Appendix 2

Territoriality, breeding philopatry, and nest site selection of Chestnut Seedeater (*Sporophila cinnamomea*) in grasslands of southern Brazil

```
################## Home Ranges and UD estimators ###############
# Set your working directory
setwd ("<path to chosen working directory")
# Load the required libraries
library (adehabitatHR)
library (sp)
library (mapview)
library (maptools)
# Load database
xy_1819_utm <- read.csv ('2018_2019_utm.csv', header=T, sep=",")
xy_1920_utm <- read.csv ('2019_2020_utm.csv', header=T, sep=",")
xy_1819_dec <- read.csv ('2018_2019_dec.csv', header=T, sep=",")
xy_1920_dec <- read.csv ('2019_2020_dec.csv', header=T, sep=",")
# Validating numbers as coordinates
coordinates (xy_1819_utm) <- ~ y + x # It corresponds to the column name
coordinates (xy_1920_utm) <- ~ y + x
# Adding a Coordinate Reference System - SRC
proj4string (xy_1819_utm) <- CRS ("+proj=utm +zone=21 +south +ellps=GRS80")
proj4string (xy_1920_utm) <- CRS ("+proj=utm +zone=21 +south +ellps=GRS80")
#################### Validating lat long ####################
####################### 2018/2019 ####################
coordinates (xy_1819_dec) <- ~ Long + Lat # Changing the position of the columns
# Adding a Coordinate Reference System - SRC
proj4string (xy_1819_dec) <- CRS ("+proj=longlat +ellps=GRS80
                +towgs84=0,0,0,0,0,0,0 +no_defs") #SIRGAS 2000
# Calling an object
xy_1819_dec
################## 2019/2020 ####################
coordinates (xy_1920_dec) <- ~ Long + Lat # Changing the position of the columns.
# Adding a Coordinate Reference System - SRC
proj4string (xy_1920_dec) <- CRS ("+proj=longlat +ellps=GRS80
                +towgs84=0,0,0,0,0,0,0 +no_defs") #SIRGAS 2000
```

```
# Calling an object
xy_1920_dec
############## Kernel Density Estimation KDE ###############
###### First, to build the KDE with smoothing factor href para 99%, 95% and 50% with
contour
###### isolines using the kernelUD() and getverticeshr() function
################################################################################
#####   Breeding season 2018-2019   #####
################################################################################
# The href was previously used to generate all reference bandwidth, and after obtaining the
# parameter's value. The median was obtained to be used as the "population smoothing
# parameter."
Khref <- kernelUD (xy_1819_utm, h = 15.92, grid = 800, extent = 2.2)
Khref10 <- getverticeshr (Khref, 10, unin = "m", unout = "ha")
Khref20 <- getverticeshr (Khref, 20, unin = "m", unout = "ha")
Khref30 <- getverticeshr (Khref, 30, unin = "m", unout = "ha")
Khref40 <- getverticeshr (Khref, 40, unin = "m", unout = "ha")
Khref50 <- getverticeshr (Khref, 50, unin = "m", unout = "ha")
Khref60 <- getverticeshr (Khref, 60, unin = "m", unout = "ha")
Khref70 <- getverticeshr (Khref, 70, unin = "m", unout = "ha")
Khref80 <- getverticeshr (Khref, 80, unin = "m", unout = "ha")
Khref90 <- getverticeshr (Khref, 90, unin = "m", unout = "ha")
Khref95 <- getverticeshr (Khref, 95, unin = "m", unout = "ha")
# Finding the value of the parameter used for each animal
Khref[["name"]]@h
# Save value in .csv
write.csv (as.data.frame (Khref10), 'namespecies_10 2018_2019.csv')
write.csv (as.data.frame (Khref20), 'namespecies_20 2018_2019.csv')
write.csv (as.data.frame (Khref30), 'namespecies_30 2018_2019.csv')
write.csv (as.data.frame (Khref40), 'namespecies_40 2018_2019.csv')
write.csv (as.data.frame (Khref50), 'namespecies_50 2018_2019.csv')
write.csv (as.data.frame (Khref60), 'namespecies_60 2018_2019.csv')
write.csv (as.data.frame (Khref70), 'namespecies_70 2018_2019.csv')
write.csv (as.data.frame (Khref80), 'namespecies_80 2018_2019.csv')
write.csv (as.data.frame (Khref90), 'namespecies_90 2018_2019.csv')
write.csv (as.data.frame (Khref95), 'namespecies_95 2018_2019.csv')
# Obtaining the areas for each of the calculated values
```

```
kernelareashref_10_90 <- kernel.area (Khref, percent = seq (10,90, by=10),
                        unin = "m", unout = "ha")
# Obtaining the area for 95% calculated values
kernelareashref_95 <- kernel.area (Khref, percent = c(95), unin = "m", unout = "ha")
# Save value in .csv
write.csv (as.data.frame (kernelareashref_10_90), 'kernelareashref 2018_2019_10_90.csv')
# Save value in .csv
write.csv (as.data.frame (kernelareashref_95), 'kernelareashref 2018_2019_95.csv')
# Plotting the limits for each percentage
mapview (Khref95) + mapview (Khref90) + mapview (Khref80) + mapview (Khref70) +
mapview (Khref60) + mapview (Khref50) + mapview (Khref40) + mapview (Khref30) +
mapview (Khref20) + mapview (Khref10) + mapview (xy_1819_utm)
# Save shape file
writePolyShape (Khref95,'Khref95 2018_2019')
writePolyShape (Khref90,'Khref90 2018_2019')
writePolyShape (Khref80,'Khref80 2018_2019')
writePolyShape (Khref70,'Khref70 2018_2019')
writePolyShape (Khref60,'Khref60 2018_2019')
writePolyShape (Khref50,'Khref50 2018_2019')
writePolyShape (Khref40,'Khref40 2018_2019')
writePolyShape (Khref30,'Khref30 2018_2019')
writePolyShape (Khref20,'Khref20 2018_2019')
writePolyShape (Khref10,'Khref10 2018_2019')
# Counting the number of points per estimated polygon
ndpontos <- over (xy_1819_utm, Khref10) # 10 to 95
table (ndpontos$id)
# Overview the lines in different % and cores
mapview (Khref95, col.regions = "red", map.type = "OpenStreetMap") + mapview (Khref90,
col.regions = "cyan1") + mapview (Khref80, col.regions = "blue") + mapview (Khref70,
col.regions = "green") + mapview (Khref60, col.regions = "yellow") + mapview (Khref50,
col.regions = "pink") + mapview (Khref40, col.regions = "orange") + mapview (Khref30,
col.regions = "white") + mapview (Khref20, col.regions = "grey") + mapview (Khref10,
col.regions = "chocolate1") + mapview (xy_1819_utm, alpha = 0.1, cex = 3)
##############################################################################
#####   Breeding season 2019-2020   #####
##############################################################################
Khref <- kernelUD (xy_1920_utm, h = 15.92, grid = 800, extent = 2.2)
```

```
Khref10 <- getverticeshr (Khref, 10, unin = "m", unout = "ha")
Khref20 <- getverticeshr (Khref, 20, unin = "m", unout = "ha")
Khref30 <- getverticeshr (Khref, 30, unin = "m", unout = "ha")
Khref40 <- getverticeshr (Khref, 40, unin = "m", unout = "ha")
Khref50 <- getverticeshr (Khref, 50, unin = "m", unout = "ha")
Khref60 <- getverticeshr (Khref, 60, unin = "m", unout = "ha")
Khref70 <- getverticeshr (Khref, 70, unin = "m", unout = "ha")
Khref80 <- getverticeshr (Khref, 80, unin = "m", unout = "ha")
Khref90 <- getverticeshr (Khref, 90, unin = "m", unout = "ha")
Khref95 <- getverticeshr (Khref, 95, unin = "m", unout = "ha")
# Finding the value of the parameter used for each animal
Khref[["name"]]@h
# Save value in .csv
write.csv (as.data.frame (Khref10), 'namespecies_Khref10 2019_2020.csv')
write.csv (as.data.frame (Khref20), 'namespecies_Khref20 2019_2020.csv')
write.csv (as.data.frame (Khref30), 'namespecies_Khref30 2019_2020.csv')
write.csv (as.data.frame (Khref40), 'namespecies_Khref40 2019_2020.csv')
write.csv (as.data.frame (Khref50), 'namespecies_Khref50 2019_2020.csv')
write.csv (as.data.frame (Khref60), 'namespecies_Khref60 2019_2020.csv')
write.csv (as.data.frame (Khref70), 'namespecies_Khref70 2019_2020.csv')
write.csv (as.data.frame (Khref80), 'namespecies_Khref80 2019_2020.csv')
write.csv (as.data.frame (Khref90), 'namespecies_Khref90 2019_2020.csv')
write.csv (as.data.frame (Khref95), 'namespecies_Khref95 2019_2020.csv')
# Obtaining the areas for each of the calculated values
kernelareashref_10_90 <- kernel.area (Khref, percent = seq (10,90, by=10),
                                      unin = "m", unout = "ha")
# Obtaining the area for 95% calculated values
kernelareashref_95 <- kernel.area (Khref, percent = c(95), unin = "m", unout = "ha")
# Save value in .csv
write.csv (as.data.frame (kernelareashref_10_90),
        'kernelareashref 2019_2020_10_90.csv')
# Save value in .csv
write.csv (as.data.frame (kernelareashref_95),
        'kernelareashref 2019_2020_95.csv')
# Plotting the limits for each percentage
```

```
mapview (Khref95) + mapview (Khref90) + mapview (Khref80) + mapview (Khref70) +
mapview (Khref60) + mapview (Khref50) + mapview (Khref40) + mapview (Khref30) +
mapview (Khref20) + mapview (Khref10) +mapview (xy_1920_utm)
# Save shape file
writePolyShape (Khref95,'Khref95 2019_2020')
writePolyShape (Khref90,'Khref90 2019_2020')
writePolyShape (Khref80,'Khref80 2019_2020')
writePolyShape (Khref70,'Khref70 2019_2020')
writePolyShape (Khref60,'Khref60 2019_2020')
writePolyShape (Khref50,'Khref50 2019_2020')
writePolyShape (Khref40,'Khref40 2019_2020')
writePolyShape (Khref30,'Khref30 2019_2020')
writePolyShape (Khref20,'Khref20 2019_2020')
writePolyShape (Khref10,'Khref10 2019_2020')
# Counting the number of points per estimated polygon
ndpontos <- over (xy_1920_utm, Khref40)
table (ndpontos$id)
# Overview the lines in different % and cores
mapview (Khref95, col.regions = "red", map.type = "OpenStreetMap") + mapview (Khref90,
col.regions = "cyan1") + mapview (Khref80, col.regions = "blue") + mapview (Khref70,
col.regions = "green") + mapview (Khref60, col.regions = "yellow") + mapview (Khref50,
col.regions = "pink") + mapview (Khref40, col.regions = "orange") + mapview (Khref30,
col.regions = "white") + mapview (Khref20, col.regions = "grey") + mapview (Khref10,
col.regions = "chocolate1") + mapview (xy_1920_utm, alpha = 0.1, cex = 3)
```