Intermediate distance sampling workshop - St Andrews 2017

Laura Marshall, David L. Miller and Len Thomas

2017-07-04

Contents

Preface 5

1 A hands on introduction to R tutorial 7

2 analyse simulated data set 9

3 Problem datasets 11

4 Creating distance sampling simulations using DSsim 13
- 4.1 AIM 13
- 4.2 Create a region object 13
- 4.3 Create a density object 14
- 4.4 Creating population description and detectability objects 16
- 4.5 Creating the survey design object 16
- 4.6 Creating the analyses object 16
- 4.7 Creating the simulation object 17
- 4.8 Now for the automated designs: Parallel lines 20
- 4.9 Results from 999 repetitions 23
- 4.10 Running distance sampling simulations using Distance 7.1 24
- 4.11 Creating simulations in Distance 25

5 Preparing data for analysis 29

6 Detection function fitting 31

7 Simple density surface models 33

8 Advanced density surface models 35

9 Prediction using fitted density surface models 37

10 Estimating precision of predictions from density surface models 39

11 Mark-recapture distance sampling of golftees 41

References 43

12 Placeholder 45

13 Simulation of distance sampling data 47

Preface

Chapter 1

A hands on introduction to R tutorial

Chapter 2

analyse simulated data set

Chapter 3

Problem datasets

Chapter 4

Creating distance sampling simulations using DSsim

Assumes home directory is directory in which exercise has been expanded

4.1 AIM

The aim of this exercise is to run simulations which will allow you to compare three different survey designs for a specific population. You should judge these designs on their accuracy and precision.

You will also need the following R packages installed on your machine: DSsim, shapefiles, splancs and mrds. Now examine the other files and folders in the “DSsim Exercise” folder. There are three files starting with the name “Region” and ending with .dbf, .shp and .shx, these files make up the shapefile for the survey region. The “density.surface.robj” file is the density surface for the survey region. The “Survey Transects” folder contains a folder for each of the designs you are asked to look at, these contain the transect shapefiles. The “Results” folder contains the results from 999 replications as this can take a good few hours to run. To setup the workspace first load the libraries DSsim and shapefiles, loading these two will automatically make splancs and mrds available.

```
library(DSsim)
library(shapefiles)
```

4.2 Create a region object

Read the Region shapefile into R using the read.shapefile function from the shapefiles library.

```
region.shapefile <- read.shapefile("Region")
```

Next you are going to create the region object using this shapefile. As there are no strata you only need to provide a name for your survey region and the units which are in metres (m). The survey region is displayed in Figure 4.1.

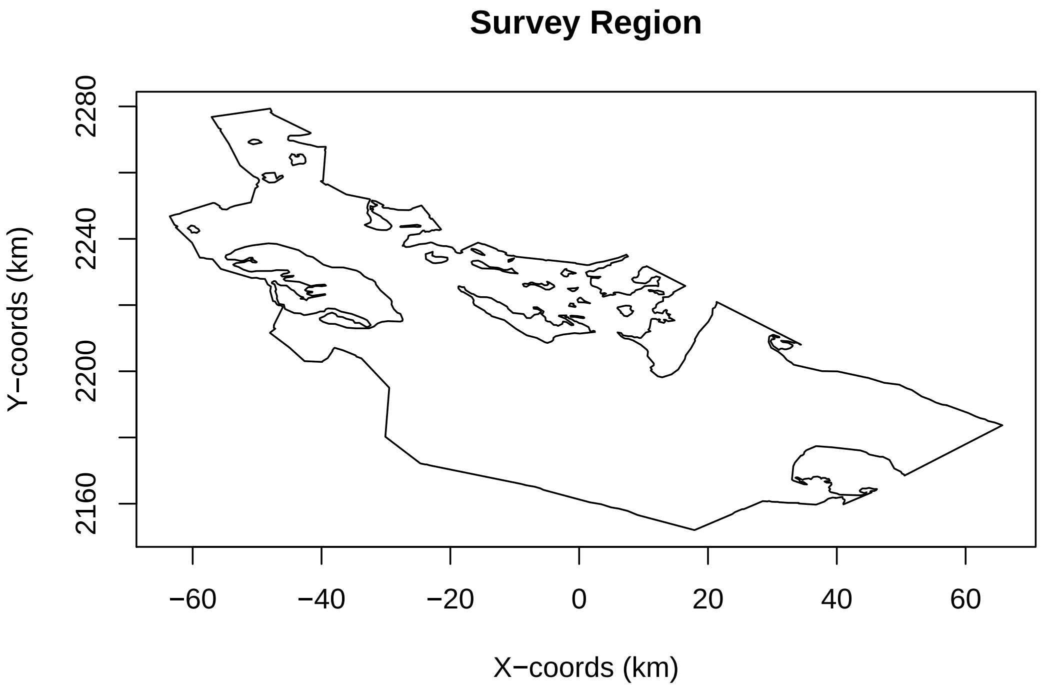


Figure 4.1: Study region for simulation

```r
region <- make.region(region.name = "Survey Region", units = "m",
                      shapefile = region.shapefile)
plot(region, plot.units = "km")
```

4.3 Create a density object

You are now going to create a density object within this region. For the purposes of this exercise a density surface has already been created and can be loaded as follows:

```r
load("density.surface.robj")
```

You will see that an object called “density.surface” has appeared in the workspace. This object is a list with one element (if the region had been divided up into strata then this list would contain an element for each strata). To have a look at what the density surface data look like type `head(density.surface[[1]])`. You can see that it is a data set of x and y locations and the densities at each point.

To create the density object you will need to provide the density surface, the region object for which it was created and the grid spacing that was used. I used a grid spacing of 1,000 m in both the x and y

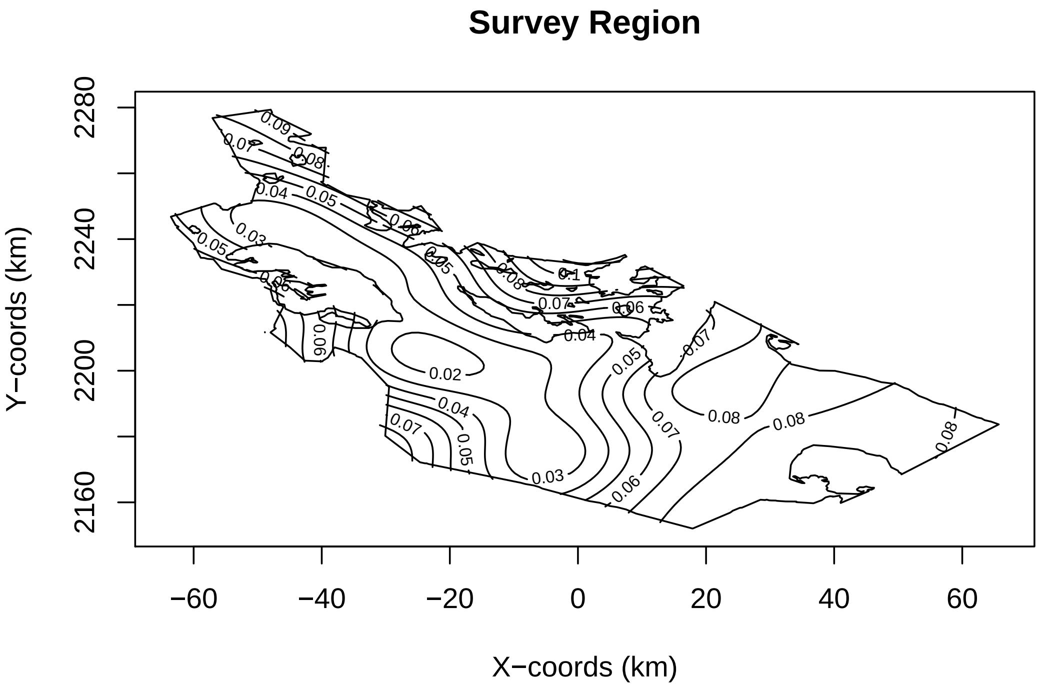


Figure 4.2: Study region with animal density superimposed Note lower density near the trail system

directions to create this density surface. The density surface describing animal distribution is shown in Figure 2.

```
pop.density <- make.density(region = region, density.surface = density.surface,
                            x.space = 1000, y.space = 1000)
plot(pop.density, plot.units = "km")
plot(region, add = TRUE)
```

Optionally, the following code can be used to define your own density surface. Firstly the density object is created with a constant value, then high and low spots can be added with varying radii of influence. The sigma parameter is used to calculate a Gaussian decay around the central point.

```
alternative.density <- make.density(region = region, x.space = 1000,
                                    y.space = 1000, constant = 0.4e-7)

alternative.density <- add.hotspot(alternative.density, centre = c(-2500, 2224000),
                                   sigma = 10000, amplitude = 0.1e-7)
alternative.density <- add.hotspot(alternative.density, centre = c(0, 2184000),
                                   sigma = 18000, amplitude = -0.5e-8)
```

4.4 Creating population description and detectability objects

For this exercise we will fix the population size at 1500 individuals. To do this set N = 1500 and tell it to generate exactly this number of individuals (fixed.N = TRUE).

```
pop.description <- make.population.description(region.obj = region,
                                                density.obj = pop.density,
                                                N = 1500, fixed.N = TRUE)
```

We will now describe the detectability of the population using a half-normal function with a sigma (scale.param) of 500 m and a truncation distance of 1000 m. This means that around 2/3 of the detections will be made within 500 m of the transect and we will exclude anything sighted further than 1000 m perpendicular distance from the transect.

```
detect <- make.detectability(key.function = "hn", scale.param = 500, truncation = 1000)
```

4.5 Creating the survey design object

We will now create a design object. For now concentrate on the subjective design, we will come back to the parallel and zigzag designs later. The subjective design was based on using some **existing paths** to make the survey easier to carry out. Additional transects were then added to achieve a more even coverage of the survey region.

NOTE: The path argument to describe where the files are located must match your previous settings add "/Survey Transects/Subjective Design".

```
subjective.design <- make.design(transect.type = "Line",
                                 design.details = c("user specified"),
                                 region = region,
                                 plus.sampling = FALSE,
                                 path = "Survey Transects/Subjective Design")
```

4.6 Creating the analyses object

The final thing we need to do before creating the simulation object is describe the analyses we wish to carry out on the simulated data. Let's try letting it choose between a half-normal and a hazard rate model based on the AIC values.

```
ddf.analyses <- make.ddf.analysis.list(
                dsmodel = list(~cds(key = "hn", formula = ~1), #half-normal model
                               ~cds(key = "hr", formula = ~1)),  #hazard rate model
                method = "ds", criteria = "AIC", truncation = 1000)
```

4.7 Creating the simulation object

We can finally put it all together and have a look at some example populations, transects and survey data. I suggest you set the number of repetitions (reps) to be fairly low or else it will take a long time to run. For the subjective design you need to specify that it will be using the same set of transects each time, single.transect.set = TRUE.

```
my.simulation.subjective <- make.simulation(reps = 10,
                                            single.transect.set = TRUE,
                                            region.obj = region,
                                            design.obj = subjective.design,
                                            population.description.obj = pop.description,
                                            detectability.obj = detect,
                                            ddf.analyses.list = ddf.analyses)
```

Before running the simulation it is a good idea to have a check to see that it is doing what you want. This little function will allow you to investigate the simulation properties. This function having created a population, transects, survey and detections, then plots them to assure you are happy with the simulation structure.

```
check.sim.setup <- function(simsetup) {
  #  small function to plot the simulation setup
  #
  #  input:
  #     simsetup - object created by make.simulation()
  #
  #  output:
  #     4-panel plot of study region with popn, study region with transects,
  #     study region with detected objects, and histogram of detection distances
  pop <- generate.population(simsetup)  #generate a population
  transects <- generate.transects(simsetup)   #generate the transects
  eg.survey <- create.survey.results(simsetup)  #simulate the survey process of detection
  dist.data <- get.distance.data(eg.survey)  #look at distance data
  par(mfrow = c(2,2))
  plot(region)
  plot(pop)
  plot(region)
  plot(transects, col = 4, lwd = 2)
  plot(eg.survey)
  hist(dist.data$distance, xlab = "Distance (m)", main = "Distance Data")
  par(mfrow=c(1,1))
  }
```

Let's check our subjective design simulation, see Figure 4.3.

```
check.sim.setup(my.simulation.subjective)
```

Once you are happy it is time to run the simulation. Please be patient as it will take a few minutes to complete.

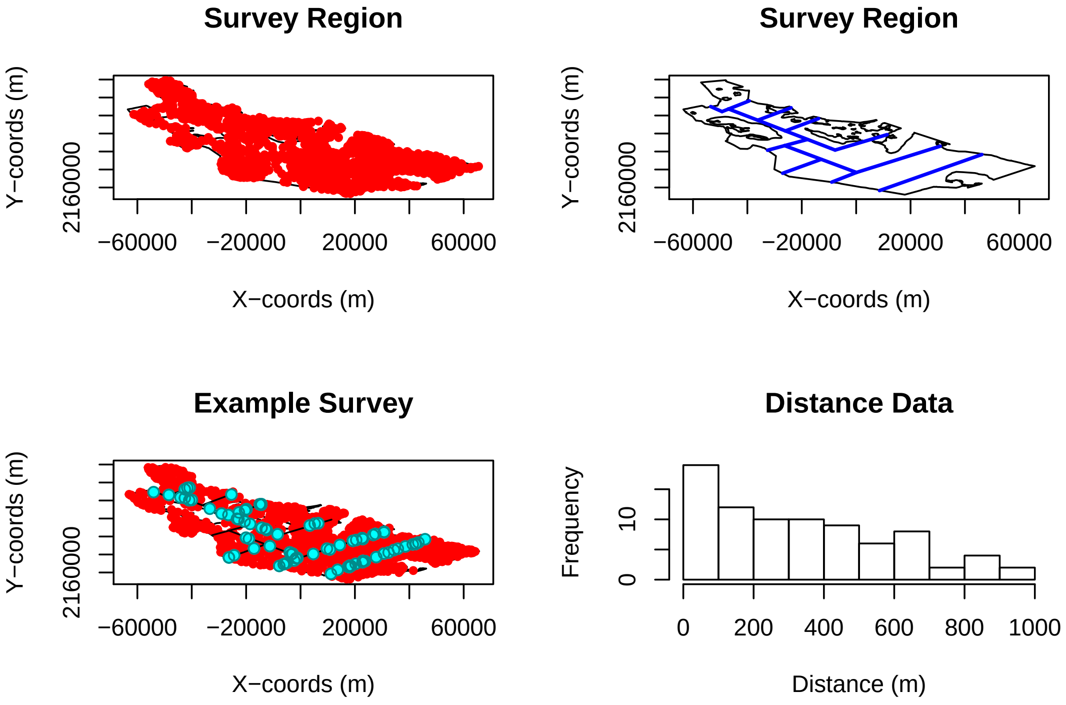


Figure 4.3: Region, population, transects, detections

```
my.simulation.subjective  <- run(my.simulation.subjective)
```

```
 1  out of  10  reps

 2  out of  10  reps

 3  out of  10  reps

 4  out of  10  reps

 5  out of  10  reps

 6  out of  10  reps

 7  out of  10  reps

 8  out of  10  reps

 9  out of  10  reps

 10  out of  10  reps
```

```
summary(my.simulation.subjective, description.summary = FALSE)
```

```
Region:  Survey Region
No. Repetitions:  10
No. Failures:  0

Design:  Subjective Line Transect
   design.axis  =  0
   spacing  =  100
   plus.sampling  =  FALSE

Population Detectability Summary:
    key.function  =  hn
    scale.param  =  500
    truncation  =  1000

Analysis Summary:
   Candidate Models:
      Model  1 :  ~ cds(key = "hn", formula = ~1)  was selected  7
 time(s).
      Model  2 :  ~ cds(key = "hr", formula = ~1)  was selected  3
 time(s).
   criteria  =  AIC
   truncation  =  1000
```

```
Summary for Individuals

Summary Statistics

  mean.Cover.Area mean.Effort mean.n      mean.ER   mean.se.ER
1       673583956      336792   84.5 0.0002508967 5.018289e-05
    sd.mean.ER
1 2.619516e-05

     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Estimates of Abundance (N)

  Truth mean.Estimate percent.bias RMSE CI.coverage.prob mean.se
1  1500       1214.33       -19.04  341                1  274.33
  sd.of.means
1      196.27

     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Estimates of Density (D)

         Truth mean.Estimate percent.bias         RMSE CI.coverage.prob
1 2.640575e-07  2.137684e-07    -19.04477 6.002856e-08                1
       mean.se  sd.of.means
1 4.829177e-08 3.455156e-08

     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Detection Function Values

  mean.observed.Pa mean.estimate.Pa sd.estimate.Pa mean.ESW sd.ESW
1              0.6              0.6           0.09   596.92   92.2
```

4.8 Now for the automated designs: Parallel lines

You will need to create a new simulation each with a new design object for the parallel design. The other objects (region, density, population description etc.) should be left the same.

NOTE: We now wish different transects to be used on each repetition (`single.transect.set = FALSE`).

```r
parallel.design <- make.design(transect.type = "Line",
                               design.details = c("Parallel","Systematic"),
                               region.obj = region, design.axis = 45,
                               spacing = 12000, plus.sampling = FALSE,
                               path = "Survey Transects/Parallel Design")
```

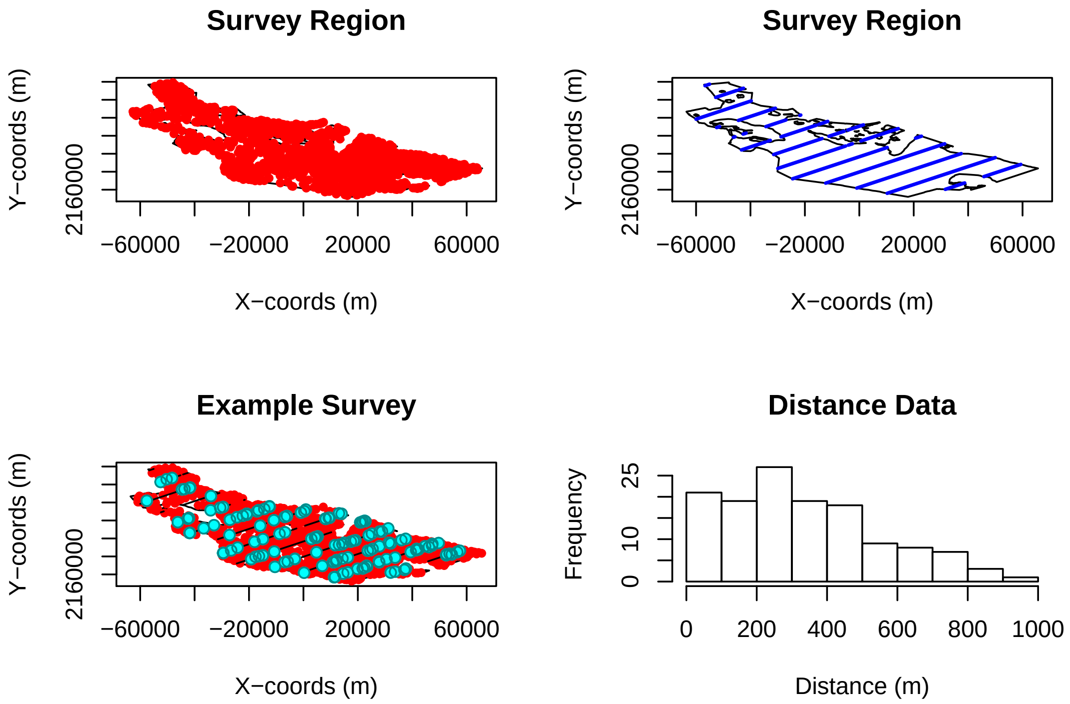


Figure 4.4: Check setup of parallel transect design simulation

```
my.simulation.parallel <- make.simulation(reps = 10,
                                          single.transect.set = FALSE,
                                          region.obj = region,
                                          design.obj = parallel.design,
                                          population.description.obj = pop.description,
                                          detectability.obj = detect,
                                          ddf.analyses.list = ddf.analyses)
```

Having created the features of the simulation, we want to check features of the simulation have been correctly specified, see Figure 4.4.

```
check.sim.setup(my.simulation.parallel)
```

When satisfied with this simulation setup, you would proceed to run your parallel design simulation.

```
my.simulation.parallel  <- run(my.simulation.parallel)
```

```
 1  out of  10  reps

 2  out of  10  reps
 3  out of  10  reps

 4  out of  10  reps

 5  out of  10  reps

 6  out of  10  reps

 7  out of  10  reps

 8  out of  10  reps

 9  out of  10  reps

 10  out of  10  reps
```

```r
summary(my.simulation.parallel, description.summary = FALSE)
```

```


Region:  Survey Region
No. Repetitions:  10
No. Failures:  0

Design:  Systematic Parallel Line Transect
   design.axis  =  45
   spacing  =  12000
   plus.sampling  =  FALSE

Population Detectability Summary:
    key.function  =  hn
    scale.param  =  500
    truncation  =  1000

Analysis Summary:
   Candidate Models:
      Model  1 :  ~ cds(key = "hn", formula = ~1)  was selected  7
 time(s).
      Model  2 :  ~ cds(key = "hr", formula = ~1)  was selected  3
 time(s).
   criteria  =  AIC
   truncation  =  1000

Summary for Individuals

Summary Statistics

  mean.Cover.Area mean.Effort mean.n      mean.ER   mean.se.ER
1       939710971    469855.5  146.7 0.0003121056 3.647356e-05
    sd.mean.ER
```

```
1 2.121697e-05

     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Estimates of Abundance (N)

  Truth mean.Estimate percent.bias   RMSE CI.coverage.prob mean.se
1  1500       1502.13         0.14 168.64                1   233.7
  sd.of.means
1      177.75

     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Estimates of Density (D)

         Truth mean.Estimate percent.bias         RMSE CI.coverage.prob
1 2.640575e-07  2.644331e-07    0.1422213 2.968748e-08                1
       mean.se  sd.of.means
1 4.113929e-08 3.129085e-08

     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Detection Function Values

  mean.observed.Pa mean.estimate.Pa sd.estimate.Pa mean.ESW sd.ESW
1             0.61              0.6           0.06   595.46   59.5
```

4.8.1 ZigZag survey design

Now have a go at creating and running a simulation using the equal spaced zigzag design transects in the “Zigzag Design” folder. The spacing used to generate these was 8250m on a design axis of 135 degrees. Use `?make.design` for help.

Having created the features of the simulation, check the features of the simulation have been correctly specified (Fig. **??**).

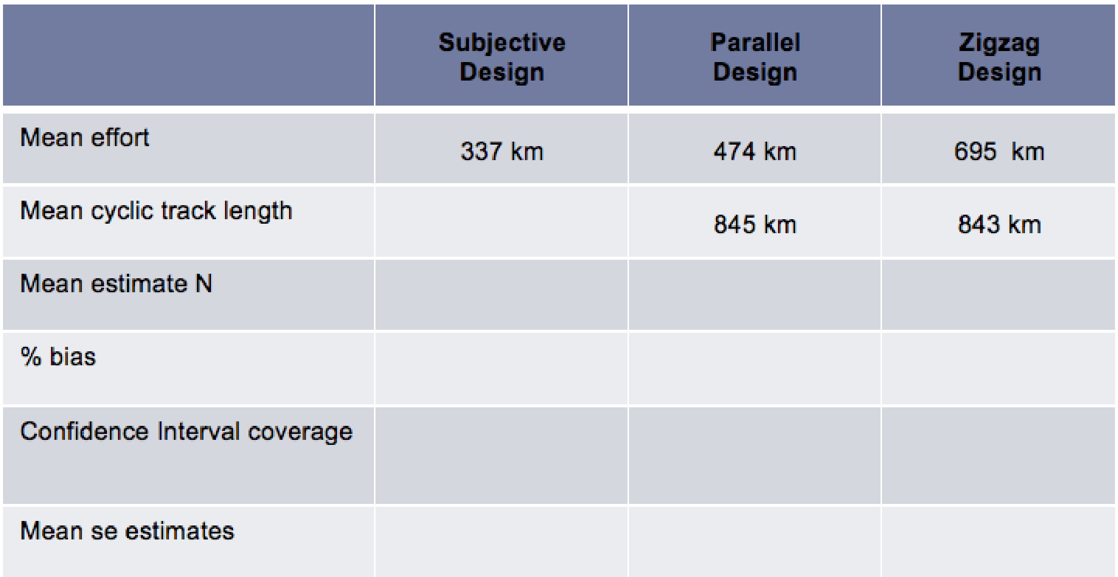
When satisfied with this simulation setup, you would proceed to run your zigzag design simulation.

```
my.simulation.zigzag  <- run(my.simulation.zigzag)
summary(my.simulation.zigzag)
```

4.9 Results from 999 repetitions

I ran each of these simulations 999 times and stored the simulations as r objects. Load these into the R workspace using the following code:

```
load("Results/simulation.subjective.robj")
load("Results/simulation.parallel.robj")
load("Results/simulation.zigzag.robj")
```



	Subjective Design	Parallel Design	Zigzag Design
Mean effort	337 km	474 km	695 km
Mean cyclic track length		845 km	843 km
Mean estimate N			
% bias			
Confidence Interval coverage			
Mean se estimates			

Figure 4.5: Results Table

The objects `simulation.subjective`, `simulation.parallel` and `simulation.zigzag` will now be in your workspace. Have a look at the results using the `summary()` function and use them to fill in the table below, Figure 5.

```
summary(simulation.subjective)
summary(simulation.parallel)
summary(simulation.zigzag)
```

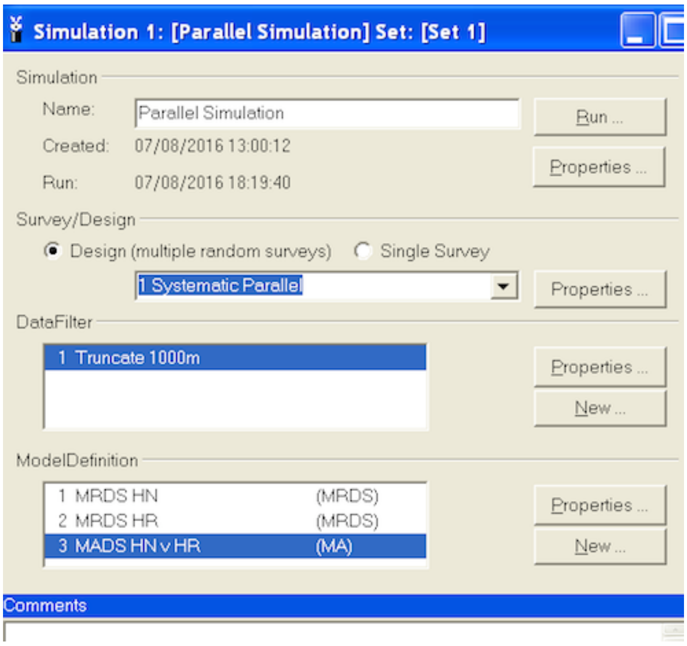
Which survey design would you recommend? Why?
What would happen if our assumptions about animal distribution were incorrect?

4.10 Running distance sampling simulations using Distance 7.1

If you would like to investigate different designs then these can be created and used in simulations in Distance 7.1. Note that currently the simulation options in Distance 7.1 are somewhat more restricted than in DSsim.

We have created a Distance project based on Exercise 1A and setup the systematic parallel and equal spaced zigzag designs as specified above. This project is named DSsimExercise. This exercise will lead you through replicating the simulations in exercise 1A in Distance, but you could choose to invesgitate different designs or even try out some simulations on your own study area if you prefer.

If you wish to try out simulations on your own study area help on importing geographic data, creating designs and analyses can be found in the Distance manual.



Simulation 1: [Parallel Simulation] Set: [Set 1]

Simulation

Name: Parallel Simulation

Created: 07/08/2016 13:00:12

Run: 07/08/2016 18:19:40

Survey/Design

● Design (multiple random surveys) ○ Single Survey

1 Systematic Parallel

DataFilter

1 Truncate 1000m

ModelDefinition

1 MRDS HN	(MRDS)
2 MRDS HR	(MRDS)
3 MADS HN v HR	(MA)

Comments

Figure 4.6: Simulation Details

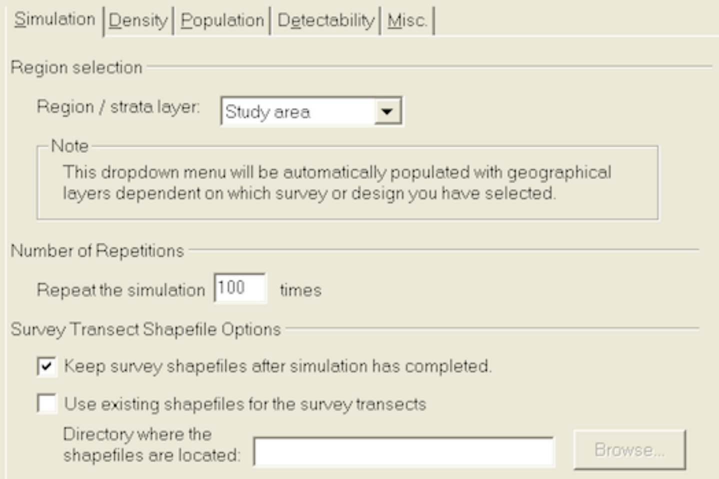
4.11 Creating simulations in Distance

4.11.1 Simulation Details

Open the DSsimExercise.dst project and navigate to the Simulation Browser tab (on the far right, with the rabbit coming out of the hat). Now create a new simulation and give it a meaningful name. Open the details for this simulation, Figure 6.
1. Select the design option for these simulations as we want to use a different survey (set of transects) in each iteration and then select which design to use from the dropdown menu. Distance will now automatically generate the required number of surveys for the simulation. (Selecting the survey option will instruct Distance to use only a single set of transects for the whole simulation.)
2. Now select a data filter with an absolute right truncation distance. The truncation distance specified in the data filter will give the greatest perpendicular distance at which an observation can be made and the distance to which the detection function model(s) will be fitted.
3. Finally we can select one or more (mrds) models to fit to the simulated data. Here we can use the MADS HN v HR model definition (ID 3) to point to both the half-normal and hazard rate model definitions. Use the Properties button to have a look at the MADS model definition properties, particularly the detection function tab. Here we will see that the model with the minimum AIC will be selected in each simulation iteration.

4.11.2 Simulation Properties

Now click on the Simulation ‘Properties’ button to set the other simulation properties. The Simulation tab (Figure 7) allows us to specify the geographic layer to use, in this example as we do not have



Region selection

Region / strata layer: Study area

Note
This dropdown menu will be automatically populated with geographical layers dependent on which survey or design you have selected.

Number of Repetitions

Repeat the simulation 100 times

Survey Transect Shapefile Options

☑ Keep survey shapefiles after simulation has completed.

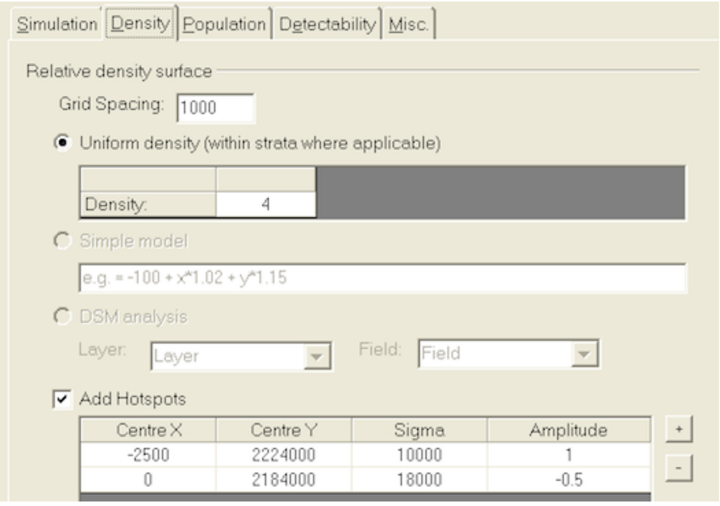
□ Use existing shapefiles for the survey transects

Directory where the shapefiles are located:

Figure 4.7: Properties Pages: Simulation tab

strata we must select the global study region layer. Here we can also tell Distance how many times to repeat the simulation and set shapefile options. It is sensible to run the simulation only once in the first instance to check the setup is correct. The shapefile options allow us to tell Distance to save the shapefiles for use in subseqent simulations using the same design. This can save some processing time. If requested the shapefiles are stored in the project .dat folder under ‘Simulation/Simulation[ID]/shapefiles’.

Next we can define our density surface which describes animal distribution (Figure 8). As in exercise 1A we can select a grid spacing of 1000. Distance has more restricted options than DSsim. Currently we are only able to specify a constant density surface with hot/low spots. Note that this density surface is just giving Distance the relative (rather than absolute) densities.



Relative density surface

Grid Spacing: 1000

● Uniform density (within strata where applicable)

Density:	4

○ Simple model

e.g. = -100 + x*1.02 + y*1.15

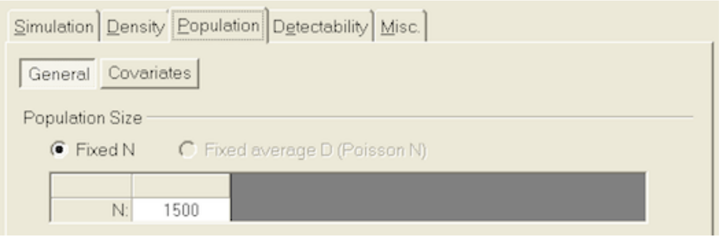
○ DSM analysis

Layer: Layer Field: Field

☑ Add Hotspots

Centre X	Centre Y	Sigma	Amplitude
-2500	2224000	10000	1
0	2184000	18000	-0.5

Figure 4.8: Properties Pages: Density tab



Population Size

● Fixed N ○ Fixed average D (Poisson N)

N:	1500

Figure 4.9: Properties Pages: Population tab

The Population tab (Figure 9) currently only requires that we provide a population size, in this case 1500.

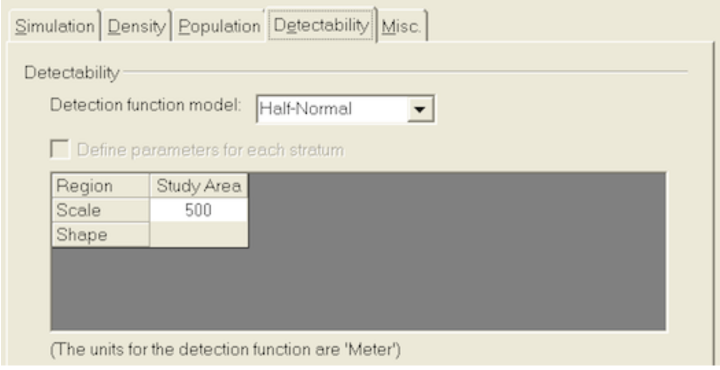
Next we describe the detectability of the animals. We will assume a half-normal detection function with sigma = 500m (Figure 10). The units of the detection function parameters must be the same as those of the study region and a reminder is provided below the table.

Finally we can select some miscelaneous options. These do not affect the output seen within distance. The option to run the simulation in parallel can speed things up if running more than a few iterations. Saving the results from each iteration to file will create csv files with the individual estimates from each repetition. Saving an example dataset will create a csv file that is ready to be read into a distance project for analysis. These files are stored in the project .dat folder under ‘Simulation/Simulation[ID]’.

Further instructions on setting up different simulations options can be found in the Distance manual.

4.11.3 Results

Solutions to this practical can be found in the DSsimExerciseSolutions.dst project. In this project both the parallel and zigzag design simulations have been run 100 times.



Detectability

Detection function model: Half-Normal

☐ Define parameters for each stratum

Region	Study Area
Scale	500
Shape	

(The units for the detection function are 'Meter')

Figure 4.10: Properties Pages: Detectability tab

Note that even though the designs were never initially run to estimate coverage, when a simulation is run this triggers the design to be run. Therefore, the design results give the coverage for the actual sets of transects used in the simulation.

Chapter 5

Preparing data for analysis

Chapter 6

Detection function fitting

Chapter 7

Simple density surface models

Chapter 8

Advanced density surface models

Chapter 9

Prediction using fitted density surface models

Chapter 10

Estimating precision of predictions from density surface models

Chapter 11

Mark-recapture distance sampling of golftees

References

Chapter 12

Placeholder

Chapter 13

Simulation of distance sampling data