# Package 'cartograflow'

October 18, 2023

**Title** Filtering Matrix for Flow Mapping

**Version** 1.0.5

**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** 7.2.3

**Suggests** knitr, rmarkdown, testthat, cartography

**VignetteBuilder** knitr,

**Imports** dplyr, graphics, ggplot2, plotly, reshape2, rlang,sf, utils,
igraph

**NeedsCompilation** no

**Author** Sylvain Blondeau [aut, cre],
Francoise Bahoken [aut]

**Maintainer** Sylvain Blondeau <`blondeau.sylvain@yahoo.fr`>

**Repository** CRAN

**Date/Publication** 2023-10-17 22:40:21 UTC

## R topics documented:

## `cartograflow` *Cartograflow Package*

### 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 package.
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 flowcontig

## `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)
```

## 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

Bahoken Françoise, 2016,« La cartographie d'une sélection globale de flux, entre 'significativité' et 'densité' », Netcom Online, 30-3/4 | 2016, Online since 23 March 2017, connection on 05 May 2019. URL : http://journals.openedition.org/netcom/2565 ; DOI : 10.4000/netcom.2565

## Examples

```
library(cartograflow)
data(flowdata)

# 1/4: Computes Gini's coefficent
tabgini <- flowgini(ODpts = flows, origin = "i", destination = "j",
                     valflow = "Fij", lorenz.plot = FALSE)
### [1] Gini's coefficent = 73.16 %

# 2/4: Plot Lorenz curve
flowgini(tabgini,
        origin = "i", dest = "j", valflow = "Fij",
        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 %"
```

## flowcarre *Builds a square matrice from geographical nodes*

### Description

Builds a square and closed matrice 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 column : 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 matrice format<br>
- format is "L" for long format, as three column dataframe

- diagonal is "TRUE" to zero the main diagonal

### Examples

```
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
)
```

---

## `flowcontig` *Builds an ordinal distance matrices from a spatial features background*

---

### 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 ;
-ordre=1 : ODs places are adjacent, ie the flow have to cross only 1 boundary
-ordre=2 : ODs places are distant from 2 borders
-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

```
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

```
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

```
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

```
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 |

## Details

flowgini(...,lorenz.plot = TRUE) for ploting 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 be not assign a variable

value of the Gini's coefficent and the table : table<-flowgini(...,missing(gini.plot) or gini.plot = FALSE )

### References

Bahoken Françoise, 2016,« La cartographie d'une sélection globale de flux, entre 'significativité' et 'densité' », Netcom Online, 30-3/4 | 2016, Online since 23 March 2017, connection on 05 May 2019. URL : http://journals.openedition.org/netcom/2565 ; DOI : 10.4000/netcom.2565.
Grasland Claude, 2014, "Flows analysis carto", unpublished R functions.

### Examples

```
library(cartograflow)
data(flowdata)
# Computes Gini's coefficent
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
```

## 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 |
| `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"
)
```

---

# `flowlowup` *Extracts the triangular sub-matrix of flows*

---

## Description

Extracts the upper or lower triangular part of a matrix

## Usage

```
flowlowup(tab, origin = NULL, destination = NULL, fij = NULL, lowup, format, x)
```

## Arguments

| | |
|---|---|
| `tab` | is the input flow dataset |
| `origin` | the place of origin code |
| `destination` | the place of destination code |
| `fij` | the flow value between origin and destination places |
| `lowup` | for selecting lower or upper triangular sub-portion of the original matrix. See Details. |
| `format` | specify the flow dataset format, "M " for square matrix [n*n] or "L" for long [i,j,data] |
| `x` | enter the Enter the triangular part to be extracted: "low", "up". See Details. |

## Details

This function compute for all pairs or origin-destination places (i,j) a lower "low" or upper "up" triangular sub-portion of the original matrix - x = "up" for the part above the main diagonal
- x = "low" for the part below the main diagonal

## Examples

```
library(cartograflow)
data(flowdata)

###Extract the upper part of the matrix : Long format
tab_up <- flowlowup(flows, format="L", lowup="up")
tab_low<-flowlowup(flows, format="L", lowup="low")
```

# 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 |
| `fij` | 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 matrice'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 betwwen (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

```
rm(list=ls())
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
)
```

## `flowplaces` *Computes flow indicators per places*

### Description

Compute indicators per places (origin and/or destination ) from the margins of the matrix. Ex/ in and out degrees, gross and net flows, asymmetry .... from an initial matrix

### Usage

```
flowplaces(tab, origin = NULL, destination = NULL, fij = NULL, format, x)
```

### Arguments

| | |
|---|---|
| `tab` | is the input flow dataset |
| `origin` | the place of origin code |
| `destination` | the place of destination code |
| `fij` | the flow value between origin and destination places |
| `format` | specify the flow dataset format, "M " for square matrix [n*n] or "L" for long [i,j,data] |
| `x` | enter the flowplaces indicator type : "allflowplaces", "ini", "outi", "degi","intra", "Oi", "Dj", voli","bali","asyi". See Details. |

### Details

This function compute for all pairs or origin-destination places (i,j) a data table that describes the flows from the point of view of Origin / destination places - x = "ini" for the number of incoming links (as in-degree)
- x = "outi" for the number of outcoming links (as out-degree)
- x = "degi" for the total number of links (as in and out degrees)
- x = "intra" for total intra zonal interaction (if main diagonal is not empty

- x = "Dj" for the total flows received by (j) place
- x = "voli" for the total volume of flow per place
- x = "bali" for the net balance of flow per place
- x = "asyi" for the asymetry of flow per place
- x = "allflowplaces" for computing all the above indicators

## Examples

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

###1:Computes the total flow volume of places : Long format
voli <- flowplaces(flows, origin ="i",destination="j",fij="Fij",format = "L", x = "voli")
###2:Computes the total flows received by destination place : Long format
tab_bali <- flowplaces(flows, origin ="i",destination="j",fij="Fij",format = "L", x = "bali")
```

---

# `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 flowdata set |
| `tab.metric` | 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 continous or not.

(1) Metric is 'continous" for distance as euclidian, 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 matrice with a distance matrice, in long format (i,j, distance)
## 1/2: Computes the matrice distances
tab.distance <- flowdist(tab, dist.method = "euclidian", result = "dist")
tab.distance <- tab.distance %>% select(i, j, distance)
## 2/2: Reduce the flow matrice
tab.flow <- flowreduct(flows, tab.distance,
  metric = "continous",
  d.criteria = "dmax", d = 8567
)
```

## flows *MOBPRO: Commuting trips in 2015*

### Description

Statistical dataset in .csv: `Extraction of the french national census : "Mobilités professionnelles en 2015 : déplacements domicile - lieu de travail" - Base flux de mobilité - for the Greater Paris area.`
`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 bo use 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

```
library(cartograflow)
data(flowdata)

dim(mat_ex) # dimension fo the original matrice
### 10 11  # first colum is fill with the ID

tab <- flowstructmat(mat_ex)
dim(tab)
## 10 10 # dimension fo the resulting matrice
```

## flowtabmat *Changing the format of a flow dataset*

### Description

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

### Usage

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

### Arguments

| | |
|---|---|
| `tab` | flow dasaset, in matrice or long format |
| `matlist` | choose "matrice" or "long" for the resulting format. See Details. |

### Details

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

### Value

a flow dataset in matrice or in long format

### Examples

```
library(cartograflow)
data(flowdata)
# 1: From long to matrix format (n*m)
matFlow <- flowtabmat(flows, matlist = "M")
# 2: From matrix to long format [i,j,Fij]
listflow <- flowtabmat(matFlow, matlist = "L")
```

## flowtype *Compute bilateral several flow types*

### Description

Compute bilateral flow type: volumn (gross), balance (net), asymetry, min/max ... from an initial asymetric matrix

## Usage

```
flowtype(
  tab,
  origin = NULL,
  destination = NULL,
  fij = NULL,
  format,
  lowup,
  net,
  x
)
```

## Arguments

`tab` is the input flow dataset

`origin` the place of origin code

`destination` the place of destination code

`fij` the flow value between origin and destination places

`format` specify the flow dataset format, "M " for square matrix [n*n] or L for long [i,j,data]

`lowup` for extracting the lower or upper triangular sub-portion of the bilateral volum matrix. See Details.

`net` for extracting the "positive" or the "negative" flow values of the bilateral balance) matrix

`x` enter the flow indicator type : "alltypes", "flux", "transpose", "bivolum", "bibal","biasym","bimin", "bimax","birange" and "bidisym"

## Details

The matrice must be squared (if not, see flowcarre). This function compute for all pairs or origin-destination places (i,j) involved in an asymetric flow matrix (Fij<> Fji) several matrix :
Param x: - x = "flux" for the initial flow: (Fij)
- x = "transpose" for the reverse flow value: (Fji) =t(Fij)
- x = "bivolum" for the bilateral volum or gross flow: FSij=(Fij+Fji)
- x = "bibal" for the bilateral balance or net flow: FBij=(Fij-Fji)
- x = "biasym" for asymetry of bilateral flow: FAij=(FBij/FSij)
- x = "bimin" for the minimum of bilateral flow: minFij=(Fij, Fji)
- x = "bimax" for the maximum of bilateral flow: Fij(Fij, Fji)
- x = "birange" for the amplitude of bilateral flows: rangeFij=(maxFij - minFij)
- x = "bidisym" for the bilateral disymetry: FDij=(rangeFij/FSij) - x = "alltypes" for computing all the available types of flows
Param lowup is for reducing the matrix:
- lowup ="up" for triangular part above the main diagonal
- lowup = "low" for triangular part below the main diagonal
Param net is for extracting positive or negative flow values of the bilateral balance (bibal matrix):
- net ="negative" values

- net ="positive" values

## References

Bahoken Francoise, 2016, L’approche cartographique de la décomposition des matrices de flux, Mappemonde, Revue sur l’image géographique et les formes du territoire, number 116, URL : https://mappemonde-archive.mgm.fr/num44/articles/art14404.html

## Examples

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

## 1a:Computes flowtypes: Matrix 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 = "bibal")

## 1b:Computes flowtypes: Long format
types_all <- flowtype(flows,origin ="i",destination="j",fij="Fij", format = "L",
x = "alltypes")
bivol<- flowtype(flows,origin ="i",destination="j",fij="Fij",format = "L",
x = "bivolum",lowup="up")
bibal_net<- flowtype(flows,origin ="i",destination="j",fij="Fij",format = "L",
x = "bibal", net="negative")
```

---

geoid *Geographical ID*

---

## 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 *Example of a small matrice*

## Description

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

# Index