

A Wildlife Simulation Package (*WiSP*)

Walter Zucchini^{*1}, David Borchers^{**}, Stefan Kirchfeld^{*} and Martin Erdelmeier^{*}

^{*} *Institut für Statistik und Ökonometrie, University of Göttingen, Germany.*

^{**} *Research Unit for Wildlife Population Assessment, University of St Andrews, Scotland.*

Abstract

The Wildlife Simulation Package (*WiSP*) is a library of functions written in **R** (Ihaka and Gentleman, 1996). It is designed to illustrate the theory and techniques for wildlife abundance estimation in closed populations described in Borchers, Buckland and Zucchini (2002).

WiSP provides an environment for students and researchers to experiment with methods for abundance estimation and to investigate their properties using simulation. It enables users to generate virtual animal populations with realistically complex spatial and individual characteristics and to apply diverse survey techniques to compute point and interval estimates of abundance. The relative merits of different designs and estimators can be assessed under different conditions, for example when the population or the sample is small; when model complexity is increased, or if the underlying assumptions are not met.

We outline the structure and the main components of *WiSP* and illustrate its use for comparing the performance of two survey strategies, and for assessing the behavior of abundance estimators with respect to specific violations of model assumptions.

Keywords. abundance estimation, **R**-library, statistical software, simulation, wildlife.

1 Introduction

The problem of estimating the abundance of wildlife populations has generated an enormous literature covering a rich variety of methods (see e.g. Seber, 1982) necessitated by divergent physical, behavioral and demographic characteristics of animals (and plants) and considerations of the costs and practicability of surveying populations. Borchers *et al* (2002) illustrate that, despite their superficial diversity, most of the standard methods are fundamentally similar and can be understood as variations on a single theme.

The main objective in developing *WiSP* was to create a convenient environment for students and researchers to experiment with different survey designs in an artificial but realistic setting,

¹Corresponding author: Walter Zucchini, Institut für Statistik und Ökonometrie, Georg-August-Universität, Platz der Göttinger Sieben 5, Göttingen, Germany. *email:* wzucchi@uni-goettingen.de

namely using generated populations with versatile spatial and animal-level characteristics. The advantage of course is that the true abundance is known and it is therefore possible to study how the properties of estimators are affected by the design parameters (survey effort, observer skill, etc.) and by specific violations of the model assumptions.

WiSP is a library of functions in **R** (Ihaka and Gentleman, 1996) that is available as free software under the terms of the Free Software Foundation’s GNU General Public License (Hornik, 2002). Important advantages of using **R** in our application are that it is easy for users of *WiSP* to modify the existing functions (e.g. change an estimator), to add new functions (e.g. a new survey design) and to utilize the full range of **R** tools for interactive data manipulation, calculation and graphical display. It has a standardized L^AT_EX-based documentation format for generating help documentation. A most convenient, if not critical, feature of **R**, from the point of view of the system design for *WiSP* is that it is an object-oriented environment.

This paper is organized as follows: Section 2 outlines the main concepts and terms used in animal abundance estimation. The distinction between *generating* observation model and *assumed* observation model, which is important for assessing the robustness of the estimators, is explained. Section 3 summarizes the implementation of these concepts in *WiSP*, i.e. the functions that are available to construct the state model, the survey design, the generating observation model, and those needed to compute point and interval estimates of abundance. Section 4 outlines the survey methods that are currently implemented in *WiSP*. Section 5 gives three examples of how *WiSP* can be used to assess the properties of estimators under different assumptions. Section 6 concludes.

2 Building Blocks

The model describing the state of a population, that is the spatial distribution and characteristics of the individual animals is called the **state model**. The term population is more complex in the context of estimating animal abundance than it is in standard statistical investigations. The individuals might congregate in groups of varying sizes; each animal has a gender, age, size, and so on. Such attributes, as well as behavioral factors, can affect the probability that a given animal will be detected or captured. In other words there may be sampling bias that needs to be taken into account when estimating abundance. The properties of abundance estimators depend on the state of the population; incorrect assumptions about the state can result in misleading estimates. Thus for the purposes of comparing survey strategies or estimators in a simulated setting, and especially for assessing the sensitivity of abundance estimators to violations in model assumptions, the software must be able to generate populations that are realistically complex in their spatial and animal-level characteristics.

The **generating observation model** specifies the probabilistic process that determines which animals are detected or captured, given the search region, the positions and other characteristics of the animals (exposure, type, etc.) as well as features of the survey (visibility conditions, survey effort, etc.). The generating observation model describes the detection (or capture) probabilities for each animal in the population. These probabilities are used to generate a **survey sample** from the virtual animal population. The generating observation model applies to the **covered region** (the sub-region that is searched), which is chosen using

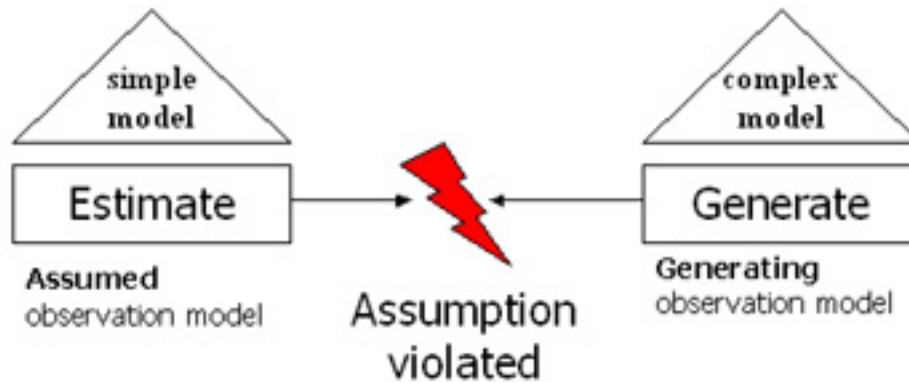


Figure 1: *Principle of assessing sensitivity to assumptions.*

a set of rules, called the **survey design**, and usually involves some randomization. The surveyor has control of the survey design, e.g. the survey effort (number of traps, observers, how long they search, etc.), the shape of the survey units (rectangles, circles, strips, etc.) and the method to select them (simple random sampling, systematic sampling on a grid, etc.). Thus, the survey design determines the covered region (*where do we look*) and the **survey method** we apply (*how do we look*). Examples for survey methods are plot sampling and line transect sampling.

The **assumed observation model** specifies our assumptions about the probabilistic process that determines which animals are detected or captured. It specifies what information we *actually* use to estimate the abundance. In other words the *assumed* observation model is the one used to estimate the detection probabilities and hence the abundance. In contrast the *generating* observation model describes the true detection probabilities that are used to generate the sample. The generating observation model is usually more complex than the assumed observation model, and by increasing the complexity of the generating observation model the user is able to violate the model assumptions of estimators (cf. Figure 1). For example, in the simplest version of mark–recapture estimation it is assumed all animals are equally likely to be captured, irrespective of whether or not they were captured on a previous occasion. In some versions of distance sampling it is assumed that the probability of detection depends only on the distance between the observer and the animal, whereas in reality this can also depend on the size of the group to which the animal belongs; large groups are easier to detect. Thus some of the assumptions that are made are not necessarily met. In other words the *assumed observation model* may be wrong in certain respects. Clearly it is desirable to know the potential consequences of such violations. It is therefore important that the software should be able to generate data that does not meet the assumptions of the assumed observation model. That is why *WiSP* makes a distinction between the *generating* observation model and the *assumed* observation model. The former generates the samples and thus reflects “reality”; the latter can involve compromises that are made by applying a method that is based on a simplifying assumption.

Incorrect simplifying assumptions are *not* necessarily disadvantageous. A simple model that is known to be incorrect may yield abundance estimators that have better properties than those resulting from a more complex model that avoids incorrect assumptions. The latter

will generally lead to a smaller bias but to a larger standard error. That is because complex models have more parameters to be estimated and, for a given sample size, the precision with which the parameters can be estimated decreases with the number of parameters in the model. Furthermore, it is also possible to use a generating observation model that is simpler than the assumed observation model, for example to apply an estimation method that takes account of animal-level heterogeneity when, in fact, there is none. This allows us to investigate the effect of applying a method that is unnecessarily elaborate.

3 Abundance estimation in *WiSP*

This section will outline how the concepts introduced in Section 2 are implemented in *WiSP*, that is the functions that are available to construct the state model, the survey design and the generating observation model, the objects that are involved in these processes, and finally

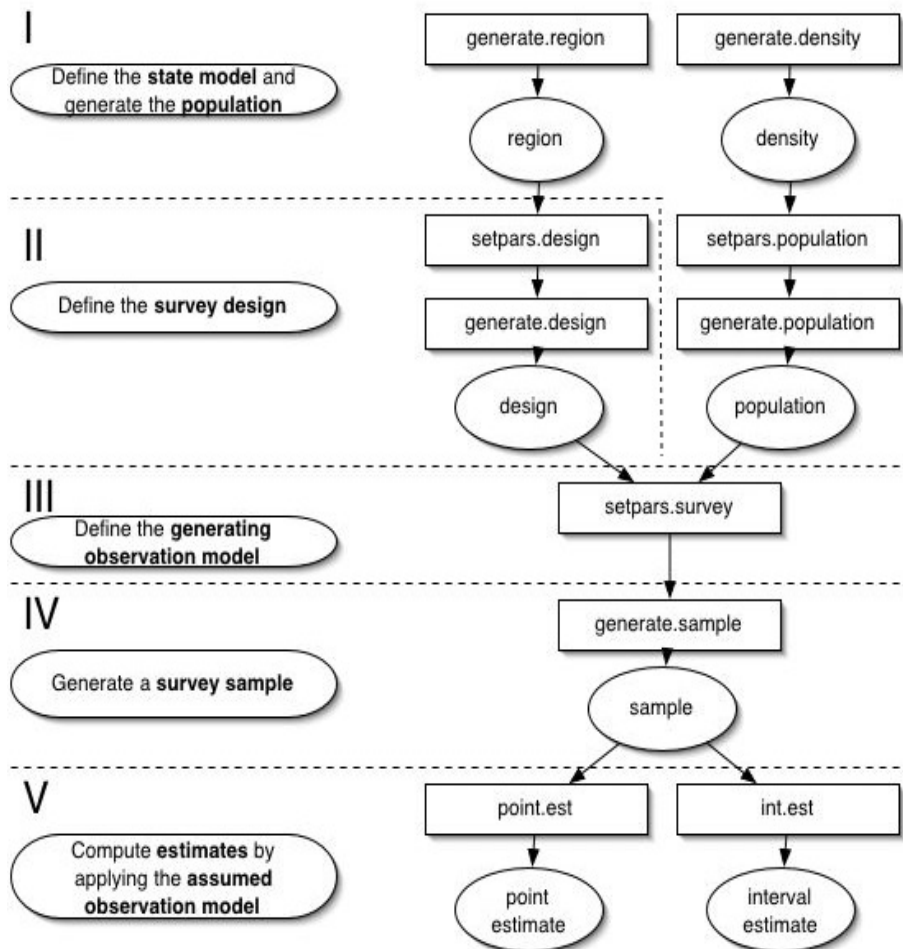


Figure 2: *The abundance estimation process in WiSP. Functions are shown as rectangles, objects as ellipses.*

the functions used to compute point and interval estimates of abundance.

Figure 2 sketches the entire process in five steps starting with the specification of the state model and ending with the estimation of abundance. It also shows the *WiSP* core functions (rectangles) and core objects (ellipses) that are involved in each of the steps. Note that each function named in Figure 2 (except for those involved in step I) actually represents a set of functions that carry out similar tasks. Thus “generate.sample” is implemented as `generate.sample.pl` in the context of plot sampling and as `generate.sample.lt` in line transect sampling. The suffix at the end of the command (“.pl”, “.lt”, etc.) indicates which survey method is to be applied.

The **state model** is specified by the functions `generate.region`, `generate.density` and `setpars.population`. In the current implementation a region is simply a rectangle of given height and width and represents the total area on which the virtual population is located. The object `density` specifies the absolute (or relative) density of animals at all points in the region. It can be represented as a surface over the survey region (Figure 3) and is used to assign positions to the animal groups in the region. (Strictly speaking the object “density” specifies the *expected density*; the actual positions are allocated at random.)

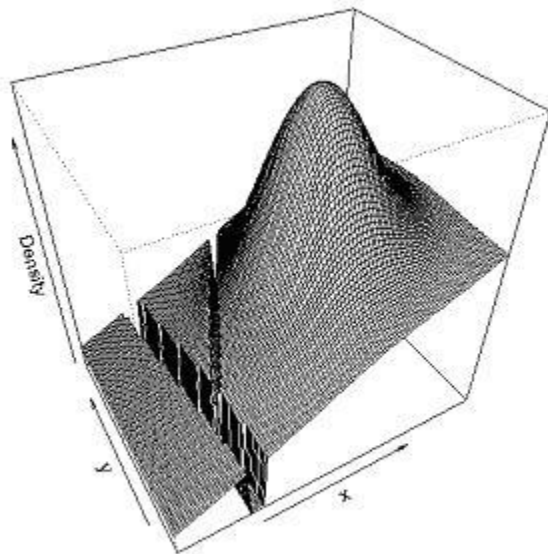


Figure 3: *Example of a complex density with a linear trend, two strips of zero density (e.g. rivers) and a hotspot (a sub-region of unusually high density).*

WiSP offers a set of functions that enable the user to conveniently generate and modify densities and thereby to investigate the properties of different survey methods and estimators under a variety of assumptions regarding the spatial distribution of the animals in the survey

region. The specific characteristics of animal groups and of the individual animals are controlled by the function `setpars.population`. This determines the number of animal groups in the population, the distribution of group sizes and the exposure distribution of each group (i.e. their detectability/catchability). The user may also associate additional characteristics with each animal (e.g. age, gender, etc.). This information is held in an object of the class `pars.population`. (Parameter objects are not displayed in Figure 2 for reasons of clarity.) The objects `region`, `density` and `pars.population` contain the information needed to generate a population object. This is done by the function `generate.population`, as displayed in Figure 4.

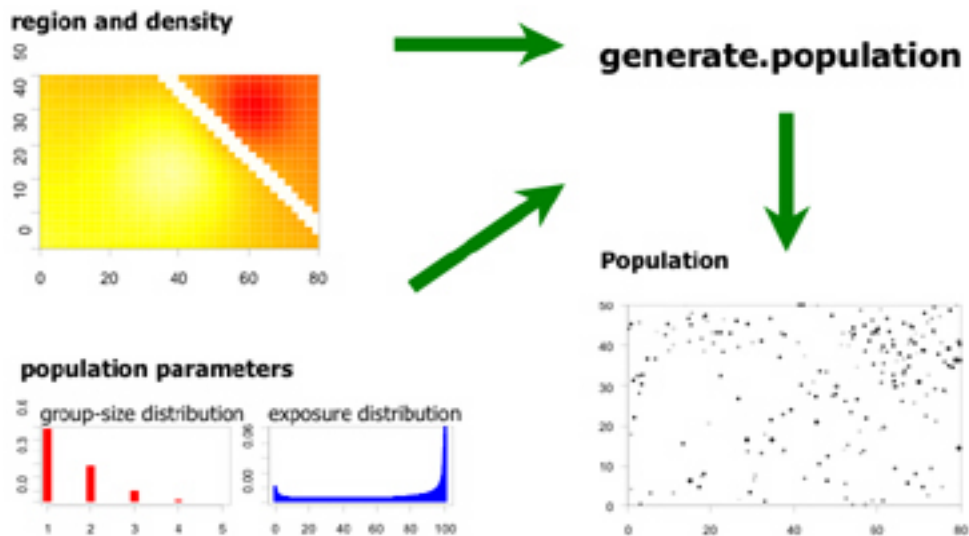


Figure 4: *Generating a realization of a population from its component objects.*

The properties of the survey design that is to be used to survey the population are specified by a function of the type `setpars.design`. The currently implemented designs are plot sampling, mark-recapture methods, removal methods (including catch-effort and change-in-ratio), distance methods (line- and point-transects) and nearest-neighbour methods. Thus, for example in the case of line-transect sampling the function `setpars.design.lt` is used to specify the number and width of the transects, how detectability of an animal is to depend on its distance from the trackline and on animal-level characteristics such as exposure, group size and gender. In the case of a mark-recapture survey the function `setpars.design.cr` is used to specify the probability that an animal will be captured on each occasion, and how that probability is to depend on its characteristics including its capture history. For example animals may become “trap-shy” or “trap-happy”. Setting up a specific survey design usually involves some randomization, a task carried out by a function of the type `generate.design`.

Having generated a virtual population and a specific survey design one is in a position to carry out the survey. The functions of the type `generate.sample` use the detection probabilities delivered by the corresponding function of the type `setpars.survey` to generate a survey sample. Figure 5 shows an example plot of a sample object.

The **generating observation model** is defined by a function of the type `setpars.survey`.

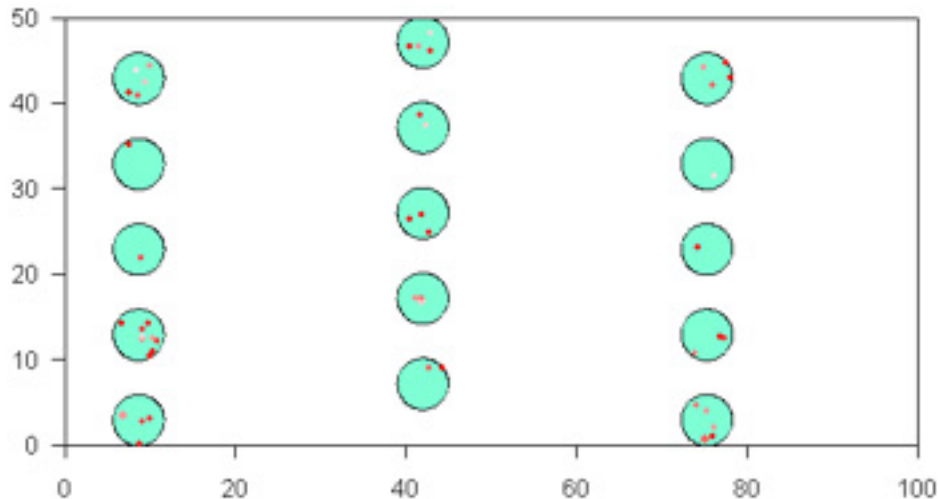


Figure 5: *Example of a point transect sample.*

It uses a detection function that computes the detection probabilities for all animals in the population. Therefore it needs the design- and the population object, as well as optional user-defined input parameters that specify additional details relating to the probability of detection. The generating observation model and the survey design are related because, among other things, the latter determines where we look and how hard we look, but there is also an important distinction between them. The former determines the detection/capture probabilities that can depend on properties of the population (e.g. the exposure distribution) that are not taken into account by the survey design. For example the detection can be made to depend on the age of an animal whereas this is not taken into account in the survey design, e.g. the age is not recorded in the survey. It is precisely this distinction that enables us to assess the performance of estimators when the assumptions of the assumed observation model are violated.

The sample, that is the particular subset of the population that is detected or captured, depends on chance in up to three distinct ways. There is the random variation in the state model, for example the positions and exposures of the animals in a population. Secondly the survey design can involve randomization, for example in the positioning of the transects in line transect sampling. Finally, for a given population and a given design, chance is involved in determining which particular animals are detected or captured. By means of simulation it is possible to investigate the effect of these sources of variation either collectively or separately. For example, by repeatedly surveying a fixed population using a fixed line transect design one can assess the effect of the remaining source of variation on the behavior of the abundance estimator.

A sample object in *WiSP* contains the information that is available to the researcher on completion of a real survey. In plot sampling one has the counts in each plot; in line transect one has the distances of each detected group to the centerline and, possibly, other information about animal-level variables such as group sizes. Sample objects also contain details about the particular survey method that was used, for example, the number and size of the sampling units (plots, transects) and other survey-related variables such as sampling effort.

In the last step of the process, the sample object is passed to estimation functions which compute point and interval estimates of abundance. Different types of surveys generate very different types of sample information and therefore require different estimators. Thus each type of survey requires a corresponding set of estimation functions. The current implementation of *WiSP* computes maximum likelihood estimation for all survey methods covered as well as a number of traditional alternatives. At least one method for interval estimation is implemented for each design. Parametric bootstrap confidence intervals are available for all cases but alternative methods (non-parametric bootstrap, asymptotic methods, profile likelihood) are also available for some designs.

4 Survey methods

We give a brief outline of the survey methods that are currently implemented in *WiSP*. Details are given in Borchers *et al* (2002).

Plot sampling covers methods such as quadrat sampling and strip sampling. Although these may differ in their implementation they are effectively identical from a statistical point of view; they differ only in the shape of the survey units. It is assumed that all animals in the covered region are detected with certainty.

Removal, catch–effort and change–in–ratio methods require at least two survey occasions and involve no assumptions regarding the state model. The design is “search everywhere”. The animals captured on each occasion are counted and removed from the population. The estimation model of the simple *removal* method assumes that capture probabilities are equal for all animals, that they remain the same on every survey occasion, and that the captures are independent. These assumptions are relaxed in *catch–effort* estimation models in which the capture probability is allowed to depend on the catch–effort on each occasion. The estimates are based on changes in the catch–per–unit–effort rather on changes in the catch. The estimation model of the *change–in–ratio* method constitutes a different kind of refinement. It makes use of changes in the proportions of animals of various types (sex, age, etc.) following removals of known numbers of animals of each type, but avoids the need to take account of the catch–effort on each survey occasion.

Mark–recapture methods are similar to removal methods but the captured animals are not removed; they are marked and replaced in the population. The methods vary in complexity depending on the model assumptions. In the simplest case one assumes that on each occasion each animal is equally likely to be captured. Other models allow for various kinds of heterogeneity, for example that the capture probabilities depend on certain (observable) features of the captured animals, on their capture history and so on.

Distance sampling methods estimate the relationship between the detection probability and the distance of detected animals to the centerline of a strip of given width (line transect sampling) or to the center of a circle of given radius (point transect sampling). It is assumed that detection is certain if the distance is zero. This assumption is unrealistic for some animals (e.g. whales) and can be relaxed by using **double platform methods** which are currently not implemented in *WiSP*. Only a single survey is required in distance sampling; randomization is usually applied to select the strips (or circles) that make up the covered region. Here too the methods vary in complexity depending on the assumptions made and

the type of information collected.

In **nearest neighbor sampling** objects (animals or plants) in the survey region are randomly sampled, and the distances to their nearest neighbors are recorded. In **point-to-nearest-object** sampling, points within the survey regions are selected at random and the distance from each point to the nearest object is measured. If we assume that objects are independently and uniformly distributed throughout the survey region then the distribution of nearest-object distances is the same under both approaches. The estimator of abundance is based on the fact there is an inverse relationship between the average distance to the nearest object and the density of objects in the region.

5 Sensitivity analyses

In this section we illustrate how *WiSP* can be applied to assess the sensitivity of two abundance estimation methods to model assumptions. We also compare two strategies for plot sampling. Details of the survey designs and the estimators are given in, e.g., Borchers *et al.* (2002).

5.1 Simple mark-recapture model

In the simplest version of mark-recapture estimation it is assumed that the capture probability is the same for all animals and on all survey occasions, irrespective of the capture history of the individual animals. We illustrate how *WiSP* can be used to assess the effect of two specific departures from these assumptions. We use three different *generating observation models*. The first one meets all the assumptions: on each survey occasion the capture probability is 5% for all animals. The second includes animal-level heterogeneity; the exposure to capture is not constant but instead varies at random over the animals in the population according to the exposure distribution shown in Figure 6.

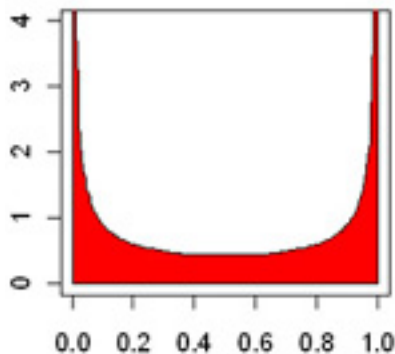


Figure 6: *Example of an exposure distribution.*

The function `setpars.population` enables the user to specify the exposure distribution, and the function `setpars.survey.cr` to specify the link between exposure and capture probability. We set the capture probability equal to one tenth of the exposure; it therefore varies

between 0 and 10%. The particular U-shaped exposure distribution displayed in Figure 6 has distinct modes at 0 and 1. Thus many animals in this virtual population are very unlikely to be captured, and many are captured with probability close to 10%.

The third generating observation model incorporates “trap-shyness”. Here we set the probability capturing an animal for the first time to 5% and reduced this to 3% for animals that had been captured on a previous occasion.

We generated 10000 populations of size 500 and surveyed each population using each of the three *generating* observation models described above. A three-occasion survey was used throughout. Estimation was carried out using the same *assumed* observation model, i.e. assuming that the capture probability is constant for all animals and on all three survey occasions. Essentially the estimator is based on the ratio of unmarked to marked animals captured on the second and third survey occasions. The results are shown in Figure 7.

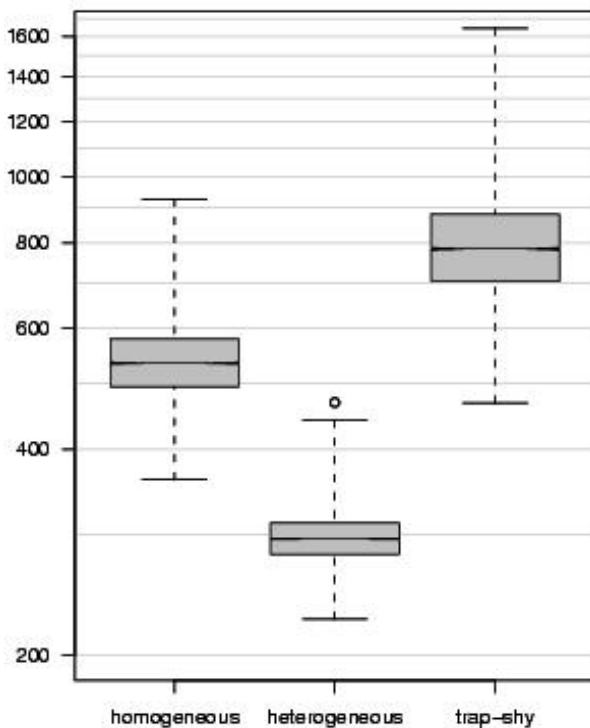


Figure 7: *Boxplots of estimates based a simple mark-recapture model estimator for three different generating observation models.*

Figure 7 shows that even in the ideal circumstance in which all assumptions are met the maximum likelihood estimator tends to overestimate abundance. But this bias is modest compared to that in the other two cases considered.

The gross underestimates in the heterogeneous case can be explained by the fact that a substantial proportion of animals have very low exposure (Figure 6.) Such animals are very seldom captured and are thus “invisible” to this type of survey. In effect they are not counted as members of the population.

Trap-shyness leads to the situation in which the subpopulation of marked animals become “under-represented” in subsequent samples, whereas the estimator is based on the assumption that the marked and unmarked subpopulations will be represented proportionately. This leads to overestimates of the number of unmarked animals and hence to overestimates of the total abundance.

5.2 Change-in-ratio method

The simplest change-in-ratio estimator relies on the assumption that the detection probability does not depend on animal-level variables, such as sex, age and so on. We assess the behavior of this estimator in the case where this assumption is met and in one case where it is violated.

We generated 10000 virtual populations each comprising 500 animals. The sex of each animal was determined at random with probability 0.5 of being male/female. The survey design involved two survey occasions. In the first experiment we used a capture probability of 50% for both males and females, i.e. the generating and the assumed observation models were the same. In the second experiment the capture probability was set to 60% for males and 40% for females. Boxplots of the resulting estimates are displayed in Figure 8.

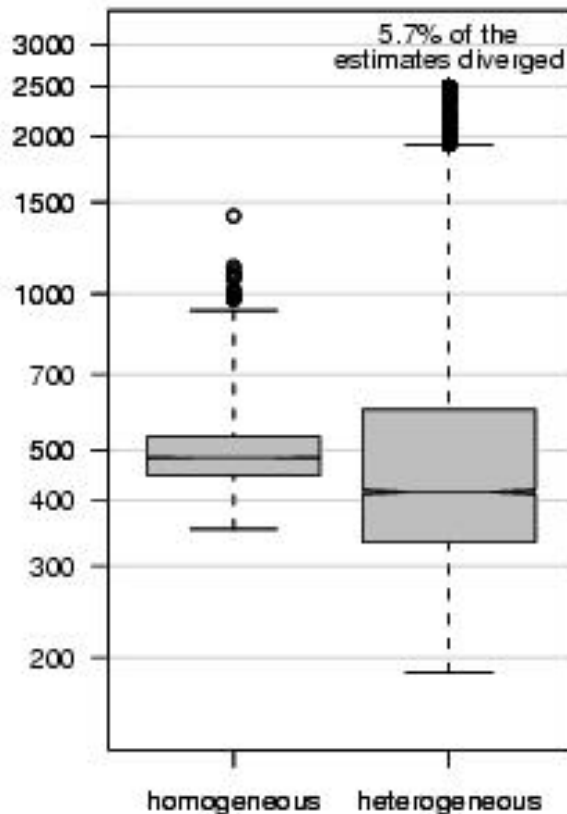


Figure 8: *Boxplots of abundance estimates using the catch-in ratio method for a population comprising 500 animals.*

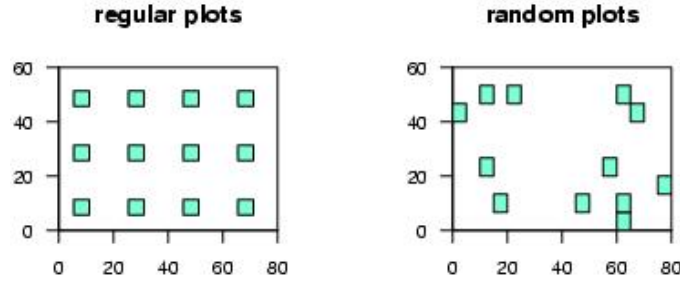


Figure 9: Plot of a regular grid design (left) and a random grid design (right), They have the same number of plots and cover the same area.

In the case where the model assumptions are met the distribution of the estimator is slightly skewed but the estimator is approximately unbiased. However, when the detection probability differs for males and females, i.e. when there is animal-level heterogeneity, the estimator underestimates abundance. There also is the marked increase in the variance of the estimator. Evidently the less rapid removal of males from the population does not compensated for the more rapid removal of females. Clearly the behavior of the estimator is not robust to violations to the assumption of homogeneity.

5.3 Plot design: random vs. regular plots

In our last example we assess the performance of two plot survey designs. The issue here is not to investigate the effect of violating model assumptions, but rather to compare the performance of two different survey designs over a range of circumstances. Plot sample estimation is very easy: From the region under consideration (of area A) one selects a subregion (of area a) and counts all the animals in the subregion. The estimator of abundance for the entire region is simply the number of animals counted in the subregion multiplied by $\frac{A}{a}$. In practice the subregion usually comprises a number of plots of equal size. Generally the plots are placed at random over the region, or they are placed systematically on the vertices of a rectangular grid covering the region (Figure 9).

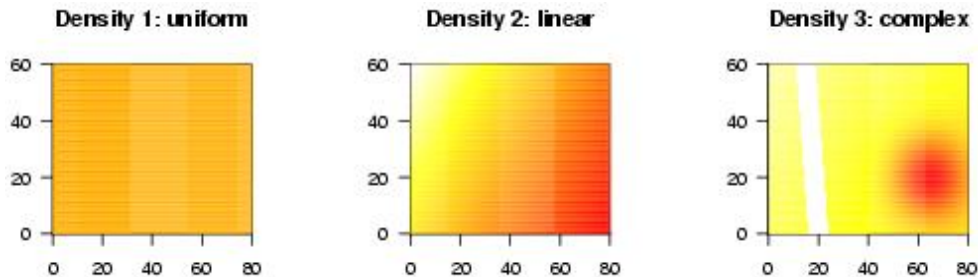


Figure 10: Three spatial densities (a) constant (b) linear (c) complex.

Here we compare the performance of the abundance estimator for these two strategies for placing the plots. In particular we compare them under three different assumptions about

the spatial distribution of the animals.

We generated 10000 replications of 3 virtual populations of size 500. The populations differ in their spatial distributions, i.e. the density of animals over the region. (See Figure 10.) In the first case the density is constant over the region; in the second it increases linearly in the direction NW to SE, and in the third case it is a more complex entity that includes a “hotspot” in the south-east corner and a river (with no animals) running approximately north-south.

Every population was surveyed using each of the two plot designs. The first design placed the plots on a regular grid and the second placed the plots at random. Typical plot placements are show in Figure 9.

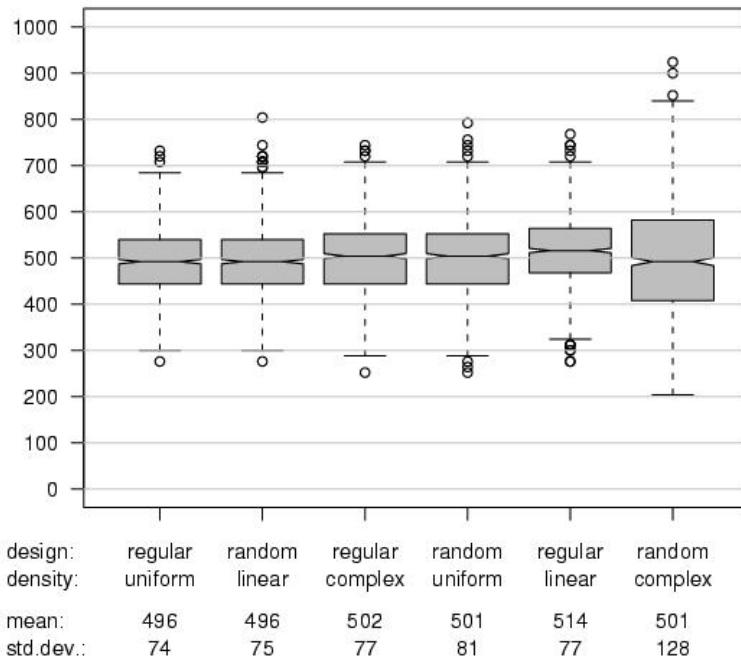


Figure 11: *Abundance estimates based on regular and random grid designs for the three densities shown in Figure 9. The population size is 500.*

A boxplot of the resulting abundance estimates are shown in Figure 11. These results indicate that there is little difference in the distribution of the estimates when the population density was constant or linear. Noticeable differences arise in the case where the density was complex. The estimators are approximately unbiased, although the regular case exhibited a tendency to overestimate slightly (3%). Theoretically the random case leads to an unbiased estimator whereas the regular case can be biased because regular plots can systematically “miss” local features of the underlying density. However, the most prominent feature in Figure 11 is the substantially larger variance of the estimator based on random plots when the density is complex. In retrospect the reason for this is obvious. The randomly placed plots sometimes all “miss” the hotspot on the SE corner of the region thus leading to an underestimate of abundance, and sometimes “too many” plots land on, or near, the hotspot thus leading to an overestimate of abundance. In contrast the regular plots are always in fixed positions and so

the estimator is not subject to the above source of random variation. Thus although regular designs are not guaranteed to lead to unbiased estimators, the variance of the latter can be substantially smaller than that resulting from randomly placed plots, especially when the spatial distribution of animals is complex. In experimental design randomization is generally regarded as a “good thing”, mainly because it avoids bias. Our experiment suggests that the bias that results by using a regular design can be insubstantial compared to the increased standard error that results by using a random design.

6 Concluding remarks

WiSP is primarily intended as a teaching tool to illustrate the basic principles underlying the most popular (closed population) abundance estimation techniques and to study their strengths and weaknesses. In its current form *WiSP* does not have the functionality of the more comprehensive software packages available for the analysis of field data. (Useful lists of links can be found at www.phidot.org/software and www.mbr.nbs.gov/software.html.)

The free availability of **R** was an important consideration in selecting it as the language for *WiSP* but the other advantages that **R** offers were even more important in terms of future development of the *WiSP* library. The code can be made available in text form, which makes it conveniently accessible to users who wish to modify it, to define new functions to cover a greater range of survey designs, or to introduce additional features in the state process. The object-orientated design of *WiSP* that is facilitated by **R** renders the task of making such modifications and enhancements incomparably less forbidding than would be the case if a language without this feature had been used.

7 References

- Borchers, D.L. Buckland, S.T. & Zucchini W. (2002). *Estimating Animal Abundance: Closed Populations*. Springer-Verlag, London.
- Hornik (2002), “The R FAQ”, <http://www.gnu.org/copyleft/gpl.html>
- Ihaka, Ross and Gentleman, Robert (1996), R: A Language for Data Analysis and Graphics, *Journal of Computational and Graphical Statistics*, 5(3), 299-314.
- Seber, G.A.F. (1982), *The Estimation of Animal Abundance and Related Information Parameters*. 2nd edition, Griffin, London.