    #         ListOOutcomes.yijrko[[y]][[i]][[1]][[r]]<-vector("list", NIntervals)

    ##For NBirds.yi==0
    for(k in 1:NIntervals){
```

```
     TimeinIntervals.yirk <- StartTimeinIntervals.yir+ (k-1) ##Intervals since
sunrise (in intervals, 0=sunrise)
     #           NoiseLevel.yirk<-Noise.yirk.List[[y]][[i]][[r]][k]
     ##Ambient noise is currently rep-specific only
     #           WindSpeed.yijrk<-as.numeric(WindSpeed.yirk.List[[y]][[i]][[r]][k])
     ##NOTE: WindIndex.yijrk does not apply: would require bird location, but
NBirds.yi==0)

     ##Generate starting values & level-specific lists for storage
     #           ListOOutcomes.yijrko[[y]][[i]][[1]][[r]][[k]]<-vector("list",
NSimultaneousObservers)

     ##For NBirds.yi==0
     for(o in 1:NSimultaneousObservers){

       if(SurveyType!="multiple"){
         ObserverIDNumber.yijrko<-ObserverIDNumber.yir.List[[y]][i,r]}

       if(SurveyType=="multiple"){
         ObserverIDNumber.yijrko<-o}

       Count.yijrko<-0  ##Currently, no false positives if NBirds.yi==0

       ## Outcomes Complete    #####################################


       ## Index all o-specific objects for NBirds.yi==0
       TempOutcomes.o<-list(
         "Count.yijrko"=Count.yijrko,
         "Pdintercept.yijrko"=NA,
```

```
         "Pdslope.yijrko"=NA,
         "Perceptibility.yijrko"=NA,

"ObserverIDNumber.yijrko"=ifelse(SurveyType=="multiple",ObserverIDNumber.yijrko,NA),
         "Detected.yijrko "=NA,
         "CorrectID.yijrko"=NA,
         "PrCorrectID.yijrko"=NA,
         "PrDoubleCount.yijrko"=NA,
         "DoubleCounted.yijrko"=NA,
         "ObsEstimatedDistance.yijrko"=NA
       )

       ##Store o-specific objects and summaries for NBirds.yi==0
       #             ListOOutcomes.yijrko[[y]][[i]][[1]][[r]][[k]][[o]]<-list(
       #               "DoubleCounted.yijrko"=NA,
       #               "Detected.yijrko"=NA,
       #               "CorrectID.yijrko"=NA,
       #               "DoubleCounted.yijrko"=NA,
       #               "Count.yijrk"=0 ##Currently, no false positives if
NBirds.yi==0
       #             )

     }  ## o for NBirds.yi==0

     ## Index all k-specific objects for NBirds.yi==0
     TempOutcomes.k<-list(
       "TimeinIntervals.yirk"=TimeinIntervals.yirk,
       #             "WindSpeed.yijrk"=WindSpeed.yijrk,
       #             "WindIndex.yijrk"=NA,
       #             "NoiseLevel.yirk"=NoiseLevel.yirk,
```

```
          "SingingMode.yijrk"=NA,
          "SingingState.yijrk"=NA,
          "PrSing.yijrk"=NA,
          "Sings.yijrk"=NA,
          "PSNS.yijrk"=NA,
          "q1.yijrk"=NA,
          "PrBirdMoves.yijrk"=NA,
          "DoesBirdMove.yijrk"=NA,
          "Distance.yijrk"=NA,
          "Location.x.yijrk"=NA,
          "Location.y.yijrk"=NA,
          "SongWeight.yijrk"=NA
        )

        #           ##Store-k-level objects and summaries for NBirds.yi==0
        #           ListKOutcomes.yijrk[[y]][[i]][[1]][[r]][[k]]<-list(
        #             "TimeinIntervals.yirk"=TimeinIntervals.yirk,
        #             "WindSpeed.yijrk"=WindSpeed.yijrk,
        #             "NoiseLevel.yirk"=NoiseLevel.yirk
        #           )

      }  ##k for NBirds.yi==0

      ##Make List of rep-specific outcomes (summary across intervals)
      #         MeanObservedWindspeed.yijr<-
   mean(unlist(TempAll.k.Outcomes.r)[names(unlist(TempAll.k.Outcomes.r))=="WindSpeed.yi
   jrk"])
      #         MeanObservedNoiseLevel.yir<-
   mean(unlist(TempAll.k.Outcomes.r)[names(unlist(TempAll.k.Outcomes.r))=="NoiseLevel.y
   irk"])
```

```
   ###NOTE: above will not work b/c TempAll.k.Outcomes.r is not set up for
NBirds.yi==0 !!

   ## Index all r-specific objects for NBirds.yi==0
   TempOutcomes.r<-list(
     #           "Leaves.yir"=Leaves.yir,
     #"ObserverIDNumber.yir"=ObserverIDNumber.yijrko,
#ObserverIDNumber.yir.List[[y]][i,r],

     ###NOTE - is above correct??
     ##Shouldn't it be ".yijrko=" ?

     #           "MeanObservedWindspeed.yijr"=MeanObservedWindspeed.yijr,
     #           "MeanObservedNoise.yijr"= MeanObservedNoiseLevel.yir,
     "Count.yijr"=0, ##Currently, no false positives if NBirds.yi==0
#max(unlist(TempOutcomes.k)["Count.yijrk"]),
     "MeanObservedDistance.yijrk"=NA
   )
   ##Store-r-level objects and summaries for NBirds.yi==0
   ListROutcomes.yijr[[y]][[i]][[1]][[r]]<-TempOutcomes.r

  }  ##r for NBirds.yi==0

}  ##bracket refers to "if(NBirds.yi==0){"

if(NBirds.yi>0){

  ##Generate Mean & SD area for bird territories at site i
  ##If density is sufficiently high, compress area so that only 50% of total area is
filled by OverlapUDTerr % UD ellipses
```

```
 ##For compressed territories, SD is multiplied by ratio (compressed
area/uncompressed territory size)
 AllowedTerrSizeOE.yi<- (0.5*Area.yi)/NBirds.yi

 ##What is the ratio in size between the %UD ellipse used to calc overlap and the
final UD for territories?
 ##See Equations 12 and 13 from Jennrich and Turner 1969
 ##For example: Calculate allowed size for 95% ellipses from 80% UD
 ##For all bivariate normal territories:
 ## Area of 80% UD ellipse = 0.53724 * Area of 95% UD ellipse
 ## Area of 95% UD ellipse = 1.86135 * Area of an 80% UD ellipse
 OverlapCoef<- log((1-OverlapUDTerr)^-2)
 FinalCoef<-log((1-UDTerr)^-2)
 RatioTerrUDOverlapUD<-FinalCoef/OverlapCoef
 AllowedTerrSize.yi<- AllowedTerrSizeOE.yi*RatioTerrUDOverlapUD
 MeanTerrArea.yi<- min(c(AllowedTerrSize.yi, MeanTerrArea)) ##input smaller of 2
values
 SDTerrArea.yi<-min(c((AllowedTerrSize.yi/MeanTerrArea)*SDTerrArea, SDTerrArea))

 ##generate all spatial.yij for site yi (i level b/c they musn't overlap)
 SpatialList.yi<-list()
 SpatialCounter<-1

 while(length(SpatialList.yi)<NBirds.yi){
   ##Make a candidate set of spatial parameters
   SpatialCandidate.yij<-GenSpatialParameters(HRAreamean=MeanTerrArea.yi,
                                              HRAreaSD=SDTerrArea.yi,
                                              PercentUD.yij=UDTerr,
                                              ylim=Ylim,
                                              xlim=Xlim,
```

```
                                                  OverlapUD=OverlapUDTerr)

   ##For birds other than the first:
   if(length(SpatialList.yi)>0){
     CandidateDistances<-DistanceCenters(SpatialCandidate.yij,SpatialList.yi)
     ClosestNeighborDistance<- min(na.omit(CandidateDistances))
     ##Max&minDiameter based on desired % UD ellipse - more closely packed
territories
     ## % UD Ellipse determined by parameter OverlapUDTerr
     MaximumDiameter<-SpatialCandidate.yij[["aOE.yij"]] +

max(unname(as.numeric(unlist(SpatialList.yi)[names(unlist(SpatialList.yi))=="aOE.yij
"])))
     MinimumDiameter<-SpatialCandidate.yij[["bOE.yij"]] +

min(unname(as.numeric(unlist(SpatialList.yi)[names(unlist(SpatialList.yi))=="bOE.yij
"])))

     ## If nearest neighbor is FARTHER than candidate major radius +
     ## largest existing major radius, save candidate
     if(ClosestNeighborDistance > MaximumDiameter){
       SpatialList.yi[[length(SpatialList.yi)+1]]<-SpatialCandidate.yij
     }

     ## If nearest neighbor is CLOSER than candidate major radius +
     ## largest existing major radius, but FARTHER than candidate
     ## minor radius + smallest existing minor radius, then go
     ## through comparisons to see if
     ##candidate overlaps any existing territories or vice versa.
     if(ClosestNeighborDistance < MaximumDiameter &
```

```
          ClosestNeighborDistance > MinimumDiameter){

       #See if candidate overlaps any bird territories already generated
       Overlap<-
AxesCheckInsideEllipsesVectorizedOE(SpatialCandidate.yij,SpatialList.yi)
       if(Overlap==FALSE){

         ##See if any existing territory overlaps candidate territory
         SumUp<-0
         for(jjj in 1:length(SpatialList.yi)){
           SumUp<-SumUp + AxesCheckInsideEllipseOE(SpatialList.yi[[jjj]],
SpatialCandidate.yij)
         }
         ##if there are no overlaps, save that territory
         if(SumUp==0) {
           SpatialList.yi[[length(SpatialList.yi)+1]]<-SpatialCandidate.yij
         }
       }
     }
   }

   ##Generate territory for bird #1
   if(length(SpatialList.yi)==0){
     SpatialList.yi[[1]]<-SpatialCandidate.yij }
   SpatialCounter<-SpatialCounter+1
   if(SpatialCounter>5000) stop("5000 attempts to generate territories exceeded -
check density & NBirds.yi")
 }  ## bracket refers to: while(length(SpatialList.yi)<NBirds.yi){

 ##Plot ellipses for site i territories
```

```
#       par(mfrow=c(1,1))
#       plot(NULL,NULL,xlim=Xlim, ylim=Ylim)
#
#       for(iii in 1:length(SpatialList.yi)){
#         lines(GenEllipsePlot(SpatialList.yi[[iii]]))
#       }
# length(SpatialList.yi)
##Generate starting values & level-specific lists for storage
ListROutcomes.yijr[[y]][[i]]<-vector("list", NBirds.yi)
#       ListKOutcomes.yijrk[[y]][[i]]<-vector("list", NBirds.yi)
#       ListOOutcomes.yijrko[[y]][[i]]<-vector("list", NBirds.yi)
TempAll.j.Outcomes.i<-vector("list", NBirds.yi)

#TempAll.roj.Outcomes.i<-vector("list", NReps)
TempAll.rj.Outcomes.i<-vector("list", NReps)
for(rr in 1:NReps){
  TempAll.rj.Outcomes.i[[rr]]<-vector("list", NBirds.yi)
}

for(j in 1:NBirds.yi){

  ##Generate Bird-specific parameters & distributions
  Spatial.yij<-SpatialList.yi[[j]]
  #           GenSpatialParameters(MeanTerrArea,SDTerrArea,
  #                                           ylim=Ylim, xlim=Xlim)

  ##Generate starting values & level-specific lists for storage
  ListROutcomes.yijr[[y]][[i]][[j]]<-vector("list", NReps)
  #         ListKOutcomes.yijrk[[y]][[i]][[j]]<-vector("list", NReps)
  #         ListOOutcomes.yijrko[[y]][[i]][[j]]<-vector("list", NReps)
```

```
   TempAll.r.Outcomes.j<-vector("list", NReps)

   for(r in 1:NReps){

     ##Generate replication-specific parameters
     StartTimeinIntervals.yir<-
round(StartTimeinMinutes.yir.List[[y]][i,r]*IntervalsPerMinute,0)
##Minutes*(intervals/Minute)
     #           Leaves.yir<-Leaves.yir.List[[y]][i,r]

     ##Generate starting values & level-specific lists for storage
     #           ListKOutcomes.yijrk[[y]][[i]][[j]][[r]]<-vector("list", NIntervals)
     #           ListOOutcomes.yijrko[[y]][[i]][[j]][[r]]<-vector("list",
NIntervals)
     TempAll.k.Outcomes.r<-vector("list", NIntervals)
     TempAll.ko.Outcomes.r<-vector("list", NSimultaneousObservers)
     #TempAll.roj.Outcomes.i[[r]]<-vector("list", NSimultaneousObservers)


     for(oo in 1:NSimultaneousObservers){
       TempAll.ko.Outcomes.r[[oo]]<-vector("list", NIntervals)
       #TempAll.roj.Outcomes.i[[r]][[oo]]<-vector("list", NBirds.yi)
     }


     ##Small k-level lists - overwritten for each rep
     #  TempOutcomes.k<-list()

     #           WasBirdCounted.yijr<-0
     #           WasBirdCountedObs.yijr<-c(0,0)
```

```
     #           DistanceFirstCount.yijr<-rep(NA,NSimultaneousObservers)
     #           MinDistance.yijr<-NA
     #
     #

     for(k in 1:NIntervals){


       ##Generate interval-specific ENVIRONMENT

       TimeinIntervals.yirk<- StartTimeinIntervals.yir+ (k-1) ##time of survey
interval (in intervals)
       #            NoiseLevel.yirk<-Noise.yirk.List[[y]][[i]][[r]][k]
       ##Ambient noise is currently rep-specific only

       ##Generate interval-specific bird BEHAVIOR
       ##Spatial: Does it Move?

       PrBirdMoves.yijrk<-PrBirdMoves(N=1,MeanMovementRate=BackgroundMovementRate)

       ##Generate initial spatial information for k=1
       if(k==1){require(arm)
                Location.yijr0<-GenLocations(1,Spatial.yij)  ##Initial Location
                Distance.yijr0<-
ObsDistance(Location.yijr0[1],Location.yijr0[2])[["Distance.yijrk"]]

                ##Assumes no movement (flush) of birds due to obs arriving
                ##Thus, location at t=1 = location at t=0
                Location.yijrk<- Location.yijr0
```

```
                ##Below: Allow birds to flush at t=0 due to observer arrival
                #                      PrBirdMovesObs.yijr1<-
PrBirdMoveObs(Distance=ObsDistance(Location.yijr0[1],Location.yijr0[2])[["Distance.y
ijrk"]])
                #                      #                      PrBirdMovesObs.yijr1<-
invlogit(1-0.1*ObsDistance(Location.yijr0[1],Location.yijr0[2])[["Distance.yijrk"]])
                #                      ##Distance-dependent Pr(bird moves due to
obs) - drops to ~0.2 at 20 m
                #                      DoesBirdMoveObs.yijr1<-
rbinom(1,1,PrBirdMovesObs.yijr1)
                #
                #                      if(DoesBirdMoveObs.yijr1==0) Location.yijrk<-
Location.yijr0 ##If no movement, location at t=1 = location at t=0
                #                      if(DoesBirdMoveObs.yijr1!=0){
##If bird moves,
                #                        Distance.yijr0<-
ObsDistance(Location.yijr0[1],Location.yijr0[2])[["Distance.yijrk"]]  ## calc
distance at t=0
                #                        CandidateDistance<-0
                #                        CandidateDistanceCounter<-0
                #                        while(CandidateDistance < Distance.yijr0)
                #                        {CandidateLocation<-
GenLocations(1,Spatial.yij)  ##generate new locations until one is farther than
location at t=0
                #                         CandidateDistance<-
ObsDistance(CandidateLocation[1],CandidateLocation[2])[["Distance.yijrk"]]
                #                         CandidateDistanceCounter<-
CandidateDistanceCounter+1
                #                         if(CandidateDistanceCounter>1000) break}
```

```
                #
if(CandidateDistanceCounter>1000){Location.yijrk<- Location.yijr0}
                #
if(CandidateDistanceCounter<=1000){Location.yijrk<- CandidateLocation}
                #                      }
       }
       ##Generate spatial information for k>1
       if(k!=1){
         DoesBirdMove.yijrk<-rbinom(1,1,PrBirdMoves.yijrk)
         if(DoesBirdMove.yijrk==1) {Location.yijrk<- GenLocations(1,Spatial.yij)}
       }
       Distance.yijrk<-
ObsDistance(Location.yijrk[1],Location.yijrk[2])[["Distance.yijrk"]]

       ##Track smallest distance between observer & bird j
       #             if(k==1){MinDistance.yijr<-Distance.yijrk}
       #             if(k!=1){MinDistance.yijr<-
min(Distance.yijrk,MinDistance.yijr)}
       #

       #             WindSpeed.yijrk<-
as.numeric(WindSpeed.yirk.List[[y]][[i]][[r]][k])
       #             WindIndex.yijrk<-
WindSpeed.yijrk*cos(ObsDistance(Location.yijrk[1],Location.yijrk[2])[["BirdAngleComp
ass.yijrk"]])
       ##WindIndex assumes that wind is from due North at all sites (ok b/c bird
territories are spatially random wrt observer)


       ##Availability: Does It Sing?
```

```
       ##Maximum daily song rate for this species on day =
SurveyDate.yir.List[[y]][i,r]
       MaxDailySongRate.yir<-MaxDailySongRate(SurveyDate.yir.List[[y]][i,r])

       ##What is the effect (weight) of time of day on the max song rate?
       SongWeight.yijrk<-TimeOfDayWeight(TimeinIntervals.yirk, IntervalsPerMinute)

       ##What is interval-specific Pr(Bird Sings within ReportedPrSingMin minutes?)
       PrSing.yijrk<-MaxDailySongRate.yir*SongWeight.yijrk

       ##Here, PrSing.yijrk refers to Pr(bird sings at least 1x in ReportedPrSingMin
min (usually 5 or 10))
       ##Therefore, use DataIntervals = ReportedPrSingMin (min) *60
(sec/min)*(1/IntervalLength) (Intervals/sec)
       ##units for DataIntervals: (min) * (60sec/min) * (Intervals/sec) = Intervals
       ##PrSing calc is AFTER spatial to allow Pa ~ Distance to obs

       OptimizePSNS.yijrk<- optimize(SolvePSNS,
                              interval=c(0,0.05),   ##NOTE: **Starting
values** are VERY important here, suggest using interval=c(0,0.05) for all
                              PrSing=PrSing.yijrk,
                              PSS=PSS.yijr,

DataIntervals=ReportedPrSingMin*60*(1/IntervalLength),
                              maximum=F)
       if(OptimizePSNS.yijrk$objective>0.05) {stop("Failure to optimize PSNS")}
       PSNS.yijrk<- OptimizePSNS.yijrk$minimum
       q1.yijrk<-MarkovSS(PSS.yijr,PSNS.yijrk)[1]
```

```
       ##Determine Singing Mode (1/0) for interval k
       if(k==1){SingingMode.yijrk<-rbinom(1,1,q1.yijrk)  }
       if(k>1){
         PreviousSingingMode<-TempAll.k.Outcomes.r[[k-1]]["SingingMode.yijrk"]
         SingingMode.yijrk<- AutoCInstant(PreviousSingingMode,PSS.yijr,PSNS.yijrk)
       }

       ## Determine Singing (1/0) for interval k, given Singing Mode
       if(SingingMode.yijrk==0) {Sings.yijrk<-0;  SingingState.yijrk<-NA}
       if(SingingMode.yijrk==1 & k==1) {
         SingingState.yijrk<-sample(SingingStates.fine,1)
         ifelse(SingingState.yijrk=="S",Sings.yijrk<-1,Sings.yijrk<-0)
       }
       if(SingingMode.yijrk==1 & k>1){
         if(PreviousSingingMode==0) SingingState.yijrk<-sample(SingingStates.fine,1)

         if(PreviousSingingMode==1) {
           PreviousSingingState<-TempAll.k.Outcomes.r[[k-1]]["SingingState.yijrk"]

           if(PreviousSingingState=="S"|PreviousSingingState=="NS1"){
             PreviousSingingStateMatrix<-
matrix(as.numeric(SingingStates.fine==PreviousSingingState),1,4)
             NewSingingStateMatrix<-
PreviousSingingStateMatrix%*%TransitionMatrix.fine
             SingingState.yijrk<-
SingingStates.fine[as.numeric(NewSingingStateMatrix%*%c(1,2,3,4))]
           }
           if(PreviousSingingState=="NS2"){
             ifelse(rbinom(1,1,TransitionMatrix.fine[3,1])==1,
                    SingingState.yijrk<-"S",
```

```
                SingingState.yijrk<-"NS3")
           }
           if(PreviousSingingState=="NS3"){
             ifelse(rbinom(1,1,TransitionMatrix.fine[4,1])==1,
                SingingState.yijrk<-"S",
                SingingState.yijrk<-"NS3")
           }
           ifelse(SingingState.yijrk=="S",Sings.yijrk<-1,Sings.yijrk<-0)
         }

       }


       ##Generate starting values & level-specific lists for storage
       #              ListOOutcomes.yijrko[[y]][[i]][[j]][[r]][[k]]<-vector("list",
NSimultaneousObservers)
       TempAll.o.Outcomes.k<-vector("list", NSimultaneousObservers)

       for(o in 1:NSimultaneousObservers){  ##For NBirds.yi>0

         ##Perceptibility: Is it detected?
         if(SurveyType!="multiple"){
           ObserverIDNumber.yijrko<-ObserverIDNumber.yir.List[[y]][i,r]}

         if(SurveyType=="multiple"){
           ObserverIDNumber.yijrko<-o}

         Pdintercept.yijrko<-(
           #
GrasslandIEffect*IsSiteGrassland.yi +
```

```
           #
DeciduousIEffect*IsSiteDeciduous.yi +
           #                                       LeavesIEffect*Leaves.yir +
           #                                       NoiseIEffect*NoiseLevel.yirk +
           #
NoiseXLeavesEffect*NoiseLevel.yirk*Leaves.yir +
           #                                       WindIndexIEffect*WindIndex.yijrk
+
           SpeciesIntercept+
             ObserverIEffects[,ObserverIDNumber.yijrko] +
             HabitatIEffect*(1-PercentHabitat.yi)
         )

         Pdslope.yijrko<- (
           #                                   WindIndexSEffect*WindIndex.yijrk +
           #                                   GrasslandSEffect*IsSiteGrassland.yi +
           #                                   DeciduousSEffect*IsSiteDeciduous.yi +
           #                                   LeavesSEffect*Leaves.yir +
           #                                   NoiseSEffect*NoiseLevel.yirk +
           DistanceEffect.yijr +
             ObserverSEffects[,ObserverIDNumber.yijrko] +
             HabitatSEffect*(1-PercentHabitat.yi)+
             SiteSEffect*SiteVariationS.yi
         )

         Perceptibility.yijrko<-
invlogit(Pdintercept.yijrko+Pdslope.yijrko*Distance.yijrk)
         #               if(o==1){Perceptibility.yijrko<-0.5}
         #               if(o==2){Perceptibility.yijrko<-0.5}
```

```
         if(Sings.yijrk==0) {Detected.yijrko <-0}
         if(Sings.yijrk==1) {Detected.yijrko <-rbinom(1,1,Perceptibility.yijrko)}

         PrCorrectID.yijrko<-PrCorrectID

         if(Detected.yijrko ==0) {CorrectID.yijrko<-NA}
         if(Detected.yijrko ==1) {CorrectID.yijrko<-rbinom(1,1,PrCorrectID.yijrko)}

         PrDoubleCount.yijrko<-PrDoubleCount

         if(is.na(CorrectID.yijrko)==T) {DoubleCounted.yijrko<-NA}
         if(is.na(CorrectID.yijrko)==F){
           if(CorrectID.yijrko==0) {DoubleCounted.yijrko<-0}
           if(CorrectID.yijrko==1) {DoubleCounted.yijrko<-
rbinom(1,1,PrDoubleCount.yijrko)}
         }
         Count.yijrko<-sum(na.omit(0+CorrectID.yijrko + DoubleCounted.yijrko))

         #               if(Count.yijrko>0 & WasBirdCountedObs.yijr[o]==0){
         #                 ##for first count of bird, record distance to obs
         #                 DistanceFirstCount.yijr[o]<-Distance.yijrk}

         #               if(Count.yijrko>0) {WasBirdCounted.yijr<-1
         #                                   WasBirdCountedObs.yijr[o]<-1}

         ## Observer estimation of distance
         ObsEstimatedDistance.yijrko<-ObserverEstDistance(Distance.yijrk,

ObserverDistanceCategories,
                                                           Output="stochastic")
```

```
         ##Prevent negative estimated distances
         if(ObsEstimatedDistance.yijrko<0){ObsEstimatedDistance.yijrko<-1}
         ## Outcomes Complete    #####################################


         ##Track Rare phenomena
         if(is.na(CorrectID.yijrko)==F){
           if(CorrectID.yijrko==0) {MisID[nrow(MisID)+1,]<-c(y,i,j,r,k,o)}
         }
         if(is.na(DoubleCounted.yijrko)==F){
           if(DoubleCounted.yijrko==1) {DoubleCounted[nrow(DoubleCounted)+1,]<-
c(y,i,j,r,k,o)}
         }

         ## Index all o-specific objects for NBirds.yi>0
         TempOutcomes.o<-list(
           "Pdintercept.yijrko"=as.numeric(Pdintercept.yijrko),
           "Pdslope.yijrko"=as.numeric(Pdslope.yijrko),
           "Perceptibility.yijrko"=as.numeric(Perceptibility.yijrko),
           "Detected.yijrko"=as.numeric(Detected.yijrko),
           "PrCorrectID.yijrko"=as.numeric(PrCorrectID.yijrko),
           "CorrectID.yijrko"=as.numeric(CorrectID.yijrko),
           "PrDoubleCount.yijrko"=as.numeric(PrDoubleCount.yijrko),
           "DoubleCounted.yijrko"=as.numeric(DoubleCounted.yijrko),
           "Count.yijrko"=as.numeric(Count.yijrko),
           "ObsEstimatedDistance.yijrko"=as.numeric(ObsEstimatedDistance.yijrko)

#"ObserverIDNumber.yijrko"=ifelse(SurveyType=="multiple",ObserverIDNumber.yijrko,NA)
         )
```

```
         ##Store o-level objects and summaries
         #               ListOOutcomes.yijrko[[y]][[i]][[j]][[r]][[k]][[o]]<-
TempOutcomes.o
         #
         #               if(SurveyType!="multiple"){
         #                 ListOOutcomes.yijrko[[y]][[i]][[j]][[r]][[k]][[o]]<-
TempOutcomes.o
         #               }


         ##Temp Store all TempOutcomes.o for each k
         TempAll.o.Outcomes.k[[o]]<-TempOutcomes.o
         TempAll.ko.Outcomes.r[[o]][[k]]<-TempOutcomes.o

       }  ##o

       ##Line below checks that parameters are properly specified
       ##Needs to be here so it will break loops if parameters are wrong
       if(SurveyType %in% SurveyOptions==FALSE) stop("SurveyType is invalid or
misspelled")

       ## Index all k-specific objects for NBirds.yi>0
       TempOutcomes.k<-list(
         #               "WindSpeed.yijrk"=as.numeric(WindSpeed.yijrk),
         #               "WindIndex.yijrk"=as.numeric(WindIndex.yijrk),
         #               "NoiseLevel.yirk"=as.numeric(NoiseLevel.yirk),
         "TimeinIntervals.yirk"=as.numeric(TimeinIntervals.yirk),
         "SingingMode.yijrk"=as.numeric(SingingMode.yijrk),
         "SingingState.yijrk"=SingingState.yijrk,
         "PrSing.yijrk"=as.numeric(PrSing.yijrk),
```

```
  "Sings.yijrk"=as.numeric(Sings.yijrk),
  "PSNS.yijrk"=as.numeric(PSNS.yijrk),
  "q1.yijrk"=as.numeric(q1.yijrk),
  "PrBirdMoves.yijrk"=as.numeric(PrBirdMoves.yijrk),
  "DoesBirdMove.yijrk"=ifelse(k>1,as.numeric(DoesBirdMove.yijrk),NA),
  "Distance.yijrk"=as.numeric(Distance.yijrk),
  "Location.x.yijrk"=as.numeric(Location.yijrk[1]),
  "Location.y.yijrk"=as.numeric(Location.yijrk[2]),
  "SongWeight.yijrk"=as.numeric(SongWeight.yijrk),
  "TempAll.o.Outcomes.k"=TempAll.o.Outcomes.k
)
#             EverDetectedByObs.yijro,
#             EverCorrectIDByObs.yijro,
#             EverDoubleCounted.yijro,
#             Count.yijro

##Create k-level summaries

Summaries.k<-
  list(
    #TempAll.o.Outcomes.k[[1]][["DoubleCounted.yijrko"]],
    #TempAll.o.Outcomes.k[[1]][["Detected.yijrko "]]

  )

#             ##Store-k-level objects and summaries
#             ListKOutcomes.yijrk[[y]][[i]][[j]][[r]][[k]]<-list(
#               "XLocation.yijrk"=Location.yijrk[1],
#               "YLocation.yijrk"=Location.yijrk[2],
#               "SingingMode.yijrk"=SingingMode.yijrk,
```

```
  #                "SingingState.yijrk"=SingingState.yijrk,
  #                "Sings.yijrk"=Sings.yijrk)

  ##Temp Store all TempOutcomes.k for each r
  TempAll.k.Outcomes.r[[k]]<-TempOutcomes.k
  #                                      append(TempOutcomes.k,
  #                                      TempAll.o.Outcomes.k)


} ##k

## Index all r-specific objects
TempOutcomes.r<-list(
  #              "Leaves.yir"=Leaves.yir,
  "StartTimeinIntervals.yir"=StartTimeinIntervals.yir,
  "PSS.yijr"=PSS.yijr,
  "MaxDailySongRate.yir"=MaxDailySongRate.yir
)

##Create r-level summaries
## summaries.yijro
#"MeanPerceptibility.yijro"=MeanPerceptibility.yijro,
if(k!=NIntervals) stop("Intervals not complete! r summaries will be weird!")

#           TempAll.ko.Outcomes.r[[o]][[k]]
PreSummaries.r<-(vector("list", NSimultaneousObservers))

for(oo in 1:NSimultaneousObservers){
  EverDetectedByObs.yijro<-(sum(as.numeric(unlist(TempAll.ko.Outcomes.r[[oo]])[
    names(unlist(TempAll.ko.Outcomes.r[[oo]]))=="Detected.yijrko"] )))
```

```
       EverDetectedByObs.yijro<-ifelse(EverDetectedByObs.yijro>0,1,0)

       EverCorrectIDByObs.yijro<-ifelse(EverDetectedByObs.yijro>0,

sum(na.omit(as.numeric(unlist(TempAll.ko.Outcomes.r[[oo]])[

names(unlist(TempAll.ko.Outcomes.r[[oo]]))=="CorrectID.yijrko"] ))),
                                         NA)
       if(!is.na(EverCorrectIDByObs.yijro)){ifelse(EverCorrectIDByObs.yijro>0,1,0)}

       EverDoubleCounted.yijro<-ifelse(EverDetectedByObs.yijro>0,

sum(na.omit(as.numeric(unlist(TempAll.ko.Outcomes.r[[oo]])[

names(unlist(TempAll.ko.Outcomes.r[[oo]]))=="DoubleCounted.yijrko"] ))),
                                        NA)
       if(!is.na(EverDoubleCounted.yijro)){EverDoubleCounted.yijro<-
ifelse(EverDoubleCounted.yijro>0,1,0)}

       Count.yijro<-max(as.numeric(unlist(TempAll.ko.Outcomes.r[[oo]])[
         names(unlist(TempAll.ko.Outcomes.r[[oo]]))=="Count.yijrko"] ))

       if(Count.yijro>0){
         FirstIntervalCounted.yijro<-
which(unlist(TempAll.ko.Outcomes.r[[oo]])[names(unlist(TempAll.ko.Outcomes.r[[oo]]))
=="Count.yijrko"]>0)[1]
         DistAtFirstDetection.yijro<-
as.numeric(unlist(unname(TempAll.k.Outcomes.r[[FirstIntervalCounted.yijro]]["Distanc
e.yijrk"])))
```

```
         ObsEstDistAtFirstDetection.yijro<-
as.numeric(unlist(TempAll.ko.Outcomes.r[[oo]])[names(unlist(TempAll.ko.Outcomes.r[[o
o]]))=="ObsEstimatedDistance.yijrko"][FirstIntervalCounted.yijro])
       }
       if(Count.yijro==0){
         FirstIntervalCounted.yijro<-NA
         DistAtFirstDetection.yijro<-NA
         ObsEstDistAtFirstDetection.yijro<-NA
       }

       PreSummaries.r[[oo]]<-list(
         "EverDetectedByObs.yijro"=EverDetectedByObs.yijro,
         "EverCorrectIDByObs.yijro"=EverCorrectIDByObs.yijro,
         "EverDoubleCounted.yijro"=EverDoubleCounted.yijro,
         "Count.yijro"=Count.yijro,
         "FirstIntervalCounted.yijro"=FirstIntervalCounted.yijro,
         "DistAtFirstDetection.yijro"=DistAtFirstDetection.yijro,
         "ObsEstDistAtFirstDetection.yijro"=ObsEstDistAtFirstDetection.yijro
       )

       #TempAll.roj.Outcomes.i[[r]][[oo]][[j]]<-PreSummaries.r
     }  ##oo

     EverDetected.yijr<-
ifelse(sum(as.numeric(unlist(PreSummaries.r)[names(unlist(PreSummaries.r))=="EverDet
ectedByObs.yijro"]))>0,
                                       1,0)
     EverCorrectID.yijr<-
ifelse(sum(as.numeric(unlist(PreSummaries.r)[names(unlist(PreSummaries.r))=="EverCor
rectIDByObs.yijro"]))>0,
```

```
                                1,0)
     EverDoubleCounted.yijr<-
ifelse(sum(as.numeric(unlist(PreSummaries.r)[names(unlist(PreSummaries.r))=="EverDou
bleCounted.yijro"]))>0,
                                    1,0)
     EverCounted.yijr<-
ifelse(sum(as.numeric(unlist(PreSummaries.r)[names(unlist(PreSummaries.r))=="Count.y
ijro"]))>0,
                              1,0)
     FirstIntervalCounted.yijrObs1<-
as.numeric(unname(unlist(PreSummaries.r[[1]]["FirstIntervalCounted.yijro"])))
     DistAtFirstDetection.yijrObs1<-
as.numeric(unname(unlist(PreSummaries.r[[1]]["DistAtFirstDetection.yijro"])))
     ObsEstDistAtFirstDetection.yijrObs1<-
as.numeric(unname(unlist(PreSummaries.r[[1]]["ObsEstDistAtFirstDetection.yijro"])))

     if(SurveyType=="multiple"){

       ###NOTE: this section does not account for Double Counting
       if(PrDoubleCount>0){stop("ERROR: Multiple Observer analysis cannot currently
accept PrDoubleCount>0")}

       ##FirstIntervalCounted.yijrObs1 etc. are above - here, add 2nd obs for
multiple obs only
       FirstIntervalCounted.yijrObs2<-
as.numeric(unname(unlist(PreSummaries.r[[2]]["FirstIntervalCounted.yijro"])))
       DistAtFirstDetection.yijrObs2<-
as.numeric(unname(unlist(PreSummaries.r[[2]]["DistAtFirstDetection.yijro"])))
       ObsEstDistAtFirstDetection.yijrObs2<-
as.numeric(unname(unlist(PreSummaries.r[[2]]["ObsEstDistAtFirstDetection.yijro"])))
```

```
       CountObs1.yijr<-
as.numeric(unlist(unname(PreSummaries.r[[1]]["Count.yijro"])))
       CountObs2.yijr<-
as.numeric(unlist(unname(PreSummaries.r[[2]]["Count.yijro"])))

       OnlyObs1.yijr<-0
       OnlyObs2.yijr<-0
       BothObs.yijr<-0

       x11.yijr<-0
       x21.yijr<-0
       x22.yijr<-0
       x12.yijr<-0

       if(CountObs1.yijr!=0 | CountObs2.yijr!=0){  #If at least 1 obs had Count=1

         if(CountObs1.yijr>0 & CountObs2.yijr>0) {BothObs.yijr<-1}
         if(CountObs1.yijr>0 & CountObs2.yijr==0) {OnlyObs1.yijr<-1}
         if(CountObs1.yijr==0 & CountObs2.yijr>0) {OnlyObs2.yijr<-1}

         if(PrimaryObsNumber.yir.List[[y]][i,r]==1){ ##ObsA is primary obs
           if(CountObs1.yijr==0) {x21.yijr<-1}
           if(CountObs1.yijr==1) {x11.yijr<-1}
         }

         if(PrimaryObsNumber.yir.List[[y]][i,r]==2){ ##ObsB is primary obs
           if(CountObs2.yijr==0) {x12.yijr<-1}
           if(CountObs2.yijr==1) {x22.yijr<-1}
         }
```

```
      }
    }

    if(SurveyType=="removal"){
      if(RemovalPeriods!=3) stop("Removal Periods does not = 3. Need to change
removal period summary in Simulation")

      if(EverCounted.yijr==0){
        EverCountedPeriod1.yijr<-0
        EverCountedPeriod2.yijr<-0
        EverCountedPeriod3.yijr<-0
        FarnsEverCountedPeriod1.yijr<-0
        FarnsEverCountedPeriod2.yijr<-0
        FarnsEverCountedPeriod3.yijr<-0
      }

      if(EverCounted.yijr>0){
        Period1<-TempAll.ko.Outcomes.r[[o]][1:RemovalPeriod1Intervals]
        Period2<-
TempAll.ko.Outcomes.r[[o]][(RemovalPeriod1Intervals+1):(RemovalPeriod1Intervals+Remo
valPeriod2Intervals)]
        Period3<-
TempAll.ko.Outcomes.r[[o]][(RemovalPeriod1Intervals+RemovalPeriod2Intervals+1):NInte
rvals]

        FarnsPeriod1<-TempAll.ko.Outcomes.r[[o]][1:FarnsRemovalPeriod1Intervals]
        FarnsPeriod2<-
TempAll.ko.Outcomes.r[[o]][(FarnsRemovalPeriod1Intervals+1):(FarnsRemovalPeriod1Inte
rvals+FarnsRemovalPeriod2Intervals)]
```

```
         FarnsPeriod3<-
TempAll.ko.Outcomes.r[[o]][(FarnsRemovalPeriod1Intervals+FarnsRemovalPeriod2Interval
s+1):NIntervals]

         EverCountedPeriod1.yijr<-
ifelse(sum(as.numeric(unlist(Period1)[names(unlist(Period1))=="Count.yijrko"]))>0,1,
0)
         EverCountedPeriod2.yijr<-
ifelse(sum(as.numeric(unlist(Period2)[names(unlist(Period2))=="Count.yijrko"]))>0,1,
0)
         EverCountedPeriod3.yijr<-
ifelse(sum(as.numeric(unlist(Period3)[names(unlist(Period3))=="Count.yijrko"]))>0,1,
0)

         FarnsEverCountedPeriod1.yijr<-
ifelse(sum(as.numeric(unlist(FarnsPeriod1)[names(unlist(FarnsPeriod1))=="Count.yijrk
o"]))>0,1,0)
         FarnsEverCountedPeriod2.yijr<-
ifelse(sum(as.numeric(unlist(FarnsPeriod2)[names(unlist(FarnsPeriod2))=="Count.yijrk
o"]))>0,1,0)
         FarnsEverCountedPeriod3.yijr<-
ifelse(sum(as.numeric(unlist(FarnsPeriod3)[names(unlist(FarnsPeriod3))=="Count.yijrk
o"]))>0,1,0)
       }
     }

     EverSing.yijr<-
ifelse(sum(as.numeric(unlist(TempAll.k.Outcomes.r)[names(unlist(TempAll.k.Outcomes.r
))=="Sings.yijrk"]))>0,
                           1,0)
```

```
     EverSingingMode.yijr<-
ifelse(sum(as.numeric(unlist(TempAll.k.Outcomes.r)[names(unlist(TempAll.k.Outcomes.r
))=="SingingMode.yijrk"]))>0,
                                  1,0)
     EverMove.yijr<-
ifelse(sum(na.omit(as.numeric(unlist(TempAll.k.Outcomes.r)[names(unlist(TempAll.k.Ou
tcomes.r))=="DoesBirdMove.yijrk"])))>0,
                           1,0)
     NumberSongs.yijr<-
sum(as.numeric(unlist(TempAll.k.Outcomes.r)[names(unlist(TempAll.k.Outcomes.r))=="Si
ngs.yijrk"]))
     NumberMoves.yijr<-
sum(as.numeric(unlist(TempAll.k.Outcomes.r)[names(unlist(TempAll.k.Outcomes.r))=="Do
esBirdMove.yijrk"][2:NIntervals])  )
     MinDistance.yijr<-
min(as.numeric(unlist(TempAll.k.Outcomes.r)[names(unlist(TempAll.k.Outcomes.r))=="Di
stance.yijrk"]))
     MeanDistance.yijr<-
mean(as.numeric(unlist(TempAll.k.Outcomes.r)[names(unlist(TempAll.k.Outcomes.r))=="D
istance.yijrk"]))
     ClosestAxesDistance.yij<-AxesClosestObs(Spatial.yij) ##For territory
Spatial.yij, distance to closest of 4 major/minor axes
     #           MeanObservedWindspeed.yijr<-
mean(as.numeric(unlist(TempAll.k.Outcomes.r)[names(unlist(TempAll.k.Outcomes.r))=="W
indSpeed.yijrk"]))
     #           MeanObservedNoise.yijr<-
mean(as.numeric(unlist(TempAll.k.Outcomes.r)[names(unlist(TempAll.k.Outcomes.r))=="N
oiseLevel.yirk"]))

     ##NOTE: Noiselevel is currently r-specific
```

```
     ## Therefore, NoiseLevel.yirk==Noise.yirk.List[[y]][[i]][[r]][k] for any k

     Summaries.r<-list(
       #             "PreSummaries.r"=PreSummaries.r,
       #
"ObserverIDNumber.yir"=ifelse(SurveyType!="multiple",ObserverIDNumber.yijrko,NA),
       #             "MeanObservedWindspeed.yijr"=MeanObservedWindspeed.yijr,
       #             "MeanObservedNoise.yijr"=MeanObservedNoise.yijr,
       "EverDetected.yijr"=EverDetected.yijr,
       "EverCorrectID.yijr"=EverCorrectID.yijr,
       "EverDoubleCounted.yijr"=EverDoubleCounted.yijr,
       "EverCounted.yijr"= EverCounted.yijr,
       "EverSing.yijr"=EverSing.yijr,
       "EverSingingMode.yijr"=EverSingingMode.yijr,
       "EverMove.yijr"=EverMove.yijr,
       "NumberSongs.yijr"=NumberSongs.yijr,
       "NumberMoves.yijr"=NumberMoves.yijr,
       "MinDistance.yijr"=MinDistance.yijr,
       "CountObs1.yijr"=ifelse(SurveyType=="multiple",CountObs1.yijr,NA),
       "CountObs2.yijr"=ifelse(SurveyType=="multiple",CountObs2.yijr,NA),
       "x11.yijr"=ifelse(SurveyType=="multiple",x11.yijr,NA),
       "x12.yijr"=ifelse(SurveyType=="multiple",x12.yijr,NA),
       "x21.yijr"=ifelse(SurveyType=="multiple",x21.yijr,NA),
       "x22.yijr"=ifelse(SurveyType=="multiple",x22.yijr,NA),
       "OnlyObs1.yijr"=ifelse(SurveyType=="multiple",OnlyObs1.yijr,NA),
       "OnlyObs2.yijr"=ifelse(SurveyType=="multiple",OnlyObs2.yijr,NA),
       "BothObs.yijr"=ifelse(SurveyType=="multiple",BothObs.yijr,NA),

"EverCountedPeriod1.yijr"=ifelse(SurveyType=="removal",EverCountedPeriod1.yijr,NA),
```

```
"EverCountedPeriod2.yijr"=ifelse(SurveyType=="removal",EverCountedPeriod2.yijr,NA),

"EverCountedPeriod3.yijr"=ifelse(SurveyType=="removal",EverCountedPeriod3.yijr,NA),

"FarnsEverCountedPeriod1.yijr"=ifelse(SurveyType=="removal",FarnsEverCountedPeriod1.
yijr,NA),

"FarnsEverCountedPeriod2.yijr"=ifelse(SurveyType=="removal",FarnsEverCountedPeriod2.
yijr,NA),

"FarnsEverCountedPeriod3.yijr"=ifelse(SurveyType=="removal",FarnsEverCountedPeriod3.
yijr,NA),
       "MeanDistance.yijr"=MeanDistance.yijr,
       "Distance.yijr0"=Distance.yijr0,
       "ClosestAxesDistance.yij"=ClosestAxesDistance.yij,
       "FirstIntervalCounted.yijrObs1"=FirstIntervalCounted.yijrObs1,
       "DistAtFirstDetection.yijrObs1"=DistAtFirstDetection.yijrObs1,
       "ObsEstDistAtFirstDetection.yijrObs1"=ObsEstDistAtFirstDetection.yijrObs1,

"FirstIntervalCounted.yijrObs2"=ifelse(SurveyType=="multiple",FirstIntervalCounted.y
ijrObs2,NA),

"DistAtFirstDetection.yijrObs2"=ifelse(SurveyType=="multiple",DistAtFirstDetection.y
ijrObs2,NA),

"ObsEstDistAtFirstDetection.yijrObs2"=ifelse(SurveyType=="multiple",ObsEstDistAtFirs
tDetection.yijrObs2,NA)
     )
```

```
      ##Consider adding - mean perceptibility (obs-specific!)
      #"Count.yijr"= max(unlist(TempOutcomes.o)["Count.yijrko"]),

      ##Store-r-level objects and summaries for NBirds.yi>0
      ListROutcomes.yijr[[y]][[i]][[j]][[r]]<-append(TempOutcomes.r,Summaries.r)
#TempOutcomes.r
      TempAll.rj.Outcomes.i[[r]][[j]]<-Summaries.r

      ##Temp Store all TempOutcomes.r for each j
      TempAll.r.Outcomes.j[[r]]<-TempOutcomes.r

    } ##r

    ## Index all j-specific objects
    if(NBirds.yi>0){
      TempOutcomes.j<-list(
        "Spatial.yij"=Spatial.yij
      )
    }

    if(NBirds.yi==0){
      TempOutcomes.j<-list(
        "Spatial.yij"=NA
      )
    }

    ##Create j-level summaries

    ##Store-j-level objects and summaries
```

```
    ## Currently there is no ListJOutcomes.yij !!


    ##Temp Store all TempOutcomes.j for each i
    TempAll.j.Outcomes.i[[j]]<-TempOutcomes.j

  } ##j


}  ##bracket refers to "if(NBirds.yi>0){"

## Index all i-specific objects
if(NBirds.yi==0){
  TempOutcomes.i<-list(
    #         "IsSiteGrassland.yi"=IsSiteGrassland.yi,
    #         "IsSiteDeciduous.yi"=IsSiteDeciduous.yi,
    #         "SpatialList.yi"=NA,
    "PercentHabitat.yi"=PercentHabitat.yi,
    "Area.yi"=Area.yi,
    "LambdaHabitat.yi"=LambdaHabitat.yi,
    "LambdaMatrix.yi"=LambdaMatrix.yi,
    "BirdsInHabitat.yi"=BirdsInHabitat.yi,
    "BirdsInMatrix.yi"=BirdsInMatrix.yi,
    "NBirds.yi"=NBirds.yi,
    "AllowedTerrSize.yi"=NA,
    "MeanTerrArea.yi"=NA,
    "SDTerrArea.yi"=NA
  )
}
```

```
if(NBirds.yi>0){
  ## Realized mean & sd of area of 95% UD ellipses for territories
  RealTerrAreaMean.yi<-
 mean(unlist(SpatialList.yi)[names(unlist(SpatialList.yi))=="Area.yij"])
  RealTerrAreaSD.yi<-
 sd(unlist(SpatialList.yi)[names(unlist(SpatialList.yi))=="Area.yij"])

  TempOutcomes.i<-list(
    #         "IsSiteGrassland.yi"=IsSiteGrassland.yi,
    #         "IsSiteDeciduous.yi"=IsSiteDeciduous.yi,
    #         "SpatialList.yi"=SpatialList.yi  ##Left out b/c info is stored at j
 level
    "RealTerrAreaMean.yi"=RealTerrAreaMean.yi,
    "RealTerrAreaSD.yi"=RealTerrAreaSD.yi,
    "PercentHabitat.yi"=PercentHabitat.yi,
    "HabitatGroup.yi"=ifelse(HabitatGroup.yi=="Low",1,0),
    "Area.yi"=Area.yi,
    "LambdaHabitat.yi"=LambdaHabitat.yi,
    "LambdaMatrix.yi"=LambdaMatrix.yi,
    "BirdsInHabitat.yi"=BirdsInHabitat.yi,
    "BirdsInMatrix.yi"=BirdsInMatrix.yi,
    "NBirds.yi"=NBirds.yi,
    "AllowedTerrSize.yi"=AllowedTerrSize.yi,
    "MeanTerrArea.yi"=MeanTerrArea.yi,
    "SDTerrArea.yi"=MeanTerrArea.yi
  )
}

##Create i-level summaries
```

```
#     TempAll.rj.Outcomes.i[[r]][[j]][["EverDetected.yijr"]]
#     length(TempAll.rj.Outcomes.i[[1]])

Summaries.ir<-vector("list", NReps)
for(rr in 1:NReps){

  EverCount.yir<-sum(as.numeric(unlist(TempAll.rj.Outcomes.i[[rr]])[
    names(unlist(TempAll.rj.Outcomes.i[[rr]]))=="EverDetected.yijr"] ))
  EverMove.yir<-sum(na.omit(as.numeric(unlist(TempAll.rj.Outcomes.i[[rr]])[
    names(unlist(TempAll.rj.Outcomes.i[[rr]]))=="EverMove.yijr"] )))
  EverDoubleCounted.yir<-sum(na.omit(as.numeric(unlist(TempAll.rj.Outcomes.i[[rr]])[
    names(unlist(TempAll.rj.Outcomes.i[[rr]]))=="EverDoubleCounted.yijr"] )))
  EverCorrectID.yir<-sum(na.omit(as.numeric(unlist(TempAll.rj.Outcomes.i[[rr]])[
    names(unlist(TempAll.rj.Outcomes.i[[rr]]))=="EverCorrectID.yijr"] )))
  EverSingingMode.yir<-sum(as.numeric(unlist(TempAll.rj.Outcomes.i[[rr]])[
    names(unlist(TempAll.rj.Outcomes.i[[rr]]))=="EverSingingMode.yijr"] ))
  EverSing.yir<-sum(as.numeric(unlist(TempAll.rj.Outcomes.i[[rr]])[
    names(unlist(TempAll.rj.Outcomes.i[[rr]]))=="EverSing.yijr"] ))

  Summaries.ir[[rr]]<-list(
    "EverCount.yir"=EverCount.yir,
    "EverMove.yir"=EverMove.yir,
    "EverDoubleCounted.yir"=EverDoubleCounted.yir,
    "EverCorrectID.yir"=EverCorrectID.yir,
    "EverSingingMode.yir"=EverSingingMode.yir,
    "EverSing.yir"=EverSing.yir
  )
}
##Store-i-level objects and summaries
```

```
  ListIOutcomes.yi[[y]][[i]]<-TempOutcomes.i
  ListIRSummaryOutcomes.yi[[y]][[i]]<- Summaries.ir

  print(paste("site", i))
  print(proc.time() - TIMER)

  ##Temp Store all TempOutcomes.i for each y
  TempAll.i.Outcomes.y[[i]]<-TempOutcomes.i

} ##i


## Index all y-specific objects (Both NBirds.yi==0 & NBirds.yi>0)
TempOutcomes.y<-list(
  # "SurveyType"=SurveyType,
  "FauxYear"=FauxYear,
  "Ns.y"= Ns.y,
  "ContinentalPopulation.y"= ContinentalPopulation.y,
  "ObserverIEffects.y1"=ObserverIEffects[,1],
  "ObserverIEffects.y2"=ObserverIEffects[,2],
  "RangeHabitatArea.y"= RangeHabitatArea.y,
  "RangeMatrixArea.y"= RangeMatrixArea.y,
  "StudyHabitatArea.y"= StudyHabitatArea.y,
  "StudyMatrixArea.y"= StudyMatrixArea.y,
  "RangeHabitatAbundance.y"=RangeHabitatAbundance.y,
  "RangeMatrixAbundance.y"= RangeMatrixAbundance.y,
  "MatrixDensity.y"= MatrixDensity.y,
  "HabitatDensity.y"= HabitatDensity.y,
  "StudyHabitatAbundance.y"= StudyHabitatAbundance.y,
  "StudyMatrixAbundance.y"= StudyMatrixAbundance.y,
```

```
  "HabitatProportionTheta.y"= HabitatProportionTheta.y,
  "StudyHabitatProportion.y"=StudyHabitatProportion.y,
  "SurveyLength"=SurveyLength,
  "RemovalPeriod1Length"=ifelse(SurveyType=="removal",RemovalPeriod1Length,NA),
  "RemovalPeriod2Length"=ifelse(SurveyType=="removal",RemovalPeriod2Length,NA),
  "RemovalPeriod3Length"=ifelse(SurveyType=="removal",RemovalPeriod3Length,NA),
  "RemovalPeriod1Intervals"=ifelse(SurveyType=="removal",RemovalPeriod1Intervals,NA),
  "RemovalPeriod2Intervals"=ifelse(SurveyType=="removal",RemovalPeriod2Intervals,NA),
  "RemovalPeriod3Intervals"=ifelse(SurveyType=="removal",RemovalPeriod3Intervals,NA)
)

## Create y-level summaries

MeanPercentHabitat.y<-
   mean(as.numeric(unlist(ListIOutcomes.yi[[y]])[names(unlist(ListIOutcomes.yi[[y]]))==
   "PercentHabitat.yi"]))
#mean(ListIOutcomes.yi[[y]][["PercentHabitat.yi"]])

Summaries.y<-list(
  "MeanPercentHabitat.y"=MeanPercentHabitat.y
)


## Store-y-level objects and summaries
ListYOutcomes.y[[y]]<- append(TempOutcomes.y,Summaries.y)
#ListYSummaryOutcomes.y[[y]]<- Summaries.y

#print(paste(SurveyType,"FauxYear", FauxYear, "Complete"))

# ###########################################################
```

```
  # ##Save dataset for year y for future analysis
#   DATE<-format(Sys.time(), "%m_%d_%Y")
  SimName<-paste(SimulationName,SIMNUM,"Scenario",Scenario,SurveyType,"FauxYear",
     FauxYear, sep="_")

  setwd(SaveDirectory)

  save(NYears, NSurveySites, NReps, SurveyType, SurveyOptions,
       NIntervals, NSimultaneousObservers, NTotalObservers, ObserverIDs,
       PrimaryObsNumber.yir.List,
       ListYOutcomes.y,
       ListIOutcomes.yi,
       ListROutcomes.yijr,
       ListIRSummaryOutcomes.yi,
       #        ListKOutcomes.yijrk,
       #        ListOOutcomes.yijrko,
       file=paste(SimName,".RData", sep=""))

  setwd(MasterDirectory)



} ##y


##Clean up data frames tracking rare phenomena
if(nrow(MisID)>1) MisID<-na.omit(MisID)
if(nrow(DoubleCounted)>1) DoubleCounted<-na.omit(DoubleCounted)
```

```
#proc.time() - TIMERALL
########END_SIMULATION###########################################



##################################
##SUMMARIZE SIMULATION RESULTS
library(reshape)

##NOTE: Lists from Sim stored but remain in memory for use in this script.
##thus: There is no "load" currently in this script.

##Melt the lists generated by the simulation
##NOTE: KK and OO not used because they're too slow
##      r-level summaries through simulation should replace them


YY<-melt.list(ListYOutcomes.y)
#head(YY)

II<-melt.list(ListIOutcomes.yi)
#head(II)

RR<-melt.list(ListROutcomes.yijr)
#head(RR)


#head(ListIRSummaryOutcomes.yi)
IIRR<-melt.list(ListIRSummaryOutcomes.yi)
#head(IIRR)
```

```
# # #TIMER<-proc.time()
# KK<-melt.list(ListKOutcomes.yijrk)
# # head(KK)

# #ListOOutcomes.yijrko[[1]][[1]][[1]][[1]][[1]]
# #TIMER<-proc.time()
# OO<-melt.list(ListOOutcomes.yijrko)
# #proc.time() - TIMER
# #head(OO)
# # nrow(OO)
# # nrow(II)
# ## With y=1, i=20, j~40 this takes 10 min.
# ## Object OO has 2 million lines with only y=1,i=20,j~150,r=1,k=60,o=2
# ##If you're going to run this, run it over lunch.
# # proc.time() - MeltTIMER
#
# # ifelse(length(unique(KK$X1))==1, KK$X1<-NULL, print("Cannot delete column - more than
     one value exists!"))
# # ifelse(length(unique(KK$L6))==1, KK$L6<-NULL, print("Cannot delete column - more than
     one value exists!"))

############################################
##Cast the data to get useful summaries
## .yi level
{
# cast(II, .~L3, length) ## number of data points for each variable,  (=y*i)
#
# cast(II, .~L3, mean) ##Mean variable values across all years
# tapply(II$value, II$L3, mean) ##Same results as above, different structure
```

```
#
# cast(II, .~L3+L1, mean) ##Mean variable values, year-specific
# cast(II, .~L3+L2, mean) ##Mean variable values, site-specific

}

CastTIMER<-proc.time()

#############################################
##Create .y data frame with all necessary columns
YY.data<-cast(YY, ...~L2)
# head(YY.data)

# unique(YY.data$L1)
names(YY.data)[names(YY.data)=="L1"]<-"y"  ##rename columns

length(unique(YY.data$y))==NYears
##Make sure this is TRUE!!!


#############################################
##Create .yi data frame with all necessary columns
II.data<-cast(II, ...~L3)
# head(II.data)
# hist(II.data$NBirds.yi)

# unique(II.data$L1)
names(II.data)[names(II.data)=="L1"]<-"y"  ##rename columns
names(II.data)[names(II.data)=="L2"]<-"i"
```

```
length(unique(II.data$y))==NYears
length(unique(II.data$i))==NSurveySites
##Make sure these are TRUE!!!

########################################
head(IIRR)
IIRR.data<-cast(IIRR, ...~L4)

head(IIRR.data)

#unique(IIRR.data$L1)
names(IIRR.data)[names(IIRR.data)=="L1"]<-"y"  ##rename columns
#unique(IIRR.data$L2)
names(IIRR.data)[names(IIRR.data)=="L2"]<-"i"
#unique(IIRR.data$L3)
names(IIRR.data)[names(IIRR.data)=="L3"]<-"r"


length(unique(IIRR.data$y))==NYears
length(unique(IIRR.data$i))==NSurveySites
length(unique(IIRR.data$r))==NReps
##Make sure these are TRUE!!!
########################################

##Create .yijr data frame with all necessary columns
head(RR)
RR.data<-cast(RR, ...~L5)
head(RR.data)
#RR.data
```

```
#unique(RR.data$L1)
names(RR.data)[names(RR.data)=="L1"]<-"y"  ##rename columns
#unique(RR.data$L2)
names(RR.data)[names(RR.data)=="L2"]<-"i"
#unique(RR.data$L3)
names(RR.data)[names(RR.data)=="L3"]<-"j"
#unique(RR.data$L4)
names(RR.data)[names(RR.data)=="L4"]<-"r"

length(unique(RR.data$y))==NYears
length(unique(RR.data$i))==NSurveySites
length(unique(RR.data$r))==NReps
##Make sure these are TRUE!!!

##RR.1 replaced by IIRR.data
##For nmixtures, need RR.1
# RR.1<-cast(RR, L1+L2~L5+L4, sum)
# # unique(RR.1$L2)
# # unique(RR.1$L1)
# names(RR.1)[names(RR.1)=="L1"]<-"y"  ##rename columns
# names(RR.1)[names(RR.1)=="L2"]<-"i"
#
# length(unique(RR.1$y))==NYears
# length(unique(RR.1$i))==NSurveySites
# ##Make sure these are TRUE!!!
#
# head(RR.1)


########################################
```

```
# ##Create .yijrk data frame with all necessary columns
# # head(KK)
# #nrow(KK)
# #unique(KK$L6)
#
# # unique(KK$L1) #y
# # unique(KK$L5) #k
#
# KK.data<-cast(KK, ...~L6)
# # head(KK.data)
#
# #unique(KK.data$L1)
# #unique(KK.data$L2)
# #unique(KK.data$L3)
# #unique(KK.data$L4)
# #unique(KK.data$L5)
# names(KK.data)[names(KK.data)=="L1"]<-"y"  ##rename columns
# names(KK.data)[names(KK.data)=="L2"]<-"i"
# names(KK.data)[names(KK.data)=="L3"]<-"j"
# names(KK.data)[names(KK.data)=="L4"]<-"r"
# names(KK.data)[names(KK.data)=="L5"]<-"k"
# #
# length(unique(KK.data$y))==NYears
# length(unique(KK.data$i))==NSurveySites
# length(unique(KK.data$r))==NReps
# length(unique(KK.data$k))==NIntervals
# ##Make sure these are TRUE!!!
#
# # head(KK.data)
# #
```

```
# # KK.data[KK.data[,"y"]==y &
# #           KK.data[,"i"]==i &
# #           KK.data[,"j"]==j &
# #           KK.data[,"r"]==r &
# #           KK.data[,"k"]==k,]

# ###############################################################

##Create .yijrko data frame with all necessary columns
# head(OO)
# nrow(OO)
# unique(OO$L6)
#
# unique(OO$L1) #y
# unique(OO$L5) #k
# unique(OO$L6) #o
#
# OO.data<-cast(OO, ...~L7)
# # head(OO.data)
# # nrow(OO.data)
# unique(OO.data$L1)
# unique(OO.data$L2)
# unique(OO.data$L3)
# unique(OO.data$L4)
# unique(OO.data$L5)
# names(OO.data)[names(OO.data)=="L1"]<-"y"  ##rename columns
# names(OO.data)[names(OO.data)=="L2"]<-"i"
# names(OO.data)[names(OO.data)=="L3"]<-"j"
# names(OO.data)[names(OO.data)=="L4"]<-"r"
# names(OO.data)[names(OO.data)=="L5"]<-"k"
```

```
# names(OO.data)[names(OO.data)=="L6"]<-"o"
#
# length(unique(OO.data$y))==NYears
# length(unique(OO.data$i))==NSurveySites
# length(unique(OO.data$r))==NReps
# length(unique(OO.data$k))==NIntervals
# length(unique(OO.data$o))==NSimultaneousObservers
# ##Make sure these are TRUE!!!
#
# # head(OO.data)
#

# proc.time()- CastTIMER
# print(paste(SurveyType, "FauxYear=",FauxYear, "Reshape Complete"))

# OO.data[OO.data[,"y"]==y &
#          OO.data[,"i"]==i &
#          OO.data[,"j"]==j &
#          OO.data[,"r"]==r &
#          OO.data[,"k"]==k,]



## Attempted to save memory by removing unecessarly large original objects.
# rm(OO)
# rm(KK)
##NOTE: Didn't seem to work.  These took a long time to generate, so don't do this in
     future.

###########################################################
```

```
##Save dataset for future use
DATE<-format(Sys.time(), "%m_%d_%Y")
#  AnalysisName<-paste("Analysis",SIMNUM,SurveyType,"FauxYear", FauxYear, sep="_")
AnalysisName<-
     paste("Analysis",SimulationName,SIMNUM,"Scenario",Scenario,SurveyType,"FauxYear",
     FauxYear, sep="_")

#setwd("C:\\Users\\rigby007\\Documents\\Detection\\Dissertation Project\\Sim 1 - BTBW
     example for AOU 2014\\")
setwd(SaveDirectory)

save(NYears, NSurveySites, NReps, SurveyType, SurveyOptions,
     NIntervals, NSimultaneousObservers, NTotalObservers, ObserverIDs,
     PrimaryObsNumber.yir.List,
     #PrimaryObsNumber.yir,
     YY.data, IIRR.data, II.data, RR.data,
     # RR.1,  KK.data, OO.data,
     file=paste(AnalysisName,".RData", sep=""))

setwd(MasterDirectory)

#   rm(ListYOutcomes.y)
#   rm(ListIOutcomes.yi)
#   rm(ListIRSummaryOutcomes.yi)
#   rm(ListROutcomes.yijr)

################################################
####END SUMMARIZE SIMULATION RESULTS
```

APPENDIX C

Supplemental Methods

Spatial Modeling

Spatial parameters (*Spatial.yij*) were generated for each bird *j*, including the center of the territory (*CenterX.yij*, *CenterY.yij*), the area of a 95% elliptical density contour (*Area.yij*), the eccentricity of the ellipse (*Ecc.yij*), and an angle of rotation (*Theta.yij*). *CenterX.yij* and *CenterY.yij* were uniform random variables: U(-1000, 1000). *Area.yij* was a lognormally-distributed random variable with parameters μ and σ^2. Using mean and standard deviation parameters for territory size (*MeanTerrArea* and *SDTerrArea*, respectively), μ and σ^2 were calculated as

$$\mu = \ln(MeanTerrArea) - \frac{1}{2}\ln(1 + \frac{(SDTerrArea)^2}{(MeanTerrArea)^2}) \text{ , (eq. C.1)}$$

and

$$\sigma^2 = \ln(1 + \frac{(SDTerrArea)^2}{(MeanTerrArea)^2}) \text{ . (eq. C.2)}$$

Ecc.yij was a uniform random variable: U(0,0.95). An ellipse with eccentricity *Ecc.yij* had a semi-major axis *a.yij* and semi-minor axis *b.yij*:

$$b.yij = a.yij\sqrt{1 - Ecc.yij^2} \text{ , (eq. C.3)}$$

$$a.yij = \frac{b.yij}{\sqrt{1-Ecc.yij^2}} \text{ , (eq. C.4)}$$

Theta.yij was a uniform random variable describing the angle at which the elliptical territory was rotated: U(0°, 360°).

The overlap comparison of territories was computationally intensive, so a precursor step was added to reduce simulation run-time. In that step, the distances between the center of the candidate territory and all existing territories (*CandidateDistances*, the smallest value of which was *ClosestNeighborDistance*) was compared to the length of the axes of all territories. The sum of the candidate major axis and largest existing major axis was *MaxmimumDiameter* and the sum of the candidate minor axis and smallest existing minor axis was *MinimumDiameter*. If *ClosestNeighborDistance* > *MaximumDiameter*, then there could be no overlap and the candidate was retained. If *ClosestNeighborDistance* < *MinimumDiameter*, then there was overlap and the candidate was rejected. If *ClosestNeighborDistance* < *MaximumDiameter* and *ClosestNeighborDistance* > *MinimumDiameter,* then the direct overlap comparison was carried out.

Using empirical estimates of territory size to create non-overlapping territories was not expedient (or even not possible) when estimates of territory size and density from the literature conflicted. For example, it would be impossible to create non-overlapping territories with mean size 1 ha if density = 3 pairs / ha. My solution had 2 parts. First, I adjusted the elliptical density contour percentage used to evaluate overlap (using a smaller percentage increased allowable overlap). Second, I compressed the mean area of territories at sites with high abundance. Via simulations, I found that computation time increased drastically when the area covered by ellipses exceeded 50% of the modeled area. I therefore used

$$MeanTerrArea.yi = \frac{0.5 \times Area.yi}{NBirds.yi} \quad \text{(eq. C.5)}$$

to parameterize *Spatial.yij* for sites where abundance was problematically high.

Availability Modeling

For the purposes of modeling, I wanted to specify parameters for P(S|S), P(S|NS), and *NIntervals* (see equation 1.11). I therefore needed to rearrange equation 1.11 in terms of those variables. By definition (Stroock 2005), the steady state vector is the vector for which the following is true, given enough time for initial states to be “forgotten”:

$$[q_1\ q_2] = [q_1\ q_2] \times P_{coarse} \quad \text{(eq. C.6)},$$

where P_{coarse} is the coarse-scale interval-specific transition matrix (Table 1.1). I also made use of the identity matrix I

$$[q_1\ q_2] = [q_1\ q_2] \times I\ , \quad \text{(eq. C.7)}$$

to obtain

$$[q_1\ q_2] = [q_1\ q_2] \times P_{coarse} = [q_1\ q_2] \times I\ . \quad \text{(eq. C.8)}$$

By rearranging equation C.8, I obtained

$$[q_1\ q_2] \times (P_{coarse} - I) = 0\ . \quad \text{(eq. C.9)}$$

Completing the matrix multiplication of equation C.9 with the contents of P_{coarse} (Table 1.1) produced

$$q_1(P(S|S) - 1) + q_2(P(S|NS)) = 0\ , \quad \text{(eq. C.10)}$$

and

$$q_1(P(NS|S)) + q_2(P(NS|NS) - 1) = 0\ . \quad \text{(eq. C.11)}$$

Combining and rearranging equation C.10 and equation C.11 gave

$$q_1 = \frac{-\,P(S|NS)}{P(S|S) - P(S|NS) - 1} \quad . \text{ (eq. C.12)}$$

Therefore, substituting equation 1.6 and equation C.12 into equation 1.7 produced

$$Z_{NIntervals} = \frac{-P(S|NS)}{P(S|S) - P(S|NS) - 1} + \left(1 - \frac{-P(S|NS)}{P(S|S) - P(S|NS) - 1}\right) \times \ (1-(1-P(S|NS))^{NIntervals})$$

,

(eq. C.13)

thus describing $Z_{NIntervals}$ in terms of 3 parameters, P(S|S), P(S|NS), and *NIntervals*. *NIntervals* is defined by the survey length of the desired scenario. I assumed that more variation in bird availability would be caused by variation in the probability of birds switching from non-singing mode into singing mode (P(S|NS)) than by variation in the tendency for birds to stay in singing mode (P(S|S)). I therefore held P(S|S) constant and varied P(S|NS) to achieve the desired $Z_{NIntervals}$. I investigated the relationship between P(S|S), P(S|NS), and $Z_{NIntervals}$ for different values of P(S|S) (Figure 1.1). P(S|S) > 0.98 produced too few transitions (birds in singing mode almost never transitioned) and P(S|S) < 0.98 produced birds that transitioned too quickly (birds were so likely to transition into singing mode that q_1 was very low and variation in $Z_{NIntervals}$ among simulations was high). I therefore held constant P(S|S) = 0.98 for all simulations. The P(S|NS) needed to produce a desired singing probability was determined with function optimize() in program R. Because values of P(S|NS) were very small, the optimization process was sensitive to starting values; all analyses used a starting interval of 0 - 0.05 for P(S|NS).

I used equation C.13, with $Z_{NIntervals}$ = *PrSing.yijrk*, P(S|S) = *PSS.yijr* = 0.98, and *NIntervals* (analysis method-specific), to determine the optimized value for P(S|NS) (*PSNS.yijrk*). The interval-specific singing mode *SingingMode.yijrk* was determined from the interval-specific transition matrix P_{coarse} (Table 1.1), using those values of *PSS.yijr* and *PSNS.yijrk*.

For example, a species with an interval length of 2 seconds and a desired singing probability $Z_{NIntervals}$ = 0.6 could be produced by using P(S|S) = 0.98 and P(S|NS) = 0.0037, creating the transition matrix

$$P_{coarse} = \begin{vmatrix} P(S|S) = 0.98 & P(NS|S) = 0.02 \\ P(S|NS) = 0.0037 & P(NS|NS) = 0.9963 \end{vmatrix} .$$

Abundance Parameters

Parameters for the scenario were based on a study area (with size *StudyArea* = 3,000 ha) within the species' range (with size *RangeArea* = 3.62×10^{7} ha, BirdLife International 2016). The simulated study area was approximately the size of the Hubbard Brook Experimental Forest, a site where much BTBW research has taken place (e.g., Sherry and Holmes 1985, Holmes et al.1986, Holmes et al. 1996). I used an empirical estimate for density of breeding pairs in BTBW habitat (*HabitatDensity.y* = 0.534 birds/ha, Holmes et al. 1986). I used Partners in Flight's (2015) estimate of continental abundance (*BaseNStar* = 1,050,000 males) and an assumed proportion of birds found in habitat (as opposed to matrix) across the species' range (*HabitatPreference.y* = 0.9) to estimate the proportion of land in the BTBW range that is habitat (as opposed to matrix), *RangeHabitatProportion.vector*, where

$$RangeHabitatProportion.vector = \frac{HabitatPreference.y \times BaseNStar}{HabitatDensity.y \times RangeArea} = 0.0489. \quad \text{(eq. C.14)}$$

From those parameters, I estimated the density of birds in matrix (*MatrixDensity.y*) as

$$MatrixDensity.y = \frac{BaseNStar \times (1 - HabitatPreference.y)}{RangeArea(1 - RangeHabitatProportion.vector)} = 0.00305 \text{ birds/ha.} \quad \text{(eq. C.15)}$$

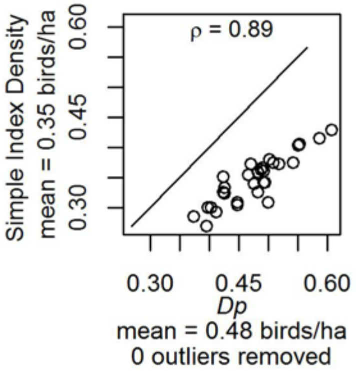
Appendix D

Density Estimators for Chapter 2

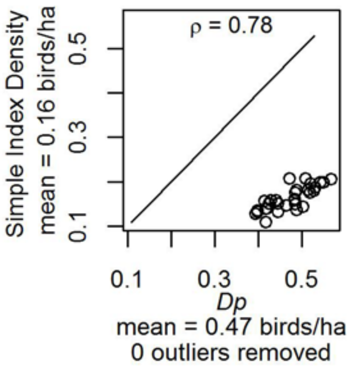
D1 Survey Type: Simple Counts Estimator: Simple Index Estimator Survey Radius 150 m

Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

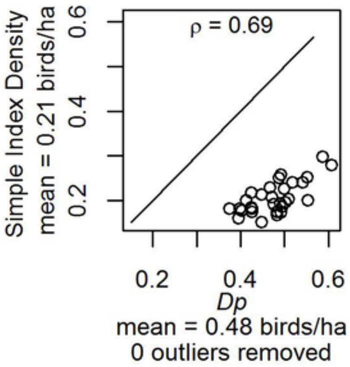
Scenario 1: high p_a high p_d habitat effect on p_d



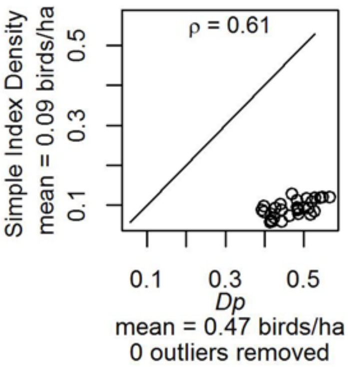
Scenario 2: low p_a high p_d habitat effect on p_d



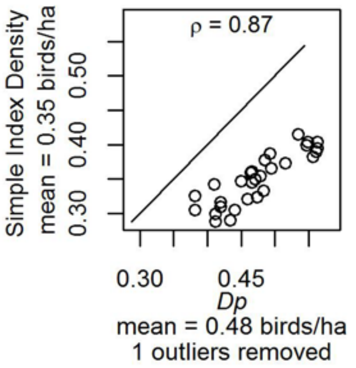
Scenario 3: high p_a low p_d habitat effect on p_d



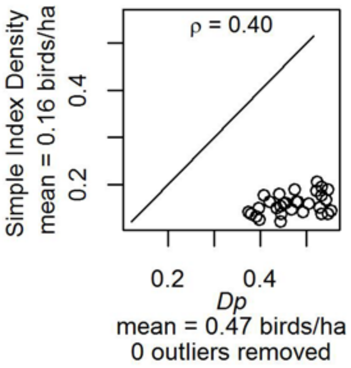
Scenario 4: low p_a low p_d habitat effect on p_d



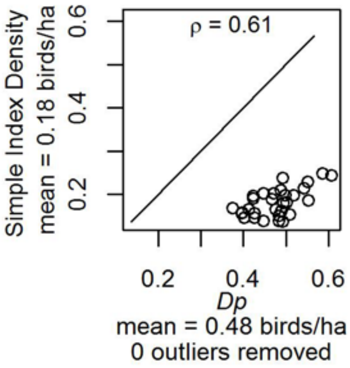
Scenario 5: high p_a high p_d no habitat effect on p_d



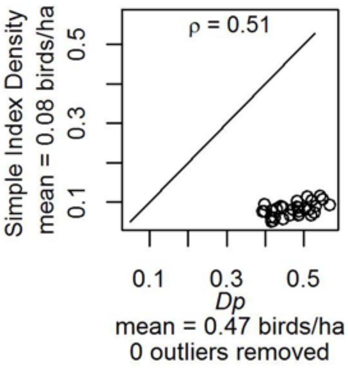
Scenario 6: low p_a high p_d no habitat effect on p_d



Scenario 7: high p_a low p_d no habitat effect on p_d



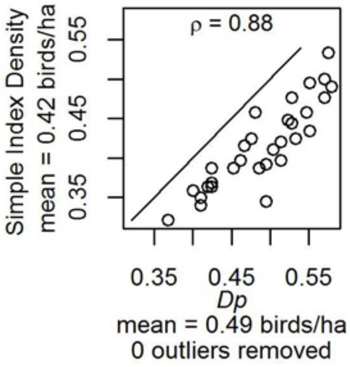
Scenario 8: low p_a low p_d no habitat effect on p_d



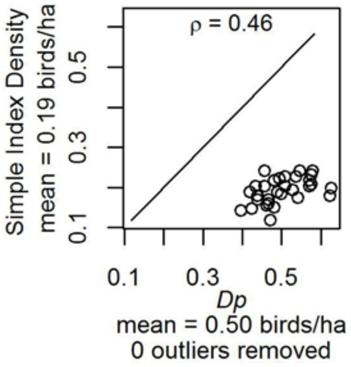
D2 Survey Type: Double-Observer Estimator: Simple Index Estimator Survey Radius 150 m

Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

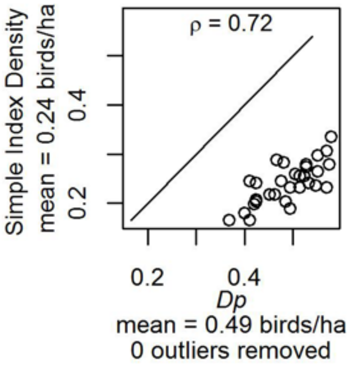
Scenario 1: high p_a high p_d
habitat effect on p_d



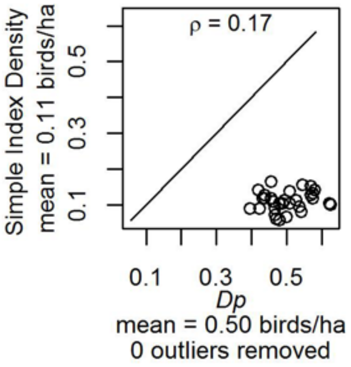
Scenario 2: low p_a high p_d
habitat effect on p_d



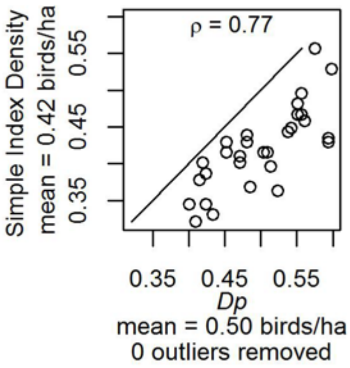
Scenario 3: high p_a low p_d
habitat effect on p_d



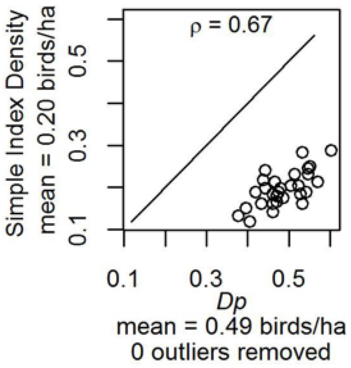
Scenario 4: low p_a low p_d
habitat effect on p_d



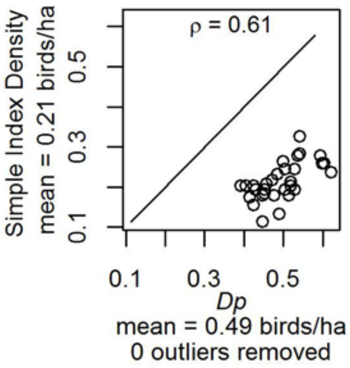
Scenario 5: high p_a high p_d
no habitat effect on p_d



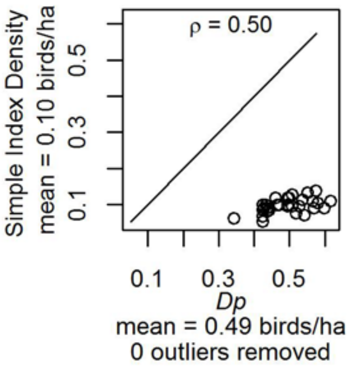
Scenario 6: low p_a high p_d
no habitat effect on p_d



Scenario 7: high p_a low p_d
no habitat effect on p_d



Scenario 8: low p_a low p_d
no habitat effect on p_d



D3 Survey Type: Replicated Counts Estimator: Simple Index Estimator Survey Radius 150 m

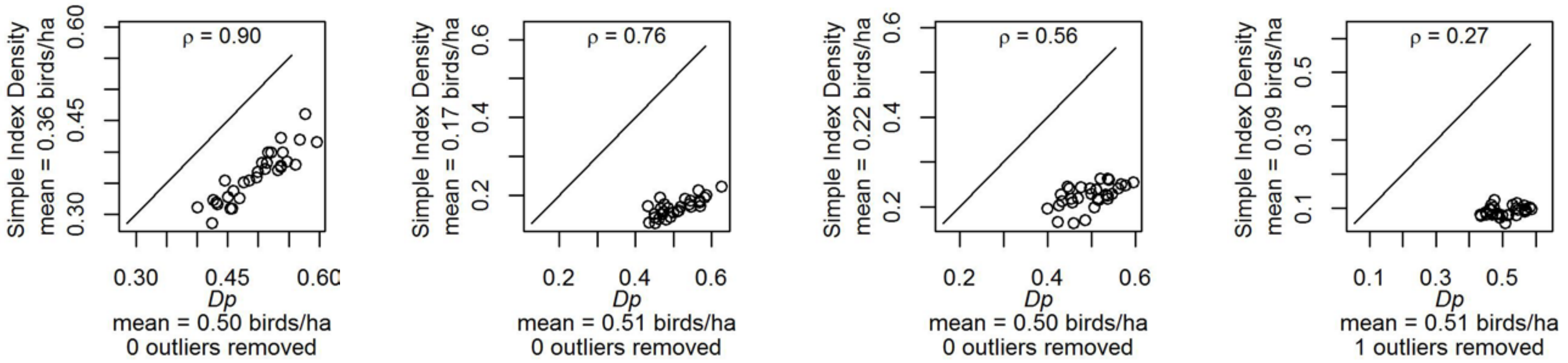
Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

Scenario 1: high p_a high p_d habitat effect on p_d

Scenario 2: low p_a high p_d habitat effect on p_d

Scenario 3: high p_a low p_d habitat effect on p_d

Scenario 4: low p_a low p_d habitat effect on p_d

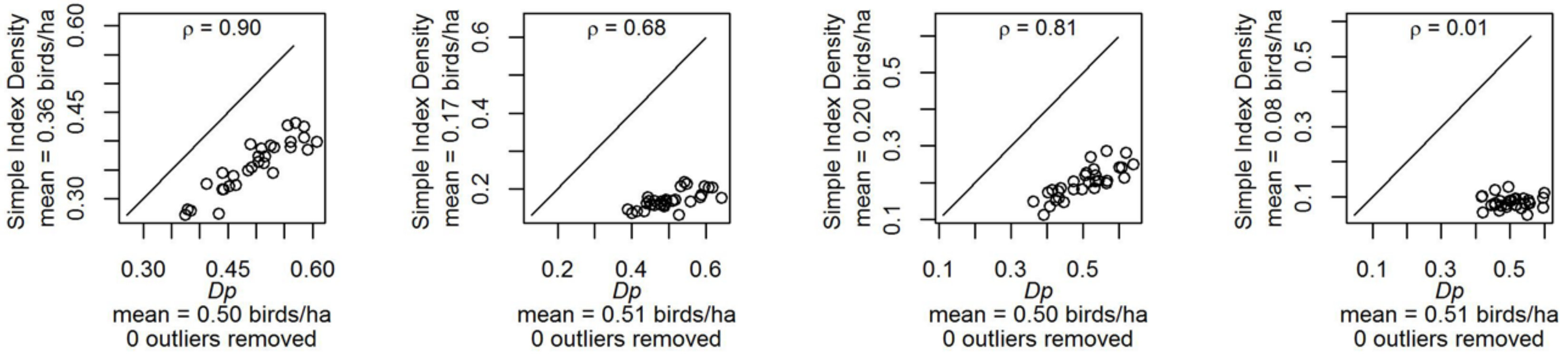


Scenario 5: high p_a high p_d no habitat effect on p_d

Scenario 6: low p_a high p_d no habitat effect on p_d

Scenario 7: high p_a low p_d no habitat effect on p_d

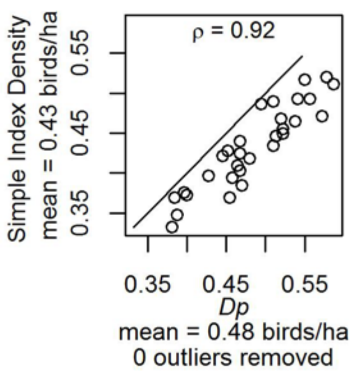
Scenario 8: low p_a low p_d no habitat effect on p_d



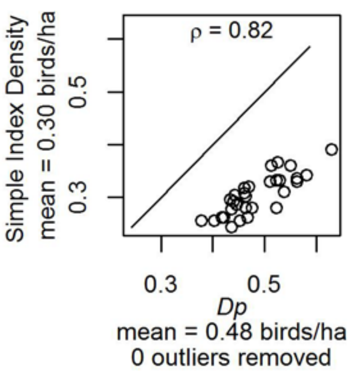
D4 Survey Type: Removal Estimator: Simple Index Estimator Survey Radius 150 m

Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

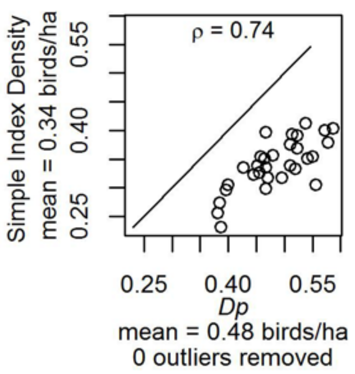
Scenario 1: high p_a high p_d
habitat effect on p_d



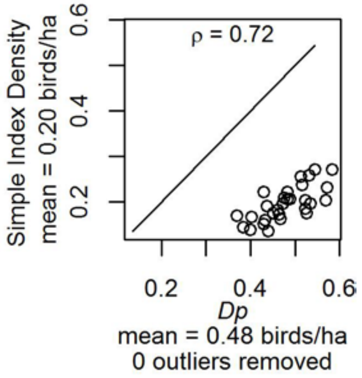
Scenario 2: low p_a high p_d
habitat effect on p_d



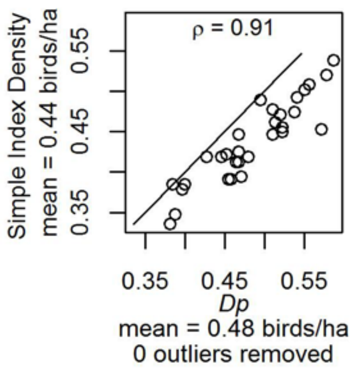
Scenario 3: high p_a low p_d
habitat effect on p_d



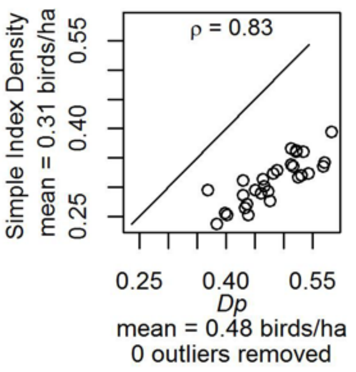
Scenario 4: low p_a low p_d
habitat effect on p_d



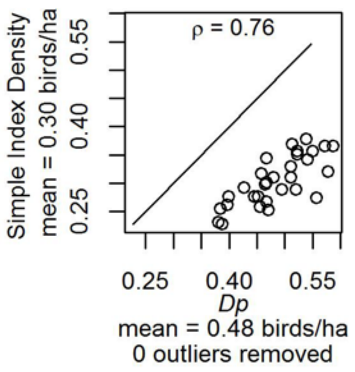
Scenario 5: high p_a high p_d
no habitat effect on p_d



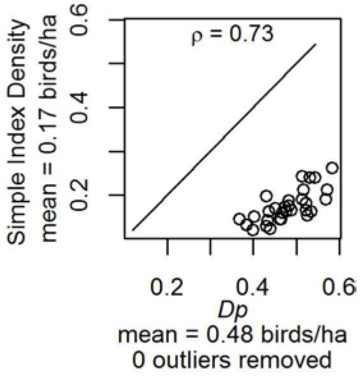
Scenario 6: low p_a high p_d
no habitat effect on p_d



Scenario 7: high p_a low p_d
no habitat effect on p_d



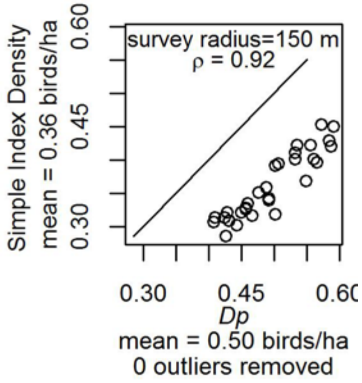
Scenario 8: low p_a low p_d
no habitat effect on p_d



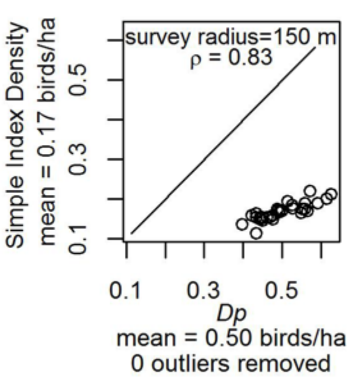
D5 Survey Type: Distance Sampling Estimator: Simple Index Estimator Survey Radius 150 m

Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

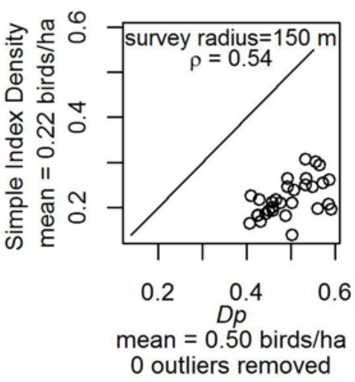
Scenario 1: high p_a high p_d
habitat effect on p_d



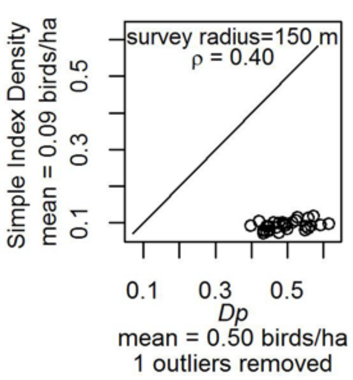
Scenario 2: low p_a high p_d
habitat effect on p_d



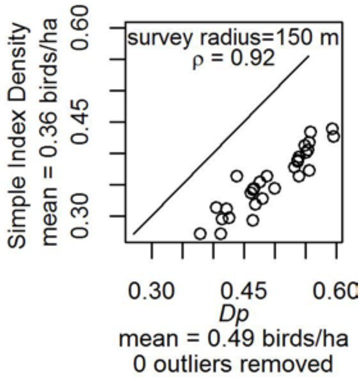
Scenario 3: high p_a low p_d
habitat effect on p_d



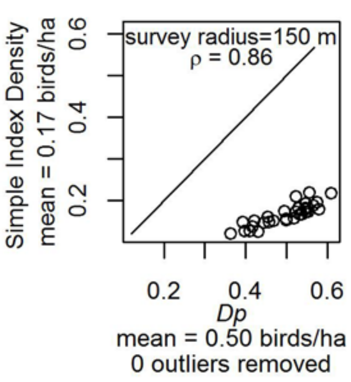
Scenario 4: low p_a low p_d
habitat effect on p_d



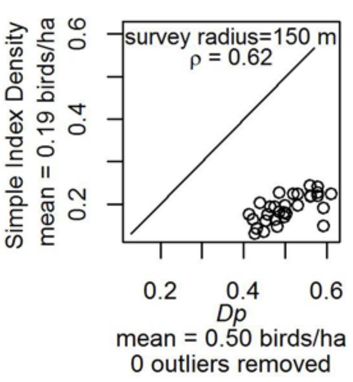
Scenario 5: high p_a high p_d
no habitat effect on p_d



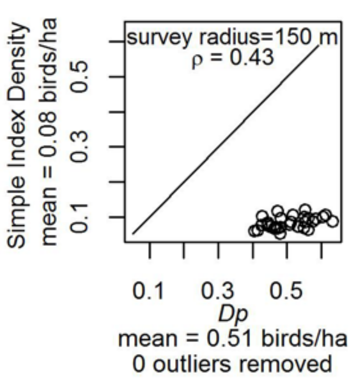
Scenario 6: low p_a high p_d
no habitat effect on p_d



Scenario 7: high p_a low p_d
no habitat effect on p_d



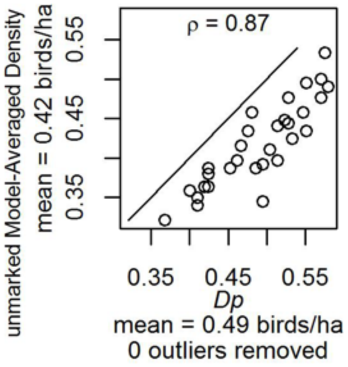
Scenario 8: low p_a low p_d
no habitat effect on p_d



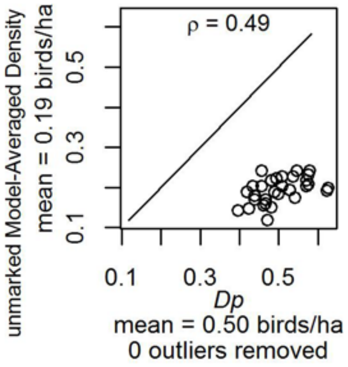
D6 Survey Type: Double-Observer Estimator: unmarked Model-Averaged Estimator Survey Radius 150 m

Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

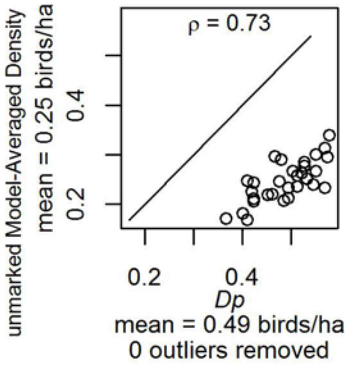
Scenario 1: high p_a high p_d
habitat effect on p_d



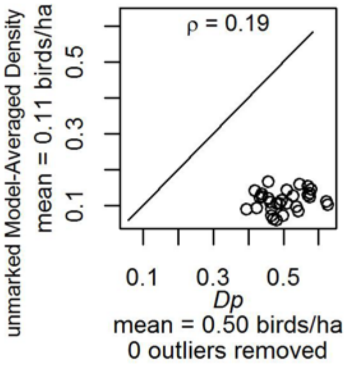
Scenario 2: low p_a high p_d
habitat effect on p_d



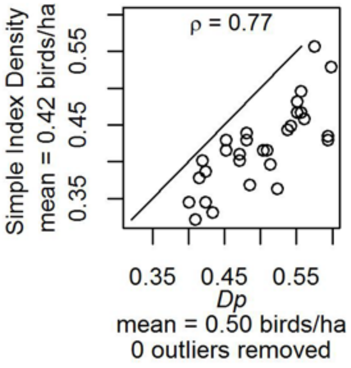
Scenario 3: high p_a low p_d
habitat effect on p_d



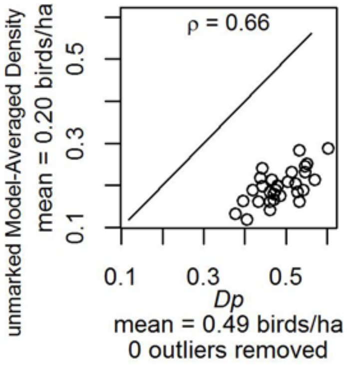
Scenario 4: low p_a low p_d
habitat effect on p_d



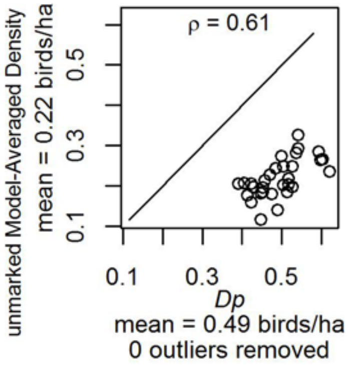
Scenario 5: high p_a high p_d
no habitat effect on p_d



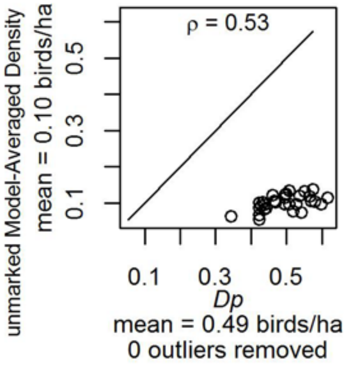
Scenario 6: low p_a high p_d
no habitat effect on p_d



Scenario 7: high p_a low p_d
no habitat effect on p_d

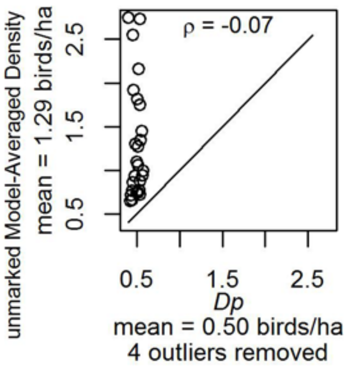


Scenario 8: low p_a low p_d
no habitat effect on p_d

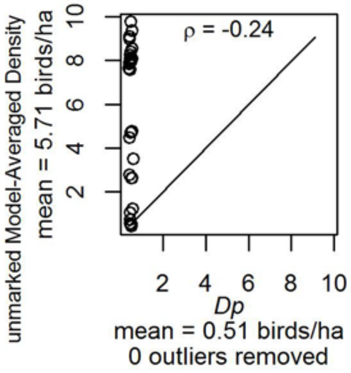


Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

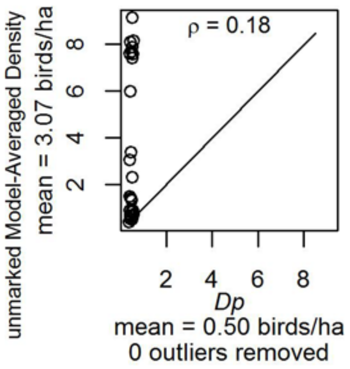
Scenario 1: high p_a high p_d
habitat effect on p_d



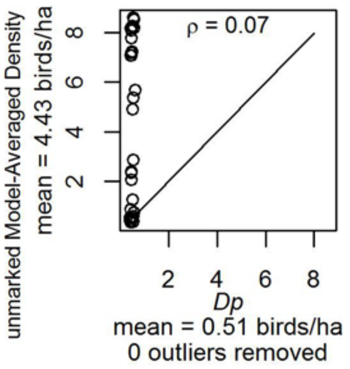
Scenario 2: low p_a high p_d
habitat effect on p_d



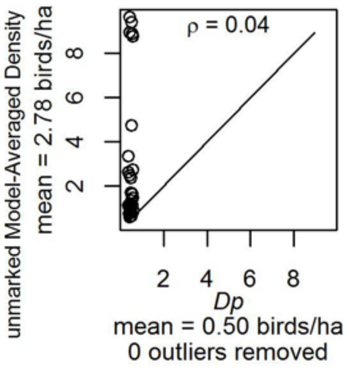
Scenario 3: high p_a low p_d
habitat effect on p_d



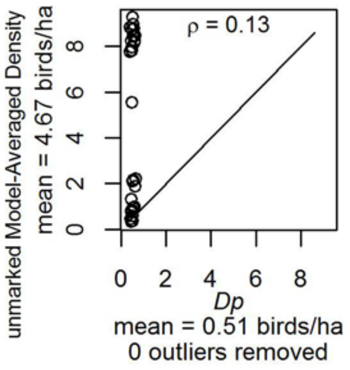
Scenario 4: low p_a low p_d
habitat effect on p_d



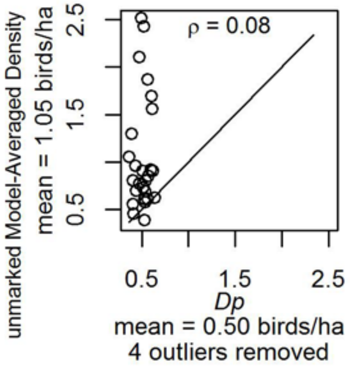
Scenario 5: high p_a high p_d
no habitat effect on p_d



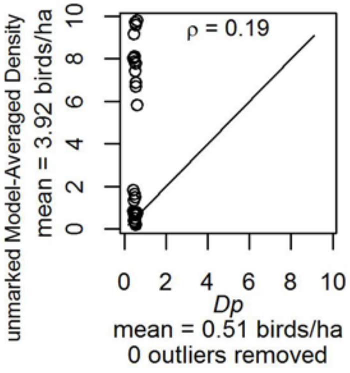
Scenario 6: low p_a high p_d
no habitat effect on p_d



Scenario 7: high p_a low p_d
no habitat effect on p_d



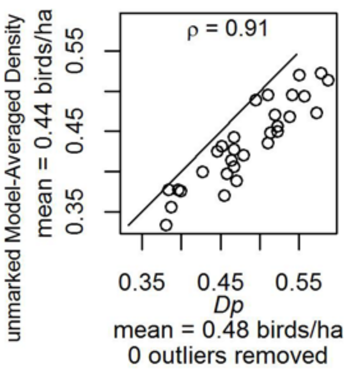
Scenario 8: low p_a low p_d
no habitat effect on p_d



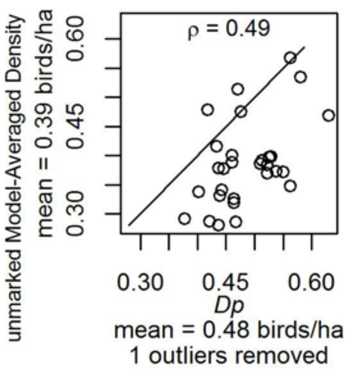
D8 Survey Type: Removal Estimator: unmarked Model-Averaged Estimator Survey Radius 150 m

Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

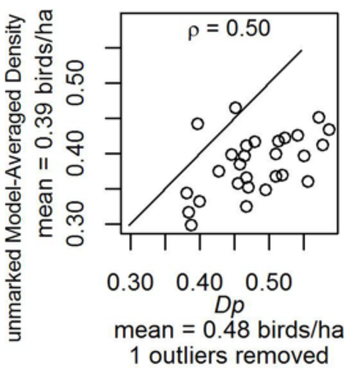
Scenario 1: high p_a high p_d
habitat effect on p_d



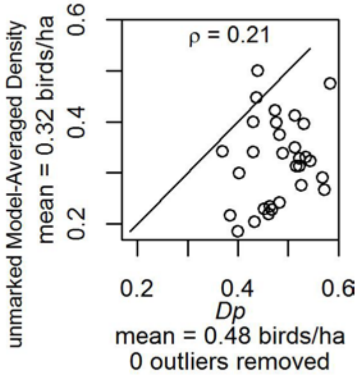
Scenario 2: low p_a high p_d
habitat effect on p_d



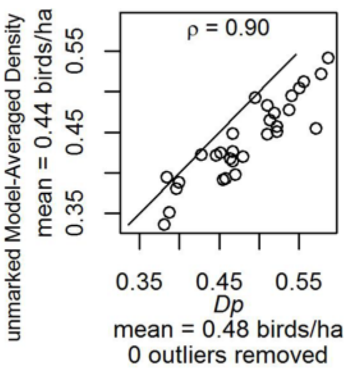
Scenario 3: high p_a low p_d
habitat effect on p_d



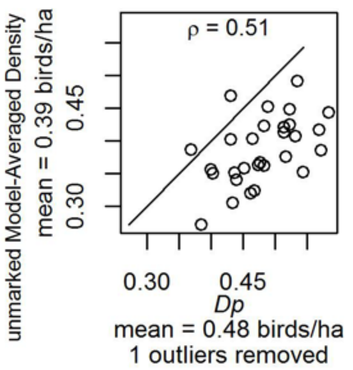
Scenario 4: low p_a low p_d
habitat effect on p_d



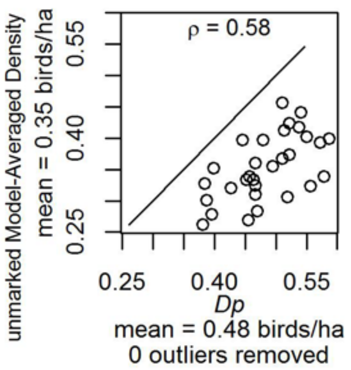
Scenario 5: high p_a high p_d
no habitat effect on p_d



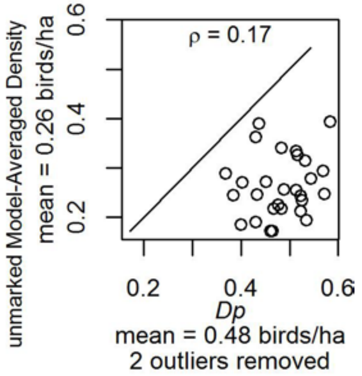
Scenario 6: low p_a high p_d
no habitat effect on p_d



Scenario 7: high p_a low p_d
no habitat effect on p_d



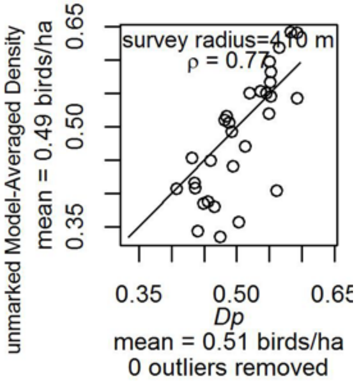
Scenario 8: low p_a low p_d
no habitat effect on p_d



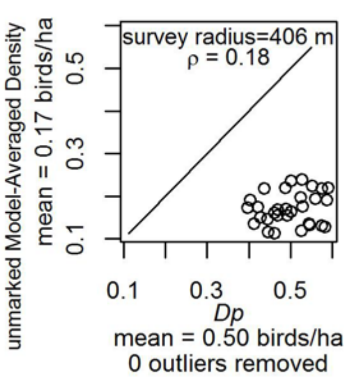
D9 Survey Type: Distance Sampling Estimator: unmarked Model-Averaged Estimator Survey Radius Determined by 10% Truncation

Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

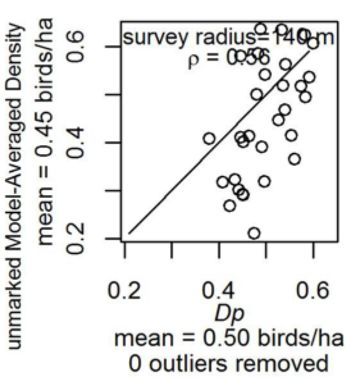
Scenario 1: high p_a high p_d habitat effect on p_d



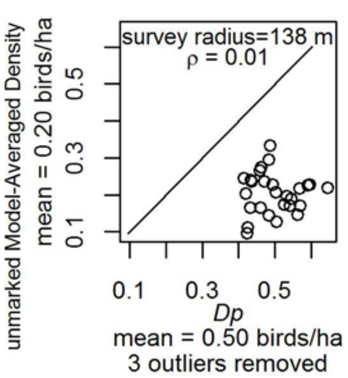
Scenario 2: low p_a high p_d habitat effect on p_d



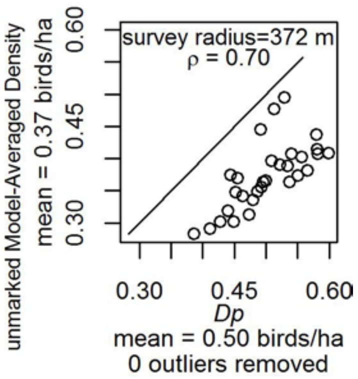
Scenario 3: high p_a low p_d habitat effect on p_d



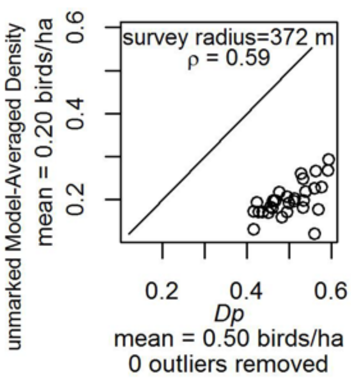
Scenario 4: low p_a low p_d habitat effect on p_d



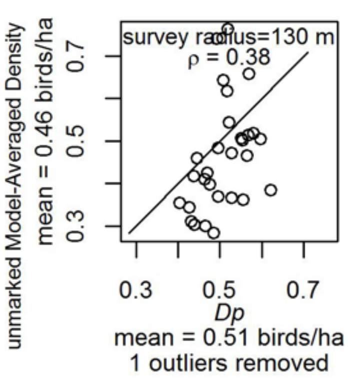
Scenario 5: high p_a high p_d no habitat effect on p_d



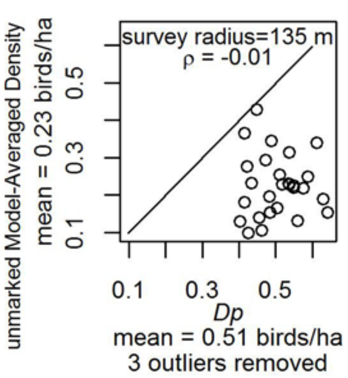
Scenario 6: low p_a high p_d no habitat effect on p_d



Scenario 7: high p_a low p_d no habitat effect on p_d



Scenario 8: low p_a low p_d no habitat effect on p_d



D10 Survey Type: Double-Observer Estimator: unmarked Top Model Estimator Survey Radius 150 m

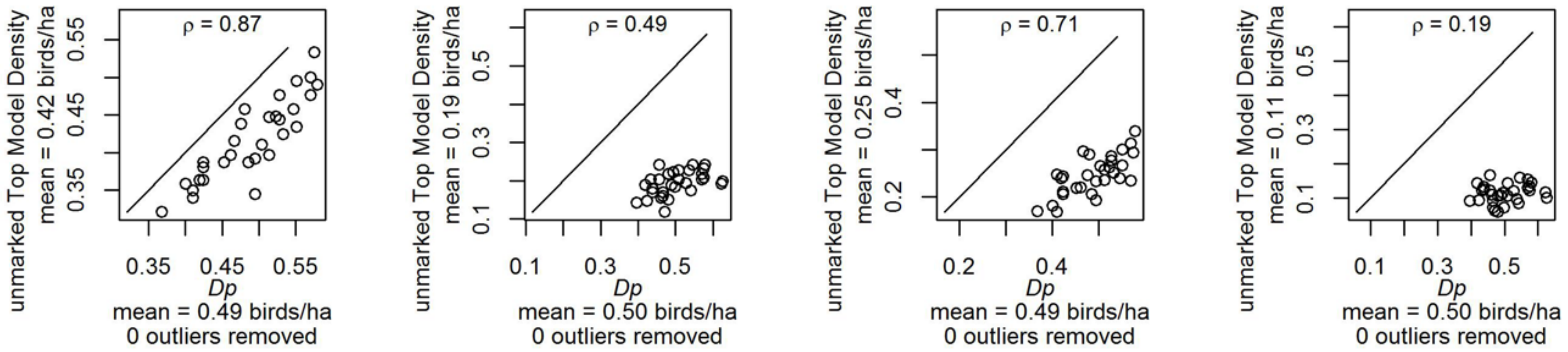
Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

Scenario 1: high p_a high p_d habitat effect on p_d

Scenario 2: low p_a high p_d habitat effect on p_d

Scenario 3: high p_a low p_d habitat effect on p_d

Scenario 4: low p_a low p_d habitat effect on p_d

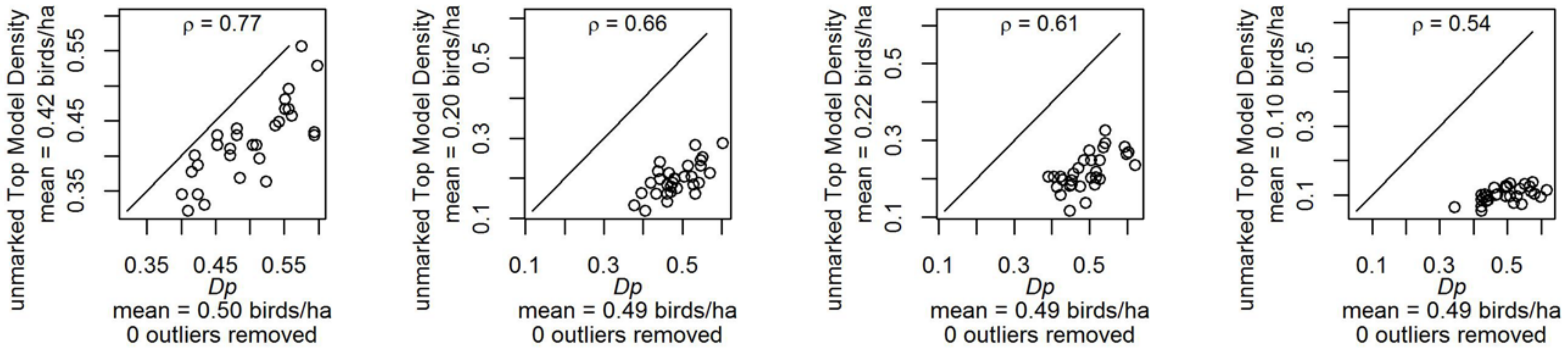


Scenario 5: high p_a high p_d no habitat effect on p_d

Scenario 6: low p_a high p_d no habitat effect on p_d

Scenario 7: high p_a low p_d no habitat effect on p_d

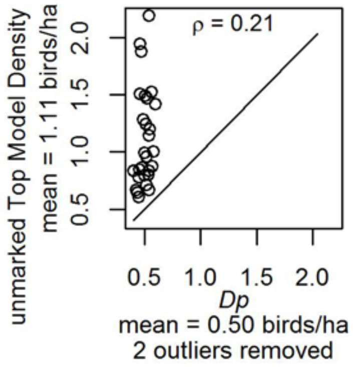
Scenario 8: low p_a low p_d no habitat effect on p_d



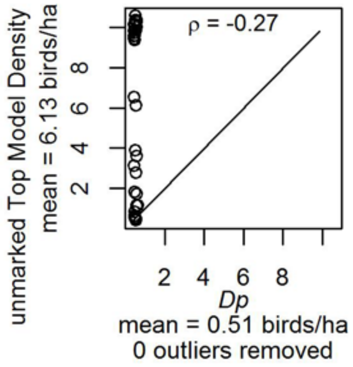
D11 Survey Type: Replicated Counts Estimator: unmarked Top Model Estimator Survey Radius 150 m

Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

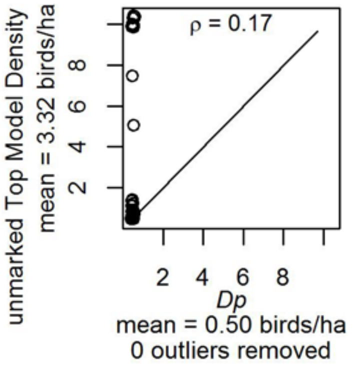
Scenario 1: high p_a high p_d
habitat effect on p_d



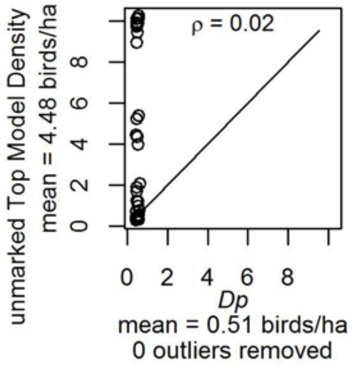
Scenario 2: low p_a high p_d
habitat effect on p_d



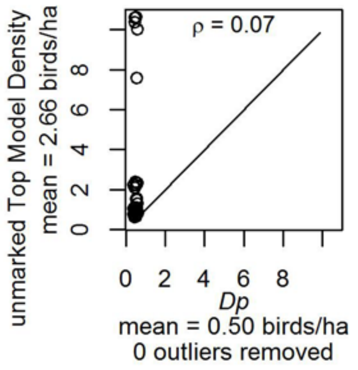
Scenario 3: high p_a low p_d
habitat effect on p_d



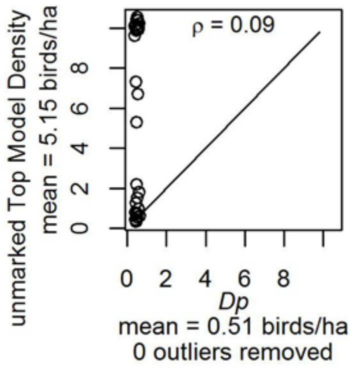
Scenario 4: low p_a low p_d
habitat effect on p_d



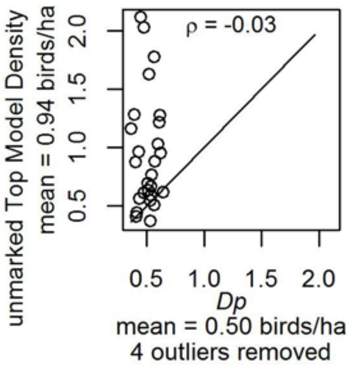
Scenario 5: high p_a high p_d
no habitat effect on p_d



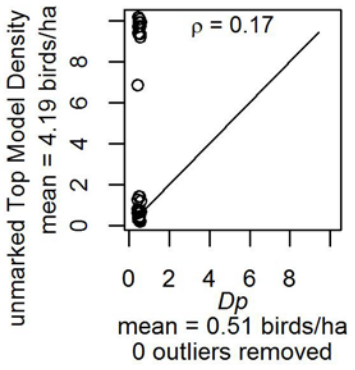
Scenario 6: low p_a high p_d
no habitat effect on p_d



Scenario 7: high p_a low p_d
no habitat effect on p_d



Scenario 8: low p_a low p_d
no habitat effect on p_d



D12 Survey Type: Removal Estimator: unmarked Top Model Estimator Survey Radius 150 m

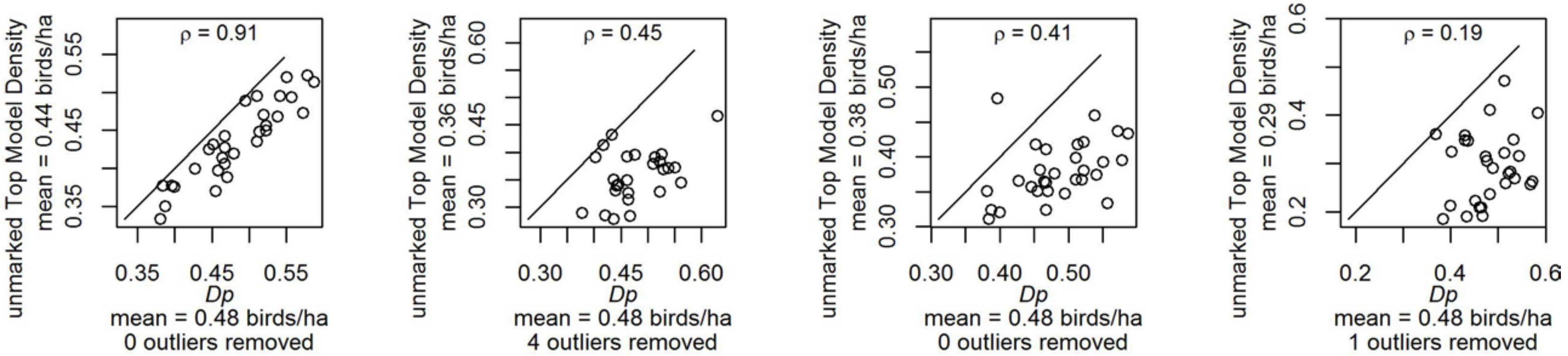
Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

Scenario 1: high p_a high p_d habitat effect on p_d

Scenario 2: low p_a high p_d habitat effect on p_d

Scenario 3: high p_a low p_d habitat effect on p_d

Scenario 4: low p_a low p_d habitat effect on p_d

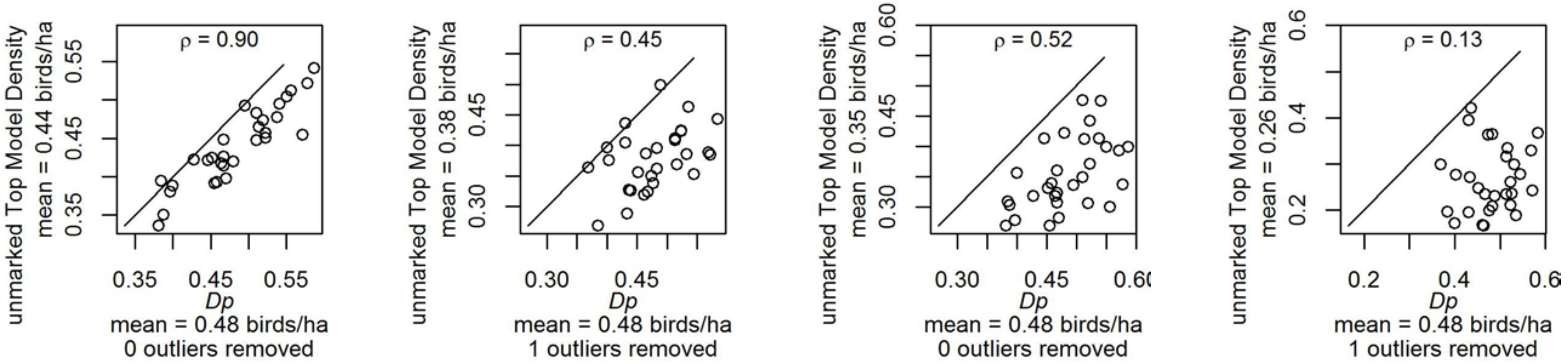


Scenario 5: high p_a high p_d no habitat effect on p_d

Scenario 6: low p_a high p_d no habitat effect on p_d

Scenario 7: high p_a low p_d no habitat effect on p_d

Scenario 8: low p_a low p_d no habitat effect on p_d



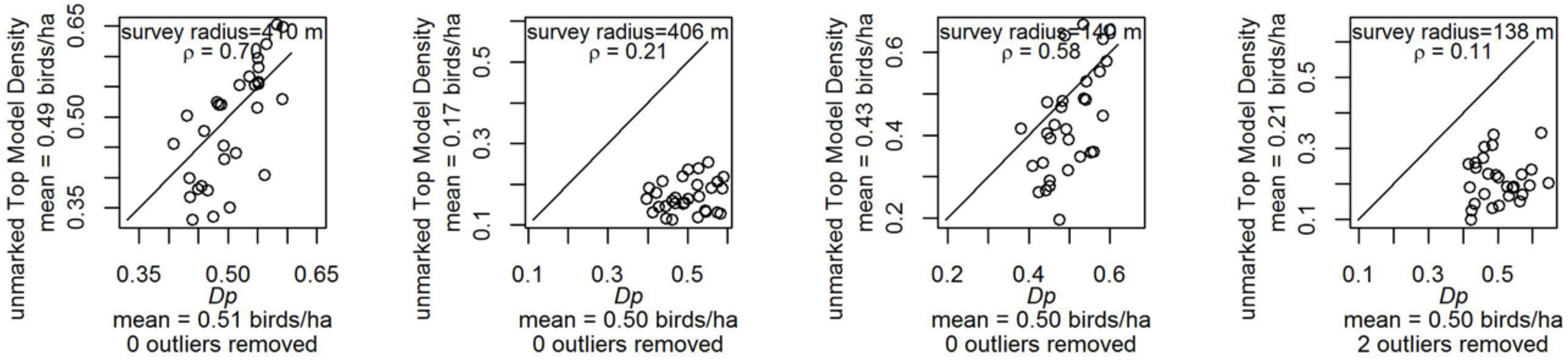
Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

Scenario 1: high p_a high p_d habitat effect on p_d

Scenario 2: low p_a high p_d habitat effect on p_d

Scenario 3: high p_a low p_d habitat effect on p_d

Scenario 4: low p_a low p_d habitat effect on p_d

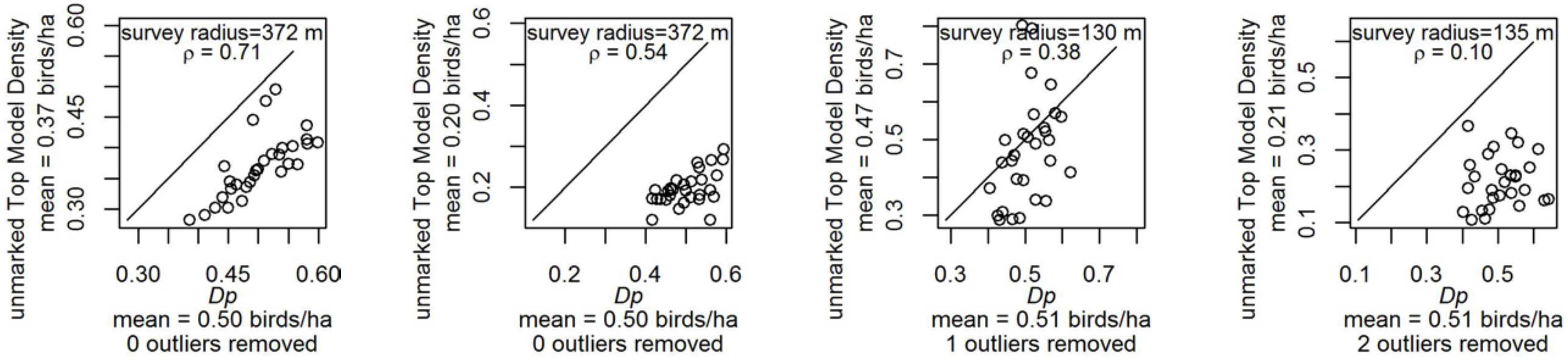


Scenario 5: high p_a high p_d no habitat effect on p_d

Scenario 6: low p_a high p_d no habitat effect on p_d

Scenario 7: high p_a low p_d no habitat effect on p_d

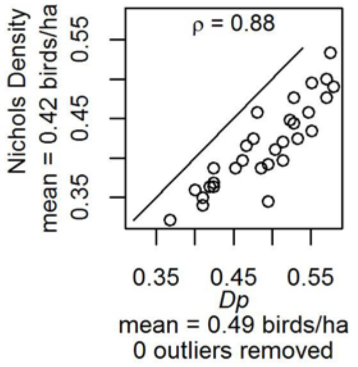
Scenario 8: low p_a low p_d no habitat effect on p_d



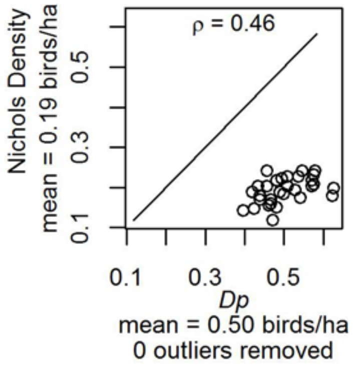
D14 Survey Type: Double-Observer Estimator: Nichols et al. (2000) Estimator Survey Radius 150 m

Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

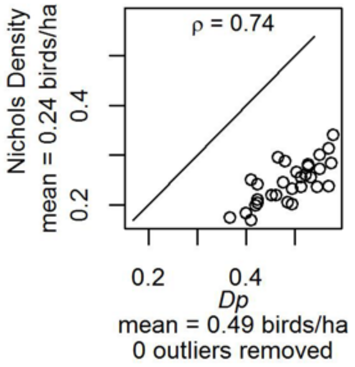
Scenario 1: high p_a high p_d
habitat effect on p_d



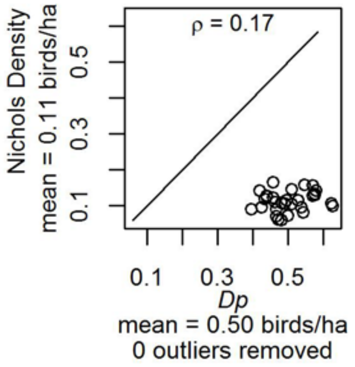
Scenario 2: low p_a high p_d
habitat effect on p_d



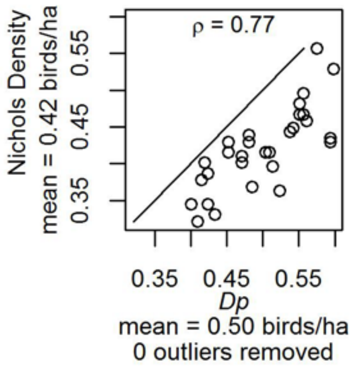
Scenario 3: high p_a low p_d
habitat effect on p_d



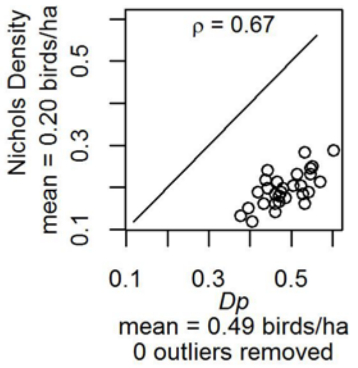
Scenario 4: low p_a low p_d
habitat effect on p_d



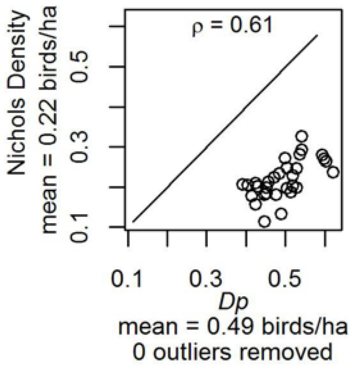
Scenario 5: high p_a high p_d
no habitat effect on p_d



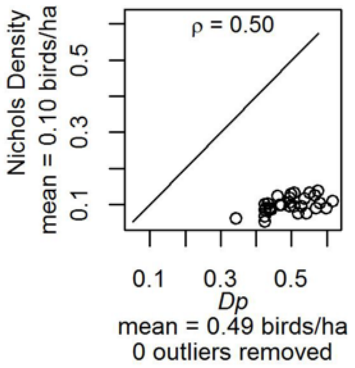
Scenario 6: low p_a high p_d
no habitat effect on p_d



Scenario 7: high p_a low p_d
no habitat effect on p_d



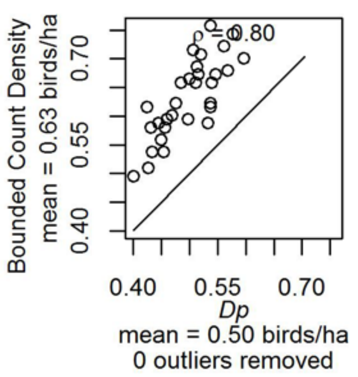
Scenario 8: low p_a low p_d
no habitat effect on p_d



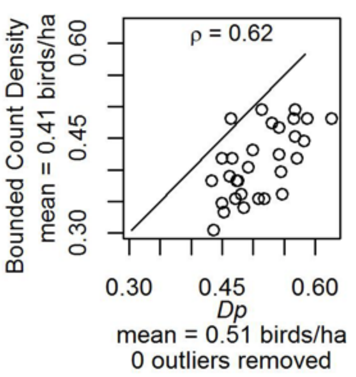
D15 Survey Type: Replicated Counts Estimator: Bounded Count Survey Radius 150 m

Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

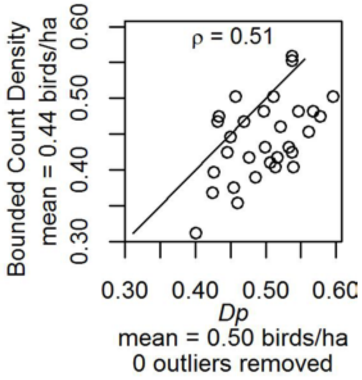
Scenario 1: high p_a high p_d habitat effect on p_d



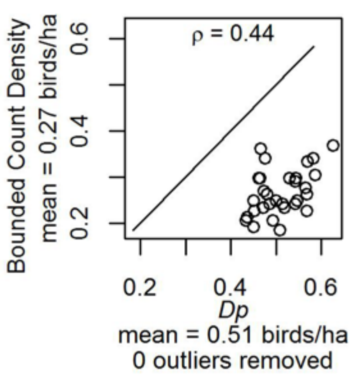
Scenario 2: low p_a high p_d habitat effect on p_d



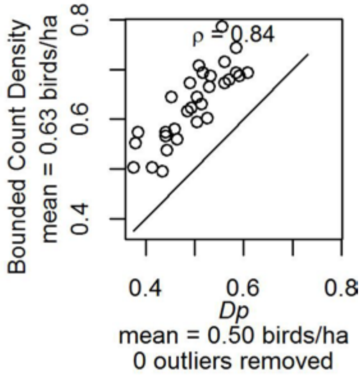
Scenario 3: high p_a low p_d habitat effect on p_d



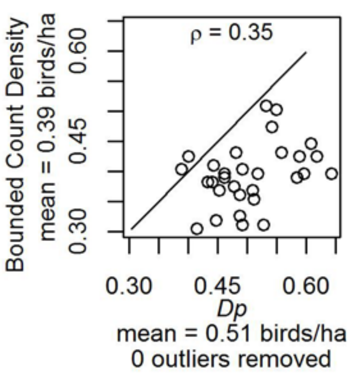
Scenario 4: low p_a low p_d habitat effect on p_d



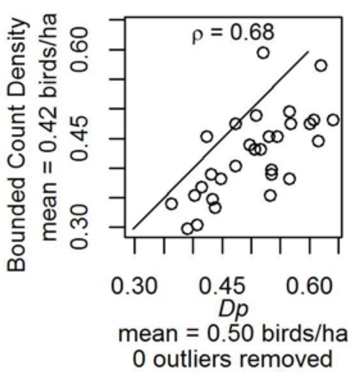
Scenario 5: high p_a high p_d no habitat effect on p_d



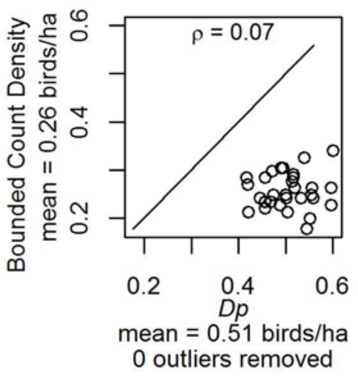
Scenario 6: low p_a high p_d no habitat effect on p_d



Scenario 7: high p_a low p_d no habitat effect on p_d



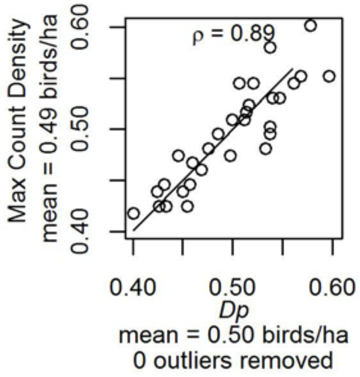
Scenario 8: low p_a low p_d no habitat effect on p_d



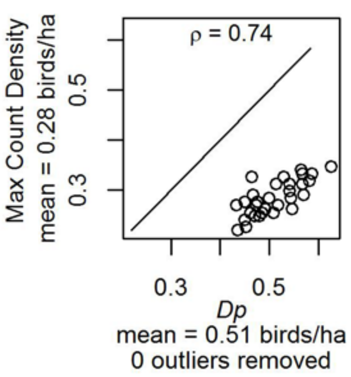
D16 Survey Type: Replicated Counts Estimator: Maximum Count Survey Radius 150 m

Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

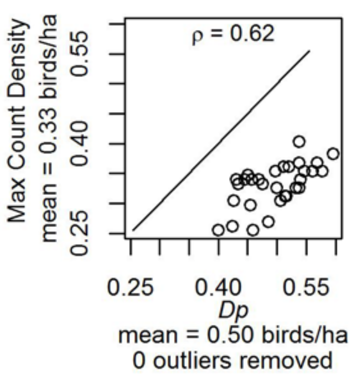
Scenario 1: high p_a high p_d habitat effect on p_d



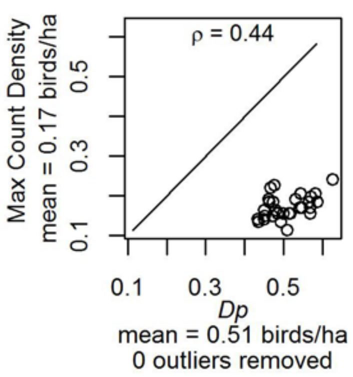
Scenario 2: low p_a high p_d habitat effect on p_d



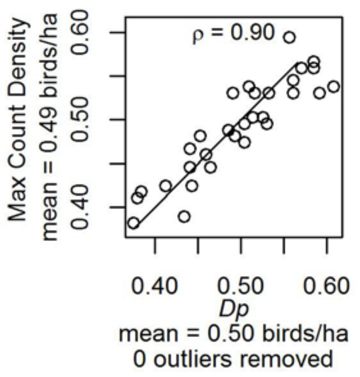
Scenario 3: high p_a low p_d habitat effect on p_d



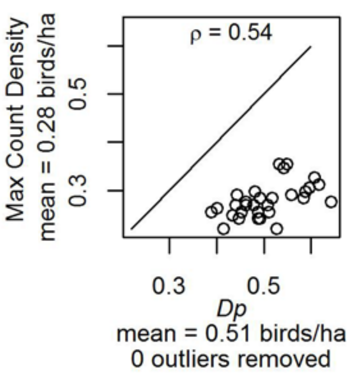
Scenario 4: low p_a low p_d habitat effect on p_d



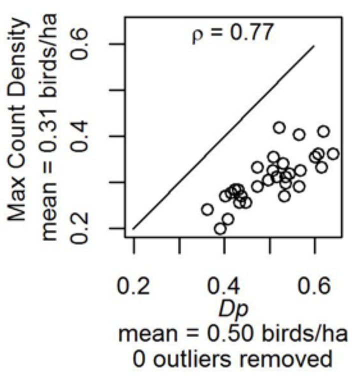
Scenario 5: high p_a high p_d no habitat effect on p_d



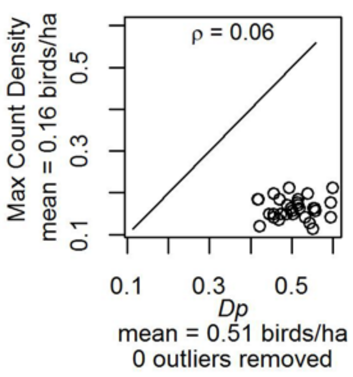
Scenario 6: low p_a high p_d no habitat effect on p_d



Scenario 7: high p_a low p_d no habitat effect on p_d



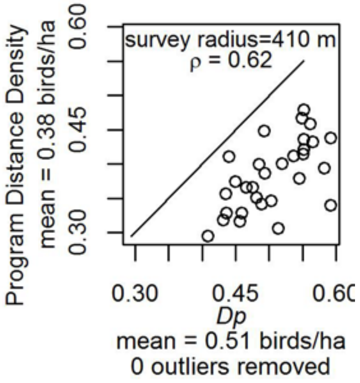
Scenario 8: low p_a low p_d no habitat effect on p_d



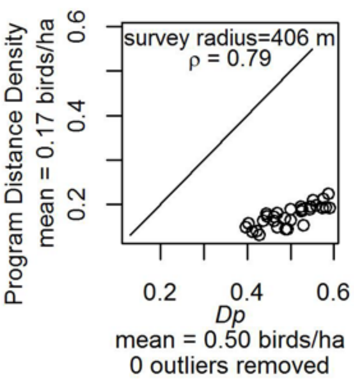
D17 Survey Type: Distance Sampling Estimator: Program Distance Top Model Estimator Survey Radius 150 m

Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

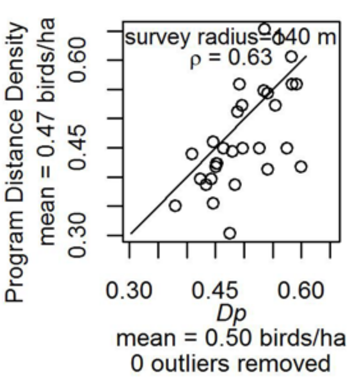
Scenario 1: high p_a high p_d
habitat effect on p_d



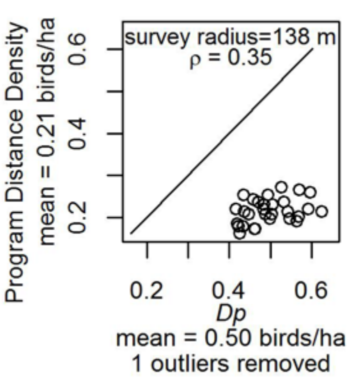
Scenario 2: low p_a high p_d
habitat effect on p_d



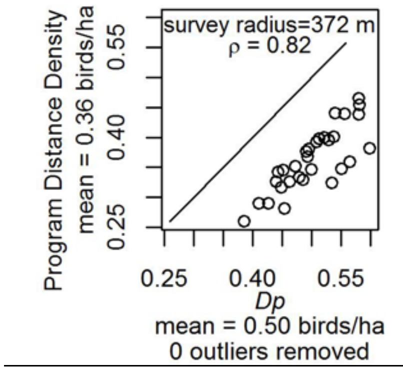
Scenario 3: high p_a low p_d
habitat effect on p_d



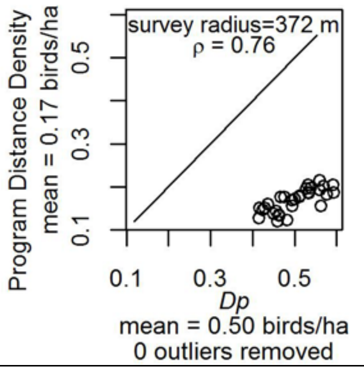
Scenario 4: low p_a low p_d
habitat effect on p_d



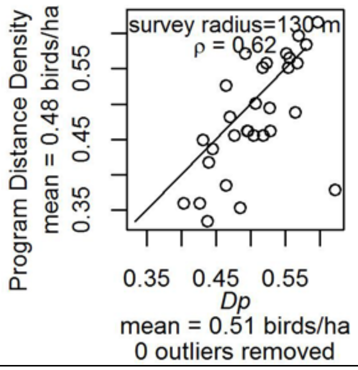
Scenario 5: high p_a high p_d
no habitat effect on p_d



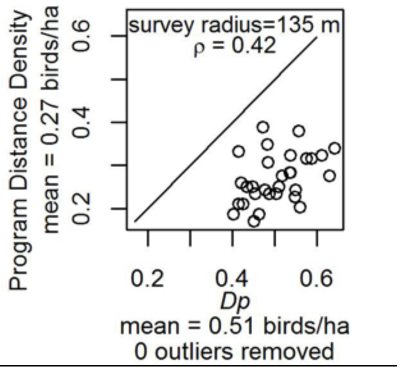
Scenario 6: low p_a high p_d
no habitat effect on p_d



Scenario 7: high p_a low p_d
no habitat effect on p_d



Scenario 8: low p_a low p_d
no habitat effect on p_d



D18 Survey Type: Distance Sampling Estimator: Program Distance Model-Averaged Estimator Survey Radius 150 m

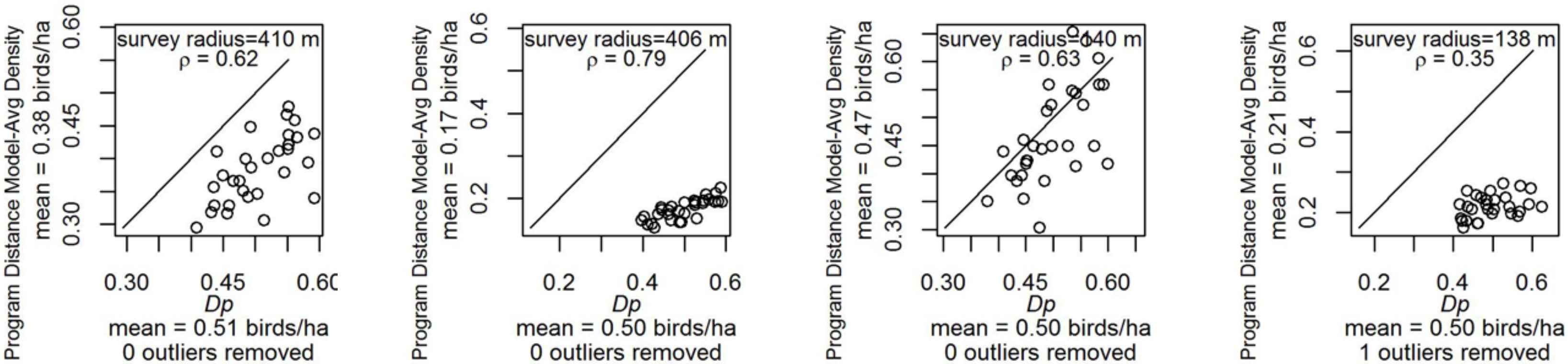
Density estimator performance, as compared to true density of birds present at survey sites (D_p). Eight scenarios (Table 1) were parameterized to reflect three binary options: high vs. low p_a (Figure 2), high vs. low p_d (Figure 3), and the presence or absence of an effect of habitat on p_d. Data points for perfect estimators would fall on the 45° line (shown for reference). Estimator performance was assessed based on correlation coefficients (ρ) and bias (estimated density – D_p).

Scenario 1: high p_a high p_d habitat effect on p_d

Scenario 2: low p_a high p_d habitat effect on p_d

Scenario 3: high p_a low p_d habitat effect on p_d

Scenario 4: low p_a low p_d habitat effect on p_d

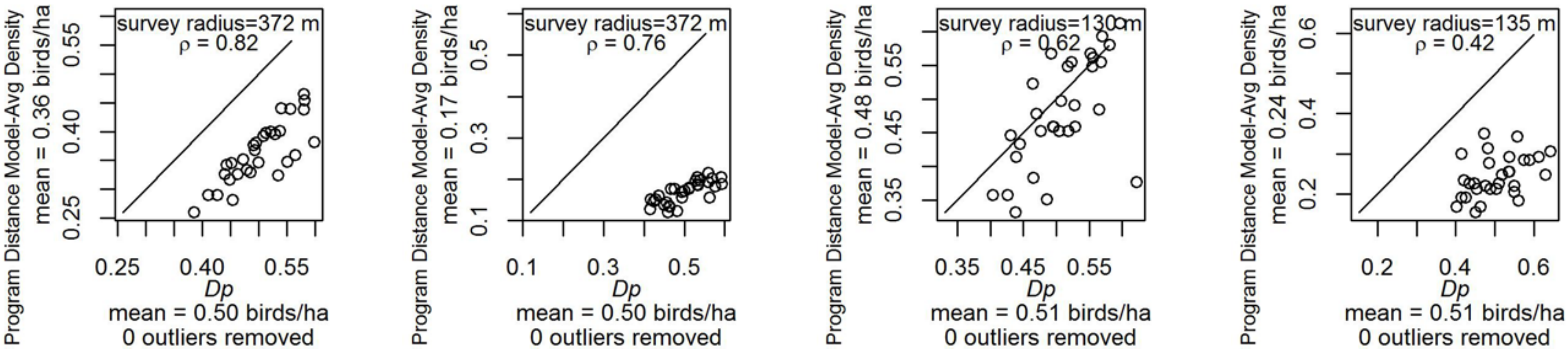


Scenario 5: high p_a high p_d no habitat effect on p_d

Scenario 6: low p_a high p_d no habitat effect on p_d

Scenario 7: high p_a low p_d no habitat effect on p_d

Scenario 8: low p_a low p_d no habitat effect on p_d



Appendix E

Predicted Probability of Detection for 10 Grassland Bird Species

Figure E1: The predicted perceptible area around the observer for BOBO. The solid black border indicates the limit inside which predicted probability of detection ≥ 0.5. Wind was modeled as coming from due north. The position of the observer at (0,0) is marked with a white +. Distances are in meters.

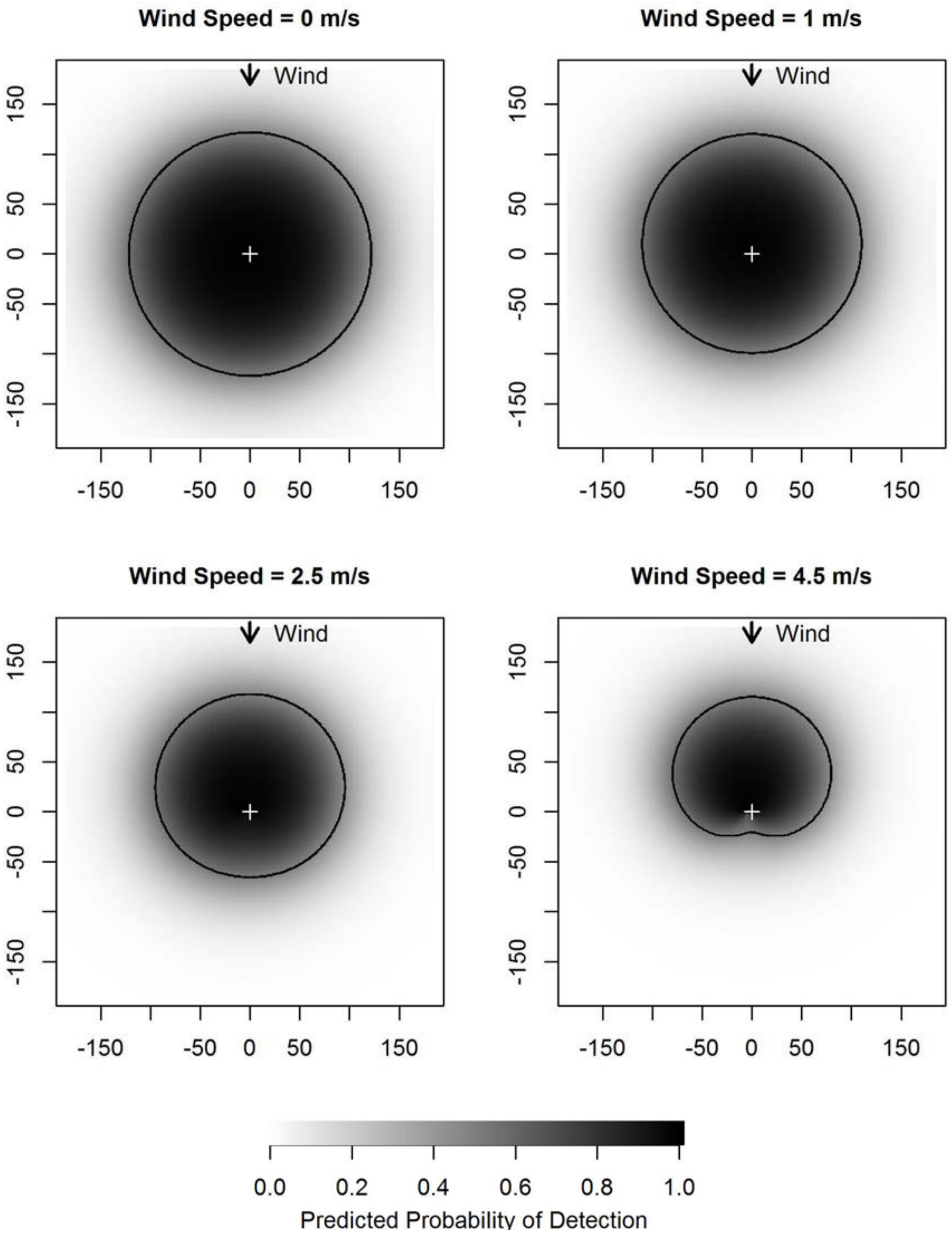


Figure E2: The predicted perceptible area around the observer for DICK. The solid black border indicates the limit inside which predicted probability of detection ≥ 0.5. Wind was modeled as coming from due north. The position of the observer at (0,0) is marked with a white +. Distances are in meters.

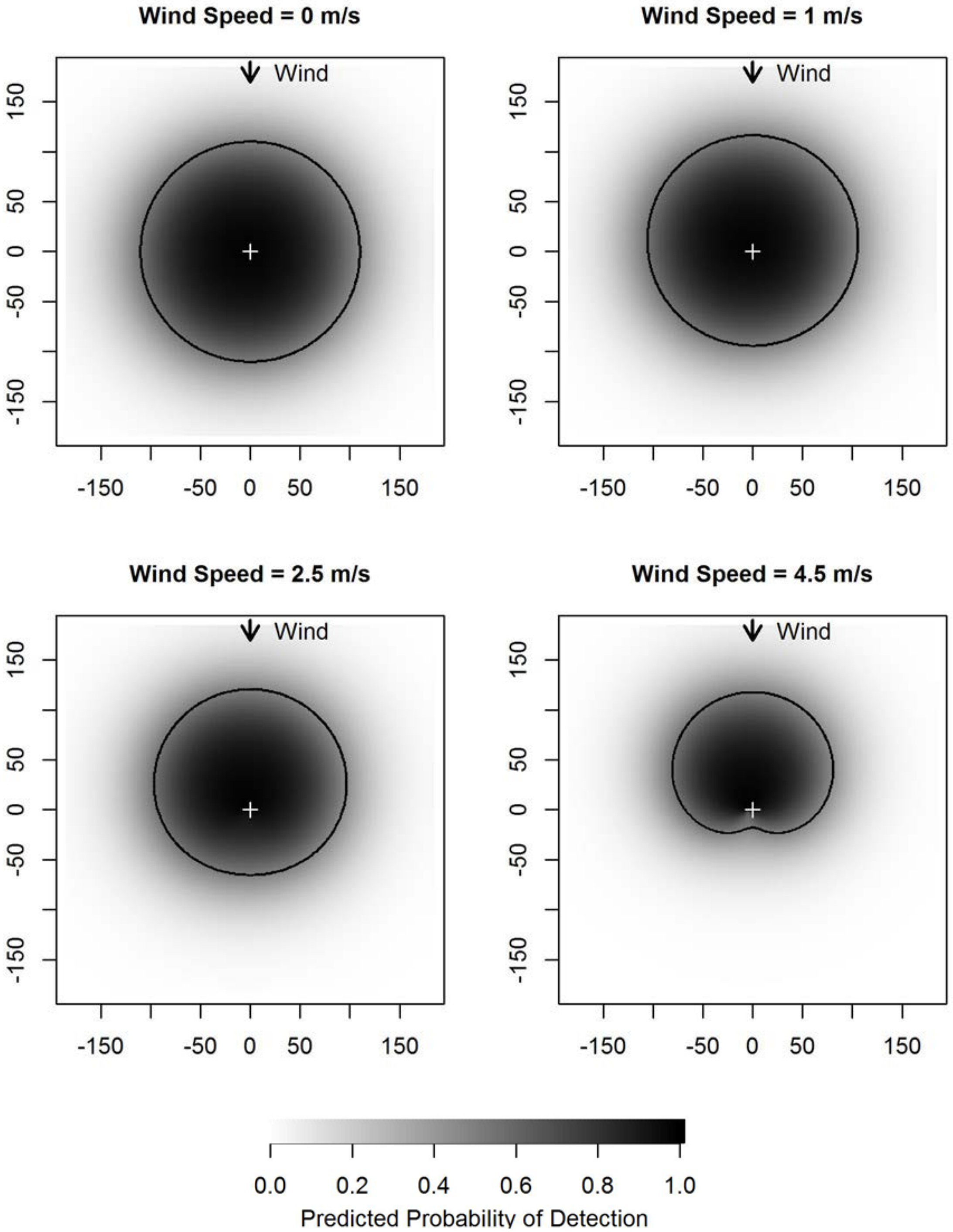


Figure E3: The predicted perceptible area around the observer for EAME. The solid black border indicates the limit inside which predicted probability of detection ≥ 0.5. Wind was modeled as coming from due north. The position of the observer at (0,0) is marked with a white +. Distances are in meters.

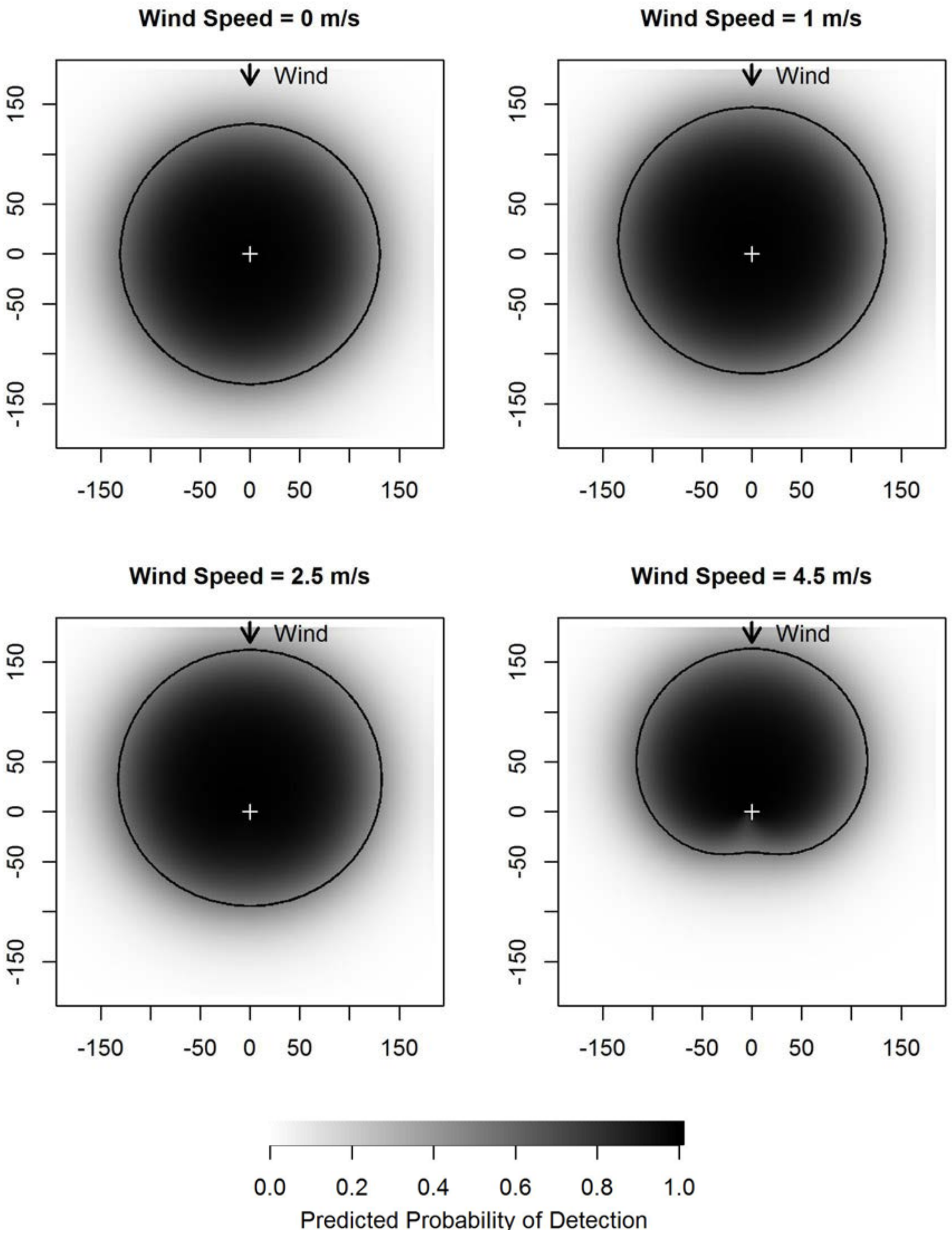


Figure E4: The predicted perceptible area around the observer for GRSP. The solid black border indicates the limit inside which predicted probability of detection ≥ 0.5. Wind was modeled as coming from due north. The position of the observer at (0,0) is marked with a white +. Distances are in meters.

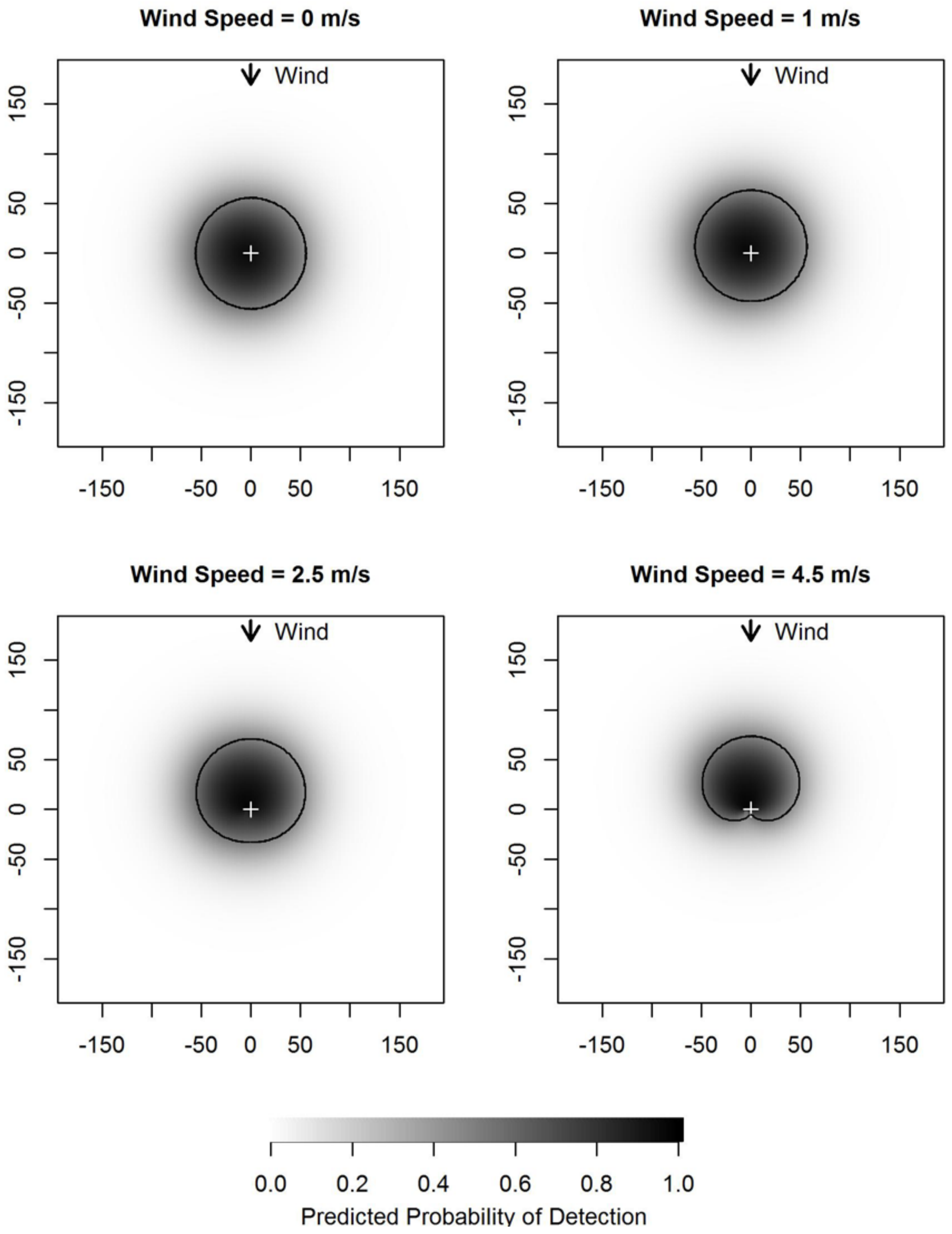


Figure E5: The predicted perceptible area around the observer for HESP. The solid black border indicates the limit inside which predicted probability of detection ≥ 0.5. Wind was modeled as coming from due north. The position of the observer at (0,0) is marked with a white +. Distances are in meters.

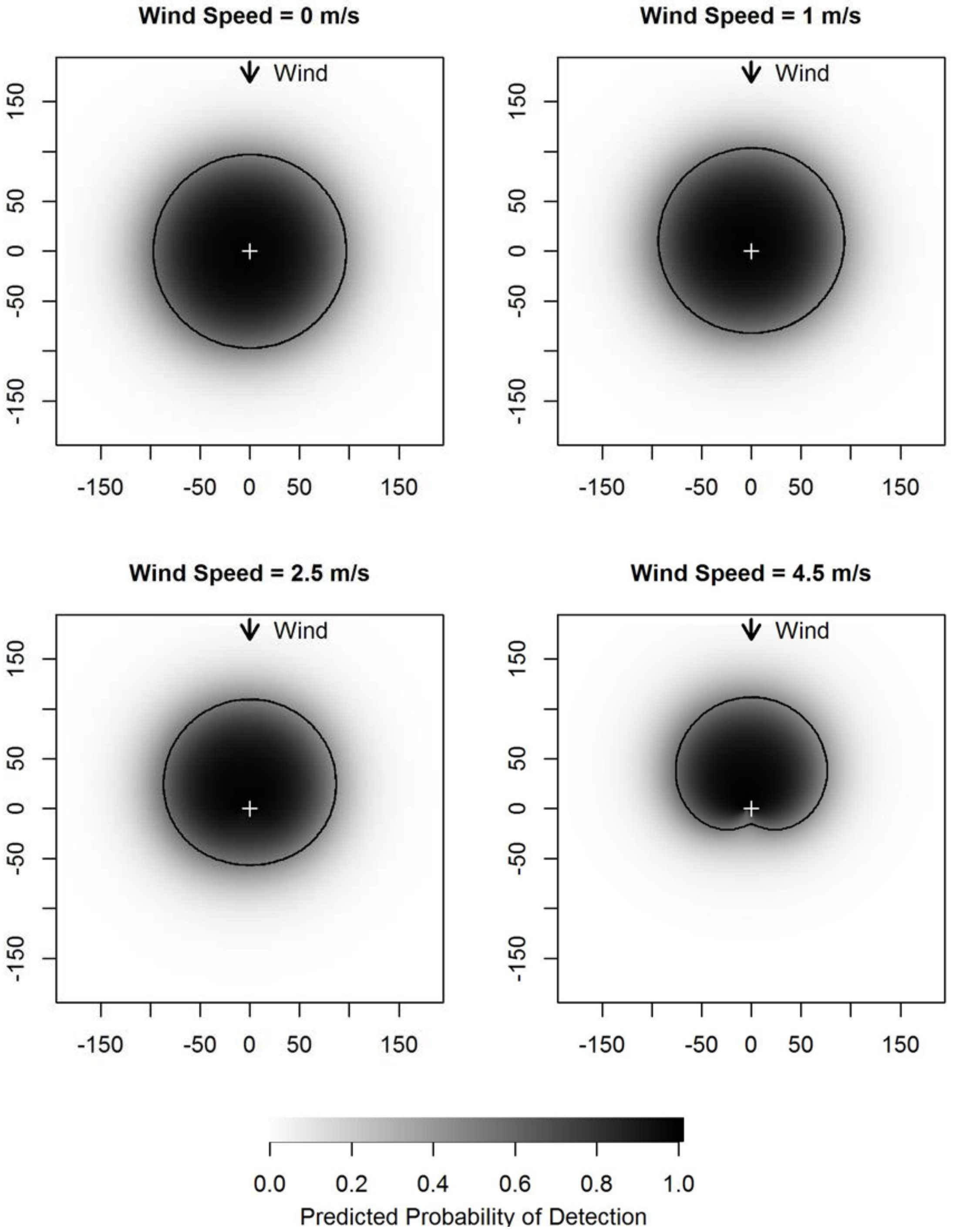


Figure E6: The predicted perceptible area around the observer for HOLA. The solid black border indicates the limit inside which predicted probability of detection ≥ 0.5. Wind was modeled as coming from due north. The position of the observer at (0,0) is marked with a white +. Distances are in meters.

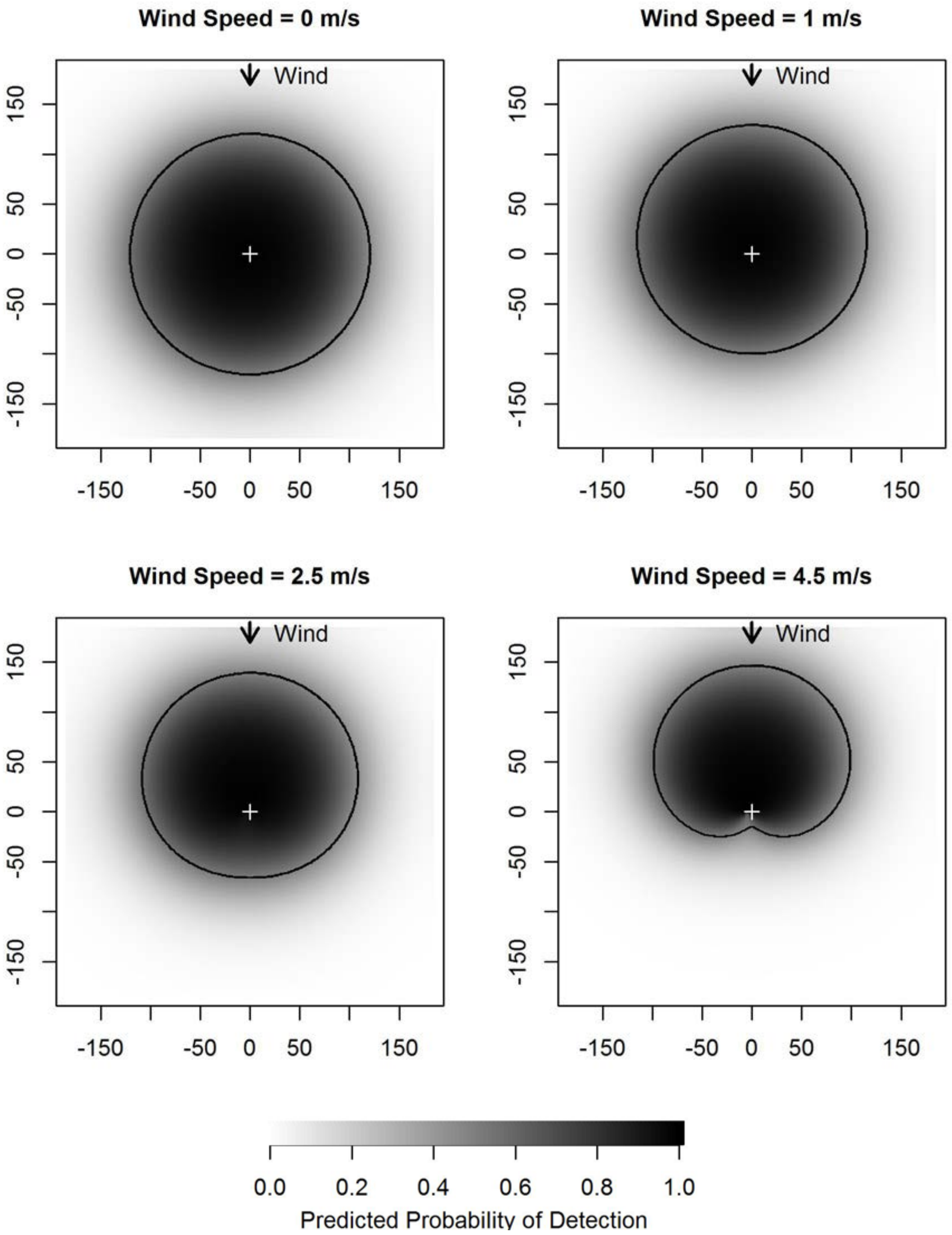


Figure E7: The predicted perceptible area around the observer for LCSP. The solid black border indicates the limit inside which predicted probability of detection ≥ 0.5. Wind was modeled as coming from due north. The position of the observer at (0,0) is marked with a white +. Distances are in meters.

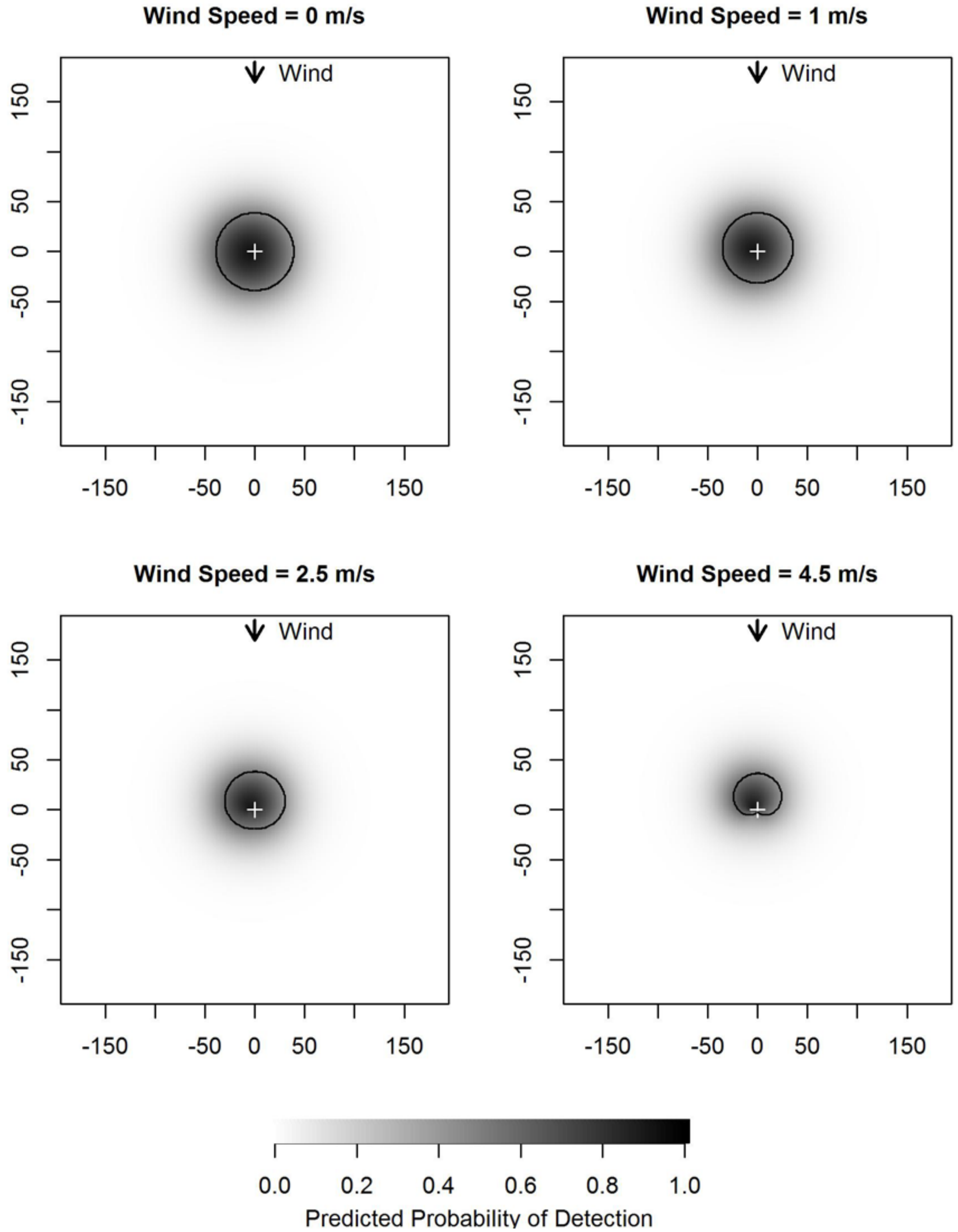


Figure E8: The predicted perceptible area around the observer for SAVS. The solid black border indicates the limit inside which predicted probability of detection ≥ 0.5. Wind was modeled as coming from due north. The position of the observer at (0,0) is marked with a white +. Distances are in meters.

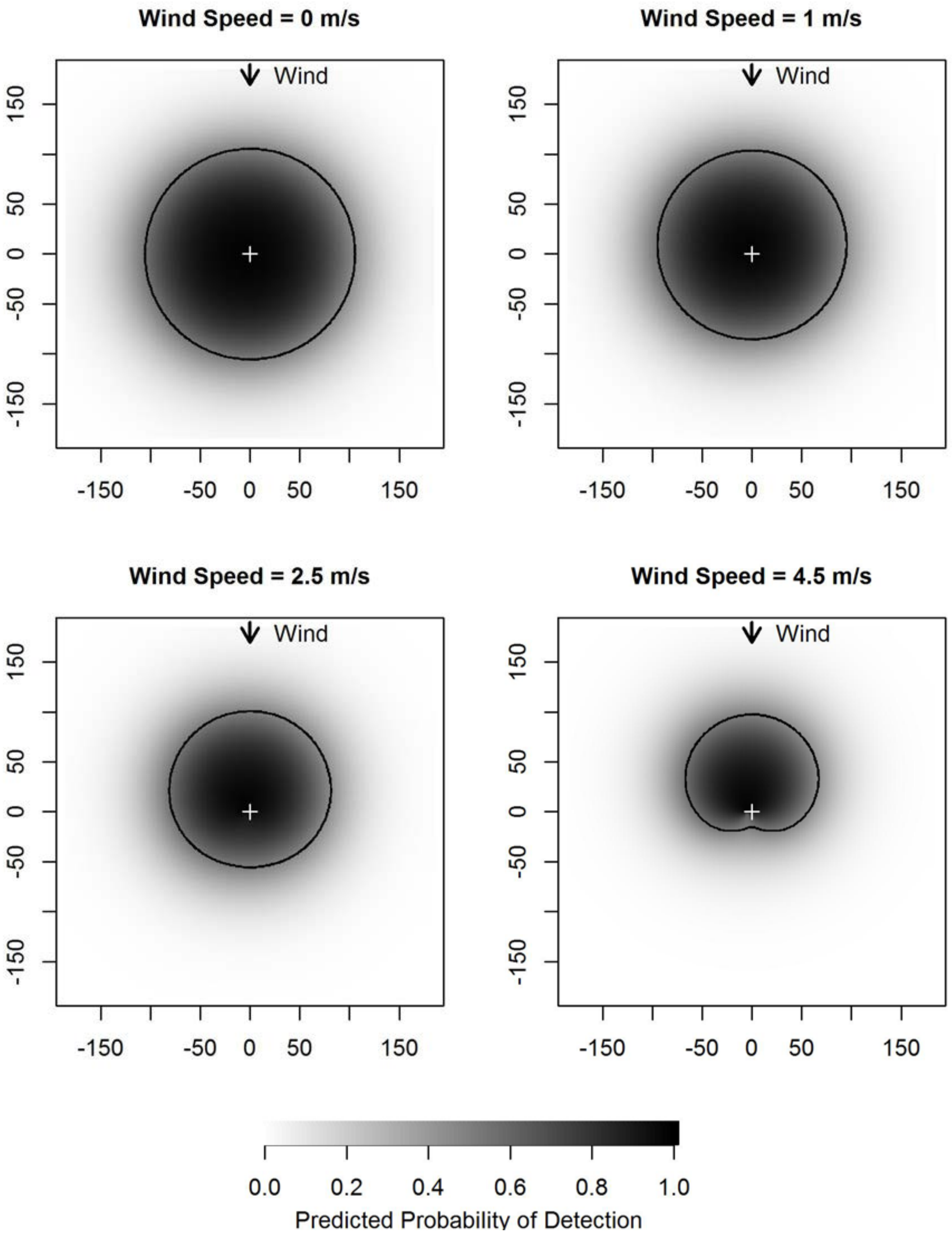


Figure E9: The predicted perceptible area around the observer for SEWR. The solid black border indicates the limit inside which predicted probability of detection ≥ 0.5. Wind was modeled as coming from due north. The position of the observer at (0,0) is marked with a white +. Distances are in meters.

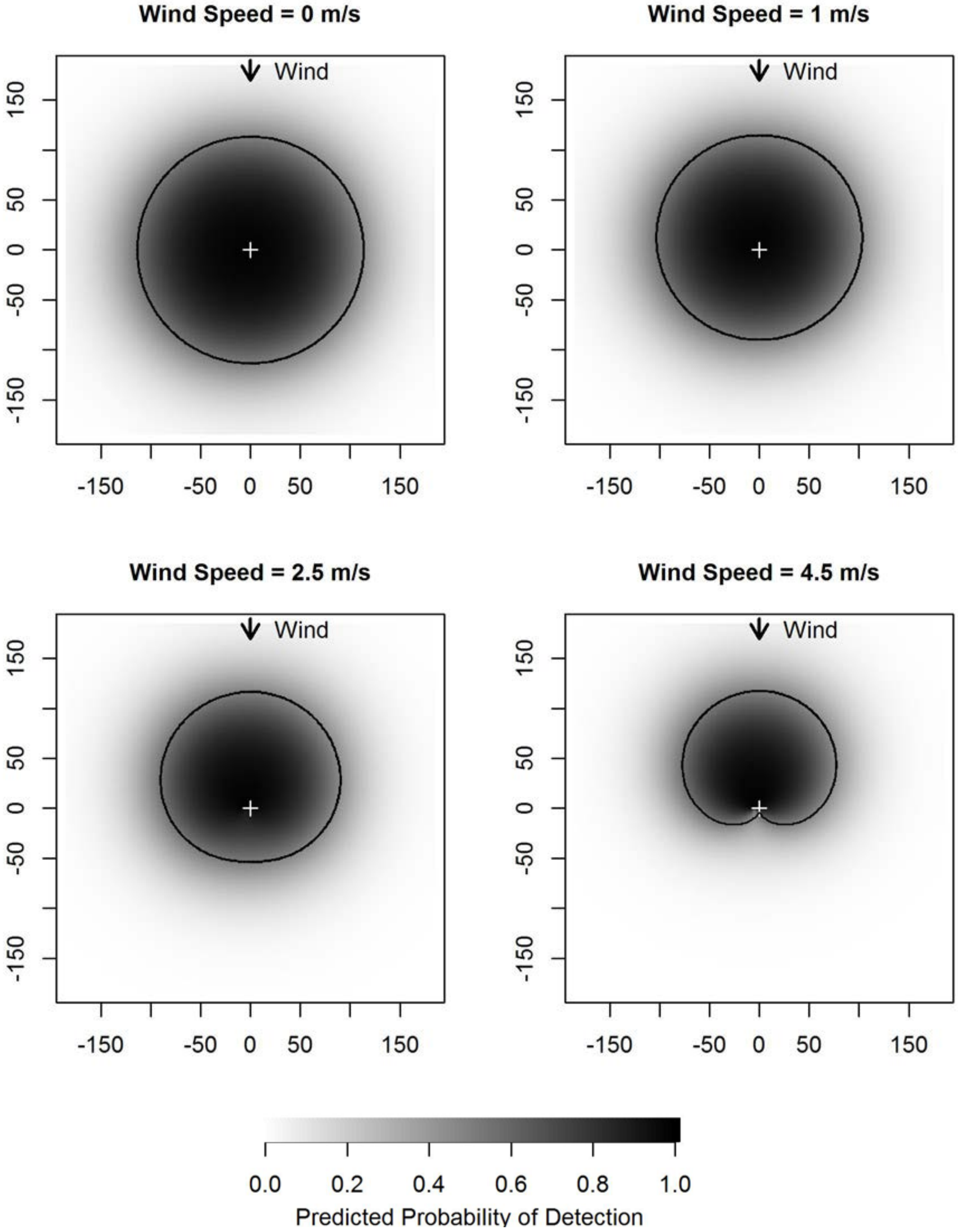


Figure E10: The predicted perceptible area around the observer for VESP. The solid black border indicates the limit inside which predicted probability of detection ≥ 0.5. Wind was modeled as coming from due north. The position of the observer at (0,0) is marked with a white +. Distances are in meters.