

REM analysis vignette

Pablo Palencia

2025-12-01

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 REM has been used for a wide range of species, with densities values (both in terms of precision and accuracy) equivalent to gold-reference methods (Palencia et al., 2022 - Remote Sens. Ecol. Conserv. 8(5): 670-682).

Briefly, the REM is based on modelling the random encounters between animals and cameras, and accounting for all the variables that affect the encounter rate (i.e. animal speed and camera's detection zone).

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 individuals entered detection zone), t is total camera survey effort, v is the average distance traveled 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, we are going to estimate population density of a fallow deer (*Dama dama*) population sampled with 40 camera traps. (Why fallow deer? Easy, is my favorite species, see **Fig. 1**). Survey design consisted in a systematic design with random origin; concretely cameras were deployed at the intersection of a grid with 2km spacing.



Ltl Acorn 041°F 005°C 01/14/2017 09:30:23

Figure 1: Fallow deer picture

It is worth noting that all parameters required to apply the REM (i.e., day range, encounter rate, and detection zone) can be estimated directly from camera-trap data, without relying on auxiliary sources. Although this approach demands additional effort—especially during image processing—it is recommended, as it allows the spatio-temporal variation in day range and the variability in the detection zone to be properly accounted for, including differences related to target species, camera-trap models, settings, and environmental conditions, among others. Finally, in spite of I'm going to use other packages, should be noted that *remBoot* R package implement REM calculations in R (Caravaggi, 2017 - J. Open Source Softw.- 2(10):176).

2. Importing data frames and functions

Three data frames 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). Additionally, we will need to import a couple of functions that are not included in an R package (these functions were developed by M. Rowcliffe and Distance Sampling folks from St. Andrews University).

```

# Load data frames
dataREM <- read.table("Data.txt", sep = ";", dec=".",
header=TRUE, as.is=TRUE) # parameters data frame
operat <- read.table("Operativity.txt", sep = ";", dec=".",
header=TRUE, as.is=TRUE) # operativity 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 data frames (available on CRAN)
library(ggplot2) # to plot encounter rates (available on CRAN)

```

3. Analysis

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

3.1 Day range

Day range is a parameter that relies on animal movement. Recent studies have described a procedure to estimate day range from camera trapping data (Palencia et al. 2021 - Methods Ecol. Evol. 12(7):1201-1212; 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 time in which each encounter was recorded.

```

# Convert time of dataREM to a numeric vector of radian time-of-day
time_rad <- gettime(dataREM$time, format = "%H:%M:%S")

# fit activity model
actmod <- fitact(time_rad, sample="model") #sample=model: Large sample size (greater than 100-200); sample=dataREM: small sample size (less than 100); o sample=none: no bootstrapping.

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

```

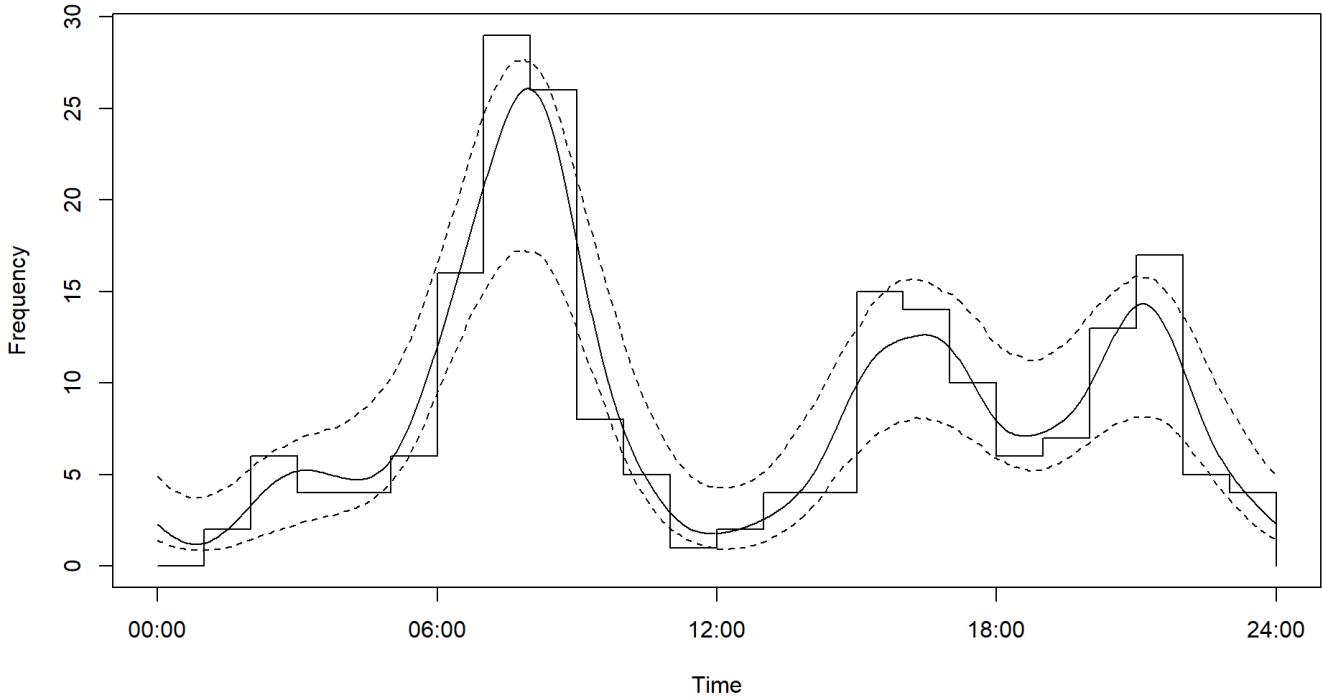


Figure 2: Activity pattern

```
actmod@act[1] # mean activity Level
```

```
##      act
## 0.3322645
```

```
actmod@act[2] # SE activity Level
```

```
##      se
## 0.046652
```

Activity results showed a clear peak around sunrise and two peaks around sunset **Fig. 2.** An activity level of 0.33 indicates that the population was active for 7.92 hours per day (0.33×24).

3.1.2 Speed

To estimate speed, we will use the column “speed”, which include speed estimations in m/s. Using the package *trappingmotion* we will follow the procedure described by Palencia et al. (2021) - Methods Ecol. Evol. 12(7): 1201-1212. 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.

```
# Explore speed distribution. Sometimes it is necessary to remove extreme-low/high values
par(mfrow=c(1,2))
boxplot(dataREM$speed); hist(dataREM$speed)
dataREM.speed <- subset(dataREM, speed < 8)
#dataREM.speed <- subset(dataREM.speed, speed > 0.04)

identbhvs(dataREM.speed$speed) # identify movement states
```

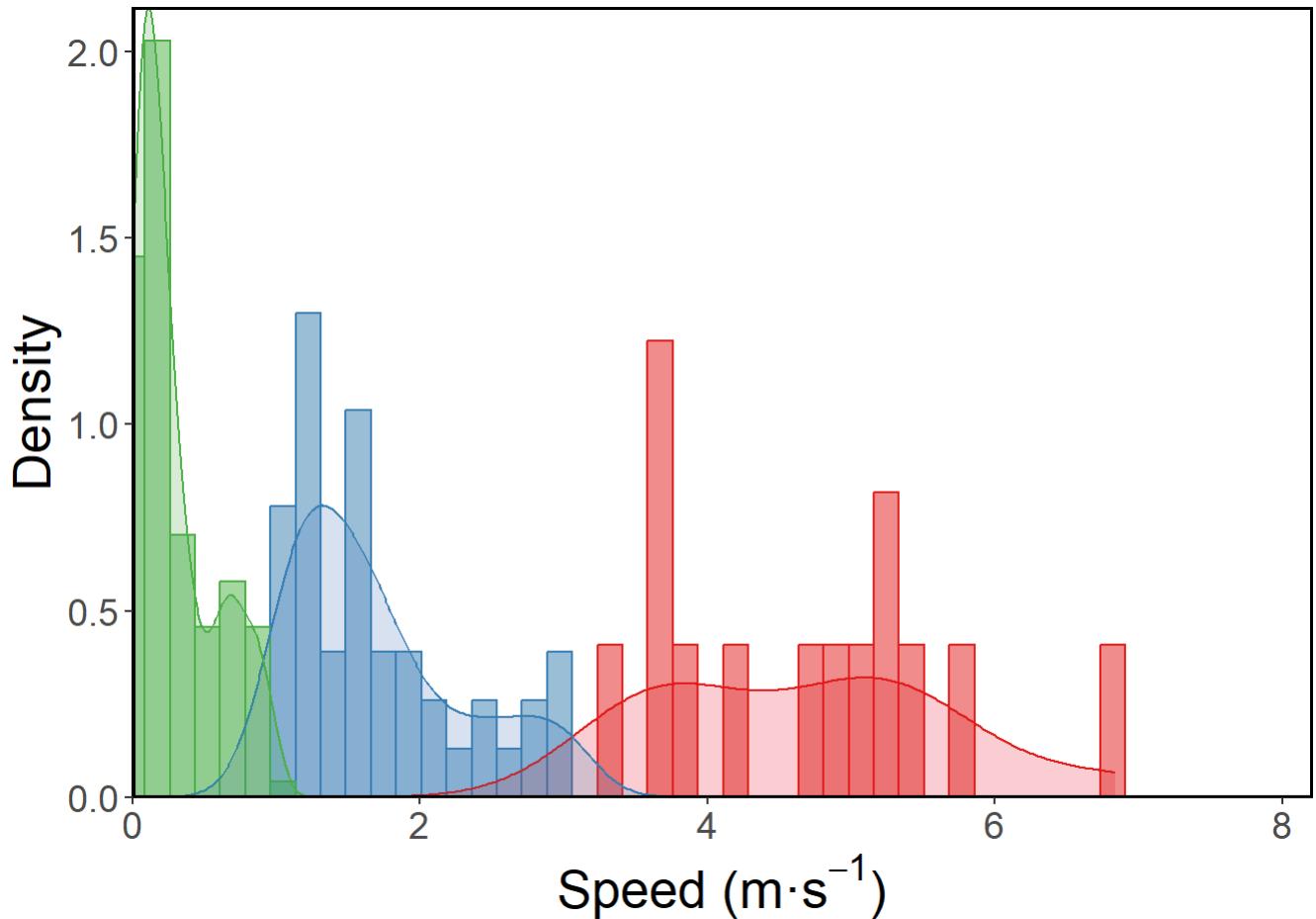


Figure 3: Movement behaviours identified

```
table behav_class$behaviour)
```

```
##  
##   1   2   3  
## 14  44 138
```

```
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=actmod@act[1], act_se=actmod@act[2], speed_data) #day range (daily distance traveled)
```

```
## Day range (Km/day) 4.185252  
## Day range SE (Km/day) 0.5354997
```

The day range estimated is 4.19 km/day (SE=0.54), which means that, in average, each fallow deer in the population travels 4.19 km per day.

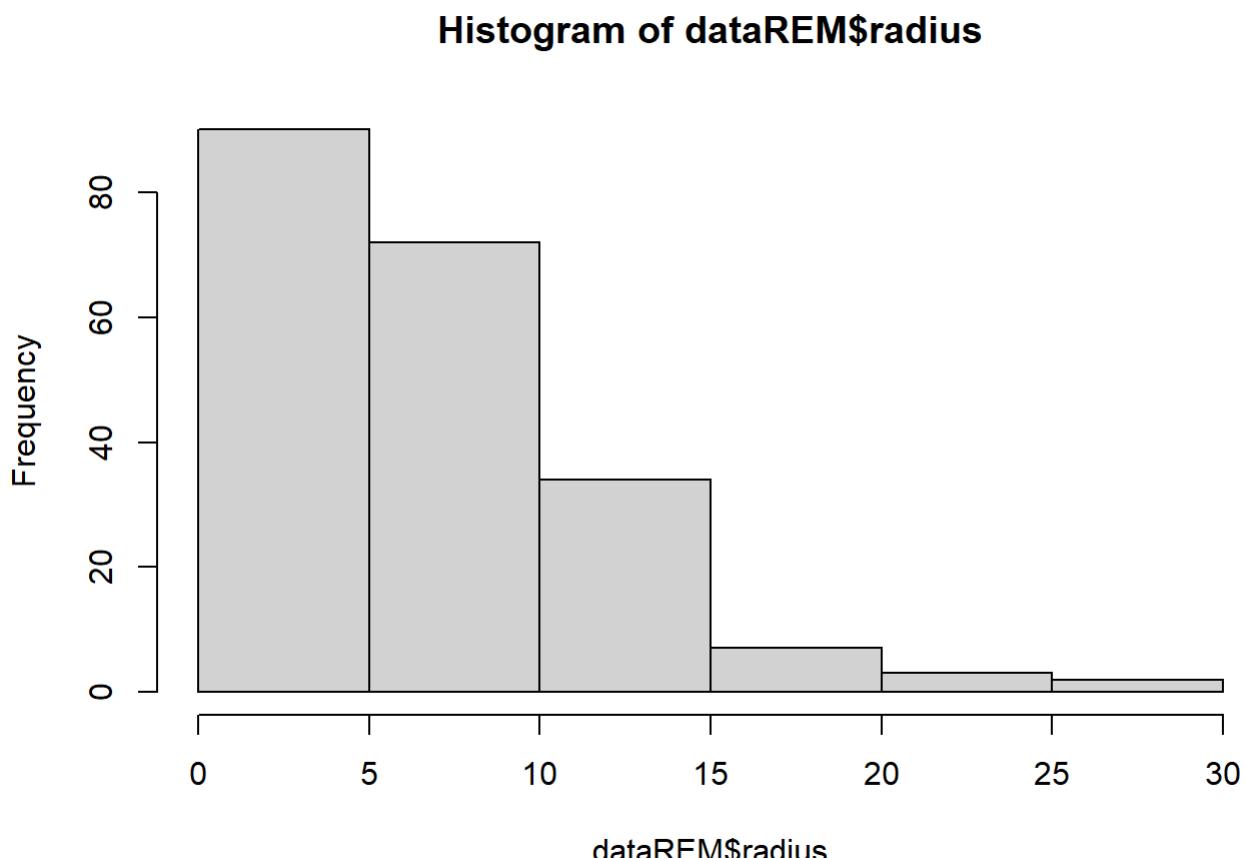
3.2 Detection zone

To estimate the effective detection zone (the area effectively monitored by the cameras), we will follow the procedure described by Rowcliffe et al. (2011) - Methods Ecol. Evol. 2(5):465-476. This approach is based on distance sampling theory (see Buckland et al. 2001 for further details), and therefore requires recording the position (distance and angle relative to the camera) at which fallow deer are first detected. I also recommend consulting Hofmeester et al. (2017 - Remote Sens. Ecol. Conserv. 3(2):81-89) which provides a practical and straightforward method for estimating detection zones when working with camera traps.

3.2.1 Radius

To estimate the effective detection radius, we will use the “radius”, column, which contains the distance (in meters) between the animal and the camera at the moment of entry. We will apply a point-transect distance sampling approach. It should be noted that more advanced models can be used by incorporating covariates such as vegetation density, camera model, etc.

```
# explore and define the truncation distance  
hist(dataREM$radius)
```



```

w_rad <- 10
# half-normal
hn <- ds(dataREM$radius, transect = "point", key="hn", adjustment = NULL, truncation=w_rad)
hn_cos <- ds(dataREM$radius, transect = "point", key="hn", adjustment = "cos", nadj = 1, truncation=w_rad)
hn_herm <- ds(dataREM$radius, transect = "point", key="hn", adjustment = "herm", nadj = 1, truncation=w_rad)
hn_poly <- ds(dataREM$radius, transect = "point", key="hn", adjustment = "poly", nadj = 1, truncation=w_rad)

# hazard-rate
hr <- ds(dataREM$radius, transect = "point", key="hr", adjustment = NULL, truncation=w_rad)
hr_cos <- ds(dataREM$radius, transect = "point", key="hr", adjustment = "cos", nadj = 1, truncation=w_rad)
hr_herm <- ds(dataREM$radius, transect = "point", key="hr", adjustment = "herm", nadj = 1, truncation=w_rad)
hr_poly <- ds(dataREM$radius, transect = "point", key="hr", adjustment = "poly", nadj = 1, truncation=w_rad)

```

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

```

# model selection
AIC(hn, hn_cos, hn_herm, hn_poly, hr, hr_cos, hr_herm, hr_poly)

```

```

##          df      AIC
## hn        1 724.4942
## hn_cos    2 726.4939
## hn_herm   2 726.4613
## hn_poly   2 726.1085
## hr        2 726.1141
## hr_cos    3 727.1004
## hr_herm   3 727.9617
## hr_poly   3 728.1141

```

```

# select best model
# (mind the fact that if your data is spiked at zero, you have to be careful with the hazard-rate model (details in Buckland et al. 2001))
best_modRad <- hn

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

EfecRad$EDR # mean (m)

```

```

## [1] 5.736822

```

```

EfecRad$se.EDR # SE (m)

```

```

##          [,1]
## [1,] 0.2601836

```

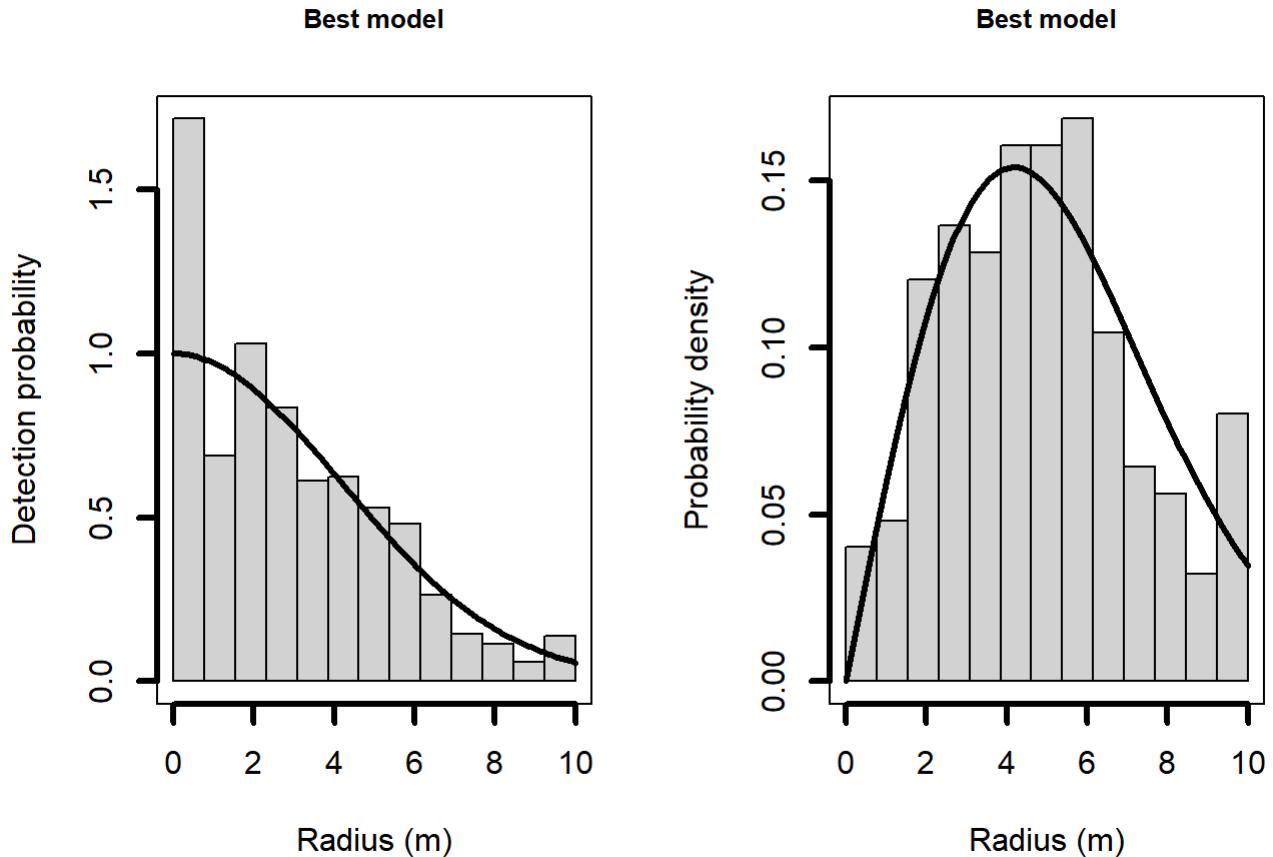


Figure 4: Detection radius plots

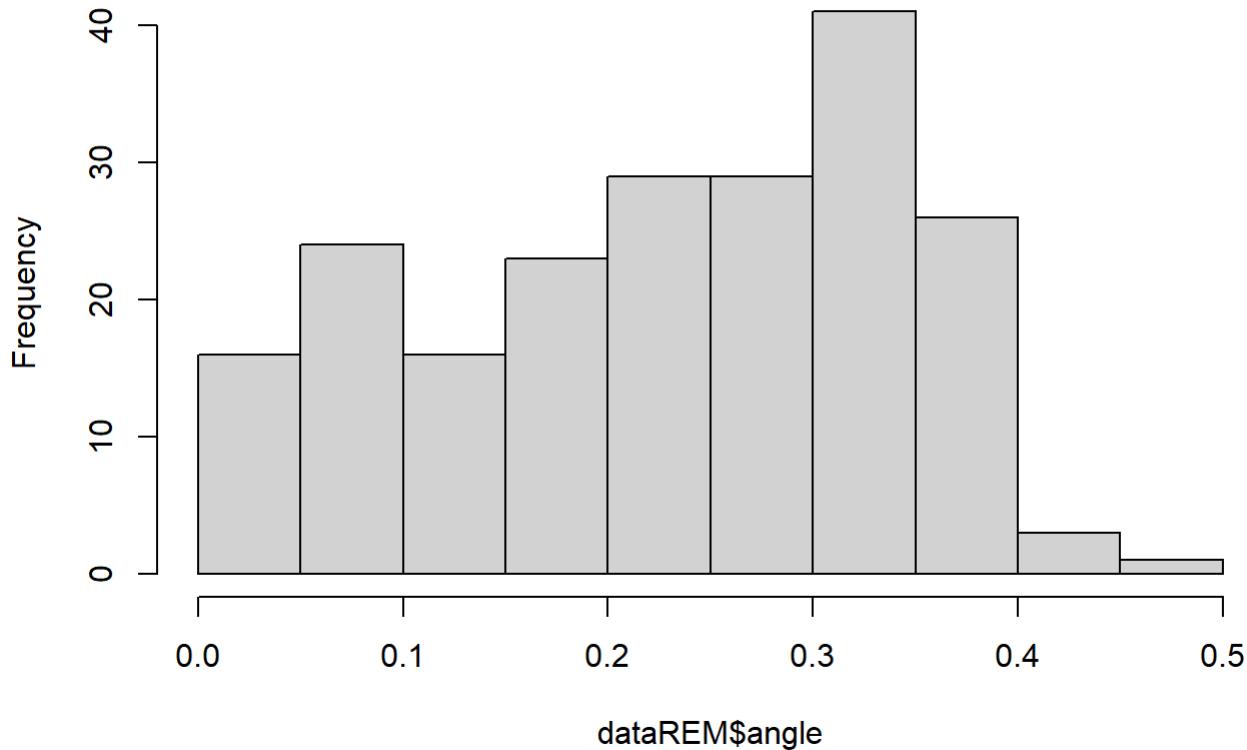
Effective detection radius (**Fig. 4**) is 5.73 m (SE=0.26), which is consistent with previous studies (e.g. Hofmeester et al. 2017 - *Remote Sens. Ecol. Conserv.* 3(2):81-89).

3.2.2 Angle

To estimate the effective detection angle, we will use the “angle”, column, which records the angle (in degrees) at which the individual enters the field of view. We considered angle = 0 to represent the center of the camera’s field of view and assumed that the detection zone is symmetric. The procedure is similar to that used for estimating the detection radius; however, in this case we apply a line-transect distance sampling approach. It should be noted that more complex models can incorporating covariates can be also fitted.

```
dataREM$angle <- abs(dataREM$angle)
hist(dataREM$angle)
```

Histogram of dataREM\$angle



```
w_ang <- 0.41 # truncation angle in radians (=23.5 grados)

# half-normal
hn_Ang <- ds(dataREM$angle, transect = "line", key="hn", adjustment = NULL, truncation=w_ang)
hn_cosAng <- ds(dataREM$angle, transect = "line", key="hn", adjustment = "cos", nadj = 1, truncation=w_ang)
hn_hermAng <- ds(dataREM$angle, transect = "line", key="hn", adjustment = "herm", nadj = 1, truncation=w_ang)
hn_polyAng <- ds(dataREM$angle, transect = "line", key="hn", adjustment = "poly", nadj = 1, truncation=w_ang)

# hazard-rate
hr_Ang <- ds(dataREM$angle, transect = "line", key="hr", adjustment = NULL, truncation=w_ang)
hr_cosAng <- ds(dataREM$angle, transect = "line", key="hr", adjustment = "cos", nadj = 1, truncation=w_ang)
hr_hermAng <- ds(dataREM$angle, transect = "line", key="hr", adjustment = "herm", nadj = 1, truncation=w_ang)
hr_polyAng <- ds(dataREM$angle, transect = "line", key="hr", adjustment = "poly", nadj = 1, truncation=w_ang)

# uniform
uni_cosAng <- ds(dataREM$angle, transect = "line", key="uni", adjustment = "cos", nadj = 1, truncation=w_ang)
uni_hermAng <- ds(dataREM$angle, transect = "line", key="uni", adjustment = "herm", nadj = 1, truncation=w_ang)
uni_polyAng <- ds(dataREM$angle, transect = "line", key="uni", adjustment = "poly", nadj = 1, truncation=w_ang)
```

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

```
# model comparison
AIC(hn_Ang, hn_cosAng, hn_hermAng, hn_polyAng, hr_Ang, hr_cosAng, hr_hermAng, hr_polyAng, uni_cosAng, uni_hermAng, uni_polyAng)
```

```
##          df      AIC
## hn_Ang     1 -365.3384
## hn_cosAng   2 -363.3381
## hn_hermAng  2 -363.3381
## hn_polyAng  2 -363.3381
## hr_Ang      2 -366.1389
## hr_cosAng    3 -362.3197
## hr_hermAng   3 -363.0055
## hr_polyAng   3 -363.0055
## uni_cosAng   1 -365.3384
## uni_hermAng  1 -365.3384
## uni_polyAng  1 -365.3384
```

```
# select best model
best_modAng <- uni_cosAng

# 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 0.41 rad (SE=0.04) (i.e. 23.49 degrees).

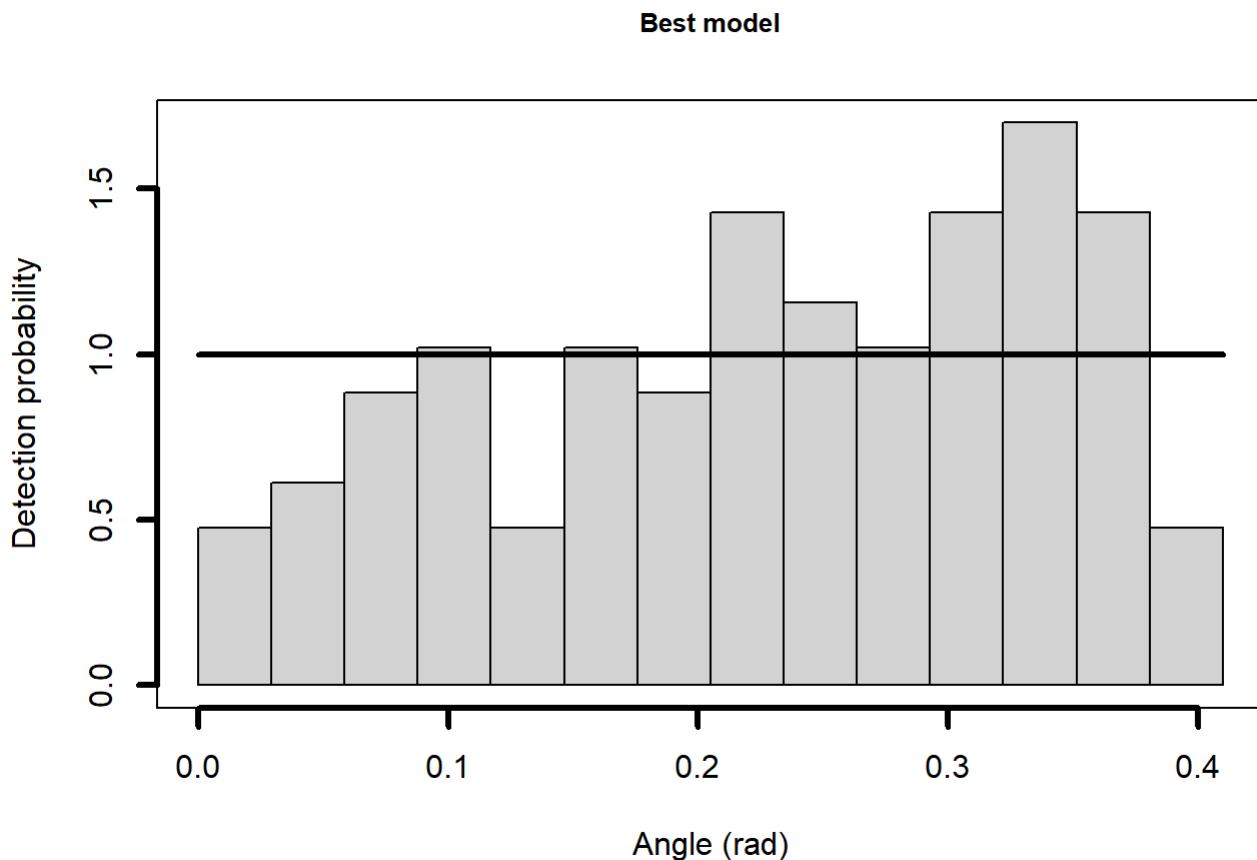


Figure 5: Detection angle plot

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 (Buckland et al. 2001).

3.3 Encounter rate

Finally, we will estimate the encounter rate. In our data frame (*dataREM*), a new row was added each time an individual entered the detection zone (i.e., each encounter), with individuals treated as the unit of observation. Based on this, we will aggregate the number of rows for each (*dataREM\$camera*). In addition, the sampling effort for each camera trap will be estimated using the information provided in (*operat*), which records the operational period of each camera.

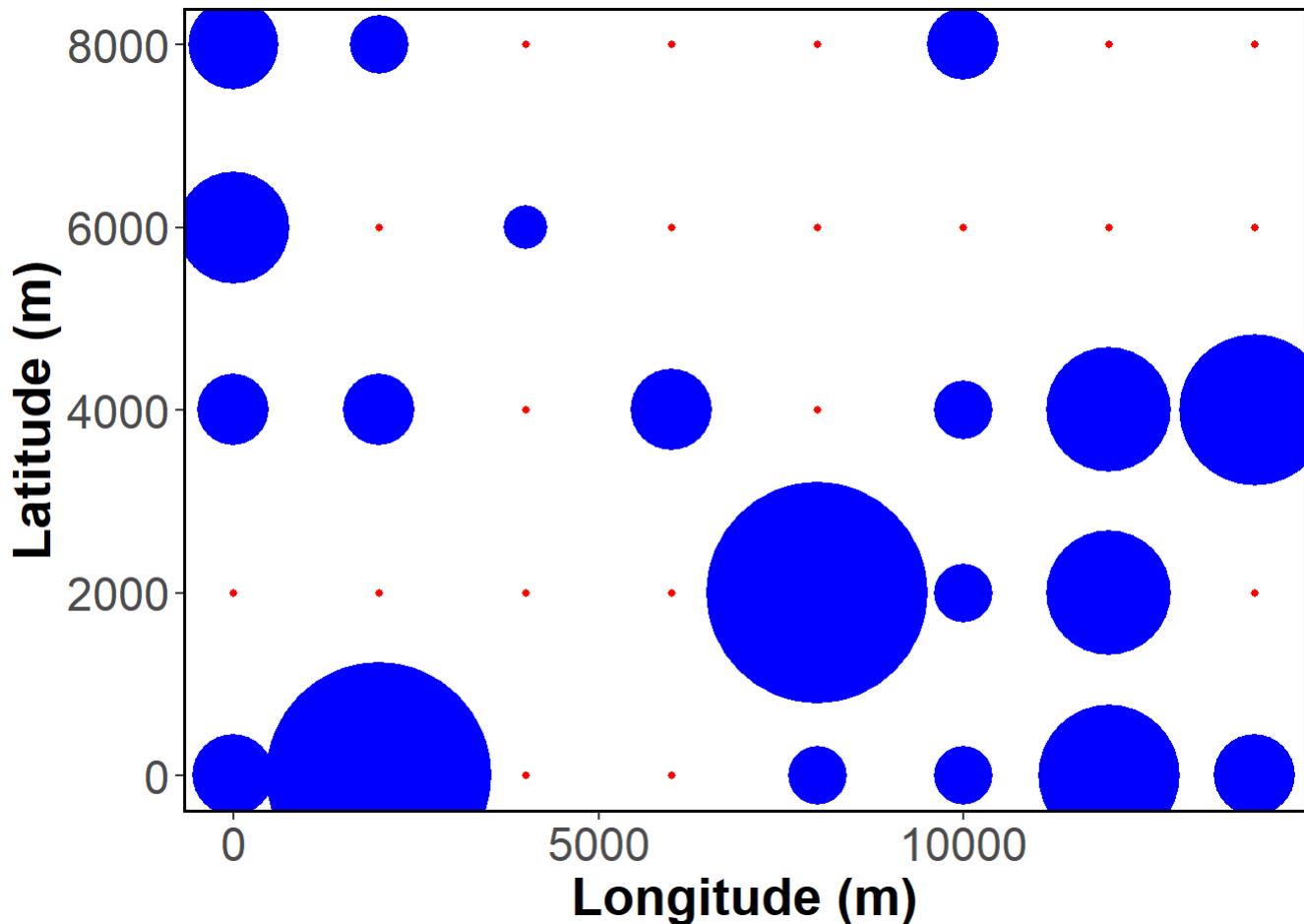


Figure 6: Encounter rate map. The areas of the blue circles are proportional to the encounter rates

In summary, we need a data frame (here *enc_rate*) in which one column represent the *y* parameter (here *Freq*) and another column represent *t* parameter (here *days*) of the REM equation.

```
head(enc_rate) # data frame to estimate encounter rate
```

```
##   Freq days
## 1    4   60
## 2   34   60
## 3    0   40
## 4    0   60
## 5    2   60
## 6    2   60
```

As **Fig. 6** shows, encounter rates are highly aggregated. In 20 of the 40 cameras, no animals were detected, whereas in two cameras more than 30 sequences were recorded at each site. This pattern is common in REM studies due to the random placement of cameras and the naturally uneven distribution of animals. Crucially, most of the variance in density estimates arises from the variation in encounter rates among camera traps.

4. Density results

After estimate all the parameters necessary to apply REM, we can estimate population density.

```
# 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(enc_rate$Freq, enc_rate$days, param, paramse); density
```

```
##      Density      SE
## [1,] 3.347081 1.096374
```

```
# CV
density[,2]/density[,1]
```

```
##      SE
## 0.3275611
```

```
# Log-normal CIs
df_lnorm_confint <- lnorm_confint(density[,1], density[,2])
```

Finally, the density of this fallow deer population is $3.35 \text{ ind}/\text{km}^2$ ($\text{SE}=1.09$). Average and SE errors of REM parameters are included in the data frame *results*:

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

```
##      seq   tm dr(km/day) dr_se(km/day)      r(m)    r_se(m) ang(rad) ang_se(rad)
## Value 160 2218     4.185252     0.5354997 5.736822 0.2601836     0.82  0.08185672
##          d(ind/km2) d_se(ind/km2) lcl_lognorm95% ucl_lognorm95%
## Value    3.347081      1.096374      1.790273      6.257679
```