

Measuring Spatial Distribution using Activity Space

Introduction

The activity space is the concept of a social place that individuals visit on a daily basis, and the concept of an activity space itself can be thought of a lot on a daily basis, but it has not been dealt with largely in the study. However, many studies have recently been conducted in that the analysis based on the activity space can realistically observe the segregation and disconnection experienced by social members in real life compared to the existing residential-centered approach(Wong and Shaw, 2011; Farber et al., 2012; ang et al., 2012; Wang and Li, 2016). However, this activity-space-based research has several problems to be overcome. In order to deal with the activity space, it is necessary to deal with data in a more complex form than the existing data, which causes difficulties in analysis and takes a lot of time.

The slice package was developed to make it easier to handle activity space data in R in this flow. The slice package can find out what area people are located at a specific time based on people's activity space, based on each travel record, and has a function that makes it easier to handle such travel records.

In this Vignette, we will use the 2016 Korea Transport Database of South Korea to deal with how and where people are distributed in each time zone through slice package.

```
library(devtools)
#> Warning: package 'devtools' was built under R version 3.6.3
#> Warning: package 'usethis' was built under R version 3.6.3
load_all()
#> Warning: package 'sp' was built under R version 3.6.3
```

Classes

Before actually starting, I will explain the two classes included in the package with data. The slice package has two classes, the names of these two classes are ASpace and ASpaces, respectively. First, let's proceed with the explanation while looking at the data.

```
data("Krtdata")

class(korea2016)
#> [1] "ASpaces"
#> attr(,"package")
#> [1] "slice"
##ASpaces class
korea2016@data[[1]]
#> An object of class "ASpace"
#> Slot "info":
#> $id
#> [1] "101771"
#>
#> $pid
#> [1] 1
#>
#> $famrel
#> [1] 1
```

```

#>
#> $yrborn
#> [1] 1964
#>
#> $sex
#> [1] 1
#>
#> $area
#> [1] "1123074000"
#>
#> $income
#> [1] 5
#>
#>
#> Slot "trip":
#>   tr_id tr_seq purpose mode o_type o_time   o_zone d_type d_time   d_zone
#> 1     1     1       4     2     1    800 1123074000     2    804 1123065000
#> 2     2     1       7     2     1   1400 1123074000     4   1430 1174068500
korea2016@attr
#> $name
#> character(0)
#>
#> $date
#> [1] "2021-01-20 20:06:50 KST"
#>
#> $desc
#> character(0)
##Not run
#korea2016@sp

```

First is the ASpaces class. The ASpaces class has a total of three slots, each of which is 'data', 'attr', and 'sp'.

- The slot 'data' contains activity space information for us to analyze, and each one is entered into the list inside the data slot.
- The slot 'attr' is the metadata of the ASpaces class, and contains the time when the data was collected or the name of the data.
- Finally, the slot 'sp' contains SpatialPolygon or SpatialPoint that contains the spatial extent of the activity space.

Note that the space unit or spatial range of the data inside the sp slot must match the data inside the data slot.

```

class(korea2016@data[[1]])
#> [1] "ASpace"
#> attr(,"package")
#> [1] "slice"
##ASpace class
korea2016@data[[1]]@info
#> $id
#> [1] "101771"
#>
#> $pid
#> [1] 1
#>
#> $famrel
#> [1] 1

```

```

#>
#> $yrborn
#> [1] 1964
#>
#> $sex
#> [1] 1
#>
#> $area
#> [1] "1123074000"
#>
#> $income
#> [1] 5
korea2016@data[[1]]@trip
#>   tr_id tr_seq purpose mode o_type o_time   o_zone d_type d_time   d_zone
#> 1     1     1       4     2     1    800 1123074000     2    804 1123065000
#> 2     2     1       7     2     1   1400 1123074000     4   1430 1174068500

```

Next is the ASpace class. Unlike the ASpaces class, ASpace has one active space and demographic data, and has a total of two slots. Each slot is Info and Trip.

- The slot 'info' contains demographic and socioeconomic variables of the respondent as a list object.
- The slot 'trip' contains information on trips made by the survey respondent.

In other words, if you briefly describe these two classes, you can say that the ASpace class is a class that has data for each person, and ASpaces is a class that has multiple data as its name suggests.

When looking at the data inside ASpace, the data starts with each individual number and has demographic characteristics such as gender and income level in the residential area, and inside the trip, it is possible to check the characteristics associated with travel.

Note that we have looked at the data, let's see how we can actually analyze the activity space based on this class.

Measuring spatial distribution

First of all, in the simplest way, it will be possible to measure whether the degree of distribution of people can vary by time zone. As the name suggests, the slice function slices consecutive data in time and allows you to see a section of time.

The slice function can be easily used as follows.

- `slice(x, at)`

In this syntax, enter the ASpaces class for x and enter your desired time for at. Like this, if you want to check the population distribution at 2 p.m. of the data you have, the function can be entered as follows.

- `slice(korea2016, 1400)`

Of course, you can simply input this, but it is expected that the operation time will take a long time due to the size of the data, so let's make it possible to perform faster operation by entering an additional statement.

It was adjusted to enable parallelized operation through the mc argument, and the function was controlled to operate in a total of 4 cores through the 'core' argument. The result appears as follows.

```

library(parallel)
all1400 <- slice(korea2016, 1400, silent = TRUE, mc = TRUE, core = 4)
head(all1400)
#>      id pid famrel yrborn sex      area income  location purpose mode

```

```

#> 1 101771 1 1 1964 1 1123074000 5 1123065000 4 2
#> 2 204786 2 2 1967 2 1168064000 5 1168064000 <NA> <NA>
#> 3 204786 3 3 1990 1 1168064000 5 1168064000 3 3
#> 4 204818 1 1 1984 1 1168064000 5 1156054000 4 1
#> 5 204818 2 5 1987 1 1168064000 5 1168064000 <NA> <NA>
#> 6 204848 1 1 1969 1 1168064000 4 1153053000 4 2
#> on.move
#> 1 FALSE
#> 2 FALSE
#> 3 FALSE
#> 4 FALSE
#> 5 FALSE
#> 6 FALSE
nrow(all1400)
#> [1] 108022

```

When you check the result, you can see that the format has changed compared to the previous one, and also that several variables have been added. In other words, a result in the form of data.frame indicating the purpose of which individuals with demographic characteristics on the left arrived at the location by means of transportation for what purpose was derived. And through the last on.move variable, you can also check whether the person is currently moving or is active in the area.

However, since the resulting data frame contains too much data (108022), it is difficult to know which regions are distributed more and which regions are less distributed. Therefore, we will refine the resulting data and find out how much it is distributed by region through the slice2df function.

The simple usage of slice2df function is as follows.

- slice2df(x, var1)

In this syntax, the data frame that is the result of slice is input in x, and the location of data.frame is input in var.

```

head(slice2df(all1400, "location"))
#>      Area count
#> 1 1111051500 263
#> 2 1111053000 356
#> 3 1111054000 101
#> 4 1111055000 102
#> 5 1111056000 67
#> 6 1111057000 117

```

Like this, you can check how many people are active by region at that time. slice2df also uses an additional variable You can check the distribution of the population by region in more detail.

```

result1 <- slice2df(all1400, "location", "sex")
head(slice2df(all1400, "location", "sex"))
#>      Area 1 2
#> 1 1111051500 140 123
#> 2 1111053000 213 143
#> 3 1111054000 41 60
#> 4 1111055000 49 53
#> 5 1111056000 33 34
#> 6 1111057000 46 71

```

Of course, you can check the results more easily than before in this way, but the table is still too long to check the results at once. It is true that it is still difficult. From now on, we will visualize with the data of sp class inside ASpaces.

Visualization

The result of slice can be visualized on the premise that the space unit of sp inside ASpaces is the same as the unit of activity space. First, merge the result and sp data for visualization.

```
korea2016@sp@data <- merge(korea2016@sp@data,
                           result1, by.x = "adm_cd", by.y = "Area")
head(korea2016@sp@data)
#>      adm_cd id objectid base_year shape_leng shape_area   1   2
#> 1 1111051500 25      16    2016   8088.371 2565649.1 140 123
#> 2 1111053000  1       1    2016   5756.737 1158538.1 213 143
#> 3 1111054000  2       2    2016   7423.825 1478147.7  41  60
#> 4 1111055000  3       3    2016   9033.824 2272861.8  49  53
#> 5 1111056000  4       4    2016  13367.893 8941185.2  33  34
#> 6 1111057000  6       5    2016   2620.108  369245.7  46  71
```

Since the number of population counted by region has been well entered into the spatial data, a simple visualization can now proceed.

The code for visualization is as follows.

```
library(RColorBrewer)
library(classInt)
#> Warning: package 'classInt' was built under R version 3.6.3
## Specify par so that three maps fit on one screen
par(mar = c(1, 1, 1, 1))
par(mai=c(1, 0, 1, 0))
par(mfrow = c(1, 3))

## Visualization for Male distribution
my_colors <- brewer.pal(5, "Reds")
class_of_freq <- classIntervals(korea2016@sp@data[, 7], 5, style = "jenks")
colors <- findColours(class_of_freq, my_colors)
plot(korea2016@sp, col = colors, main = "Male", border = F)

## Visualization for Female distribution
my_colors <- brewer.pal(5, "Reds")
class_of_freq <- classIntervals(korea2016@sp@data[, 8], 5, style = "jenks")
colors <- findColours(class_of_freq, my_colors)
plot(korea2016@sp, col = colors, main = "Female", border = F)

## Visualization for total population distribution
my_colors <- brewer.pal(5, "Reds")
class_of_freq <- classIntervals(korea2016@sp@data[, 7] + korea2016@sp@data[, 8],
                               5, style = "jenks")
colors <- findColours(class_of_freq, my_colors)
plot(korea2016@sp, col = colors, main = "Both", border = F)
```



When checking the visualization results, it was found that there was no significant difference in the distribution between men and women at 2:00 pm. Of course, these results may have a large influence on working hours due to the specificity of time. If so, what would be the result of different times?

```
library(parallel)
## Set the time to 8:00 am and 20:00 pm
all800 <- slice(korea2016, 800, silent = TRUE, mc = TRUE, core = 4)
all2000 <- slice(korea2016, 2000, silent = TRUE, mc = TRUE, core = 4)

## refining results
result2 <- slice2df(all800, "location", "sex")
result3 <- slice2df(all2000, "location", "sex")

## merging results to sp slot
korea2016@sp@data <- merge(korea2016@sp@data,
                           result2, by.x = "adm_cd", by.y = "Area")
korea2016@sp@data <- merge(korea2016@sp@data,
                           result3, by.x = "adm_cd", by.y = "Area")

head(korea2016@sp@data)
#>      adm_cd id objectid base_year shape_leng shape_area 1.x 2.x 1.y 2.y 1
#> 1 1111051500 25      16     2016   8088.371  2565649.1 140 123  77  67  73
#> 2 1111053000  1       1     2016   5756.737  1158538.1 213 143  93  67 105
#> 3 1111054000  2       2     2016   7423.825  1478147.7  41  60  22  36  28
#> 4 1111055000  3       3     2016   9033.824  2272861.8  49  53  45  46  42
#> 5 1111056000  4       4     2016  13367.893  8941185.2  33  34  44  37  42
#> 6 1111057000  6       5     2016   2620.108   369245.7  46  71 105 119  99
#>      2
#> 1  77
#> 2  89
#> 3  29
```

```
#> 4 38
#> 5 36
#> 6 110
```

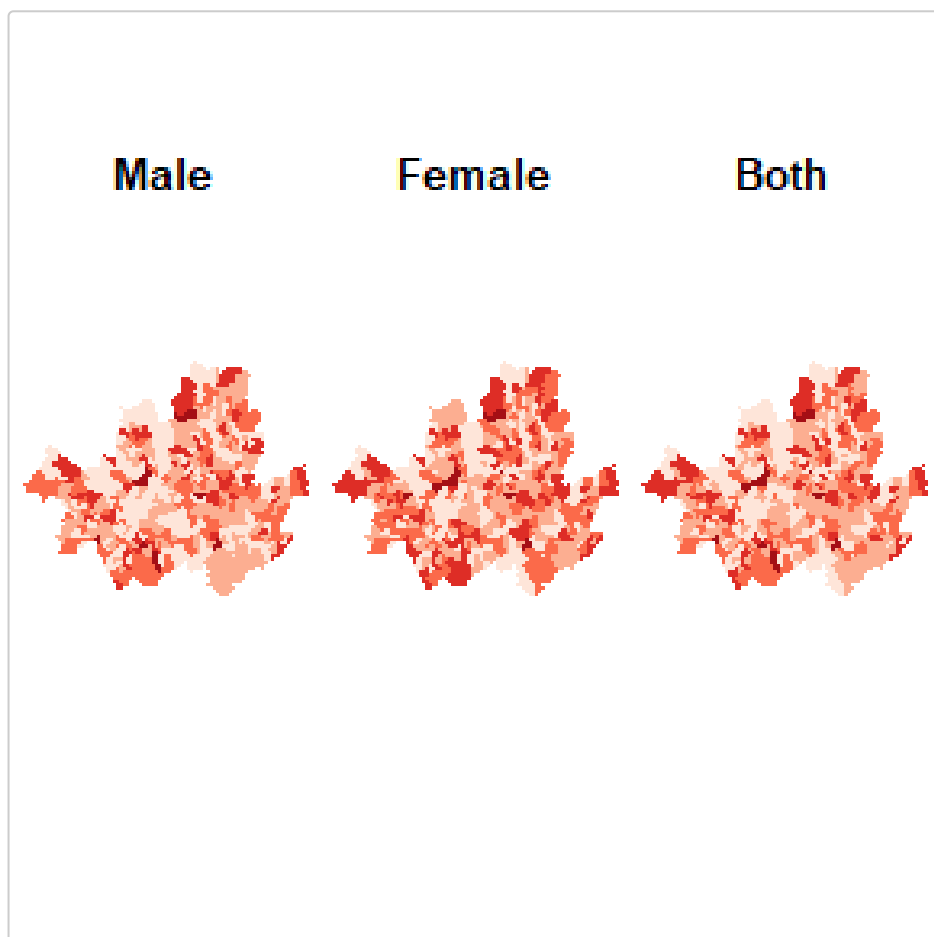
```
par(mar = c(1, 1, 1, 1))
par(mai=c(1, 0, 1, 0))
par(mfrow = c(1, 3))
```

```
## Visualization of 8 AM results
```

```
my_colors <- brewer.pal(5, "Reds")
class_of_freq <- classIntervals(korea2016@sp@data[, 9],
                                5, style = "jenks")
colors <- findColours(class_of_freq, my_colors)
plot(korea2016@sp, col = colors, main = "Male", border = F)
```

```
my_colors <- brewer.pal(5, "Reds")
class_of_freq <- classIntervals(korea2016@sp@data[, 10],
                                5, style = "jenks")
colors <- findColours(class_of_freq, my_colors)
plot(korea2016@sp, col = colors, main = "Female", border = F)
```

```
my_colors <- brewer.pal(5, "Reds")
class_of_freq <- classIntervals(korea2016@sp@data[, 9] + korea2016@sp@data[, 10],
                                5, style = "jenks")
colors <- findColours(class_of_freq, my_colors)
plot(korea2016@sp, col = colors, main = "Both", border = F)
```



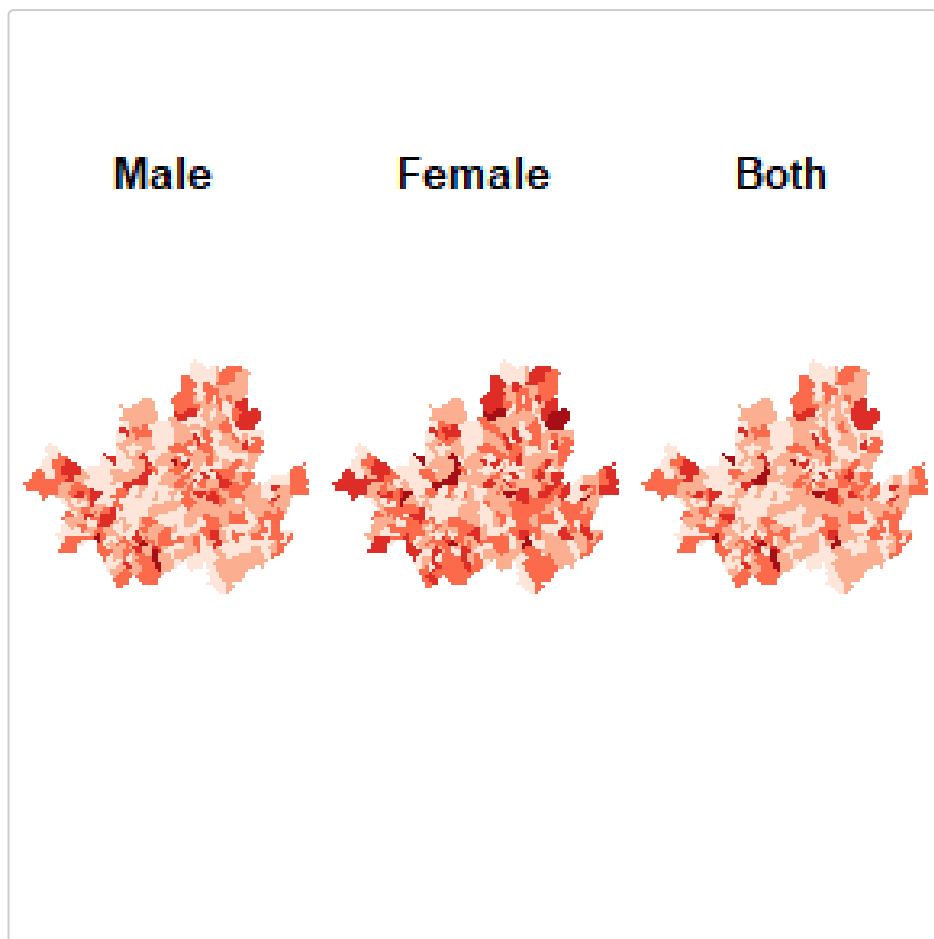
```

par(mar = c(1, 1, 1, 1))
par(mai=c(1, 0, 1, 0))
par(mfrow = c(1, 3))
## Visualization of 8 PM results
my_colors <- brewer.pal(5, "Reds")
class_of_freq <- classIntervals(korea2016@sp@data[, 11],
                                5, style = "jenks")
colors <- findColours(class_of_freq, my_colors)
plot(korea2016@sp, col = colors, main = "Male", border = F)

my_colors <- brewer.pal(5, "Reds")
class_of_freq <- classIntervals(korea2016@sp@data[, 12],
                                5, style = "jenks")
colors <- findColours(class_of_freq, my_colors)
plot(korea2016@sp, col = colors, main = "Female", border = F)

my_colors <- brewer.pal(5, "Reds")
class_of_freq <- classIntervals(korea2016@sp@data[, 11] + korea2016@sp@data[, 12],
                                5, style = "jenks")
colors <- findColours(class_of_freq, my_colors)
plot(korea2016@sp, col = colors, main = "Both", border = F)

```



In this way, you can see how it changes over time. This can be confirmed as a result of spatially differentiated residences and jobs in Seoul, Korea. However, it was confirmed that the difference between women and men still did not appear noticeably. It can be interpreted that the spatial distribution of men and women is not very different in terms of activity space.