Package ‘cartograflow’

May 17, 2020

Title  Filtering Matrix for Flow Mapping
Version  1.0.1
Description  Functions to prepare and filter an origin-destination matrix for thematic flow mapping purposes. This comes after Bahoken, Francoise (2016), Mapping flow matrix a contribution, PhD in Geography - Territorial sciences. See Bahoken (2017) <doi:10.4000/netcom.2565>.
Depends  R (>= 3.4.0)
License  GPL-3
URL  https://github.com/fbahoken/cartogRaflow
Encoding  UTF-8
LazyData  true
RoxygenNote  7.1.0
Suggests  knitr, rmarkdown, testthat, cartography
VignetteBuilder  knitr
Imports  dplyr, g.data, graphics, ggplot2, maptools, plotly, rgdal, rgeos, reshape2, rlang, sf, sp, utils, igraph
NeedsCompilation  no
Author  Sylvain Blondeau [aut, cre], Francoise Bahoken [aut]
Maintainer  Sylvain Blondeau <blondeau.sylvain@yahoo.fr>
Repository  CRAN
Date/Publication  2020-05-17 19:50:03 UTC

R topics documented:
cartograflow .......................................................... 2
flowanalysis .......................................................... 2
flowcarre ............................................................... 4
flowcontig ............................................................ 5
flowdist .............................................................. 6
Description

This package contains functions to prepare and filter origin-destination matrix for thematic flow mapping purposes. The spatial objects processing are those of sf.

This comes after Bahoken, Francoise (2016) Contribution à la cartographie d’une matrice de flux, Phd in Geography, Sorbonne Paris Cité, Paris 7.

Details

To learn more about cartograflow, see the vignette cartograflow.html

Main functions:

- flowanalysis
- flowcarre
- flowcontig
- flowdist
- flowgini
- flowjointure
- flowmap
- flowreduct
- flowstructmat
- flowtabmat
- flowtype
- geoid
- mat_ex

---

flowanalysis

Computation of a global concentration criterion of flows values or features

Description

Computation of a global selection criterion for filtering flows values or flow features.
To be use after flowgini and before flowmap.

Usage

flowanalysis(tab, fij = NULL, critflow, critlink, result)
flowanalysis

Arguments

- **tab**: input flow dataset from `flowgini`
- **fij**: flow value between origin and destination places
- **critflow**: desired level of information significativity. See Details.
- **critlink**: desired level of features density. See Details.
- **result**: resulting filtering criterion value. See Details.

Details

- `critflow` = desired level of flow's information significativity (e.g. 80)
- `critlink` = desired level of flow's features density (e.g. 20) of the flow features that represents the more significant information.

- `result = "density"` returns the desired level of features density as a
- `result = "significativity"` returns the level of flow significativity as a

References


Examples

```r
library(cartograflow)
data(flowdata)

# 1/4: Computes Gini's coefficient

tabgini <- flowgini(ODpts = flows, origin = "i", destination = "j",
                  valflow = "Fij", lorenz.plot = FALSE)
### [1] Gini's coefficient = 73.16 %

# 2/4: Plot Lorenz curve

flowgini(tab_gini,
         format = "L", origin = "i", dest = "j", valflow = "ydata",
         bkg, code = "EPT_NUM", lorenz.plot = TRUE)

# 3/4: Compute critflow filtering parameter
# critflow = 0.8 #selected criterion

flowanalysis(tabgini, critflow = 0.8, result = "signif")
### [1] "threshold = 11238 --- flows = 80 % --- links = 22.94 %"

# 4/4: Plot the flowmap

flowmap(
         tab = tabflow, fij = "Fij", origin.f = "i", destination.f = "j",
         bkg = map, code = "EPT_NUM", nodes.X = "X", nodes.Y = "Y",
         filter = TRUE,
         threshold = 11238,
         threshcol = "grey"
)
```

flowcarre

Builds a square matrix from geographical nodes

Description

Builds a square and closed matrix from a dataframe of spatial nodes

Usage

flowcarre(liste, tab, origin, dest, valflow, empty.sq, format, diagonale)

Arguments

- **liste**: list of all the spatial codes as a single dataframe
- **tab**: the non squared input flow dataset with three columns: origin, destination, flow value
- **origin**: node / place of origin of the flow
- **dest**: node / place of destination of the flow
- **valflow**: is the flow value between origin and destination places
- **empty.sq**: Builds an empty matrix or not. See Details.
- **format**: is the desired squared flow dataset output format. See Details.
- **diagonale**: to zero or not the main diagonal. See Details.

Details

- empty.sq is "TRUE" builds an empty matrix; else is "FALSE" or missing
- format is "M" for matrix format
- format is "L" for long format, as three column dataframe
- diagonal is "TRUE" to zero the main diagonal
Examples

```r
library(cartograflow)
data(flowdata)
var1 <- geoid
var2 <- flows

# 1/2 Compute an empty square matrice with ID code, and sets the value to zero
# Example for matrice format (same procedure for the long format)
mat <- flowcarre(var1, var2,
          origin = "i", dest = "j", valflow = "Fij",
          format = "M", empty.sq = TRUE)

# 2/2 Fill in the matrice with external flow values
mat <- flowcarre(var1, var2,
          origin = "i", dest = "j", valflow = "Fij",
          format = "M", empty.sq = FALSE)

# Square a matrice and zero the main diagonal
mat <- flowcarre(var1, var2,
          origin = "i", dest = "j", valflow = "Fij",
          format = "M", empty.sq = FALSE, diagonale = FALSE)
```

---

**Description**

From a layer of areal spatial features, compute an ordinal distance matrice based on a k order criterion of adjacency or contiguity between origin and destination places.
The result is a neighbourhood graph that can be used for filtering flow values before flow mapping (flowmap)

**Usage**

`flowcontig(bkg, code, k, algo)`

**Arguments**

- `bkg` a layer of areal spatial features (eg. the map background)
- `code` spatial areal features code
- `k` order of adjacency or contiguity between two areal spatial features
- `algo` algorithm to use for ordinal distance calculation. Default is "Dijkstra’s" algorithm. See Details.
Details

The \((k=1,2,...,k)\) order of adjacency or contiguity, of an areal spatial features background, is the number of spatial boundaries to be crossed between a couple of origin-destination (ODs) places. The \(k\) number can be assimilated to a shortest path between two pair of nodes Argument ‘\(k\)’ is to enter the number \(k\) of the contiguity matrix to be constructed; -\(k\)ordre=1 : ODs places are adjacent, ie the flow have to cross only 1 boundary.  
-\(k\)ordre=2 : ODs places are distant from 2 borders  
-\(k\)ordre=k : ODs places are distant from \(k\) borders  
The function returns also the \((k)\) number of the layer

Value

a contiguity matrice with the \(k\) orders of adjacency

Examples

```r
library(cartograflow)
library(sf)
data(flowdata)
map <- st_read(system.file("shape/MGP_TER.shp", package = "cartograflow"))
graph_ckij_1 <- flowcontig(bkg = map, code = "EPT_NUM", k = 1, algo = "automatic")
flowmap(
  tab = graph_ckij_1,
  fij = "ordre", origin.f = "i", destination.f = "j",
  bkg = map, code = "EPT_NUM", nodes.X = "X", nodes.Y = "Y",
  filter = FALSE
)
```

flowdist

Builds a continuous distance matrices from a spatial features background

Description

From a layer of areal spatial features, compute and threshold a continuous distance matrix. The result is either a matrice of distances between ODs, or a flow matrix based on the distance travelled between ODs; both can be used for filtering flow before flow mapping (flowmap)

Usage

```r
flowdist(tab, dist.method, result)
```

Arguments

- **tab** the input flow dataset
- **dist.method** distance calculation algorithm, default is euclidian calculation
- **result** Choose Building a "flowdist" or a simple "dist" matrice. See Details
Details

- result = "dist" is the simple resulting distance matrice.
- result = "flowdist" is the resulting distance matrice with additional calculated parameters.
- It is also possible to filter flow by a level of distance travelled.

Value

(1) A flowdata set with continuous distances calculations. See dist.method parameter
(2) A flowdata set with movement from euclidian distances calculations

Examples

```r
library(cartograflow)
library(sf)
data(flowdata)
map <- st_read(system.file("shape/MGP_TER.shp", package = "cartograflow"))
tabflow <- flowjointure(
  geom = "area", bkg = map, DF.flow = flows, origin = "i",
  destination = "j", id = "EPT_NUM", x = "X", y = "Y"
)

# Format long with only origin, destination and distance parameters:
tab.distance <- flowdist(tabflow, dist.method = "euclidian", result = "dist")
# Format long with with all parameters: coordinates, distance, mouvement
tab.distance <- flowdist(tabflow, dist.method = "euclidian", result = "flowdist")
```

---

**flowgini**  
*Analysis of flow concentration (Gini coefficient)*

Description

Calculates Gini coefficient, plot Lorenz curve and threshold the matrice according to a global concentration criterion for mapping flow intensity or flow density.
To be use before `flowanalysis`

Usage

```r
flowgini(ODpts, origin, destination, valflow, lorenz.plot)
```

Arguments

- **ODpts**  
  the input dataset with : nodes code, flow values and XY coordinates
- **origin**  
  ID origin place, in long format
- **destination**  
  ID destination place, long format
- **valflow**  
  flow value between origin and destination places
- **lorenz.plot**  
  to plot or the Lorenz curve. See Details
flowjointure

Builds a spatial join with a flow dataset

Description
Builds a spatial join between a flow dataset and a spatial features layer (as a map background)

Usage
flowjointure(geom, bkg, DF.flow, origin, destination, DF.point, id, x, y)

Arguments
- geom: the geometry of the spatial features layer: points or areas
- bkg: the spatial features layer
- DF.flow: the input flow dataset as a dataframe
- origin: the place of origin code

Details
flowgini(...,lorenz.plot = TRUE) for plotting Lorenz curve associate to the gini coefficient, from cumulated flows and links.

Value
plot Lorenz curve for the cumulated flow and links: flowgini(...,gini.plot = TRUE), warning: the function must not assign a variable
value of the Gini’s coefficient and the table: table<-flowgini(...,missing(gini.plot) or gini.plot = FALSE)

References

Examples
library(cartograflow)
data(flowdata)
# Computes Gini's coefficient
tabgini <- flowgini(ODpts = flows, origin = "i", destination = "j",
valflow = "Fij", lorenz.plot = FALSE)
# Plot Lorenz curve
flowgini(ODpts = flows, origin = "i", dest = "j", valflow = "Fij", lorenz.plot = TRUE)
# See \link{flowanalysis} for viewing the tab_gini table
destination  the place of destination code
DF.point  a dataframe of points or places
id  dataframe of points or places file code
x  the X coordinate of the point or places
y  the Y coordinate of the point or places

Value
the corresponding joint table between the flow dataset and the spatial feature layer

Examples
library(cartograflow)
library(sf)
data(flowdata)
map <- st_read(system.file("shape/MGP_TER.shp", package = "cartograflow"))
tabflow <- flowjointure(
  geom = "area", bkg = map, DF.flow = flows, origin = "i", destination = "j",
  id = "EPT_NUM", x = "X", y = "Y"
)

flowmap  Mapping of an origin-destination flow matrix

Description
Mapping of an origin-destination flow matrix

Usage
flowmap(
  tab,
  fij,
  origin.f,
  destination.f,
  bkg = NULL,
  crs,
  nodes = NULL,
  code,
  nodes.X,
  nodes.Y,
  filter,
  plota,
  threshold,
  taille,
  a.head,
  a.length,
a.angle,
a.col,
add = NULL,
... )

Arguments

*tab* the input flow dataset in .csv format. See Details

*fi j* the flow value between origin and destination places

*origin.f* the place of origin code

*destination.f* the place of destination code

*bkg* a spatial feature layer, as a map background, in .shp or .json or other format

*crs* the coordinate reference system (CRS)

*nodes* the input points file in .csv format

*code* the spatial features code

*nodes.X* the X coordinate of the point or places

*nodes.Y* the Y coordinate of the point or places

*filter* is to filter or not the flow values. See details

*plota* is to add spatial features as map background to the flows’s plot

*threshold* the value of the threshold criterion to filter flows. Default is 1.

*taille* the value of the width of the flow feature

*a.head* for arrow’s head is the arrow head parameter code. It allows to choose the kind of arrow. See Details

*a.length* for arrow’s length is the length of the edges of the arrow head (in inches)

*a.angle* for arrow’s angle is the angle from the shaft of the arrow to the edge of the arrow head

*a.col* for arrow’s color

*add* is to allow to overlay flow features on external spatial features background

... Adds the set of variables of the arrow function

Details

The input .csv flow dataset must be first converted to a dataframe for optimal performance (troubles remains with tibble format)

- filter is "FALSE" means that all the flow value will be plot as segments \((n*(n-1))\), i.e. all the OD matrix’s cells out of the main diagonal will be plot.
- filter is "TRUE" means only non-zero values will be plot, i.e. existing links with or without threshold.

The default threshold is set to 1.
Flow features are plot as segments between (x0,y0) and (x1,y1)
- a.head is for applying an arrow or not to a segment:
  - code="0" : the link has no head - no arrow
  - code="1" : an arrow is draw at (x0[i], y0[i])
  - code="2" : an arrow is draw at (x1[j], y1[j])
  - code="3" : an arrow is draw at both nodes.

Value

a matrix or a list with the correct flow dataframe ID code
The resulting flowmap

Examples

library(cartograflow)
library(sf)
data(flowdata)
# example with the background
map <- st_read(system.file("shape/MGP_TER.shp", package = "cartograflow"))
par(bg = "NA")
plot(st_geometry(map), col = "blue")
flowmap(
  tab = flows, fij = "Fij", origin.f = "i", destination.f = "j",
  bkg = map,add=TRUE, code = "EPT_NUM", nodes.X = "X", nodes.Y = "Y",
  filter = FALSE
)

# example with nodes files
map <- st_read("MGP_territoires.json")
pt <- read.csv2("points.csv")# points files origin destination
flows<-red.cs2("flows.csv") # flows files
par(bg = "NA")
plot(st_geometry(map), col = "blue")
flowmap(tab = flows, fij = "d", origin.f = "a", destination.f = "b",
    crs=4326,nodes = pt, code = "EPT_NUM", nodes.X = "X", nodes.Y = "Y",
    filter = TRUE,add=TRUE,threshold = 10, taille = 8,arr.length = 0.1)

flowreduct

Flow matrix reduction according to another matrix

Description

Reduces a flow dataset according to an external matrix, eg. a matrix of travelled distance.
Builds geographical movements, by weighting a flow dataset according to a distance criterion.

Usage

flowreduct(tab, tab.metric, metric, d.criteria, d)
Arguments

tab is the input flow data set

tab.metric is the distance dataset

metric select "continuous" or "ordinal" metric. See Details

d.criteria is for selecting "dmin" or "dmax" distance criteria for "continuous" metric. See Details.

d is the value of the selected "dmin" or "dmax". see Details

Details

The involved metric can be continuous or not.

(1) Metric is 'continuous" for distance as euclidean, maximum, manhattan, etc. See flowdist
- Metric is 'ordinal" for computing neighbourhood ordinal distance matrix. – Select ="dmin" for reducing flow dataset to flow values that are up or equal to the dmin distance parameter (Fij>=dmin);
- select ="dmax" for reducing flow dataset to values that are less or equal to the dmax distance parameter(Fij=<dmin).

- Metric is 'ordinal' for computing neighbourhood ordinal distance with k contiguity. See flowcontig for computing ordinal distance matrix

Value

A flow dataset with distances computations and flow reduction

Examples

library(cartograflow)
library(sf)
library(dplyr)
data(flowdata)
map <- st_read(system.file("shape/MGP_TER.shp", package = "cartograflow"))

tab <- flowjointure(
    geom = "area", bkg = map, DF.flow = flows, origin = "i", destination = "j",
    id = "EPT_NUM", x = "X", y = "Y"
)

# Example for reducing a flow matrix with a distance matrix, in long format (i,j, distance)
## 1/2: Computes the matrix distances
tab.distance <- flowdist(tab, dist.method = "euclidian", result = "dist")
tab.distance <- tab.distance %>% select(i, j, distance)
## 2/2: Reduce the flow matrix
tab.flow <- flowreduct(flows, tab.distance, metric = "continuous")
flows

    d.criteria = "dmax", d = 8567
)

flows          MOBPRO: Commuting trips in 2015

Description

Citation: INSEE - RP MOBPRO, 2015.
Variable (i) is the place of origin of the flow.
Variable (j) is the place of destination of the flow.
Variable (Fij) is the flow value between (i, j).
Variable (count) is the frequency of the (i, j) couple of places.

Source

https://www.insee.fr/fr/statistiques/fichier/3566008/rp2015_mobpro_txt.zip

flowstructmat          Structuring a matrix

Description

Fixes an ID shift in the flow matrix (to be used with flowjointure if necessary and flowtabmat)

Usage

flowstructmat(z)

Arguments

z            The input flow dataset in the matrice format where the first column is filled with the ID

Value

A flow dataset with an usable format
Examples

```r
library(cartograflow)
data(flowdata)

dim(mat_ex) # dimension of the original matrix
### 10 11 # first column is filled with the ID

tab <- flowstructmat(mat_ex)
dim(tab)
## 10 10 # dimension of the resulting matrix
```

### flowtabmat

*Changing the format of a flow dataset*

**Description**

Transform a flow dataset from long to matrix format, and vice versa. Square matrix.

**Usage**

```r
flowtabmat(tab, matlist)
```

**Arguments**

- `tab`: flow dataset, in matrix or long format
- `matlist`: choose "matrix" or "long" for the resulting format. See Details.

**Details**

- `matlist="M"` from long (3 columns: origin, destination, flow) to matrix format \([n \times n]\);
- `matlist="L"` from matrix to long format.

**Value**

A flow dataset in matrix or in long format

**Examples**

```r
library(cartograflow)
data(flowdata)

# 1: From long to matrix format \((n \times m)\)
matFlow <- flowtabmat(flows, matlist = "M")

# 2: From matrix to long format \([i,j,F_{ij}]\)
listflow <- flowtabmat(matFlow, matlist = "L")
```
flowtype

**Description**

Compute gross (volumn) and net (balance) flows from initial asymmetric flow values

**Usage**

```r
flowtype(tab, format, x)
```

**Arguments**

- `tab` is the input flow dataset
- `format` specify the flow dataset format, "M" for square matrix [n*n] or L for long [i,j,data]
- `x` enter the computation type : "flux", "transpose", "bivolum" and "bisold".

**Details**

The matrix must be squared (if not, see `flowcarre`). This function compute for all pairs or origin-destination places (i,j) involved in an asymmetric flow matrix (Fij<> Fji) several matrix :
- `x = "flux"` for remaining initial flow (Fij)
- `x = "transpose"` for reverse flow value (Fji)
- `x = "bivolum"` for bilateral gross flow Vij=(Fij+Fji)
- `x = "bisold"` for bilateral net flow Sij=(Fij-Fji)

**References**


**Examples**

```r
library(cartograflow)
data(flowdata)
bkg <- system.file("shape/MGP TER.shp", package = "cartograflow", lib.loc = NULL, mustWork = TRUE)

## 1a:Computes flowtypes: Matrice format
matflow <- flowtabmat(flows, matlist = "M")
m <- flowtype(matflow, format = "M", x = "flux")
m <- flowtype(matflow, format = "M", x = "transpose")
m <- flowtype(matflow, format = "M", x = "bivolum")
```
m <- flowtype(matflow, format = "M", x = "bisold")

## 1b: Computes flowtypes: Long format
list <- flowtabmat(matflow, matlist = "L")
colnames(list) <- c("i", "j", "Fij")
l_all <- flowtype(list, format = "L", x = "all")
l_sold <- flowtype(list, format = "L", x = "bisold")

# 2: Flowmapping: example of bisold
flowmap(l_sold,  
  format = "L", bkg, code = "EPT_NUM",  
  filter = TRUE, threshold = 20, taille = 5
)

<table>
<thead>
<tr>
<th>geoid</th>
<th>Geographical ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

One column dataframe in .csv.
Variable (COD_GEO_EPT) is the geographical code of the territory
Citation: APUR, 2018

Source

https://www.insee.fr/fr/statistiques/fichier/3566008/rp2015_mobpro_txt.zip

mat_ex

<table>
<thead>
<tr>
<th>mat_ex</th>
<th>Example of a small matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

Example of a small Origin-Destination flow dataset, in a matrice format
Index

cartograflow, 2
flowanalysis, 2, 2, 7
flowcarre, 2, 4, 15
flowcontig, 2, 5, 12
flowdist, 2, 6, 12
flowgini, 2, 3, 7
flowjointure, 2, 8, 13
flowmap, 2, 5, 6, 9
flowreduct, 2, 11
flows, 13
flowstructmat, 2, 13
flowtabmat, 13, 14
flowtype, 15

geoid, 16

mat_ex, 16