

ESTIMATING SMALL MAMMAL AND WINTER NEST DENSITIES USING CAPTURE-RECAPTURE AND DISTANCE SAMPLING WITH R

Version 1

Dominique Fauteux
Gilles Gauthier
Marie-Christine Cadieux



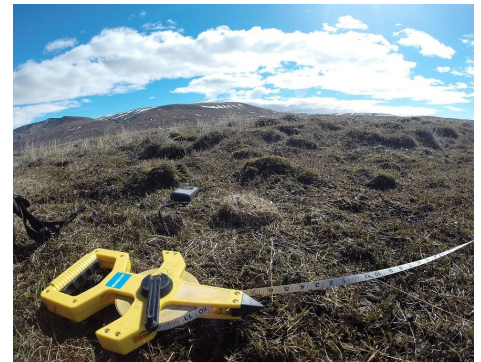
UNIVERSITÉ
LAVAL



CENTRE D'ÉTUDES NORDIQUES
CEN Centre for Northern Studies



canadian museum of nature
nature
musée canadien de la nature



A collaborative document produced with the financial support of Polar Knowledge Canada

About the authors

Dominique Fauteux, Ph.D., was a post-doc at Université Laval, Québec (QC) and is currently a research scientist at the Canadian Museum of Nature, Ottawa (ON). He has been studying small mammal ecology in both boreal and arctic ecosystems for 10 years.

Gilles Gauthier, Ph.D., is a professor at Université Laval and has been conducting ecological research in the Arctic for more than 25 years. He is leader of the Ecological and Environmental Monitoring Program at Bylot Island, Sirmilik National Park, NU.

Marie-Christine Cadieux, M.Sc., is a research assistant at Université Laval and has been working extensively on the ecology of Arctic vertebrates for more than 15 years.



UNIVERSITÉ
LAVAL

The first version of this manual has been written in 2017 and published in 2018.

It was printed at Université Laval, Québec (QC), Canada.

© All copyrights are retained by the authors. This manual was intended to be publicly available.

Citation: Fauteux, D., G. Gauthier, M.-C. Cadieux. 2018. Estimating small mammal and winter nest densities using capture-recapture and distance sampling with R – Version 1. Centre d'études nordiques, Université Laval, Québec, 39 pages.

Photo credit of cover page: *Dominique Fauteux*

TABLE OF CONTENT



1. Introduction	1
2. Estimating small mammal densities with spatially-explicit capture-recapture methods	3
2.1. How to format capture-recapture datasets for R	4
2.1.1. Capture files	4
2.1.2. Trap files.....	7
2.2. Data analysis with SECR models implemented in R	9
2.3. Model selection and model-averaging density estimates	10
3. Estimating lemming densities with SECR: an example	13
3.1. Context of the example.....	13
3.2. R codes to estimate SECR densities with the Bylot Island example.....	13
4. Estimating winter nest densities with distance sampling methods.....	23
4.1. How to format winter nest datasets for <i>Distance</i>	25
4.2. Data analysis with the package <i>Distance</i> in R.....	27
5. Estimating winter nest densities with <i>Distance</i> : an example	31
5.1. Context of the example.....	31
5.2. R codes to estimate winter nest densities with the Bylot Island example.....	31
6. References	39

1. INTRODUCTION



Ecology is the study of changes in animal population abundance. Calculating the abundance of a natural population living in the wild is not an easy task because counting every single individual is often impossible. Indeed, vertebrate populations can be dispersed over hundreds of kilometers, their numbers can go from a few to millions, and individuals can be hard to observe (e.g. animals that are very small, nocturnal, or cryptic).

Estimating abundance vs. densities

One way to estimate a population's abundance is to count the number of individuals in a fraction of the whole population by restricting the observations to a small area, a process called sampling. Once the abundance has been estimated for this small area with a certain amount of confidence, the population abundance can be assessed by adjusting the sampled abundance to a standardised area such as the number of individuals per hectare or square kilometers. This is a simple example of how we can estimate densities.

Imperfect detection of animals

An important aspect that has to be considered when estimating abundances or densities is that some individuals will never be seen or heard even though they are present. Thus, a trained worker who is watching or listening to count the number of individuals in a given area may be able to observe only approximately 60% of all the animals that are present because some are hidden underground, by vegetation, or uneven topography. Modelling imperfect detection probabilities adequately has been the subject of many studies during the past few decades and is now included in several analytical methods such as capture-mark-recapture methods. Using those methods, however, requires some skills and guidance.

Spatially-explicit capture-recapture and Distance-based modelling of densities

In this manual, we present in details two statistical methods commonly used with small mammals to estimate densities considering imperfect detection probabilities. These methods are the spatially-explicit capture-recapture (SECR) method and the line-transect method, which considers that the probability of detecting a feature decreases with the distance to the observer. In general, the SECR method is applied to small mammal live-trapping data collected on grids following a Cartesian plane and the line-transect method is applied when transects, which are imaginary lines in the field, are used to sample small mammal signs such as winter nests. For more information on applying these sampling methods in the field, please consult our other

manual entitled *Technical manual for sampling small mammals in the Arctic* (Cadieux et al. 2015; http://www.cen.ulaval.ca/bylot/files/Small_Mammal_Sampling_v1.pdf).

The analytical methods presented here are fairly advanced and require some notions in likelihood-based statistics or in the theory of model selection and model-averaging. We also assume that readers have a basic knowledge of the software R, which is freely available on the internet. However, we strived to make the presentation of these methods as simple as possible and to provide ample explanations on how to proceed with the analysis in a step by step approach. For each method, we first present in a chapter the general principles of the method and in a second chapter we guide the reader through the analysis of an example dataset obtained from the Bylot Island project (Gauthier et al. 2013). We also present all the R codes required to perform an analysis, explaining what each line of codes means and does and presenting screen captures of the software.

2. ESTIMATING SMALL MAMMAL DENSITIES WITH SPATIALLY-EXPLICIT CAPTURE-RECAPTURE METHODS



Capture-recapture methods have been used for a long time to estimate population sizes and densities (Williams et al. 2002). These methods perform best on closed populations and require that some assumptions are met (see text box). One of the most recent capture-recapture methods to estimate densities is the spatially-explicit model (Efford and Fewster 2013). This method considers that the detection is imperfect and models it. In addition, it uses the movements of individuals among traps to estimate the average distance moved and the spatial scale of those movements in and around trapping grids. Overestimation of densities, which is common when the scale of movements is not properly considered, are here limited by using the concepts of home range and competition for space.

There are two software packages that can be used to estimate densities with the spatially-explicit method: DENSITY and the *secr* library, available on the R platform (Table 2.1; Efford 2013, Efford 2015). Although the same algorithms are implemented in both softwares, the instructions provided in this manual are based on the *secr* library. These two tools were designed by Murray Efford, the author of considerable work on this method.

In this manual, we propose a standardized approach to estimating small mammal densities with capture-mark-recapture models year after year and to ensure that the estimates are comparable. Following the procedure proposed in this document should make it possible to obtain reliable estimates and minimize the risk of having errors creeping into the analyses. However, it is important to check the output of each model to ensure, for example, that the analyses worked properly or that there were no programming errors. If the estimates give unusual results (a standard error of 0 or greater than the density estimate), a common situation when sample sizes and especially the number of recaptures are too low, it may be necessary to use the minimum number of individuals alive (see details below in section 2.3). In the following sections of this document, the instructions for using the *secr* library are described in details and it is important to follow them step by step to ensure that the models converge properly. It is essential to first install this package on your computer before proceeding further.

* TIPS! – Assumptions of closed-population models

Before applying closed-population models to capture-recapture data, it is important to verify that the following assumptions are met:

- No birth, death, immigration or emigration occurred during the trapping period (or can be considered negligible). If an animal dies during trapping, its capture history must be censored (see box entitled "Censoring trapping histories" below);
- No tag was lost or read incorrectly;
- Capture probabilities are equal among individuals.

Table 2.1. Websites where softwares can be downloaded and additional help on how to use them can be found. Installation instructions are provided on these websites (checked in September 2017).

Software	Website
DENSITY	http://www.otago.ac.nz/density/
R	http://www.r-project.org/
<i>secr</i>	http://cran.r-project.org/web/packages/secr/index.html
Help forum ¹	http://www.phidot.org/forum/index.php

¹There is a high probability that problems you may encounter have already been dealt with and solved. Search the forum thoroughly with keywords before posting a new question.

2.1. How to format capture-recapture datasets for R

It is important to correctly format your dataset so that the *secr* package can recognize it. We recommend having the data in a data file in text format, each column being separated by a tabulation. Two files must be provided: one for the capture data and one with the spatial position of traps.

1. The file containing captures can be easily built by using data entered directly from field books to Excel spreadsheets or exported from a database. Typically, each capture or recapture of an individual with the accompanying information is entered on a different line.
2. The exact positions of all traps in trapping grids should be recorded every year in a separate data file. The position of each trap will appear on a separate line but only once because trap position are fixed in a given year.

If several species are captured on a given grid (e.g. brown and collared lemmings) or several grids are trapped, we recommend creating a different capture text file for each species and trapping grid each year. For instance, there should be 4 capture text files and 2 files for trap positions for year x if there are 2 species and 2 trapping grids. Making several text files makes SECR models less heavy, which reduces computing time.

2.1.1. Capture files

An example data file from the Bylot Island study is provided in Figure 2.1. In this case, the data were entered in an Excel file and later exported in a text file. Each column must be separated by a tabulation in the text file. Note that the order of the columns and the column titles are important and must be EXACTLY the same because these titles correspond to the names of the variables in the program that follows.

	A	B	C	D	E	F	G	H	I	J
1	#Session	ID	Occasion	Detector	species	sex	age	mass	repro	
2	LG2PP1	152239140A	1	LG2J07	BrownLemming	F	A	48	PF	
3	LG2PP1	152249612A	1	LG2H11	BrownLemming	M	J	23	AB	
4	LG2PP1	152242690A	1	LG2K08	BrownLemming	M	A	47	SC	
5	LG2PP1	152239652A	1	LG2A07	BrownLemming	M	J	26	SC	
6	LG2PP1	152249491A	1	LG2G09	BrownLemming	F	J	24	NRP	
7	LG2PP1	BR_01	-1	LG2C09	BrownLemming	M	A	62	AB	
8	LG2PP1	152243077A	2	LG2F02	BrownLemming	F	J	26	NRP	
9	LG2PP1	152248371A	2	LG2D11	BrownLemming	F	J	18	NRP	
10	LG2PP1	152240424A	2	LG2H05	BrownLemming	M	A	64	SC	
11	LG2PP1	152240052A	2	LG2H08	BrownLemming	M	A	68	SC	

#Session	ID	Occasion	Detector	species	sex	age	mass	repro
LG2PP1	152239140A	1	LG2J07	BrownLemming	F	A	48	PF
LG2PP1	152249612A	1	LG2H11	BrownLemming	M	J	23	AB
LG2PP1	152242690A	1	LG2K08	BrownLemming	M	A	47	SC
LG2PP1	152239652A	1	LG2A07	BrownLemming	M	J	26	SC
LG2PP1	152249491A	1	LG2G09	BrownLemming	F	J	24	NRP
LG2PP1	BR_01	-1	LG2C09	BrownLemming	M	A	62	AB
LG2PP1	152243077A	2	LG2F02	BrownLemming	F	J	26	NRP
LG2PP1	152248371A	2	LG2D11	BrownLemming	F	J	18	NRP
LG2PP1	152240424A	2	LG2H05	BrownLemming	M	A	64	SC
LG2PP1	152240052A	2	LG2H08	BrownLemming	M	A	68	SC

Figure 2.1. Format of the capture data as an Excel spreadsheet (top) and the resulting text file (tab-delimited) to use for capture-mark-recapture analyses with the *secr* package in R (bottom). Note that the columns in the text file (bottom) may not be aligned because the information of some cells, especially some ID and Detector names cover the space of two tabs (the information of the other columns is therefore spaced to the right), which would give a misaligned aspect.

* TIPS! – Censoring trapping histories

If an animal was not released in the field after being captured because it was kept in captivity or accidentally died in the trap, its capture history must be censored. To do this, **a minus sign should be placed in front of the occasion number.** Censoring a capture history avoids including artificial non-captures in the detection histories used to estimate detection probabilities and relaxes the assumption that the trappable population should not change during primary trapping sessions (closed-population assumption, see above).

The columns contain the following information:

1. *#Session*: This column identifies the trapping session by the name of the grid and the primary capture session involved because several trapping sessions can be conducted in a given year. In this example, “LG2PP1” stands for Lemming Grid 2 and Primary Period 1 (June in this case); the comment character # MUST be written before *Session* in the column title.
2. *ID*: Unique identifier of each individual animal captured.
3. *Occasion*: The secondary capture occasion that an individual was captured within the primary trapping session. Secondary trapping occasions are typically consecutive days or consecutive 12-h or 6-h intervals and are numbered consecutively starting at 1 for the first

occasion of a trapping session (e.g. if traps are visited in the morning and evening for 3 consecutive days, this gives a total of 6 occasions numbered from 1 to 6 for each trapping session).

4. *Detector*: It indicates in which trap the animal was captured. Each trap has a unique identifier, which is often a simple alphanumeric coordinate on a Cartesian plane (e.g. A01, E08, D10; Figure 2.2). The identifier of the trapping grid can be added before the trap identifier (e.g. LG1F04) if models that are used include a grid effect. The models described in this document do not require such naming, but it can still be useful to help distinguish capture and trap files when several are created. It is important that the code used to identify each trap is the same in the capture and the trap file.

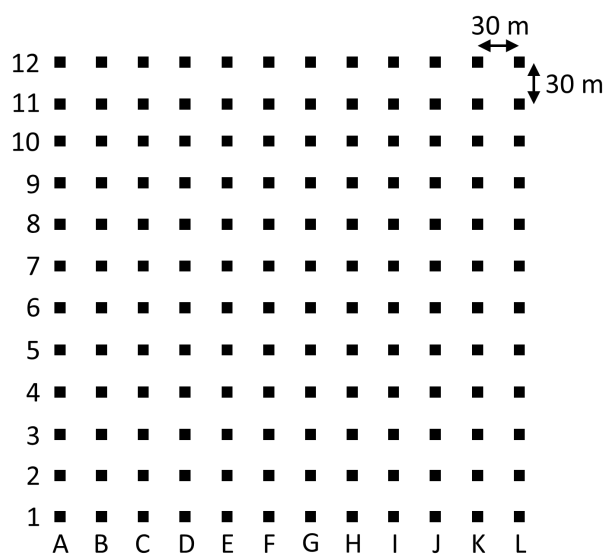


Figure 2.2. Example of a 330×330 m grid with 144 stations at 30-m intervals. Letters are used to label the columns and lines are labelled using numbers.

The first 4 columns are the minimum information required to run SECR models. Additional columns contain information ancillary to the density analyses (i.e. covariates) and are not used by the basic models. However, these data can be left in the text file as they could be used for further analysis. In our example, they are:

5. *species*: name of the species captured. It is **very important that there is no space** between the words here or the software will consider it as two separate columns. For example, “Brown Lemming” is problematic, “BrownLemming” is not.
6. *sex* : sex of the individual captured.
7. *age*: relative age of the individual based on body mass (for brown lemmings, J: females <28g, males <30 g; A: all other individuals).

8. *mass*: body mass of the individual captured.
9. *repro*: reproductive condition of the individual (NRP: non-reproductive female or male; PF: female with perforated hymen; LA: lactating female; PRG: pregnant female; AB: male with testes in abdomen; SC: male with testes in scrotum).

Data included in each column (or covariate) should always be verified with the function `levels(object$column)` to avoid erroneous data. For example, some analyses may be case sensitive or a typing mistake could have been introduced in the data file. Using the function `levels()` will show all the data that were entered in the specified column. Once an error has been located, it is then easy to look for that specific error in the data.

2.1.2 Trap files

Formatting the trap file adequately is also important to ensure that the program recognizes the data structure. Again, the data were entered in an Excel file and later exported in a text file (Figure 2.3). Each column must be separated by one tab. It will require a different text file for each trapping grid each year. A procedure that saves time is to use the files from previous years as initial template and to replace the values with those of the current year. The order of the columns and the column titles must again be EXACTLY the same as in the example file in Figure 2.3 for the same reason as in the previous file. The positions of the traps are determined by a Cartesian plane (X, Y). **Note that in *secr*, units are in meters by default.**

	A	B	C			
1	# Detector	X	Y	# Detector		
2	LG2A01	-3.4	-4.25	LG2A01	-3.4	-4.25
3	LG2A02	5.95	34.25	LG2A02	5.95	34.25
4	LG2A03	-5.1	57.45	LG2A03	-5.1	57.45
5	LG2A04	5.1	86.6	LG2A04	5.1	86.6
6	LG2A05	4.25	126.8	LG2A05	4.25	126.8
				LG2A06	1.7	144.9
				LG2A07	4.25	182.55
				LG2A08	-8.5	206.6
				LG2A09	-3.4	237.45

Figure 2.3. Format of trap position data in Excel (left) and the resulting file in text (tab-delimited; right) to be used for spatially-explicit analyses in R.

The file must contain the following information:

1. *#Detector* : A unique identifier for each trap in a trapping grid, which is often a simple alphanumeric coordinate on a Cartesian plane. These identifiers must perfectly match those written in the *Detector* column of the capture file (see section 2.1.1 above).
2. *X* : the position in relation to the origin along the X axis; the coordinate is in meters (see below for additional details).

3. *Y* : the position in relation to the origin along the *Y* axis; the coordinate is in meters (see below for additional details).

As an example, we can use a trapping grid on Bylot Island with 144 traps (12×12) arranged along perpendicular *X*- and *Y*-axes spaced by 30 m, resulting in 330×330 m for the entire grid. The coordinate (0,0) has to be assigned to a corner of the grid. For example, the origin of the grid shown in Figure 2.2 is A01. In the field it is customary to order the lines in *X* alphabetically, from A to L, and the lines in *Y* by numbers, from 1 to 12. Thus, the position will increase positively in *X* from 0 to 330 m according to the alphabet (A to L) while in *Y*, the position increases from 0 to 330 m according to the number of the trap (1 to 12).

It is important to note that the traps are not always placed exactly at each intersection of the grid (also called trapping stations) but often within a radius of ~15 m from the station, preferably at a site showing signs of use by animals. In the field, the distance of the trap in *X* and *Y* with respect to each point of intersection of the grid is often measured in number of steps made in each of these two directions to reach the trap and then transformed into meters. This makes it possible to position each trap exactly in space and explains why the values in Figure 2.3 are not exact factors of 30 m and are sometimes negative (i.e. if a trap is positioned slightly to the left of the first line in *Y* or slightly below the first line in *X*, which defines the 0,0 coordinate).

*** TIPS! – What to do when some traps cannot be activated.**

When more than one trapping sessions is conducted annually, the first session may begin during snow melt. In those situations, some traps may not be activated during the first secondary occasions if some stations are still covered by snow and will only be activated later. In the *secr* package, it is possible to consider trap activation and deactivation by providing an activation history (e.g. 000111), where 1 indicates a trap enabled while 0 indicates a trap disabled. A trapping history of "000111" indicates that, for a trapping session with 6 occasions, the trap was inactive for the first three occasions, whereas it was activated for the three remaining occasions. To do this, a new column entitled *Usage* must be created in the trap file (Figure 2.3). In cases where a high proportion of traps could not be activated, this could be useful. However, if no more than 2 or 3 traps cannot be activated for a trapping session, it can be ignored because this method complicates the models, which already tend to be over parameterized. We therefore recommend not using this procedure except under extreme conditions (>10% of traps cannot be activated).

2.2. Data analysis with SECR models implemented in R

The parameter of primary interest for the researcher estimated by *secr* is the animal density (D). However, in order to correctly estimate this parameter, *secr* must estimate two other parameters: the detection probability (g_0) and the amplitude of movement (σ). A general definition of g_0 is the mean probability that an animal is captured during a primary period. σ is the mean distance traveled by animals estimated by the algorithm and is calculated as a function of the distances separating each trap in which animals were recaptured consecutively. This distance is used to determine the spatial structure of the animals' home range in and around the grid.

To analyse datasets, we propose to fit six different models in *secr* when more than one trapping session has been conducted on the same grid in the same year and three when a single annual session was performed (Table 2.2). Single-session models are also useful for situations when sample size is too low (e.g. <5 individuals captured and <2 recaptures in total) during most sessions conducted in the same year. The models vary in number of parameters in order to allow estimation of densities when sample size is limited because the more parameters there are, the more poorly the models perform with small sample size, which is typical during the low phase of lemming cycles. However, when sample size is large, more complex models may yield more accurate density estimates.

In the models proposed, the parameter related to the probability of detection (g_0) varies for each trapping session when applicable (but is identical for each secondary occasion within each trapping session) and between the newly marked and recaptured individuals, which considers the effect of a previous capture on the current capture probability (b). The same effects are applied to the movement parameter (σ). All combinations of reduced models without

Table 2.2. Candidate models used to estimate small mammal densities with spatially-explicit capture-recapture methods when two or more trapping sessions have been conducted on the same grid in the same year, or when a single session was conducted.

Number of sessions per year	Model	Density (D)	Detection prob. (g_0)	Movements (σ)
2 or more	Model 1	~session	~session	~session
	Model 2	~session	~session+ b	~session
	Model 3	~session	~session+ b	~session+ b
	Model 4	~session	~session	~1
	Model 5	~session	~session+ b	~1
	Model 6	~session	~1	~1
Single	Model 1	~1	~ b	~1
	Model 2	~1	~ b	~ b
	Model 3	~1	~1	~1

Note: b = effect of previous capture on detection probability; session = temporal effect (categorical variable for primary sessions); 1 = constant (no behavioral or temporal effect).

those effects are also considered (Table 2.2). More complex models could in theory be fitted, such as different probability of detection for each secondary occasion within each trapping session. However, based on our experience with analysing this type of data on Bylot Island, these models are over parameterised and almost always produce systematic errors, which prevent them from converging correctly.

Here, we use the half-normal distribution for the detection probabilities (g_0), as recommended by Krebs et al. (2011) for lemming trapping. The buffer zone is fixed at 100 m because this corresponds to 3-4 times the sigma parameter (σ). All models use the maximum likelihood approach to estimate the parameters. This takes a certain amount of time (a few minutes to an hour for each model) before it is completed because many iterations may be required for models to converge when number of captures are high. We recommend David Anderson's book (Anderson 2008), especially chapter 2 and 3, for more information on the use of likelihood statistics because it is well written for biologists and other professionals who are not specialized in statistics.

2.3. Model selection and model-averaging density estimates

Once the *secr* script has been run (see Chapter 3 for an example), it is VERY IMPORTANT to eliminate all models with estimation problems. For example, models showing standard errors much higher than the parameter itself (e.g. standard error on density estimates being 10 times higher than the density estimate), with probability detection of 1 (or 100%), or showing movements being ridiculously high (e.g. >1,000 m) are erroneous and must be eliminated. All these errors will bias the final density estimate, which must be avoided. In some cases, algorithms of some models will not converge because of a local minimum during the likelihood estimations or problems with the data such as sample size being too small.

These problems can sometimes be easily detected when the second-order Akaike's Information Criterion (AICc; the criterion used for model selection; see details in Chapter 3) of a model is completely off compared to the other models (e.g. a model is probably incoherent when it has a difference in AICc 10 times higher than the second worst model). When this is the case, it is necessary to eliminate this problematic model from the list of models to consider or to use during the multi-model inference process (explained below).

If no model can provide reliable densities for a particular session because the standard error is unusually low or high or the sample size is too small (typically <5 individuals captured and <2 recaptures in total), it is preferable not to estimate densities with *secr* for this period and to eliminate all data from this session from the data file. In this case, it will be necessary to use the minimum number alive as an estimate of density (see text box).

Sometimes, the algorithm may not converge or the function is unable to start the iteration process. In this case, it may be wise to restart these models with different starting values

because it is possible that new starting values will allow the iterative procedure to reach its maximum likelihood. You can specify starting values with the argument (`start = c()`) as detailed in the following example. If we have three primary capture sessions, three starting values (one for each parameter D , g_0 and σ to be estimated) must be provided for each session for a total of nine. D and σ values must be between $[0, \infty]$ and g_0 values must be between $[0, 1]$. Values must be provided in this order: three values for D , three values for g_0 , and three values for σ . Here are the codes:

```
model.1<-secr.fit(cmr.data, model=list(D~session, g0~session, sigma~session),
  buffer=100, trace=T, start=c(2,1,3,0.1,0.3,0.2,15,20,25))
```

It is also possible to launch a null model to rapidly estimate starting values that could then be used in the `start = c()` argument:

```
model.null<- secr.fit(cmr.data, model=list(D~1, g0~1, sigma~1), buffer=100,
  trace=T, ncores=4)
model.1<-secr.fit(cmr.data, model=list(D~session, g0~session, sigma~session),
  buffer=100, trace=T, ncores=4, start=model.null)
```

* TIPS! – Minimum number alive.

The minimum number alive is defined as the number of individuals captured divided by the effective sampling area (ESA). This is obtained by using the effective sampling area estimated by the top model ($\Delta AIC_c = 0$), which can be extracted from the `secr.fit` object with the function `esa()`. The ESA is calculated in hectare and is the value used to divide the total number of individuals captured in order to obtain densities (individuals/ha) in those situations. If no ESA can be estimated for the current year, the mean ESA calculated for past years could be used. If this is the first time the population is sampled, the area covered by the trapping grid should be used as a last resort.

3. ESTIMATING LEMMING DENSITIES WITH SECR: AN EXAMPLE



3.1. Context of the example

We now present a case study conducted on Bylot Island. The data files (capture and trap files) as well as the file with the R codes used for this example are provided on our website (i.e. where this manual can be found). We live-trapped lemmings on several trapping grids each summer (see Fauteux et al. 2015 for details of the field methods). The example files contain the trapping data from 2 grids with 144 traps (12×12 , spaced out at 30-m interval), one located in wet habitat and one in mesic habitat, during summer 2016. There were 3 trapping sessions (June, July, and August) and at each sessions there was 6 secondary trapping occasions (traps were checked at 12-h intervals during 3 consecutive days). We included 5 covariates (i.e. *species*, *sex*, *age*, *mass*, and *repro*) to be specified in the loading codes. The covariate *species* defines which species was captured, *sex* is a binary variable defining the sex of the individual, *age* is also a binary variable indicating whether the individual is an adult or juvenile based on body mass, *mass* is the body mass of the individual, and *repro* is a five-category variable for non-reproductive individuals, abdominal males (scrotum well-developed but testes in abdomen), scrotal males (testes in scrotum), lactating females (i.e. mammaries are inflated), and pregnant females.

The 5 covariates are not used in the following analyses because our goal is to obtain a single density estimate per trapping session and grid. However, using these variables would make it possible to stratify the population according to sex, age, or any categorical variables to obtain specific densities. This can easily be done by creating one session (column *Session* in the capture file) per category for each monthly trapping session. For example, if we want to estimate densities for males and females separately for the months of June, July, and August, we would create 6 different trapping sessions instead of 3 (i.e. 3 sessions for males only and 3 sessions for females only). In the following example, 3 sessions are sufficient because we are interested in obtaining a single density estimate per monthly trapping session and grid.

3.2. R codes to estimate SECR densities with the Bylot Island example

The first steps are to open the R software, install (command `install.packages`) and load (command `library`) the packages available on the CRAN server. The package *secr* is used to conduct spatially-explicit capture-recapture analyses and the package *parallel* reduces computing time by using several cores of the computer's processor.

```
install.packages("secr")
install.packages("parallel")
library(secr)
library(parallel)
```

The next step is to load the files that were previously formatted (see section 2.1 and Figure 2.1 for more details). We will proceed with a first analysis where we want to obtain a single density estimate for the months of June, July, and August in our trapping grid located in the wet tundra habitat (file name `LemmusBylot_Wet2016_Captures.txt`). Note that the function to load the data file provided below is complete except for the pathname (e.g. `captfile="C:/Bylot/LemmusBylot_Wet2016_Captures.txt"`, `trapfile="C:/Bylot/LemmusBylot_Wet2016_Traps.txt"`) that must be specified according to where the file is stored on your computer hard drive. Note that R use the slash in pathnames "/" instead of backslash "\". Also, note that the argument `covnames=` should be verified to make sure that the names match the columns names in the capture text file. If an error occurs regarding the number of column names that do not match the number of covariates (or simply the number of columns), make sure that there is no space between the species names (e.g. change "Brown Lemming" for "BrownLemming" in the data file).

```
trap.wet<-read.caphist(captfile="LemmusBylot_Wet2016_Captures.txt",
  trapfile="LemmusBylot_Wet2016_Traps.txt", covnames=c
  ("species","sex","age","mass","repro"), detector="single", fmt="trapID")
```

There will be a warning that `Levels of factor covariate(s) differ between sessions` but this is normal and only concern covariates, which are not used in this example. Thus, we can ignore this message. Once the file has been loaded, we will run the 6 models proposed in Table 2.2. Those models will provide one density estimate for each monthly trapping session. They will also allow us to verify if probabilities of capturing lemmings and the amplitude of movements varied between monthly sessions and if previously captured individuals were easier/harder to recapture or moved more/less than new ones. Here are some information on the different codes used in the scripts below:

Command `secr.fit`

- This command calls the spatially-explicit capture-recapture likelihood algorithm used to estimate densities, detection probabilities, and movements.
- Arguments of the command are:
 1. `trap.wet`: name of the input object that was loaded with the `read.caphist` command (see above).
 2. `model`: a list object that specifies the model. For each of the 3 parameters (D = animal density; g_0 = capture probability; σ = amplitude of movements), we must specify the effects imposed on them. Tilds (~) are used to code 'as a function of'; `session` and `b` are keywords recognized by `secr` (see Table 2.2 for an explanation of those effects); plus signs

(+) are used to indicate additive effects and commas (,) separate the three parameters in the parenthesis.

3. `buffer`: distance used to estimate a buffer zone around traps in which home range centers of individuals can be found. In other words, it is the maximum distance moved from a trap when the capture probability is not 0. Here, we use 100 m (meters are the default units in *secr*) because lemmings rarely move beyond this distance within grids. This equals approximately 2-4 times sigma (Krebs et al. 2011).
4. `detectfn`: detection function used to estimate the capture probabilities, as the model assumes that this probability declines the further an animal is from a trap. The detection function is specified by a keyword; 0 = half-normal distribution, 1 = hazard-rate distribution, and 2 = exponential distribution. Here, we use the half-normal detection function.
5. `trace`: argument used to specify that partial results of each iteration are displayed on the screen during the iterative process. Use the keyword `TRUE` to activate this function.
6. `ncores`: the number of processor cores that should be used in the computer to make estimations. Typically, the number of cores specified should be a multiple of the number of sessions in the analysis (e.g. if we estimate densities for 3 or 6 sessions, then we should use 3 cores).

Note: lines starting with a # are comments. Comments are facultative and are not interpreted by R.

If we want to run all 6 models of Table 2.2 for the trapping grid found in the wet tundra habitat, we need to repeat the `secr.fit` command with its argument 6 times, one for each model. Note that running these 6 models could take several minutes to a few hours depending on the sample size of the dataset and speed of the computer processor. The R codes for these 6 models are as follows:

```
# Model 1: Density~session; detection probability~session; movements~session
secr.wet.1<-secr.fit(trap.wet, model=list(D~session,
g0~session,sigma~session), buffer=100, detectfn=0, trace=TRUE, ncores=3)

# Model 2: Density~session; detection probability~session+previous capture;
# movements~session
secr.wet.2<-secr.fit(trap.wet, model=list(D~session,
g0~session+b,sigma~session), buffer=100, detectfn=0, trace=TRUE, ncores=3)
```

```
# Model 3: Density~session; detection probability~session+previous capture;
# movements~session+previous capture
secr.wet.3<-secr.fit(trap.wet, model=list(D~session,
g0~session+b,sigma~session+b), buffer=100, detectfn=0, trace=TRUE, ncores=3)

# Model 4: Density~session; detection probability~session; movements~constant
secr.wet.4<-secr.fit(trap.wet, model=list(D~session,g0~session,sigma~1),
buffer=100, detectfn=0, trace=TRUE, ncores=3)

# Model 5: Density~session; detection probability~session+previous capture;
# movements~constant
secr.wet.5<-secr.fit(trap.wet, model=list(D~session,g0~session+b,sigma~1),
buffer=100, detectfn=0, trace=TRUE, ncores=3)

# Model 6: Density~session; detection probability~constant;movements~constant
secr.wet.6<-secr.fit(trap.wet, model=list(D~session,g0~1,sigma~1),buffer=100,
detectfn=0, trace=TRUE, ncores=3)
```

After running the models, the results of each of them should be looked at to assess whether errors occurred such as abnormally high/low values for densities, detection probabilities, movements or for their standard errors. If one or more model(s) has erroneous results, they should be eliminated from the model selection procedure that follows. Looking at the output of each model is simply done by calling the resulting object created by the `secr.fit()` command. Here, we will look at the output from model `secr.wet.1`:

```
>secr.fit(capthist = trap.wet, model = list(D ~ session, g0 ~
session, sigma ~ session), buffer = 100, detectfn = 0, trace = TRUE,
ncores = 3)
secr 3.1.0, 15:27:35 24 oct. 2017

$LG1PP1
Object class      traps
Detector type     single
Detector number   144
Average spacing   22.64703 m
x-range           -11.9 336.37 m
y-range           -8.5 342.74 m
(...)

```

In the first part of the output, the model that has been run and the date are presented, and information about the trapping session are displayed. `Object class` indicates what kind of detector was used, `Detector type` indicates if the traps were single-catch or multiple-catch, `Detector number` indicates the number of traps used, `Average spacing` indicates the mean minimum distance between traps, and the `x-` and `y-ranges` indicate the minimum and maximum positions of traps on both axes. In the same output, we then find more information about the animals trapped during each primary session, on the options that we specified for the model and the maximum likelihood and AICc values of the model:

```

>
      LG1PP1 LG1PP2 LG1PP3
Occasions      6      6      6
Detections     21     38     23
Animals        7     13      6
Detectors     144    144    144

Model          : D~session g0~session sigma~session
Fixed (real)   : none
Detection fn   : halfnormal
Distribution    : poisson
N parameters   : 9
Log likelihood : -341.2644
AIC            : 700.5288
AICc          : 711.7788

```

This part of the output is useful to detect any problems in the input data or in the model. For example, were there really 6 occasions for each primary sessions? Were there really 6 individuals captured in primary session LG1PP3 (e.g. if not, could there be mistakes in column ID)? The next information in the output are `Beta parameters`, `Variance-covariance matrix of beta-parameters`. We will not present this output here because those statistical details are not of direct interest to us. However, note that this information can be useful to detect problems of estimations when some estimates are really high or low. In the current example, the Beta parameter of `g0.sessionLG1PP3` has a high value (19.46) and an extremely low standard error (0.00000046). The consequence of this is that `g0` is estimated at 1 (i.e. at the boundary on the $[0,1]$ scale), which means that detection was perfect. This could cause potential estimation errors and to assess this, we can look at the parameters on the 'real' scale. These parameters are presented at the end of the output:

```

> Fitted (real) parameters evaluated at base levels of covariates

  session = LG1PP1
      link      estimate      SE.estimate      lcl      ucl
D      log      0.3990285      0.15915754      0.1879068      0.8473546
g0     logit     0.2230026      0.07397058      0.1105664      0.3985433
sigma  log      25.9477402      3.95414549      19.2810055      4.9196114

```

session = LG1PP2					
	link	estimate	SE.estimate	lcl	ucl
D	log	0.65219756	0.19176915	0.3709078	1.1468123
g0	logit	0.08602192	0.02310094	0.0502549	0.1434012
sigma	log	41.05998129	5.21574047	32.0424179	52.6153197
session = LG1PP3					
	link	estimate	SE.estimate	lcl	ucl
D	log	0.4212054	1.803255e-01	0.1884949	0.9412135
g0	logit	1.0000000	5.271669e-09	1.0000000	1.0000000
sigma	log	11.4153969	1.041246e+00	9.5501486	13.6449484

Even though the link functions are shown here, estimates are on the real scale: individuals/ha for D , probability of detection for g_0 , and movements in meters for σ . Most estimates shown here are plausible except for g_0 of session LG1PP3, which indicates perfect detection. This is a situation potentially caused by the complexity of the model (9 parameters), low sample size (6 individuals only) and the very high recapture rate (23 captures; individuals were captured 4 times on average during LG1PP3). Although this situation is unusual, it is not impossible. Moreover, D and σ estimates are also plausible, so we can keep this model for the next steps. Taking a look at the output of model `secr.wet.2`, we obtain:

```
> Fitted (real) parameters evaluated at base levels of covariates
```

session = LG1PP1, b = 0					
	link	estimate	SE.estimate	lcl	ucl
D	log	1.25875361	11.53261808	2.022798e-02	78.3301559
g0	logit	0.01869709	0.04838448	1.084447e-04	0.7699688
sigma	log	26.01009744	3.76709791	1.961101e+01	34.4972170
session = LG1PP2, b = 0					
	link	estimate	SE.estimate	lcl	ucl
D	log	4.190799344	62.17353809	4.411009e-02	398.1582776
g0	logit	0.004129858	0.01058066	2.678586e-05	0.3909935
sigma	log	39.128772628	4.51351448	3.123447e+01	49.0183036
session = LG1PP3, b = 0					
	link	estimate	SE.estimate	lcl	ucl
D	log	0.4439666	0.1951722	0.1947850	1.011918
g0	logit	0.5255861	0.2805761	0.1088039	0.909528
sigma	log	12.2219801	1.2621045	9.9879214	14.955744

In this case, we can see that D for sessions LG1PP1 and LG1PP2 are much higher compared to estimates from model `secr.wet.1` and have very high standard errors (~10 times D). Clearly, this model went through some problems when maximising the likelihood. Looking at the output of `secr.wet.3` reveals similar estimation problems. However, models `secr.wet.4`, `secr.wet.5`, and `secr.wet.6` all give plausible results with relatively small standard errors.

Overall, models 1, 4, 5, and 6 can be kept for the next step in which we will proceed with model selection and model averaging.

Once we have the results from our list of candidate models without error, they should be compared to determine which one provides the best fit to our data. This can be done with the AICc. Models having the lowest AICc are considered to be those providing the best compromise between complexity and precision of the estimates and are preferred. This information can be obtained with the following statement:

```
AIC(secr.wet.1, secr.wet.4, secr.wet.5, secr.wet.6)
```

>	model	detectfn	npar	logLik	AIC	AICc	dAICc	AICcwt
secr.wet.1	D~session g0~session sigma~session	halfnormal	9	-341.2644	700.529	711.779	0.000	0.9603
secr.wet.5	D~session g0~session + b sigma~1	halfnormal	8	-346.8394	709.679	718.149	6.370	0.0397
secr.wet.6	D~session g0~1 sigma~1	halfnormal	5	-364.5324	739.065	742.065	30.286	0.0000
secr.wet.4	D~session g0~session sigma~1	halfnormal	7	-363.2599	740.520	746.742	34.963	0.0000

Information found in this table are model names (`model`), the detection function (`detectfn`), the number of parameters (`npar`), the log-likelihood value (`logLik`), the AIC and the AICc values, the Δ AICc (`dAICc`), and the AICc weight (`AICcwt`). In this table, models will always be ordered from the model with the lowest AICc value first to the model with the highest AICc value last. Indeed, AICc values are related to the information lost by the model (the lowest the AICc, the closest to reality is the model). If difference in AICc (`dAICc` or Δ AICc) between the first model (i.e. the one with the lowest AICc value) and the second one is >4 or if the top model has an AICc weight of >0.90 (`AICcwt`), this means that there is little ambiguity in model selection. In this situation, the first model is preferred and parameters estimates of this model alone are retained. However, it will often be the case that 2 or more models will have Δ AICc values <4 . This indicates that several models are considered competitive and in some cases (e.g. if Δ AICc <2 among the top models), models can be considered close competitors. In such situations, it is recommended to use the model-averaging procedure, which adjust parameter estimates from the models and their variance to take into account the uncertainty in model selection (see text box at the end of this chapter). In the example above, model `secr.wet.1` is the top model and the second best model is `secr.wet.5` with a Δ AICc of 6.37. Since the second best model has a Δ AICc of >4 , the procedure can stop here and we use the output of model 1 as the final results.

Now, let's run the same codes for the second trapping grid and see if the results are similar and if models perform well in this situation. We provide below the R codes for this second example.

```

### Models for trapping grid 2 located in the mesic habitat. Load file first.

trap.mesic<-read.caphist("Captures_mesic.txt", "Traps_mesic.txt", covnames=c
("species","sex","age","mass","repro"), detector="single", fmt="trapID")

# Model 1: Density~session; detection probability~session; movements~session
secr.mesic.1<-secr.fit(trap.mesic, model=list(D~session, g0~session,
sigma~session), buffer=100, detectfn=0, trace=TRUE, ncores=3)

# Model 2: Density~session; detection probability~session+previous capture;
# movements~session
secr.mesic.2<-secr.fit(trap.mesic, model=list(D~session, g0~session+b,
sigma~session), buffer=100, detectfn=0, trace=TRUE, ncores=3)

# Model 3: Density~session; detection probability~session+previous capture;
# movements~session+previous capture
secr.mesic.3<-secr.fit(trap.mesic, model=list(D~session, g0~session+b,
sigma~session+b), buffer=100, detectfn=0, trace=TRUE, ncores=3)

# Model 4: Density~session; detection probability~session; movements~constant
secr.mesic.4<-secr.fit(trap.mesic, model=list(D~session, g0~session,
sigma~1), buffer=100, detectfn=0, trace=TRUE, ncores=3)

# Model 5: Density~session; detection probability~session+previous capture;
# movements~constant
secr.mesic.5<-secr.fit(trap.mesic, model=list(D~session, g0~session+b,
sigma~1), buffer=100, detectfn=0, trace=TRUE, ncores=3)

# Model 6: Density~session; detection probability~constant;
# movements~constant
secr.mesic.6<-secr.fit(trap.mesic, model=list(D~session, g0~1, sigma~1),
buffer=100, detectfn=0, trace=TRUE, ncores=3)

```

As explained above, we need to look at the AICc and the Δ AICc to assess model support and determine whether some models should be eliminated or not from the model-averaging step. In this case, all models gave plausible D and g_0 estimates without aberrant standard errors, therefore all models are retained.

```

AIC(secr.mesic.1, secr.mesic.2, secr.mesic.3, secr.mesic.4, secr.mesic.5,
secr.mesic.6)

```

The output obtained is:

model	detectfn	npar	logLik	AIC	AICc	dAICc	AICcwt
secr.mesic.2 D~session g0~session + b sigma~session	halfnormal	10	-1038.218	2096.435	2098.993	0.000	0.4704
secr.mesic.5 D~session g0~session + b sigma~1	halfnormal	8	-1041.220	2098.441	2100.077	1.084	0.2736
secr.mesic.3 D~session g0~session + b sigma~session + b	halfnormal	11	-1037.551	2097.103	2100.209	1.216	0.2561
secr.mesic.1 D~session g0~session sigma~session	halfnormal	9	-1060.800	2139.600	2141.669	42.676	0.0000
secr.mesic.4 D~session g0~session sigma~1	halfnormal	7	-1064.895	2143.789	2145.048	46.055	0.0000
secr.mesic.6 D~session g0~1 sigma~1	halfnormal	5	-1070.151	2150.301	2150.961	51.968	0.0000

Here, the output indicates that more than one model have good statistical support and that two models (`secr.mesic.5` and `secr.mesic.3`) have a $\Delta AICc$ of <4 . This indicates that we should proceed with model averaging. A simple line of code provided by the `secr` package does the dirty job, or more precisely the calculations with the delta method, for us. **Do not forget that erroneous models should not be included in this calculation** (not the case here). Here, we specify `betanames = "D"` in the argument to ask the model to return the model-averaged estimates of density, and not of the detection probability (`g0`) or movements (`sigma`). We also indicate in the argument the name of the models to include in the procedure. In this case, all models with $\Delta AICc$ of <4 were included for model-averaging:

```
model.average(secr.mesic.2, secr.mesic.3, secr.mesic.5, realnames = "D")
```

Running this command gives the following output:

	estimate	SE.estimate	lcl	ucl
session=LG2PP1,b=0	3.209448	0.7804074	2.006402	5.133845
session=LG2PP2,b=0	2.412587	0.5048589	1.607923	3.619934
session=LG2PP3,b=0	2.504588	0.5444891	1.643691	3.816388

In this output, we can find the estimates on the real scale (individuals/ha), their standard errors as well as their respective confidence intervals. Because these results are based on models that had no aberrant values or estimation problems, they can be considered final. Compare model-averaged densities obtained for each trapping grid and each session. Did density increase or decline during summer? Are there any major differences between trapping grids? Feel free to

explore this example in details to familiarise yourself with the results and see how it would apply to your own study system.

*** TIPS! – Model averaging density estimates**

In the model-averaging procedure, we do a weighted average of the parameter estimate of interest across several models using the AICc weight of each model as the weighing factor. Thus, the highest the AICc weight of a model, the more weight the parameter value from this model will have in the average. The standard errors are also adjusted to reflect not only the uncertainty associated with the estimate in each model but also the uncertainty associated with the model selection procedure itself (see Burnham and Anderson 2002 for more details). The end-result is that model-averaged standard errors will be inflated.

Whether model-averaging should be applied or not is currently a research topic in the biostatistical literature and we prefer to warn readers about this debate. However, we believe that the advantages of this method outweigh its potential disadvantages in the sense that each hypothesis (or model) can be true, but some are more statistically supported than others. Model-averaging accounts for this uncertainty based on the principle of parsimony and helps us to remain objective and prudent in our inferences. We therefore recommend its use.

4. ESTIMATING WINTER NEST DENSITIES WITH DISTANCE SAMPLING METHODS



As explained in the introduction, one has sometimes to rely on indirect methods to survey small mammals and obtain abundance indices. These alternative methods rely on counting signs left by the presence of animals such as scats, tracks, burrows or nests made of vegetation built by small mammals under the snow in winter (refer to as winter nests). When counting these signs of animal presence in the field, it is often not possible to find all of them, which raises again the problem of imperfect detection as in the case of live-trapping data. A statistical method called line-transect (or distance sampling) has been developed to address this problem of imperfect detection. This method can be applied to sample the abundance of all sorts of signs left by animals but in the context of small mammals and especially lemmings, it is most often applied to the sampling of their winter nests. As with the SECR method described previously, distance sampling requires the use of advanced statistical models and some assumptions must be met (see text box below). In this section, we explain how to apply this statistical method to sample winter nest density and we propose a standardized approach to perform such analysis.

The basic principle of distance sampling consists of walking along a straight, geolocated line (i.e. the transect) in the field and noting every object that is seen on either side of the transect (Figure 4.1). Obstacles such as uneven topography, shrubs, or bad weather may prevent the observer from seeing an object, which would lead to a false absence. Once an object is detected, the perpendicular distance from the transect is measured. The observer then returns to the transect and continue walking along it. It is important to record ONLY objects detected while walking along the transect line itself and NOT those detected while walking off this line, such as when measuring the distance from the transect and a previously-detected object. Distance sampling assumes that the probability of detecting an object decreases with distance from the transect according to a probability distribution (e.g. half-normal, hazard-rate, uniform). The maximum perpendicular distance measured between an object and the transect is used to estimate the area surveyed by the observer. Then, the number of undetected object present in this surveyed area is estimated as a function of all observed objects and their distances to the transect, which are used to model detection probabilities over the area surveyed. The sum of detected and undetected features divided by the area surveyed gives a density per unit area.

TIPS! – Assumptions of distance sampling models

Before applying distance sampling models to sample signs of animal presence, it is important to verify that the following assumptions are met:

- All objects present on the transect line itself are detected;
- Objects do not move during sampling (i.e. no double-counting);
- Distances measured between the objects and the transect line are exact.

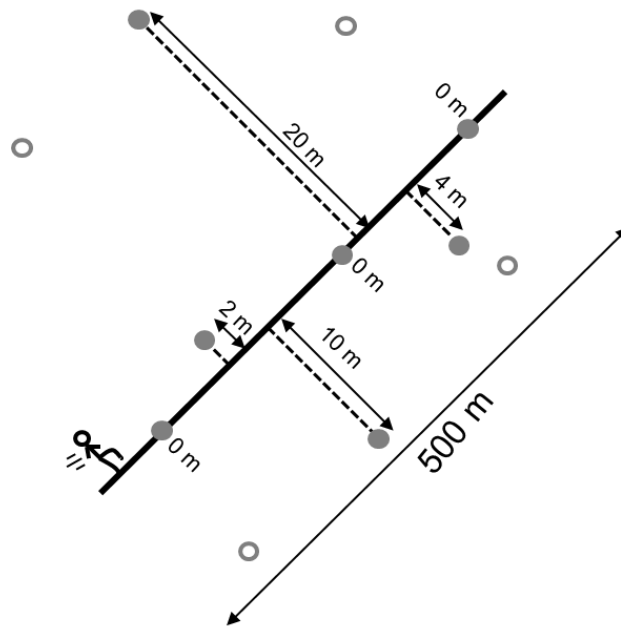


Figure 4.1. Schematic view of a line transect sampling. The thick line represents the transect. The solid dots are objects detected by the observer when walking along the transect while open dots were not detected either because of the uneven topography or their distance from the transect. The dashed line represents the perpendicular distance from each element to the transect line.

There are two software to analyze data obtained with the line-transect method. *DISTANCE* was developed many years ago and can still be used without problem because it is maintained up to date by the development team. However, an R package called *Distance* has also been developed to perform the same analyses (Thomas et al. 2009, Miller 2016). Because it is faster to program in R with lines of code than in *DISTANCE* and because we can benefit from other packages available in R, we recommend this option. Both applications were developed by the same team and use the same algorithms. We provide useful websites to download the softwares and get help in Table 4.1. In this manual, we focus our explanations on how to use the package *Distance* in R.

Tableau 4.1. Websites useful to download the softwares and obtain help. Instructions provided in the websites should be followed to correctly install the softwares (checked in September 2017).

Software	Website
<i>DISTANCE</i>	http://distancesampling.org/
R	http://www.r-project.org/
<i>Distance</i>	https://cran.r-project.org/web/packages/Distance/index.html
Help Forum	https://groups.google.com/forum/#!forum/distance-sampling

4.1. How to format winter nest datasets for *Distance*

It is important to correctly format your dataset so that the *Distance* package can recognize it. To import the data into R, it is possible to prepare only one text file per year even if your data are stratified by habitat, sites or other variables (as opposed to several files with the *secr* packages when the data are separated by habitat or species; see Chapter 3). Each line in the data file corresponds to one winter nest detected along one transect and the information associated with that nest. As with *secr*, you can prepare your data in an Excel file and export it in a text file. Each column must be separated by a tabulation in the text file. Note that the order of the columns and the column titles are important and must be EXACTLY the same as in the example file because these titles correspond to the names of the variables in the program codes that follow. We show in Figure 4.2 an example data file that comes from the sampling of lemming winter nests on Bylot Island. In the text file provided in R, it is necessary to include the following columns with the names written EXACTLY as follows:

1. *Region.Label*: The name of the site (hereafter called region) and of the species. On Bylot Island for instance, we sampled winter nests for two species (brown and collared lemmings). This column will tell the software to estimate density separately for each different label recognized in this column (in our case, for each lemming species in each region). In other words, the regions can be anything that requires a different estimation of density.
2. *Area*: Size of the surveyed area if known. Here we specify 0 because the size of the surveyed area on either side of the transects is unknown *a priori*. In this case, the software will estimate the surveyed area based on the longest distance separating a nest from the transect along which it was found.
3. *Sample.Label*: The name/number of the transect along which the nest was found. This implies that each transect has a unique identifier. For example in the Bylot dataset, the first transect made in the mesic habitat of region C2 was named "C2M01".
4. *Effort*: Length of the transect along which the nest was found in meters. The length of the transect is usually estimated from the GPS tracks recorded in the field. A pre-determined theoretical length (e.g. 500 m) can be used for straight transects. Transects can be non-linear (e.g. when following a stream bank), but require that their length be estimated from GPS tracks.
5. *distance*: Perpendicular distance between a nest found and the transect in meters. This distance is measured directly in the field with a measuring tape or afterwards according to GPS positions and geographic analysis systems. Here, "NA" is noted for transects where no nest has been sampled.
6. *habitat*: Habitat type associated with each transect.

7. *stratum*: Name of the small mammal species using the nest. On Bylot Island, two species are present, brown and collared lemmings, and it is possible to distinguish in the field which species used the nest (here, unlike the file used for *secr*, there can be spaces in labels without creating new columns). Because we already specify in the *Region.Label* that the data is stratified by region and species, this column is facultative.
8. *Nest_number*: A unique identifier for each nest. Here "NA" is used for transects where no nest was sampled. Note that only RECENT nests (i.e. those built during the previous winter) are used to estimate densities because nests older than 1 year are not included in the estimates. Old nests must be removed from the text file before the analyses if present. Typically, the *Nest_number* is unique in the data file. For example, a nest found in transect 2 of habitat mesic in the GP region would be named GPM02-01.
9. *Year*: Year of sampling. This column is facultative because one text file per year is usually constructed. Having this column or not in the data file will have no consequence on the analyses as long as a year covariate is not specified in the R codes.

	A	B	C	D	E	F	G	H	I	J
1	Region.Label	Area	Sample.Label	Effort	distance	habitat	region	stratum	Nest_number	Year
2	C2_brown	0	C2M01	500	2	mesic	Camp2	Brown Lemming	C1M01-02	2016
3	C2_brown	0	C2M02	500	NA	mesic	Camp2	Brown Lemming	NA	2016
4	C2_brown	0	C2M03	500	NA	mesic	Camp2	Brown Lemming	NA	2016
5	C2_brown	0	C2M04	500	NA	mesic	Camp2	Brown Lemming	NA	2016
6	C2_brown	0	C2M05	500	3	mesic	Camp2	Brown Lemming	C2M05-01	2016
7	C2_brown	0	C2M05	500	6	mesic	Camp2	Brown Lemming	C2M05-02	2016
8	C2_brown	0	C2M05	500	0	mesic	Camp2	Brown Lemming	C2M05-03	2016
9	C2_brown	0	C2M06	500	NA	mesic	Camp2	Brown Lemming	NA	2016
10	C2_brown	0	C2M07	500	NA	mesic	Camp2	Brown Lemming	NA	2016
11	C2_brown	0	C2M10	500	NA	mesic	Camp2	Brown Lemming	NA	2016
12	C2_brown	0	C2M10	500	6.8	mesic	Camp2	Brown Lemming	C2M10-01	2016
13	C2_brown	0	C2M10	500	4	mesic	Camp2	Brown Lemming	C2M10-02	2016
14	C2_brown	0	C2M10	500	2.6	mesic	Camp2	Brown Lemming	C2M10-03	2016

Region.Label	Area	Sample.Label	Effort	distance	habitat	region	stratum	Nest_number	Year
C2_brown	0	C2M01	500	2	mesic	Camp2	Brown Lemming	C1M01-02	2016
C2_brown	0	C2M02	500	NA	mesic	Camp2	Brown Lemming	NA	2016
C2_brown	0	C2M03	500	NA	mesic	Camp2	Brown Lemming	NA	2016
C2_brown	0	C2M04	500	NA	mesic	Camp2	Brown Lemming	NA	2016
C2_brown	0	C2M05	500	3	mesic	Camp2	Brown Lemming	C2M05-01	2016
C2_brown	0	C2M05	500	6	mesic	Camp2	Brown Lemming	C2M05-02	2016
C2_brown	0	C2M05	500	0	mesic	Camp2	Brown Lemming	C2M05-03	2016
C2_brown	0	C2M06	500	NA	mesic	Camp2	Brown Lemming	NA	2016
C2_brown	0	C2M07	500	NA	mesic	Camp2	Brown Lemming	NA	2016
C2_brown	0	C2M10	500	NA	mesic	Camp2	Brown Lemming	NA	2016
C2_brown	0	C2M10	500	6.8	mesic	Camp2	Brown Lemming	C2M10-01	2016
C2_brown	0	C2M10	500	4	mesic	Camp2	Brown Lemming	C2M10-02	2016
C2_brown	0	C2M10	500	2.6	mesic	Camp2	Brown Lemming	C2M10-03	2016

Figure 4.2. Excel file format (top) and text file (bottom, tab-delimited) used by the *Distance* package in the R software. Note that the column headings must be exactly as in the example so that the software detects the required information correctly.

*** WARNING! – Transects sampled but with no nest**

It is extremely important that **ALL transects** have been entered in the data file **FOR BOTH SPECIES, even those where no nest has been found** because the total effort must be taken into account. Such transect should be entered only once (single line in the data file) with the required information.

4.2. Data analysis with the package *Distance* in R

To estimate winter nest densities, it is assumed that the probability of detection is 100% for nests located directly on the transect line (column *distance* = 0 m) and that it decreases further away from the transect. The most common probability distributions that are used to estimate the detection function is the hazard-rate or half-normal. However, the latter detection function generally adjusts poorly (based on chi-square tests) to the data when a covariate is used (e.g., habitat, region). The uniform distribution, which assumes a constant probability of detection, is also possible but is generally avoided because it considers that all nests are equally detectable regardless of the perpendicular distance (although an adjustment factor can change that to some extent, see below) and it cannot be used with covariates. This distribution can sometimes be useful when the sample size is very small. To learn more about probability distribution functions, we refer readers to Chapter 5 of the book on distance sampling by Buckland et al. (2015).

In sampling distance models, an adjustment factor should also be specified. Possible factors are the cosine, the Hermite polynomial and the simple polynomial. These factors allow the model to be adjusted to situations where the probability of detection does not exactly follow the specified distribution (e.g. hazard-rate). For example, nests located within 5 meters may be highly visible from the transect but detection may sharply decline when nests are farther than 5 meters. The adjustment term allows a certain flexibility to better fit the observations.

It is necessary to specify in the models that we wish to estimate a density of nests per hectare. To do this, a conversion factor of 0.0001 must be included in the models because original data in the data file (for length of the transect and perpendicular distance to the transects) are in meters and thus the total area estimated by the model is by default in square meters. Other conversion factors can be used if other units are used in the data file.

Analyses can be repeated for different subgroups in the data (e.g. if one wishes to estimate nest density of each species in each study region separately). This can be easily achieved by using different labels in the *Region.Labels* column of the data file or by using covariates. When specifying several labels, a separate density estimate will be obtained for each different subgroup specified in that column. Habitat is a frequent covariate to consider because both the density of winter nests and the detection function will frequently vary across habitats. This

configuration adjusts different detection probability for each habitat and allows simultaneous estimation of densities in all regions and for each lemming species, one habitat at a time. We recommend testing a minimum of three models for each level of the covariate (here habitat), one using the half-normal detection function, one using the hazard-rate function, and one using the uniform function. This list is in no way exhaustive but represents simple and broadly applicable models.

After the models have been run, it is necessary to check the fit of each model to the data with goodness of fit tests. The software performs three fit tests and allow visual check of the adjustment on a graph combining the values predicted by the selected detection function and the observed data. The first test is a simple chi-square, which checks the fit by comparing the predicted and observed values. This test is the most useful for winter nest sampling whereas the other two are more specific. The Kolmogorov-Smirnov test verifies the fit using the greatest deviation obtained between values of the predicted relationship and the observed values. This deviation is used as a statistical value for the test. Finally, the Cramer-von Mises test uses the sum of positive and negative deviations between values of the predicted relationship and the observed values. The problem with the last two tests is that sometimes the number of nests found on the transect line itself (variable *distance* = 0) is disproportionate compared to the nests found away from the line, a situation that can greatly inflate the values of the Kolmogorov-Smirnov and Cramer-von Mises statistics. Thus, although all tests can be considered, the chi-square test is the most useful here.

Table 4.2 summarizes the detection functions, adjustment factors and fit tests available in the package *Distance* and their order of preference when estimating winter nest densities. Even though the combination of the hazard-rate function with a cosine adjustment usually performs best, it is recommended to adjust models with other detection functions also. Based on past experience, we recommend the cosine adjustment term because it is often the only term that increases model fit. We also do not recommend omission of adjustment terms because uniform detection functions cannot be used without it. Furthermore, the command `ds` in R (see below)

Table 4.2. Detection functions, adjustment factors and fit tests available in the package *Distance* to model the probability of detection in order of preference. Note that the preference order applies to each column separately (i.e. they are in no way orders of combined scenarios). This order is suggested because the model using the combination of preferred options generally performs better for winter nests on Bylot Island. Keywords for detection functions and adjustment factors to be used in R are in parentheses.

Preference order	Detection function	Adjustment factor	Fit test
1	Hazard-rate (<code>hr</code>)	Cosine (<code>cos</code>)	Chi-square
2	Half-normal (<code>hn</code>)	Polynomial (<code>poly</code>)	Kolmogorov-Smirnov
3	Uniform* (<code>unif</code>)	Hermite (<code>herm</code>)	Cramer-von Mises

* The uniform distribution does not allow inclusion of covariates and requires an adjustment factor.

automatically tests whether adding the adjustment factor increases the model fit or not when using the half-normal or hazard-rate detection functions. When alternative models are used, we use the AICc value to determine which one provides the best fit. Density estimates generated by all these models should always be checked because, although the adjustment may sometimes improve when changing detection function or adjustment factor, there is still a risk of obtaining aberrant densities or standard errors (i.e. far too high). If this happens, it is necessary to eliminate this(ese) model(s) from the selection.

*** TIPS! - Estimation problems in the candidate models**

Sometimes, all models with one or more covariates will give erroneous parameters, especially when sample size is too small (e.g. non-convergence, extremely large standard errors, failed fit tests). In this case, only the null model can be considered, but it should still be fitted with different probability distributions (half-normal, hazard-rate, uniform) and the best model selected based on the AICc values.

5. ESTIMATING WINTER NEST DENSITIES WITH DISTANCE: AN EXAMPLE



5.1. Context of the example

We now present a case study conducted on Bylot Island. The data file as well as the file with the R codes used for this example are provided on our website (i.e. where this manual can be found). On Bylot Island, we are using the line transect method to sample winter nests of brown and collared lemmings annually in 3 habitats: wet polygons, mesic tundra, and along stream gullies in mesic habitat. Nests are sampled in 5 regions (sites), each separated by ~10 km from each other. Transects are 500-m long, except in a few cases where topography (e.g. a lake) restricted their length, are randomly located in each region, are permanent (i.e. the same transects are sampled every year), and are sampled as soon as possible after snow-melt. The number of transects is variable across regions due to logistic constraints (Table 5.1) and not all habitats are present in each region. The example used in this chapter comes from the year 2016.

Table 5.1. Number of 500-m transects used to sample brown and collared lemming winter nests on Bylot Island according to the region and habitat in summer 2016.

Region	Habitat	Number of transects
Camp 1	Wet	20
	Mesic	23
	Stream	17
Camp 2	Mesic	8
	Stream	10
Camp 3	Mesic	15
	Stream	5
Dufour Point	Mesic	8
	Stream	5
Goose Point	Mesic	6
	Stream	1

5.2. R codes to estimate winter nest densities with the Bylot Island example

The first step is to open the R software and load the packages available on the CRAN servers. The package required to conduct the distance sampling analyses is *Distance*. We also need to load the package *MuMIn* so we can easily obtain the AICc values for our models.

```
library(Distance)
library(MuMIn)
```

In contrast with the SECR analyses, we load a single data file for the whole analysis (LemmingsBylot_NestTransects2016.txt). Note that the function to load the data file for distance sampling is simpler than with *secr* (see previous sections) as we need to load only one table in which the columns are separated by tabulations. Moreover, the names of covariates (or columns) do not have to be specified with the loading command `read.delim`, which is again different than for *secr* analyses, as it takes the first line of the text file to create the column names. The function provided below is complete except for the pathname (e.g. `file="C:/Bylot/LemmusBylot_NestTransects2016.txt"`) that must be specified according to where the file is stored on your computer hard drive. The data file is loaded with the following command:

```
neests2016 <- read.delim(file = "../LemmusBylot_NestTransects2016.txt")
```

In R, one distance sampling model can be written with a single code line with the command `ds` followed by several arguments in parentheses: the subsamples based on one or several covariates such as `habitat` (`data`), the conversion units (`convert.units`), the adjustment term (`adjustment`), and the probability function (`key`). In this example, because we have sampled winter nests in three habitats, we are interested in adjusting separate detection probabilities for each of them, hence we create subsamples based on this covariate. We first start with the mesic habitat. Note that we assume that the probability of detecting a winter nest in a given habitat is the same for both species and all regions and this probability is thus modelled jointly. Indeed, we have no reason to believe that this probability differs between the two species because their nests are very similar and species are identified *a posteriori* based on faeces present in the nest (Duchesne et al. 2011). Moreover, the mesic and stream gully habitat are fairly homogeneous throughout our study area.

We run a first set of three models, each one with a different detection function but the same adjustment factor (see Table 4.2), for the mesic tundra habitat:

```
mod.2016.hn.mesic <- ds(data=neests2016[neests2016$habitat=="mesic",],
  convert.units=0.0001, adjustment = "cos", key="hn")
mod.2016.hr.mesic <- ds(data=neests2016[neests2016$habitat=="mesic",],
  convert.units=0.0001, adjustment = "cos", key="hr")
mod.2016.un.mesic <- ds(data=neests2016[neests2016$habitat=="mesic",],
  convert.units=0.0001, adjustment = "cos", key="un")
```

During the calculation process, no error occurred for the first two models but one occurred for the third model using the uniform detection function. However, the message given by R

indicates that only the last attempt to adjust the algorithm did not converge; the other attempts did converge and the one with the lowest AICc was retained. To see if the models fitted the data and determine whether the results are statistically sound, we use the command `ds.gof`. The output from this command gives the results of the goodness-of-fit tests, the chi-square, Kolmogorov-Smirnov, and Cramer-von Mises tests. We have to repeat the command for each model. We start with the model using the half-normal function:

```
ds.gof(mod.2016.hn.mesic)
```

The output of this command is as follows:

```
> Goodness of fit results for ddf object

Chi-square tests
(...)

P = 0.036364 with 8 degrees of freedom

Distance sampling Kolmogorov-Smirnov test
Test statistic = 0.12389 P = 0.062293

Distance sampling Cramer-von Mises test (unweighted)
Test statistic = 0.14045 P = 0.42005
```

In the output, we see the results, the test statistics and p-values, from the three tests. As stated above, we focus on the chi-square test because it is more appropriate in the case of winter nests (see section 4.2). Here, the p-value is <0.05 indicating that expected values are statistically different than the observed values, which is not good. The goodness-of-fit tests for the hazard-rate model give the following results:

```
ds.gof(mod.2016.hr.mesic)
```

```
> Goodness of fit results for ddf object

Chi-square tests
(...)

P = 0.09012 with 8 degrees of freedom

Distance sampling Kolmogorov-Smirnov test
Test statistic = 0.12389 P = 0.062293

Distance sampling Cramer-von Mises test (unweighted)
Test statistic = 0.14505 P = 0.40502
```

The hazard-rate model gives a better fit as revealed by the p-value >0.05 . Repeating the fit tests for the third model (uniform detection function) reveals that this model has a poor fit (not shown). Thus, the hazard-rate model is the only model providing a good fit to the data and should be preferred. It is also possible to have a visual assessment of the fit of the detection function by using the command `plot` on the model itself:

```
plot(mod.2016.hr.mesic)
```

The resulting figure (Figure 5.1) shows that the detection function appears to follow well the frequencies of distances where nests were found.

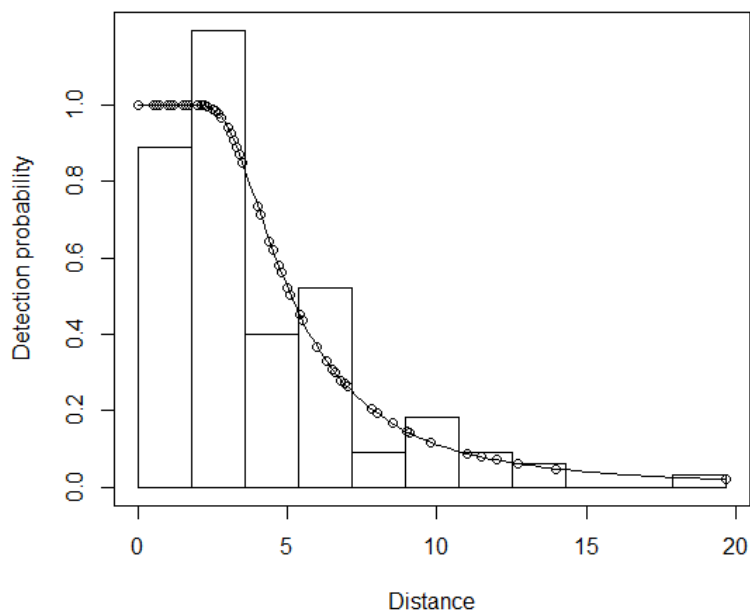


Figure 5.1. Hazard-rate detection function curve for the mesic tundra habitat on Bylot in 2016 and frequency distribution of winter nests found in relation to their distance from the transect line. This model has the highest fit for this habitat.

The next step is to look at the output of the model to determine if it yielded coherent results. To do this, the command `summary` gives a short description of the results and presents the essential information such as the coefficients and the density estimates with their standard errors. When more than one model fits the data well, summary outputs of all models should be checked to ensure that they all have converged properly and that results are coherent (i.e. density estimates or their standard errors are not abnormally large). Problematic models should be identified and eliminated right away.


```
summary(mod.2016.hr.mesic)
```

This command gives the following output:

```
> Summary for distance analysis
Number of observations : 113
Distance range       : 0 - 19.7

Model : Hazard-rate key function
AIC   : 539.9559

Detection function parameters
Scale coefficient(s):
      estimate      se
(Intercept)  1.493797  0.1492135

Shape coefficient(s):
      estimate      se
(Intercept)  0.9684474  0.1894533

Average p          Estimate      SE      CV
N in covered region 359.407850  44.66859548  0.12428386

Summary statistics:
      Region      Area CoveredArea Effort  n  k      ER      se.ER      cv.ER
1  C1_brown  44.89236  44.89236  11394  23  23  0.0020186063  0.0004772357  0.2364184
2  C1_collared  44.89236  44.89236  11394  19  23  0.0016675443  0.0005336011  0.3199922
3  C2_brown  44.89236  15.76000  4000  7  8  0.0017500000  0.0009589801  0.5479886
4  C2_collared  44.89236  15.76000  4000  7  8  0.0017500000  0.0007007649  0.4004371
5  C3_brown  44.89236  29.55000  7500  2  15  0.0002666667  0.0001817027  0.6813851
6  C3_collared  44.89236  29.55000  7500  23  15  0.0030666667  0.0006723567  0.2192467
7  GP_brown  44.89236  12.52526  3179  10  6  0.0031456433  0.0013555251  0.4309214
8  GP_collared  44.89236  11.82000  3000  18  6  0.0060000000  0.0017888544  0.2981424
9  PD_brown  44.89236  15.76000  4000  0  8  0.0000000000  0.0000000000  0.0000000
10 PD_collared  44.89236  15.76000  4000  4  8  0.0010000000  0.0005345225  0.5345225
      Total  448.92360  236.26998  59967  113  120  0.0018843697  0.0002389701  0.1268170

Density:
      Label      Estimate      se      cv      lcl      ucl      df
1  C1_brown  1.6295381  0.4163237  0.2554857  0.9749521  2.723615  29.836424
2  C1_collared  1.3461402  0.4500504  0.3343266  0.6896010  2.627742  26.171421
3  C2_brown  1.4127032  0.7861422  0.5564807  0.4197923  4.754090  7.443646
4  C2_collared  1.4127032  0.5820082  0.4119820  0.5650875  3.531719  7.841154
5  C3_brown  0.2152691  0.1481553  0.6882332  0.0568960  0.814482  14.570606
6  C3_collared  2.4755943  0.5933603  0.2396840  1.5118233  4.053759  19.900822
7  GP_brown  2.5393488  1.1215547  0.4416702  0.8839485  7.294874  5.517217
8  GP_collared  4.8435540  1.5183454  0.3134775  2.2972035  10.212423  6.107774
9  PD_brown  0.0000000  0.0000000  0.0000000  0.0000000  0.000000  0.000000
10 PD_collared  0.8072590  0.4385234  0.5432251  0.2462322  2.646555  7.466621
      Total  1.6682110  0.2765594  0.1657820  1.1973701  2.324200  45.084804
```

First, a short description of the dataset is presented such as the Number of observations (here, number of nests found) and the Distance range in which nests were found. This information is important to look at to make sure typing mistakes did not occur while entering

data in the original Excel data file. Below this, the type of detection function (`Model`) is specified, followed by the `AIC`. This `AIC` is not the one we are interested in because it is not corrected for small sample size. Instead, we use the `AICc` command, as shown later in this example. Next are the statistics for the detection function. The scale coefficient is related to the slope of the function and the shape coefficient is used only in hazard-rate functions to prolong the shoulder of the detection probabilities at very short distances from the transect (i.e. close to 0 m). If cosine adjustments were retained by the model, the coefficients will be shown below the scale coefficients. We then find the `Average p`, which is the average detection probability in the area surveyed along with its standard error. The `N in covered region` is simply the total number of nests in all subgroups (i.e. all labels specified in the `Region.Label` column) combined divided by the probability of detection (i.e. $113 / 0.314$).

The `Summary statistics` table gives plenty of raw information for each subgroup previously specified in the `Region.Label` column. In our case, this yields estimates for each lemming species in each region because each label corresponds to a unique combination of species and region in that column. The line `Total` corresponds to the summary statistics for all subgroups combined. Here is a description of the information contained in each column of the table:

1. `Region`: The different labels (i.e. subgroups) that were entered in the `Region.Label` column of the data file.
2. `Area`: maximum area in hectares covered for each combination of region and species. In the data file, there were only zeroes specified in the column `Area` because we do not wish to estimate the nest abundance in an area larger than the sampled area (`CoveredArea`). In our case, we don't use this information.
3. `CoveredArea`: Total area in hectares surveyed according to the `Effort`, as well as the maximum `Distance range` (in this case 19.7 m, see above) to which a nest was observed. In this case, we can obtain the `CoveredArea` of a subgroup by multiplying 19.7 by 2 (left and right of transect), multiplied by the `Effort` of the subgroup, and divided by 10,000 to transform units in hectares.
4. `Effort`: Sum of transect lengths in meters specified in the column `Effort` of the data file.
5. `n`: Number of nests found.
6. `k`: Number of transects completed.
7. `ER`: Encounter rate of nests, which is `n` divided by `Effort`.
8. `se.ER`: Standard error for `ER`. This variability stems from the different number of nests found per transect.
9. `cv.ER`: Coefficient of variation for `ER`, which is `ER` divided by `se.ER`.

The last section of the output, `Density`, gives, for each subgroup, the density estimates on the real scale (nests/ha; column `Estimate`) along with their standard error (`se`), coefficient of variation (`cv`), lower (`lcl`) and upper (`ucl`) limits of the 95% confidence interval, and the degrees of freedom (`df`). At this stage, winter nest densities obtained with the model using the hazard-rate detection function are considered final for the mesic tundra habitat.

In the wet tundra habitat, we only observed one winter nest of brown lemming on all transects surveyed in 2016, which is clearly not enough to estimate a detection probability. Thus, we should expect problems when trying to model it with our three candidate models:

```
mod.2016.hn.wet <- ds(data=nids2016[nids2016
$habitat=="wetland",],convert.units=0.0001,adjustment = "cos", key="hn")
mod.2016.hr.wet <- ds(data=nids2016[nids2016
$habitat=="wetland",],convert.units=0.0001,adjustment = "cos", key="hr")
mod.2016.un.wet <- ds(data=nids2016[nids2016
$habitat=="wetland",],convert.units=0.0001,adjustment = "cos", key="un")
```

Running the half-normal model, we obtain the error message "All models failed to fit!", which was expected. Idem for the model using the uniform detection function. In contrast, the algorithm of the model using the hazard-rate detection functions initially showed convergence but showed non-convergence when adding the cosine adjustment. Here is the output obtained for the hazard-rate model:

```
> Starting AIC adjustment term selection.
Fitting hazard-rate key function
Key only model: not constraining for monotonicity.
AIC= 3.997
Fitting hazard-rate key function with cosine(2) adjustments
Error :
gosolnp-->Could not find a feasible starting point...exiting

Error in model fitting, returning: hazard-rate key function
Error: Error in -lt$value : invalid argument to unary operator
```

Looking at the fit of the model (command `ds.gof`) reveals that the chi-square fit test could not be performed due to the very small sample size. When a detection probability cannot be estimated, one can use the number of nests observed divided by the area surveyed as an estimate of the minimum number of nests present. Detection probabilities of nests are typically high within 5 m of the transects as it is usually $\geq 80\%$ in years with sufficient sample size on Bylot Island. Even though this distance can be debated, what is important is to be consistent and use this distance every time the minimum number of nests observed has to be calculated due to low

sample size. Here, we assume that the total area surveyed with high confidence is 9.73 ha ($5 \text{ m} \times 2 \times 9,730 \text{ m}$; the last number being the total length of transects surveyed). Thus, the minimum density of brown lemming winter nests in the wet tundra habitat was 0.1 nest/ha.

The stream gully habitat is generally where winter nests of lemmings are found in greater number on Bylot Island and 2016 was no exception to this trend. This allowed the algorithms of all three models with different detection functions to converge correctly. Moreover, all three models had a good fit to the data (chi-square p-values >0.05). In this case, we need to go through model selection using the AICc. The command `AICc` calls the AIC corrected for small sample sizes. Here are the commands followed by their respective outputs:

```
AICc(mod.2016.hn.stream)
> 899.6271
```

```
AICc(mod.2016.hr.stream)
> 900.5045
```

```
AICc(mod.2016.un.stream)
> 900.6723
```

The results indicate that the model using the half-normal detection function (`mod.2016.hn.stream`) has the highest statistical support but by a slight margin over other models. As discussed by Miller et al. (2016), density estimates of models with ΔAICc of <2 are generally very similar. We can confirm this by looking at the `summary` output of each model. In order to simplify the procedure and because model selection is not encouraged by the authors of the *Distance* package, model-averaging is not mandatory here. The `summary` output of the half-normal model gives the following densities:

```
> Density:
      Label Estimate      se      cv      lcl      ucl      df
1  C1_brown 1.9095345 0.6314549 0.33068527 0.9680017 3.766855 17.093597
2  C1_collared 0.7638138 0.2785534 0.36468759 0.3622828 1.610376 16.891409
3   C2_brown 1.7611425 0.5102431 0.28972275 0.9340333 3.320677  9.814222
4  C2_collared 3.8745135 0.8098676 0.20902432 2.4534961 6.118557 10.663913
5   C3_brown 1.5368649 1.0155672 0.66080449 0.2920485 8.087541  4.066051
6  C3_collared 5.7083554 2.6987334 0.47276899 1.6658221 19.561106  4.130564
7   GP_brown 1.1554434 0.1927791 0.16684427 0.3412927 3.911743  1.314785
8  GP_collared 5.7772169 0.5300150 0.09174226 4.3160061 7.733130  2.995671
9   PD_brown 2.6220797 1.2261916 0.46764084 0.7747626 8.874076  4.133514
10 PD_collared 6.9127556 1.0847546 0.15692072 4.6760654 10.219316  5.463403
11      Total 3.2021720 0.3979328 0.12426965 2.4683873 4.154091 17.825047
```

The densities obtained by the half-normal model are plausible and errors are a fraction of the density estimates. These results can thus be considered final.

6. REFERENCES



1. Anderson, D.R. 2008. Model based inference in the life sciences. Springer Science+Business Media, New York.
2. Buckland, S.T., E.A. Rexstad, T.A. Marques, and C.S. Oedekoven. 2015. Modelling Detection Functions. Pages 53-103 *in* S.T. Buckland, E.A. Rexstad, T.A. Marques, and C.S. Oedekoven, editors. Distance Sampling: Methods and Applications. Springer International Publishing, Cham, Switzerland.
3. Burnham, K.P., and D.R. Anderson. 2002. Model selection and multimodel inference: a practical information-theoretic approach. 2nd edition, Springer, New York.
4. Duchesne, D., G. Gauthier, and D. Berteaux. 2011. Evaluation of a method to determine the breeding activity of lemmings in their winter nests. *Journal of Mammalogy* 92:511-516.
5. Efford, M. 2013. DENSITY 5. Spatially explicit capture-recapture. University of Otago, New Zealand.
6. Efford, M. G. 2015. secr: spatially explicit capture-recapture models. R package version 2.9.4. <http://CRAN.R-project.org/package=secr>.
7. Efford, M.G., and R.M. Fewster. 2013. Estimating population size by spatially explicit capture-recapture. *Oikos* 122:918-928.
8. Fauteux, D., G. Gauthier, and D. Berteaux. 2015. Seasonal demography of a cyclic lemming population in the Canadian Arctic. *Journal of Animal Ecology* 84:1412-1422.
9. Gauthier, G., J. Bêty, M.-C. Cadieux, P. Legagneux, M. Doiron, C. Chevallier, S. Lai, A. Tarroux, and D. Berteaux. 2013. Long-term monitoring at multiple trophic levels suggests heterogeneity in responses to climate-change in the Canadian Arctic tundra. *Philosophical Transactions of the Royal Society B-Biological Sciences* 368:1-12.
10. Krebs, C.J., R. Boonstra, S. Gilbert, D.G. Reid, A.J. Kenney, and E.J. Hofer. 2011. Density estimation for small mammals from livetrapping grids: rodents in northern Canada. *Journal of Mammalogy* 92:974-981.
11. Miller, D.L. 2016. Distance: Distance sampling detection function and abundance estimation. R package version 0.9.6. <http://CRAN.R-project.org/package=Distance>.
12. Miller, D.L., E. Rexstad, L. Thomas, L. Marshall, and J.L. Laake. 2016. Distance Sampling *in* R. bioRxiv preprint server, <https://doi.org/10.1101/063891> .
13. Thomas, L., J.L. Laake, E. Rexstad, S. Strindberg, M.F.F.C., S.T. Buckland, D.L. Borchers, D.R. Anderson, K.P. Burnham, M.L. Burt, S.L. Hedley, J.H. Pollard, J.R.B. Bishop, and T.A. Marques. 2009. Distance 6.0. Release 2. Research Unit for Wildlife Population Assessment, University of St. Andrews, UK.
14. Williams, B.K., J.D. Nichols, and M.J. Conroy. 2002. Analysis and management of animal populations. Academic Press, San Diego.



FAUTEUX, D., G. GAUTHIER, M.-C. CADIEUX. 2018. ESTIMATING SMALL MAMMAL AND WINTER NEST DENSITIES USING CAPTURE-RECAPTURE AND DISTANCE SAMPLING WITH R - VERSION 1. CENTRE D'ÉTUDES NORDIQUES, UNIVERSITÉ LAVAL, QUÉBEC, 39 PAGES.