REM analysis vignette

Pablo Palencia

2021-04-14

1. Introduction

1.1 REM background

The Random Encounter Model (**REM**) is a method for estimating animal density from camera trap data without the need for individual recognition (Rowcliffe et al., 2008 - J. App. Ecol. 45: 1228-1236). Over the last years it has been used for a wide range of unmarked species (Cusack et al., 2015 - J. Wild. Manag. 79(6): 1014-1021; ENETWILD-consortium et al., 2019 - EFSA Supporting Publications - 16(9); Pfeffer et al., 2018 - Remote Sens. Ecol. Conserv. 2:84-94; Zero et al., 2013 - Oryx 47(3): 410-419). Most of these studies highlighted the potencial of REM as a promising method to monitor unmarked species.

Briefly, the REM is based on modelling the process of random encountres between animals and cameras, and accounting for all the variables that affect the encounter rate (animal movement speed and camera's detection viewshed).

REM equation:

$$D = \frac{y}{t} \frac{\pi}{v \cdot r \cdot (2 + \alpha)}$$

in which *y* is the number of encounters (i.e. number of independent photographic sequences), *t* is total camera survey effort, *v* is the average distance travelled by an individual during a day (day range), and *r* and α are the radius and angle of the camera traps detection zone, respectively.

1.2 Practical exercise

Here, I'm going to estimate population density of fallow deer (*Dama dama*) population samped with 30 camera traps. (Why fallow deer? Easy, is my favourite species, see **Fig. 1**). Survey design consisted in a systematic design with random origin; concretely cameras were deployed at the intesection of a grid with 2km spacing.

Figure 1: Fallow deer picture

Should be noted that all the necessary parameters to apply REM (i.e. day range, encounter rate and detection zone) will be derived from camera trap data, without the need of auxiliary data. This required additional effort, especialy during image processing, but it is recommended taking into account the spatio-temporal variations in day range, and the variability in detection zone as function of target species, camera trap brand, settings and enviromental variables between others. Finally, in spite of I'm going to use other packages, should be noted that *remBoot* R package implemet REM calculations in R (Caravaggi, 2017 - J. Open Source Softw.- 2(10):176).

2. Importing dataframes and functions

Three dataframes are needed to run the analysis: i) the raw data of day range, detection zone and encounter rate, ii) the operativity matrix (information about camera traps functionality), and iii) camera trap coordinates (just for plots and exploratory analysis). Adittionally, we will need to import a couple of functions that are not included in an R package (these functions were devoloped by M. Rowcliffe and Distance Sampling folks from St. Andrews University).

```
# Load dataframes
dataREM <- read.table("Data.txt", sep = ";", dec=".", header=TRUE, as.is=TRUE) # parameters dataframe
operat <- read.table("Operativity.txt", sep = ";", dec=".", header=TRUE, as.is=TRUE) # operatity matrix (to estimate survey effort)
df_coord <- read.table("Coordinates.txt", sep = ";", dec=".", header=TRUE, as.is=TRUE) # camera trap locations (plots, maps etc.)

# Load functions
source("REM_functions.R") # importing some key functions to run the analysis


# Packages required to run the analyses
library(activity) # to estimate activity pattern and day range (available on CRAN)
library(trappingmotion) # to estimate speed and day range (available on github https://github.com/PabloPalencia/trappingmotion)
library(Distance) # to estimate detection zone (available on CRAN)
library(dplyr) # to work with dataframes (available on CRAN)
library(ggplot2) # to plot encounter rates (avilable on CRAN)
```

3. Analysis

As described above, all the parameters needed to apply REM will be derived from camera trap data (*dataREM* dataframe). I have included a specific section for each parameter.

3.1 Day range

Day range is a parameter that relies on movement and behaviour. Recent studies have described a porcedure to estimate day range from camera trapping data (Palencia et al. 2021 - Methods Ecol. Evol. doi: 10.1111/2041-210X.13609; Rowcliffe et al. 2016 - Remote Sens. Ecol. Conserv. 2:84-94). Briefly, day range is estimated as the product of speed (average speed of travel while active) and activity rate (proportion of day that the population spent active).

3.1.1 Activity

To estimate activity, we will use the columns "G_size" and "H_first". "G_size" includes the number of animals observed per group, and "H_first" includes the time of the first photo of each group.(Adittionally, to avoid the bias caused by shorter detection distances at night (see Rowcliffe et al. 2014 Methods Ecol. Evol, 5(11): 1170-1179), we are only going to considered for activity estimate sequences closer than 5m to the cameras - i.e. "Interval.min == 1 | 2").

```
data.acti <- as.data.frame(dataREM)

# Estimating radian time of day
data.acti$T_sec2 <- as.numeric(strptime(data.acti$H_first, format="%H:%M:%S") - as.POSIXct(format(Sys.Date())),
 units="secs")
data.acti$T_0_1 <- data.acti$T_sec2/86400; data.acti <- subset(data.acti, G_size!="NA") # remove NAs

# Replicating each sequence based on the group size
activity.repli <- data.acti[rep(row.names(data.acti), data.acti$G_size), 1:16]

# Discariding observations further than 5 meters (see Rowcliffe et al. 2014 Methods Ecol. Evol, 5(11): 1170-117
9)
activity.repli <- subset(activity.repli, Interval.min =="1"  | Interval.min =="2")

# Estimating activity rate
activityRES <- 2*pi*activity.repli$T_0_1
mod1 <- fitact(activityRES, sample="data")

# Plot activity patterns
par(mfrow=c(1,1)); plot(mod1)
```

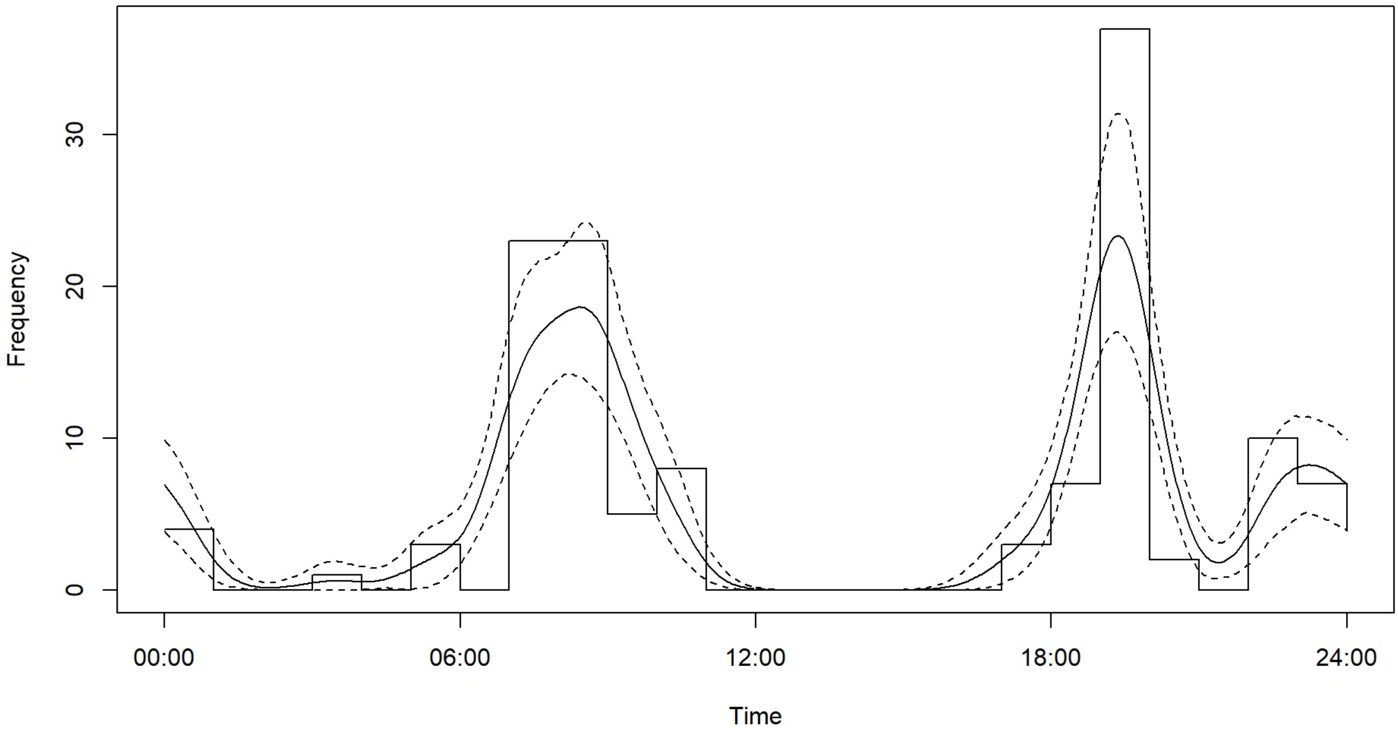

Figure 2: Activity pattern

```
mod1@act[1] # mean activity level
```

```
##       act
## 0.2372434
```

```
mod1@act[2] # SE activity level
```

```
##         se
## 0.03109462
```

Activity results evidenced two peaks of activity during sunrise and sunset **Fig. 2**, which is expected for ungulates. Activity level of 0.23 can be interpreted as the population spent acitive (0.23 x 24) 5.5 hours per day.

3.1.2 Speed

To estimate speed, we will use the column "Speed.m.s", which include speed estimations in m/s of those animals that did not react to the camera traps. Using the package *trappingmotion* we will follow the procedure described by Palencia et al. (2021). Briefly, we will identify different movement behaviours on the basis of the speeds (**Fig. 3**); and for each behaviour, we will estimate the average speed. Those sequences in which animals react to the camera trap cannot be considered for speed estimation (Rowcliffe et al. 2016 - Remote Sens. Ecol. Conserv. 2:84-94).

```
data.speed <- subset(dataREM, Behaviour != "Curiosity") # discard animals that react to the camera
data.speed <- subset(data.speed, Speed.m.s!="NA"); data.speed$Speed.m.s <- as.numeric(as.character(data.speed$Speed.m.s)) # Remove NAs

identbhvs(data.speed$Speed.m.s) # identify movement state
```

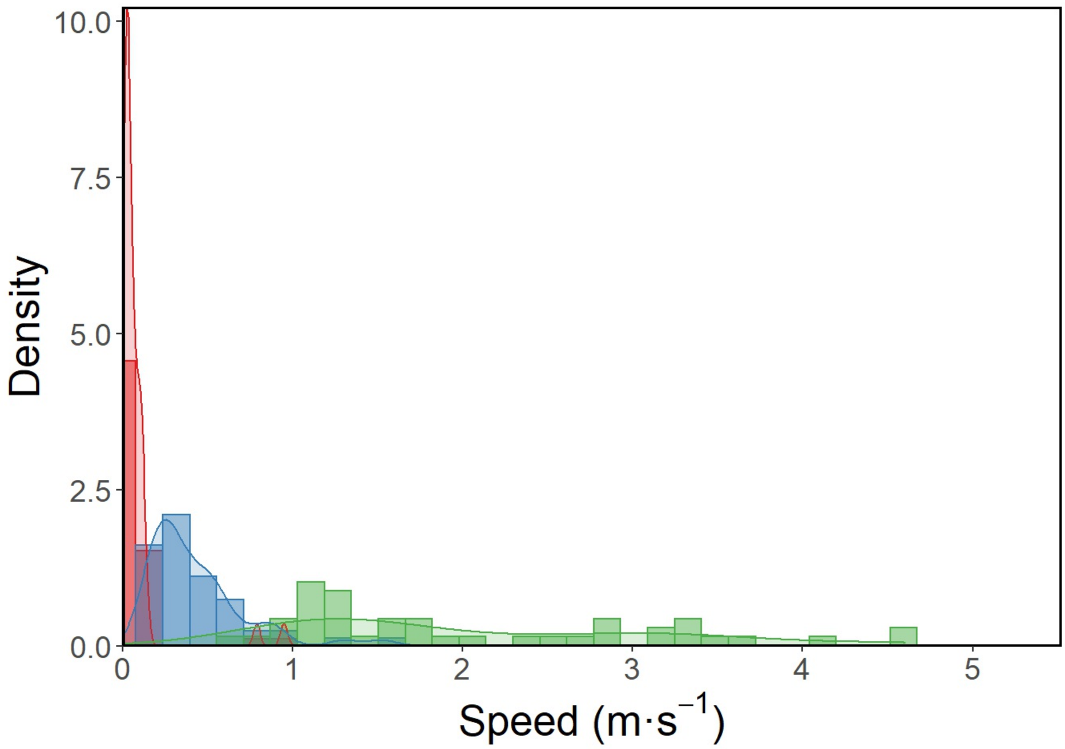

Figure 3: Movement behaviours identified

```
meanspeed(behav_class) # average movement speed of each state
```

3.1 Day range

Finally, day range is estimated as the sum of the product of the mean speed and the proportion of the activity level associated with each behaviour.

```
dayrange(act=mod1@act[1], act_se=mod1@act[2], speed_data) #day range (daily distance travelled)
```

```
## Day range (Km/day) 1.368014
## Day range SE (Km/day) 0.1484657
```

The day range estimated is 1.37 km/day (SE=0.1), which means that, in average, each fallow deer in the population travels 1.37km per day.

3.2 Detection zone

To estimate effective detecion zone (area effectively monitored by cameras), we will follow the procedure described by Rowcliffe et al. (2011) - Methods Ecol. Evol. 2(5):465-476. These procedure borrows from distance sampling theory (see Buckland et al. 2001 for further details), and for that it is needed to record the position (distance and angle relative to the camera) where the fallow deers are first detected. I also recommend to have a look to Hofmeester et al. 2017 - Remote Sens. Ecol. Conserv. 3(2):81-89 for practical and simple method to estimate detection zone when working with camera traps.

3.2.1 Radius

To estimate effective detection radius we will use the column "Dist_det", which include the distance -in meters- between animals and cameras in the first photo of each sequence. When more than one animal appeared in the first photo, we recorded the distance to the closer one. We will apply a point-transect distance sampling. Should be noted that advanced models can be considered by including covariates as sampling point or vegetation cover, betweeen others.

```
data_dz_r<-subset(dataREM, Dist_det >= 0 ) # selecting data to estimate effective detection radius

w_rad <- 10 # truncation distance (in meters)

# half-normal
hn_cos0 <- ds(data_dz_r$Dist_det, transect = "point", key="hn", adjustment = "cos", order = 0, truncation=w_rad
)
```

```
## Fitting half-normal key function
```

```
## Key only model: not constraining for monotonicity.
```

```
## AIC= 417.566
```

```
## No survey area information supplied, only estimating detection function.
```

```
#hazard-rate
hr_cos0 <- ds(data_dz_r$Dist_det, transect = "point", key="hr", adjustment = "cos", order = 0, truncation=w_rad
)
```

```
## Fitting hazard-rate key function
```

```
## Key only model: not constraining for monotonicity.
```

```
## AIC= 413.435
```

```
## No survey area information supplied, only estimating detection function.
```

Here I've included just two models as example, but should be tested as minimun all the combinations between 'half-normal' and 'hazard-rate' functions, 'cos', 'herm' and 'poly' adjustments, and orders 0 and 2.

After testing all these models, we will select the best one on the basis of AIC:

```
#model comparison
AIC(hn_cos0, hn_cos2, hn_herm0, hn_herm2, hn_poly0, hn_poly2, hr_cos0, hr_herm0, hr_herm2, hr_poly0, hr_poly2)
```

```
##          df      AIC
## hn_cos0   1 417.5662
## hn_cos2   2 416.6139
## hn_herm0  1 417.5662
## hn_herm2  2 419.0958
## hn_poly0  1 417.5662
## hn_poly2  3 413.0855
## hr_cos0   2 413.4352
## hr_herm0  2 413.4352
## hr_herm2  3 415.3991
## hr_poly0  2 413.4352
## hr_poly2  4 416.1015
```

```
# select best model
# (mind the fact that if your data is spiked at zero, you have to be carefoul with the hazard-rate model (detai
ls in Buckland et al. 2001))
best_modRad <- hr_cos0 # AIC 413.44

# Estimating effective detection radius and (SE)
EfecRad <- EDRtransform(best_modRad)

EfecRad$EDR # mean (m)
```

```
## [1] 4.83145
```

```
EfecRad$se.EDR # SE (m)
```

```
##            [,1]
## [1,] 0.8768747
```

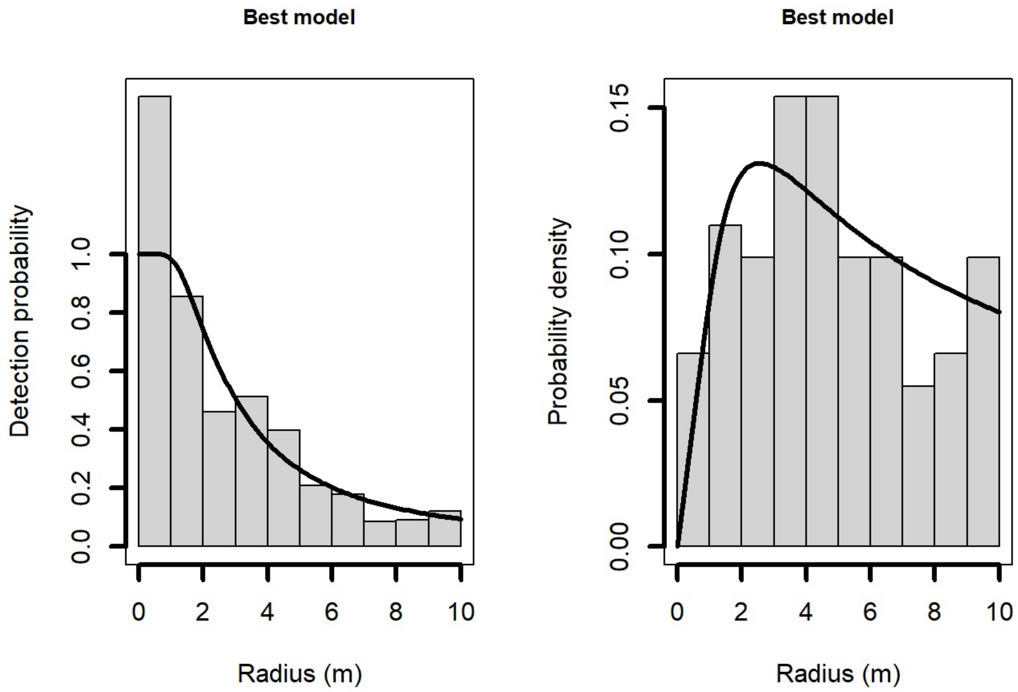

Figure 4: Detection radius plots

Effective detection radius (**Fig. 4**) is 4.83m (SE=0.88) which is consistent with previous studies (e.g. Hofmeester et al. 2017 - Remote Sens. Ecol. Conserv. 3(2):81-89).

3.2.2 Angle

To estimate effective detection angle we will use the column "Angle_det", which include the angle -in degrees- in the first photo of each sequence. When more than one animal appeared in the first photo, we recorded the angle to the closer one. We considered angle=0 in the centrer of the field of view, and we assumed that detection zone is symmetric We will proceed in a similar way than when estimating radius. Now, we apply a line-transect distance sampling. Should be noted that advanced models can be considered by including covariates as sampling point or vegetation cover, betweeen others.

```
data_dz_ang<-subset(dataREM, Ang_det != "NA" ) # selecting data to estimate effective detection angle

data_dz_ang$Ang_rad <- abs(data_dz_ang$Ang_det*0.0174533) # transform degrees to radians
FOV <- 42 # field of view of the cameras (degrees)
w_ang <- FOV/2*0.0174533 # truncation angle (in radians)

# half-normal
hn_cos0Ang <- ds(data_dz_ang$Ang_rad, transect = "line", key="hn", adjustment = "cos", order = 0, truncation=w_ang)
```

```
## Fitting half-normal key function
```

```
## Key only model: not constraining for monotonicity.
```

```
## AIC= -227.147
```

```
## No survey area information supplied, only estimating detection function.
```

```
#hazard-rate
hr_cos0Ang <- ds(data_dz_ang$Ang_rad, transect = "line", key="hr", adjustment = "cos", order = 0, truncation=w_ang)
```

```
## Fitting hazard-rate key function
```

```
## Key only model: not constraining for monotonicity.
```

```
## AIC= -236.431
```

```
## No survey area information supplied, only estimating detection function.
```

Again, all the combinations between 'half-normal' and 'hazard-rate' functions, 'cos', 'herm' and 'poly' adjustments, and orders 0 and 2 should be tested.

After testing all these models, we will select the best one on the basis of AIC:

```
#model comparison
AIC(hn_cos0Ang, hn_cos2Ang, hn_herm0Ang, hn_herm2Ang, hn_poly0Ang, hn_poly2Ang, hr_cos0Ang, hr_cos2Ang, hr_herm
0Ang, hr_herm2Ang, hr_poly0Ang, hr_poly2Ang)
```

```
##              df       AIC
## hn_cos0Ang    1 -227.1472
## hn_cos2Ang    2 -226.4685
## hn_herm0Ang   1 -227.1472
## hn_herm2Ang   2 -225.2827
## hn_poly0Ang   1 -227.1472
## hn_poly2Ang   3 -223.6056
## hr_cos0Ang    2 -236.4309
## hr_cos2Ang    3 -225.4981
## hr_herm0Ang   2 -236.4309
## hr_herm2Ang   3 -223.0488
## hr_poly0Ang   2 -236.4309
## hr_poly2Ang   4 -233.2294
```

```
# select best model
# (mind the fact that if your data is spiked at zero, you have to be carefoul with the hazard-rate model (detai
ls in Buckland et al. 2001))
best_modAng <- hr_cos0Ang  # AIC -208.2947

# Estimating effective detection radius and (SE)
summary_ang<- summary(best_modAng$ddf)

EfecAng_mean <- summary_ang$average.p*w_ang # mean (radians)
EfecAng_SE <-  summary_ang$average.p.se*w_ang # SE (radians)
```

Effective detection radius (**Fig. 5**) is 0.33rad (SE=0.01)

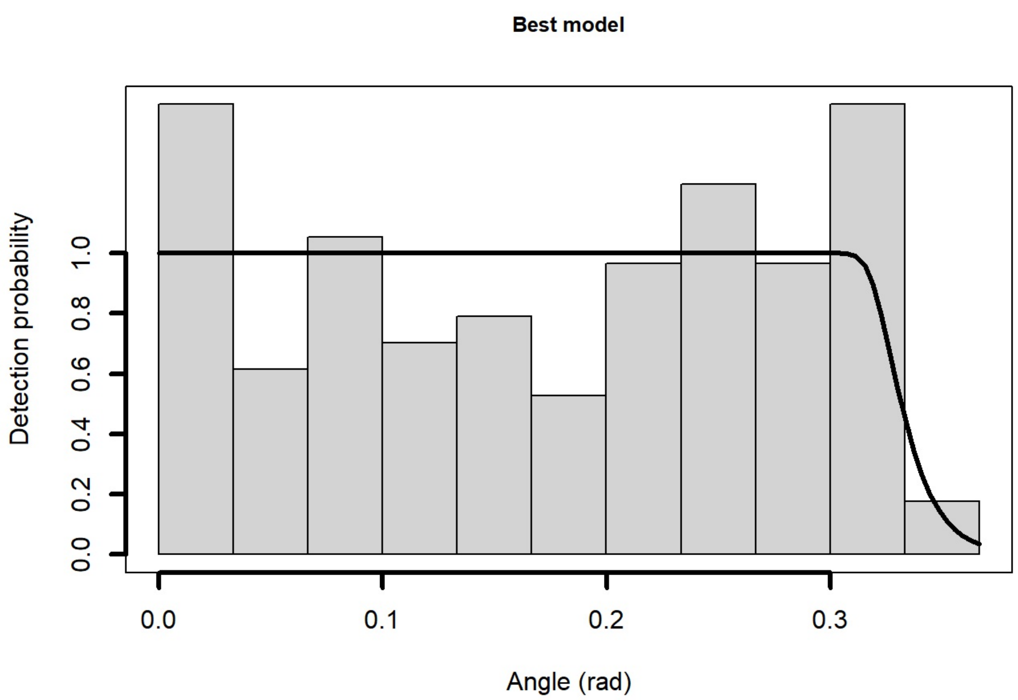

Figure 5: Detection angle plot

Finally, and just to clarify, the interpretation of effective detection radius (or angle) is the threshold value at which the expected number of missed within is equal to the expected number detected beyond.

3.3 Encounter rate

Finally, we will estimate encounter rate. In our dataframe (*dataREM*), each time that an individuals enters in the detecion zone (sequence) we added a row. For that, I considered individuals as the unit of observation. Considering that, we will aggregate the number of rows per each "Point_ID" which is our camera trap ID. Adittionally, we will estimate the sampling effort of each camera trap using the information provided in (*df_coord*).

In summary, we need a dataframe (here *tr*) in wich one column represent the *y* parameter (here *Freq*) and another column represent *t* parameter (here *oper_days*) of the REM equation.

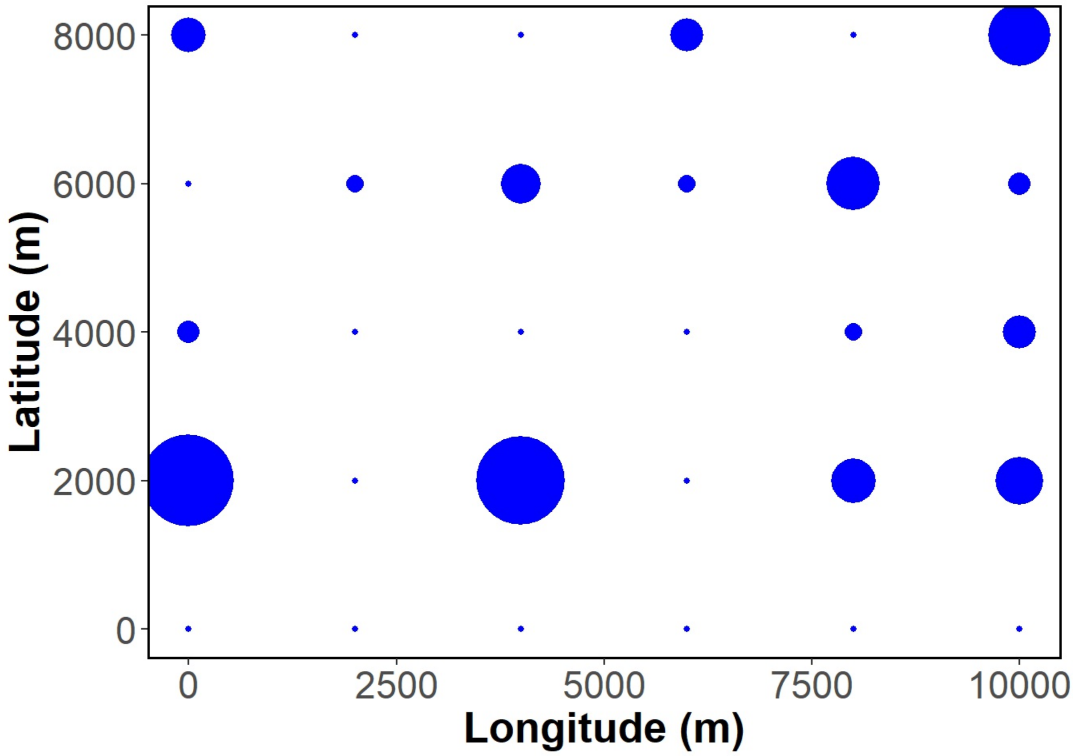

Figure 6: Encounter rate map. The areas of the blue circles are proportional to the encounter rates

```
head(tr) # dataframe to estimate encounter rate
```

```
##   Freq oper_days
## 1    0       151
## 2    0       169
## 3    0       169
## 4    0       169
## 5    0       151
## 6    0       169
```

As **Fig. 6** evidenced, encounter rates are aggregated (and usually overdispersed).In 15 of the 30 cameras, no animals were registered; while in two cameras more that 50 sequences were recorded in each one. This is habitual in REM studies becuase of the random design and the irregular distribution of the animals. The most important practical consideration is that most of the density variance is attributable to the variation in encounter rate between camera traps. In conclusion, to improvements in encounter rate precision is needed to improve density variances estimates.

4. Density results

After estimate all the parameters necessary to apply REM, we can estimate population density. Overall varaince of REM density is estimated using delta method. The variance associated with the encounter rate is estimated by resampling camera locations with replacement 1000 times.

```
# We include in a list average values of day range and detection zone
param <- list(DR = DR,
              r = EfecRad$EDR / 1000,
              theta = EfecAng_mean*2)

# We include in a list standard error values of day range and detection zone
paramse <- list(DR = DR_se,
                r = EfecRad$se.EDR / 1000,
                theta = EfecAng_SE*2)

# Density estimation
density<-bootTRD(tr$Freq, tr$oper_days, param, paramse); density
```

```
##       Density       SE
## [1,] 7.930052 3.304931
```

Finally, the density of this fallow deer population is 7.92 ind/km^2 (SE=3.33). Average and SE errors of REM parameters are included in the dataframe *results*:

```
# Saving results
head(results)
```

```
##       y(seq) t(days) s(km/day) s_se(km/day)    r(m)   r_se(m)  ang(rad)
## Value    229    4808  1.368014    0.1484657 4.83145 0.8768747 0.6678117
##       ang_se(rad) d(ind/km2) d_se(ind/km2)
## Value  0.02610699   7.930052      3.304931
```