Introduction to spatsoc

Alec Robitaille

2017-09-13

The spatsoc package aims to provide functionality for analysing animal location data in time and space to find potential interactions or social groups.

The package contains a set of Group\* functions that are used for finding groups in input data depending on spatial location and (optionally) time. In addition, the package contains functions for determining the nearest neighbor to each individual, as well as the mean pairwise distance between a set of individuals (both optionally across time).

## Functions

* Grouping
  + GroupPts
  + GroupLines
  + GroupHRs / GroupPolys
  + GroupClusters
  + CalcCentroids
  + GroupTime
* Mean pairwise distance
  + PairwiseDist
* Nearest neighbor
  + Nearest
* Randomization of IDs within groups
  + Randomizations
* Helper functions
  + BuildPts
  + BuildLines
  + BuildHRs
  + BuildClusters
  + RoundTime

## Grouping

Let's read in spatsoc's example data.

library(spatsoc); library(data.table)  
data(locs)  
locs[1:5]

## ID idate itime X\_COORD Y\_COORD EASTING NORTHING  
## 1: ind1 2010-01-16 00:00:49 -54.54954 48.17941 682152.7 5339145  
## 2: ind1 2010-01-16 05:00:54 -54.55161 48.17919 681999.1 5339116  
## 3: ind1 2010-01-16 10:00:50 -54.55159 48.17925 682000.9 5339122  
## 4: ind1 2010-01-16 15:00:55 -54.56311 48.17490 681159.4 5338612  
## 5: ind1 2010-01-16 20:00:54 -54.57921 48.15350 680037.8 5336195

The data consists of the location of 23 individuals over 15 days. The location was recorded using GPS collars with two-hour fixes. Using this data, we can compare the various grouping methods available in spatsoc.

### GroupPts

The GroupPts function compares the location of all individuals at each time step in the input data and groups those within a provided distance.

Before grouping, we will ensure that the time field we intend to group on is suitable. First, we will check how many locations are taken more than 5 minutes off of the nearest hour.

locs[minute(itime) > 5, .N]

## [1] 20

locs[minute(itime) > 5]

## ID idate itime X\_COORD Y\_COORD EASTING NORTHING  
## 1: ind1 2010-01-30 05:40:43 -54.63622 48.06663 676094.8 5326409  
## 2: ind5 2010-01-29 00:40:54 -54.65353 48.03558 674910.3 5322918  
## 3: ind6 2010-01-17 00:40:50 -54.98782 47.89605 650397.7 5306706  
## 4: ind6 2010-01-19 20:40:54 -54.93632 47.95971 654058.0 5313884  
## 5: ind9 2010-01-31 15:40:54 -54.67109 48.07432 673471.7 5327184  
## 6: ind10 2010-01-27 10:41:23 -55.23654 48.26501 630868.4 5347258  
## 7: ind11 2010-01-18 10:40:42 -54.87401 48.01855 658528.9 5320549  
## 8: ind11 2010-01-23 15:50:50 -55.27833 48.17091 628001.7 5336730  
## 9: ind11 2010-01-27 15:50:55 -54.94322 48.00784 653400.0 5319218  
## 10: ind12 2010-01-16 00:40:26 -54.67240 48.16536 673067.9 5337299  
## 11: ind12 2010-01-22 10:40:50 -54.91741 48.28590 654488.1 5350174  
## 12: ind12 2010-01-28 05:50:47 -54.63666 48.07749 676025.0 5327615  
## 13: ind14 2010-01-22 20:40:53 -55.26529 48.30066 628645.6 5351173  
## 14: ind14 2010-01-28 00:50:47 -55.26328 48.23642 628956.3 5344036  
## 15: ind17 2010-01-25 10:40:47 -54.66341 48.06641 674070.3 5326322  
## 16: ind18 2010-01-31 15:50:53 -55.04466 48.05157 645711.4 5323882  
## 17: ind20 2010-01-21 05:40:49 -54.60086 48.15089 678437.1 5335855  
## 18: ind21 2010-01-18 00:40:47 -55.26763 48.32441 628412.5 5353808  
## 19: ind22 2010-01-24 10:40:48 -54.75892 48.02256 667097.1 5321238  
## 20: ind22 2010-01-26 00:40:25 -54.76696 48.01598 666518.7 5320489

If we want to group individuals in time and space, it is clear that these 20 locations may not be relevant to those recorded ~40-50 minutes earlier. We will round the times to the nearest hour using RoundTime and returning them as.POSIXct datetimes if a date field is provided. Alternatively, if the temporal scale of your input data is different, time fields can be rounded using different round units such as '15 minutes' or '4 hours'.

locs[, roundDateTime := RoundTime(.SD, 'itime', roundUnit = '1 hour', dateField = 'idate')]

We can then use the GroupPts function to find overlapping spatial location of buffered points. Here, we buffer each point by 50m to match temporal and spatial overlap of potential interation or social group.

GroupPts(locs, 50, timeField = 'roundDateTime',   
 projection = '+proj=utm +zone=21 ellps=WGS84',   
 idField = 'ID', coordFields = c('EASTING', 'NORTHING'))

## Notes

Package dependencies for spatsoc are primarily sp, rgeos, and data.table. data.table provides efficient methods for manipulating large (or small) datasets. As a result, input DT for all spatsoc functions must be a data.table and if it isn't, you can simply use setDT(df) to convert it by reference.

In addition, since the rgeos package is used in most functions (notably GroupPts and GroupLines) the input DT's coordinate system is important. rgeos expects planar coordinates and this requirement is carried forward for spatsoc.