

# Package ‘slice’

January 20, 2021

**Version** 0.1-2

**Date** 2021-01-20

**Title** Manipulating Activity Space Data for the Measurement of Segregation

**Author** Seong-Yun Hong [aut, cre], Changlock Choi [ctb]

**Maintainer** Seong-Yun Hong <syhong@khu.ac.kr>

**Depends** R (>= 3.4.0), methods, stats, sp

**Imports** splancs

**Suggests** spdep, spgrass6, rgdal, parallel, pbapply

**Description** Measuring spatial segregation. The methods implemented in this package include White's P index (1983) <doi:10.1086/227768>, Morrill's D(adj) (1991), Wong's D(w) and D(s) (1993) <doi:10.1080/00420989320080551>, and Reardon and O'Sullivan's set of spatial segregation measures (2004) <doi:10.1111/j.0081-1750.2004.00150.x>.

**License** GPL (>= 3)

**URL** <https://github.com/syunhong/slice>

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2019-12-18 15:20:02 UTC

**RoxygenNote** 7.1.1

**Collate** 'ASpace-class.R'  
'ASpaces-class.R'  
'ASpaces-methods.R'  
'as.Tracks.R'  
'as.data.frame.Asplaces.R'  
'as.tracksCollection.R'  
'checkslicemc.R'  
'data.R'  
'get\_xy.R'  
'locate.R'  
'select.R'  
'slice.R'  
'slice2df.R'  
'subset.R'

## R topics documented:

.locate . . . . .	2
as.data.frame.ASpaces . . . . .	3
as.Tracks . . . . .	4
as.tracksCollection . . . . .	5
ASpace-class . . . . .	6
ASpaces-class . . . . .	7
show,ASpaces-method . . . . .	8
slice . . . . .	8
slice2df . . . . .	10
slicedata . . . . .	11
subset . . . . .	12
summary,ASpaces-method . . . . .	13
update,ASpaces-method . . . . .	13
[[,ASpaces,ANY,ANY-method . . . . .	13
[[<-,ASpaces,ANY,ANY-method . . . . .	14
<b>Index</b>	<b>15</b>

---

.locate	<i>Find Individual's Location by Time</i>
---------	---

---

### Description

Find where an individual is located at a specific time during continuous trip

### Usage

```
.locate(trip, id, at, na.rm, silent)
```

### Arguments

trip	an object of data.frame containg person's continuous trip.
id	a vector of length 1, indicating person's ID
at	a numeric vector of length 1, specifying the time at which the location of an individual needs to be extracted
silent	logical. If TRUE, if any trip records have NAs in the time fields, a warning message will be provided.

### Value

a single row data.frame containing correspond with 'at'

### Author(s)

Seong-Yun Hong (syhong@khu.ac.kr)

### See Also

[slice](#)

**Examples**

```
#creates sample trip data
testtrip <- data.frame(tr_id = 1:10, tr_seq = rep(NA, 10), purpose = rep(NA, 10),
                      mode = rep(NA, 10), o_type = rep(NA, 10),
                      o_time = c(200,300,400,500,600,700,800,900,1000,1100),
                      o_zone = c(1,2,3,4,5,6,5,4,3,2),
                      d_type = rep(NA, 10),
                      d_time = c(300,400,500,600,700,800,900,1000,1100,1200),
                      d_zone = c(2,3,4,5,6,5,4,3,2,1))

#Find where at 800 via .locate
.locate(testtrip, 1, 600, na.rm = TRUE, silent = FALSE)
```

---

as.data.frame.ASpaces *Convert ASpaces to Data Frame*


---

**Description**

Coerces the Input Object of Class ASpaces to a Data Frame Object

**Usage**

```
as.data.frame.ASpaces(ASpaces, remove.notrip = TRUE)
```

**Arguments**

ASpaces            an object of class ASpaces  
remove.notrip    logical. See details.

**Details**

This conversion function converts the ASpaces class composed of a relatively complex structure into a simple data.frame. The existing ASpaces class is divided into info and trips inside the data and is stored in a list form for each individual, so it is easy to distinguish between data but has a limitation that it is difficult to see at a glance.

This function converts to data.frame to make it easier to check. In this process, the conversion is performed in a way that the information of the corresponding info for each trip is connected equally.

Among the arguments, remove.notrip determines whether to keep the trip as it is or to delete it for an individual that does not exist. If set to TRUE, which is the default value, the relevant individuals are deleted, and if set to FALSE, the trip data of NA is attached to the information of personal info.

**Value**

An object of class data.frame

**Author(s)**

Changlock Choi (hihi7100@khu.ac.kr), Seong-Yun Hong (syhong@khu.ac.kr)

**Examples**

```
#load data
data(slicedata)

#convert ASpaces into data.frame
df.slicedata <- as.data.frame.ASpaces(slicedata)

#compare with original data and result
slicedata@data[[1]]
df.slicedata[1:2, ]
```

as.Tracks

*Convert Objects to Class Tracks***Description**

Coerces the Input Object of Class ASpaces to a Tracks Object

**Usage**

```
as.Tracks(x, sp, varname)
```

**Arguments**

x	an object of class ASpace
sp	an object of class sp used in x
varname	name of the column in sp's data.frame which can be matched

**Details**

Function as.Track accepts converts ASpace objects to an onject of class Tracks. When converting a single ASpace object, unlike the ASpaces class, there is basically no sp given, so it is essential to enter the sp argument.

Also, since the input of the time is fixed to the POSIXct or POSIXt class in the Tracks class, attention is needed that the existing time format is changed to the type of the corresponding class.

**Value**

An object of class Tracks

**Author(s)**

Seong-Yun Hong (syhong@khu.ac.kr)

**See Also**

[as.Tracks](#)

## Examples

```
## Not run:
# creates sample info data
testinfo <- list()

# creates sample trip data
testtrip <- data.frame(tr_id = 1:10, tr_seq = rep(NA, 10), purpose = rep(NA, 10),
                      mode = rep(NA, 10), o_type = rep(NA, 10),
                      o_time = c(200,300,400,500,600,700,800,900,1000,1100),
                      o_zone = c(1,2,3,4,5,6,5,4,3,2),
                      d_type = rep(NA, 10),
                      d_time = c(300,400,500,600,700,800,900,1000,1100,1200),
                      d_zone = c(2,3,4,5,6,5,4,3,2,1))

#constructs an object of class 'ASpace'
testASpace <- new("ASpace", info = list(id = 1), trip = testtrip)

testTracks <- as.tracksCollection(testASpace, varname = "adm_dr_cd")
```

---

as.tracksCollection      *Convert Objects to Class tracksCollection*

---

## Description

Coerces the Input Object of Class ASpaces to a tracksCollection Object

## Usage

```
as.tracksCollection(x, sp, varname)
```

## Arguments

x	an object of class ASpaces
sp	an object of class sp used in x
varname	name of the column in sp's data.frame which can be matched with o_zone and d_zone in ASpaces

## Details

Function as.trackCollection accepts converts ASpaces objects to an onject of class tracksCollection. In the conversion since the individual's travel position matches the position on the spatial data, care must be taken in selecting 'sp'. If 'sp' is not given, the 'sp' contained in ASpaces is selected.

## Value

An object of class tracksCollection

## Author(s)

Seong-Yun Hong (syhong@khu.ac.kr)

**See Also**[as.Tracks](#)**Examples**

```
## Not run:
# creates sample info data
testinfo <- list()

# creates sample trip data
testtrip <- data.frame(tr_id = 1:10, tr_seq = rep(NA, 10), purpose = rep(NA, 10),
                      mode = rep(NA, 10), o_type = rep(NA, 10),
                      o_time = c(200,300,400,500,600,700,800,900,1000,1100),
                      o_zone = c(1,2,3,4,5,6,5,4,3,2),
                      d_type = rep(NA, 10),
                      d_time = c(300,400,500,600,700,800,900,1000,1100,1200),
                      d_zone = c(2,3,4,5,6,5,4,3,2,1))

#constructs an object of class 'ASpace'
testASpace <- new("ASpace", info = list(id = 1), trip = testtrip)

#constructs an object of class 'ASpaces'
testtASpaces <- new("ASpaces", data = list(testaspace), sp = testshp)

testTC <- as.tracksCollection(testASpaces, varname = "adm_dr_cd")
```

ASpace-class

*Class ASpace***Description**

A S4 class for storing activity-space data; Contains information of one individual person

**Details**

The slot 'info' contains demographic and socioeconomic variables of the respondent as a list object. The list object must include the following named element(s):

.. \$id

If this information is not given by user at the time of creation of 'ASpace', an empty list will be supplied.

The slot 'trip' contains information on trips made by the survey respondent. It a data frame object with the following variables:

```
.. $tr_id: Trip ID (integer)
.. $tr_seq: Sequence ID of the trip (integer)
.. $purpose: Purpose of the trip (factor)
.. $mode: Mode of the trip (factor)
.. $o_type: Type of the origin (factor)
.. $o_time: Time departed from the origin (integer, or POSIXct)
.. $o_zone: Area code for the origin (character)
.. $d_type: Type of the destination (factor)
.. $d_time: Time arrived at the destination (integer, or POSIXct)
.. $d_zone: Area code for the destination (character)
```

**Slots**

`info` an object of list. representing the demographic characteristic of individual.

`trip` an object of data.frame containing a trip data made by the person described in the info slot.

**Author(s)**

Seong-Yun Hong (syhong@khu.ac.kr)

**See Also**

[ASpaces](#)

**Examples**

```
#example of creating ASpace class
#creates sample info data
testinfo <- list()

#creates sample trip data
testtrip <- data.frame(tr_id = 1:10, tr_seq = rep(NA, 10), purpose = rep(NA, 10),
                      mode = rep(NA, 10), o_type = rep(NA, 10),
                      o_time = c(200,300,400,500,600,700,800,900,1000,1100),
                      o_zone = c(1,2,3,4,5,6,5,4,3,2),
                      d_type = rep(NA, 10),
                      d_time = c(300,400,500,600,700,800,900,1000,1100,1200),
                      d_zone = c(2,3,4,5,6,5,4,3,2,1))

#constructs an object of class 'ASpace'
testaspace <- new("ASpace", info = list(id = 1), trip = testtrip)
```

---

ASpaces-class

---

Class ASpaces

---

**Description**

A S4 class for storing activity-space data; Contains information of one individual person

**Details**

The slot 'data' holds a list of ASpace objects (i.e., activity space information for an individual).

The slot 'attr' has supplementary information about the data. For example, it may contain the name of the survey, the date it was conducted, or the code book for the variables.

The slot 'sp' contains spatialpolygon or spatialpoint class data that can indicate the specific location of o\_zone and d\_zone in trip.

**Slots**

`data` an object of list containing ASpace object.

`attr` an object of list containing metadata.

`sp` an object of spatialpolygon or spatialpoint class data.

Author(s)

Seong-Yun Hong (syhong@khu.ac.kr)

See Also

[ASpace](#)

Examples

```
#example of creating ASpace class
#creates sample info data
testinfo <- list()

#creates sample trip data
testtrip <- data.frame(tr_id = 1:10, tr_seq = rep(NA, 10), purpose = rep(NA, 10),
                      mode = rep(NA, 10), o_type = rep(NA, 10),
                      o_time = c(200,300,400,500,600,700,800,900,1000,1100),
                      o_zone = c(1,2,3,4,5,6,5,4,3,2),
                      d_type = rep(NA, 10),
                      d_time = c(300,400,500,600,700,800,900,1000,1100,1200),
                      d_zone = c(2,3,4,5,6,5,4,3,2,1))

#constructs an object of class 'ASpace'
testASpace <- new("ASpace", info = list(id = 1), trip = testtrip)

#constructs an object of class 'ASpaces'
testtASpaces <- new("ASpaces", data = list(testaspace), sp = testshp)
```

---

show,ASpaces-method	<i>generic method for class ASpaces</i>
---------------------	---

---

Description

generic method for class ASpaces

Usage

```
## S4 method for signature 'ASpaces'
show(object)
```

---

slice	<i>Retreiving Population Distribution at Given Time</i>
-------	---

---

Description

Identifies the location of each individual at the given time instant

Usage

```
slice(x, at, vars, showProgress = TRUE, na.rm = TRUE, silent = FALSE,
mc = FALSE, core)
```



## Arguments

<code>x</code>	an object of class ASpaces
<code>at</code>	a numeric vector of length 1, specifying the time at which the location of an individual needs to be extracted
<code>vars</code>	an optional character vector. See Details.
<code>showProgress</code>	logical. If TRUE, a progress bar appears on the R console while iterating.
<code>silent</code>	logical. If TRUE, if any trip records have NAs in the time fields, a warning message will be provided.
<code>mc</code>	logical. If TRUE, function is operated in parallel. See Details.
<code>core</code>	numeric vector of length 1, specifying the Number of cores to be utilized in parallel computation.

## Details

the `slice()` function attempts to retrieve the locations of people in the input object 'x' at the given time instant 'at'.

The function looks into the slot 'trip' in each of the elements in the slot 'data' of the object 'x' (i.e., `x@data[[i]]@trip` where `i = 1 ... n`).

The 'trip' slot contains a sequence of trips made by each person, and it is assumed that the trips are sorted in ascending order of the departure time (i.e., `$o_time`). If NOT, this function may not work as intended.

'vars' are used to refine the information in the resulting ASpace's info slot. When a specific character vector is input to 'vars', the info slot of ASpace resulting from the function has only the input variables. If 'vars' is not input, the result is same with the original info slot.

The 'mc' and 'core' arguments are used to enable parallelization of the slice function. If the option of mc is set to TRUE, the function works in a multi-core environment. At this time, the number of cores (threads) used can be adjusted with the 'core' argument, and if the set number exceeds the user's environment, it is adjusted to -1 of the maximum number.

The result of the slice function is provided in the form of a single data.frame, and one variable is added excluding the variables of info and trip. For this, a variable called `on.move` is added, and the variable is information about whether a person existing at the location is passing at the time specified by the user. If `on.move` is TRUE, it means that the person is moving.

## Value

A data.frame

## Author(s)

Seong-Yun Hong (syhong@khu.ac.kr)

## Examples

```
# load data
data(slicedata)

# extracting population distribution at 1:00PM
slice(slicedata, at = 1300)

# observation of changes in population distribution over time
```

```

result <- list()
time <- seq(800, 2000, 100)
for(i in 1:length(time)){
  result[[i]] <- slice(sliceddata, at = time[i])}

# running function with parallelization
if (require(parallel)){
  result <- slice(newtestset, 1200, silent = TRUE, mc = TRUE, core = 2)

  result <- slice(newtestset, 1200, silent = TRUE, mc = TRUE, core = 6)

  result <- slice(newtestset, 1200, silent = TRUE, mc = TRUE, core = 100)
}

```

---

slice2df

Retreiving Population Distribution at Given Time

---

### Description

Summarises the Input Data Frame by Area and an Optional Grouping Variable

### Usage

```
slice2df(x, var1, var2)
```

### Arguments

x	a data.frame object
var1	a character vector of length 1, indicating the name of the column in 'x' that contains area codes or names. This argument must be provided.
var2	an optional character vector of length 1, indicating the name of a grouping variable

### Details

The result of the slice function shows where people exist at a given time in the form of data.frame, and each person is displayed in one row. slice2df is used when you want to aggregate these results the way you want.

Specifically, slice2df is a function indicating how many people exist for each space unit through x given as a result of the slice function, and additional classification can be performed by demographic characteristics through var2.

In other words, the result of slice2df is data.frame indicating how many populations exist in each area at a specific time, and this is a form that can be used directly in a function that measures spatial segregation inside a seg package.

However, slice2df should be noted that when the spatial unit of the data of the initially given ASpace class is measured in points or very small spatial unit, the number of all regions may appear as one.

### Value

A data.frame that is suitable for the functions in the seg package

**Author(s)**

Seong-Yun Hong (syhong@khu.ac.kr)

**See Also**

seg, dissim

**Examples**

```
# load data
data(slicedata)

# run slice function to create suitable data.frame
result1 <- slice(slicedata, at = 800)
result2 <- slice(slicedata, at = 1400)
result3 <- slice(slicedata, at = 2000)

# converting the result
slice2df(result1, "location")
slice2df(result2, "location")
slice2df(result3, "location")
```

---

slicedata

*Sythetic Patterns of Activity Space*


---

**Description**

A dataset containing the demographic attributes and trip record of almost 100 people

**Usage**

```
data(slicedata)
```

**Format**

An object of class ASpaces. The dataset contains 84 people's demographic attribute and 236 trip records.

**data** include list object which contains info and trip data

**info** 14 different demographic characteristics, including individual ID

**trip** 10 essential information which is needed to configure ASpace class

**sp** No sp data is given since it is synthetic travel information ...

**Examples**

```
#load data
data(slicedata)

#validate whether data has normal condition
validateASpaces(slicedata)
```

```
#check data's basic information
summary(slicedata)
show(slicedata)
```

subset

*Subsets the Input object of Class ASpaces*

## Description

Extract ASpace objects with characteristics that meet given conditions from ASpaces

## Usage

```
subset(x, condition, vars, all = TRUE, showProgress = TRUE)
```

## Arguments

<code>x</code>	an object of class ASpaces
<code>condition</code>	a list that indicate characteristics.
<code>vars</code>	an optional character vector. See details.
<code>all</code>	logical. See details.
<code>showProgress</code>	logical. If TRUE, a progress bar appears on the R console while iterating.

## Details

subset function is a function that refines a given ASpaces class object.

ASpace classes contain the characteristics of the activity space subject in the info slot, and subset function extracts only ASpaces that meet the given conditions based on the characteristics of# this info slot and returns them to ASpaces.

'condition' is entered in the form of a list, and the ASpace of the corresponding condition is extracted by entering each desired condition in the list (i.e., `list(var1 = x, var2 = y)`)

'vars' are used to refine the information in the resulting ASpace's info slot. When a specific character vector is input to 'vars', the info slot of ASpace resulting from the function has only the input variables. If 'vars' is not input, the result is same with the original info slot.

'all' determines whether to extract only the results that satisfy all of them when various conditions are given in the 'condition', or to extract the results that apply at least one. The default value is set to extract only when all conditions are satisfied as true.

## Value

An object of class ASpaces

## Author(s)

Changlock Choi (hihi7100@khu.ac.kr), Seong-Yun Hong (syhong@khu.ac.kr)

**Examples**

```
# load data
data(slicedata)

# subsetting ASpaces
subsetasp1 <- subset(slicedata, condition = list(sex = 1))
subsetasp2 <- subset(slicedata, condition = list(sex = 1, hhsize = 4))
subsetasp3 <- subset(slicedata, condition = list(sex = 1, hhsize = 4),
                     all = FALSE)
```

---

summary,ASpaces-method

*generic method for class ASpaces*


---

**Description**

generic method for class ASpaces

**Usage**

```
## S4 method for signature 'ASpaces'
summary(object)
```

---

update,ASpaces-method *generic method for class ASpaces*


---

**Description**

generic method for class ASpaces

**Usage**

```
## S4 method for signature 'ASpaces'
update(object, data, attr, sp)
```

---

[[,ASpaces,ANY,ANY-method  
*generic method for class ASpaces*


---

**Description**

generic method for class ASpaces

**Usage**

```
## S4 method for signature 'ASpaces,ANY,ANY'
x[[i]]
```

---

[[<-,ASpaces,ANY,ANY-method  
*generic method for class ASpaces*

---

**Description**

generic method for class ASpaces

**Usage**

```
## S4 replacement method for signature 'ASpaces,ANY,ANY'  
x[[i]] <- value
```

# Index

## \*Topic **datasets**

- slicedata, [11](#)
- .locate, [2](#)
- [[, ASpaces, ANY, ANY-method, [13](#)
- [[<-, ASpaces, ANY, ANY-method, [14](#)
- as.data.frame.ASpaces, [3](#)
- as.Tracks, [4](#), [4](#), [6](#)
- as.tracksCollection, [5](#)
- ASpace, [8](#)
- ASpace-class, [6](#)
- ASpaces, [7](#)
- ASpaces-class, [7](#)
- show, ASpaces-method, [8](#)
- slice, [2](#), [8](#)
- slice2df, [10](#)
- slicedata, [11](#)
- subset, [12](#)
- summary, ASpaces-method, [13](#)
- update, ASpaces-method, [13](#)