Package ‘MTA’

October 16, 2019

Title Multiscalar Territorial Analysis
Version 0.2.0
Description Build multiscalar territorial analysis based on various contexts.
License GPL-3

URL https://github.com/riatelab/MTA/
BugReports https://github.com/riatelab/MTA/issues/
LazyData true
Depends R (>= 3.5)
Imports methods, stats, sp, igraph, sf, cartography
Suggests knitr, rmarkdown, ineq, png, reshape2
VignetteBuilder knitr
Encoding UTF-8

RoxygenNote 6.1.1
NeedsCompilation no

Author Ronan Ysebaert [aut],
Nicolas Lambert [aut],
Timothée Giraud [aut, cre]
Maintainer Timothée Giraud <timothée.giraud@cnrs.fr>
Repository CRAN

Date/Publication 2019-10-16 10:00:02 UTC

R topics documented:

  cardist ................................................................. 2
  com ................................................................. 2
  ept ................................................................. 3
  gdev ................................................................. 4
  mapmst .......................................................... 5
  mas ............................................................... 7
### Description

Travel time between Grand Paris Metropole communes’ centroids by car, in minutes. Row names and column names match the DEPCOM field in `com`.

### Source

The matrix is computed using the osrm package ([https://cran.r-project.org/package=osrm](https://cran.r-project.org/package=osrm)).

Data (c) OpenStreetMap contributors, ODbL 1.0. http://www.openstreetmap.org/copyright

Routes: OSRM. http://project-osrm.org/

### Examples

```r
data(GrandParisMetropole)
cardist[1:10,1:10]
```

### Description

Data on the Grand Paris Metropole communes, included in a sf object.

### Format

A data frame with 150 rows and 10 variables:

- **DEPCOM**: Commune identifiers
- **LIBCOM**: Commune names
- **EPT**: EPT identifiers of the commune
- **LIBEPT**: EPT names of the commune
- **DEP**: Identifiers of the department
- **INC**: Amount of income tax reference (in euros)
- **TH**: Number of tax households
- **geometry**: Commune geometry
Source

Direction générale des finances publiques, income tax 2014 (2013 incomes):
http://www.impots.gouv.fr/portal/dgi/public/statistiques.impot?espId=-4&pageId=
stat_donnees_detaill[éees&sfid=4503
Atelier parisien d’urbanisme, Grand Paris communal composition (2015-12-17):
http://www.apur.org/article/composition-12-territoires-metropole-grand-paris

Examples

data(GrandParisMetropole)
head(com)

---

<table>
<thead>
<tr>
<th>ept</th>
<th>Grand Paris Metropole EPTs</th>
</tr>
</thead>
</table>

Description

sf object. Grand Paris Metropole EPTs. EPTs (Etablissements Publics Territoriaux) are groups of communes.

Format

EPT  EPT identifiers
LIBEPT  EPT names

Source

Atelier parisien d’urbanisme, Grand Paris communal composition (2015-12-17):
http://www.apur.org/article/composition-12-territoires-metropole-grand-paris

Examples

library(sf)
data(GrandParisMetropole)
plot(st_geometry(ept))
**gdev**  

*General Deviation*

**Description**

Compute the deviation of each territorial unit as regards to all the study area (or a reference value).

**Usage**

```r
gdev(x, var1, var2, type = "rel", ref)
```

**Arguments**

- `x`: a data.frame, an sf object or a SpatialPolygonsDataFrame, including `var1` and `var2`.
- `var1`: name of the numerator variable in `x`.
- `var2`: name of the denominator variable in `x`.
- `type`: type of deviation; "rel" for relative deviation, "abs" for absolute deviation (see Details).
- `ref`: ratio of reference; if missing, the ratio of reference is the one of the whole study area (`sum(var1) / sum(var2)`).

**Details**

The relative global deviation is the ratio between `var1/var2` and `ref` (\(100 \times (var1 / var2) / ref\)). Values greater than 100 indicate that the unit ratio is greater than the ratio of reference. Values lower than 100 indicate that the unit ratio is lower than the ratio of reference.

The absolute global deviation is the amount of numerator that could be moved to obtain the ratio of reference on all units. (\((var1 - (ref \times var2))\)).

**Value**

A vector is returned.

**Examples**

```r
library(sf)
library(cartography)
# load data
data("GrandParisMetropole")

# compute absolute global deviation
com$gdevabs <- gdev(x = com, var1 = "INC", var2 = "TH", type = "abs")

# compute relative global deviation
com$gdevrel <- gdev(x = com, var1 = "INC", var2 = "TH", type = "rel")

# Deviations maps
```
# set graphical parameters
par(mar = c(0,0,1.2,0))
# set breaks
bks <- c(min(com$gdevrel),50,75,100,125,150,max(com$gdevrel))
# set color palette
cols <- carto.pal(pal1 = "blue.pal", n1 = 3,
                  pal2 = "wine.pal", n2 = 3)
# plot a choropleth map of the relative global deviation
choroLayer(x = com, var = "gdevrel",
           legend.pos = "topleft",
           legend.title.txt = "Relative Deviation\n(100 = general average)",
           breaks = bks, border = NA,
           col = cols)
# add symbols proportional to the absolute general deviation
com$sign <- ifelse(test = com$gdevabs<0, yes = "negative", no = "positive")
propSymbolsTypoLayer(x = com, var = "gdevabs",var2 = "sign",
                      legend.var.pos = "left",legend.values.rnd = -2,
                      legend.var2.values.order = c("positive", "negative"),
                      legend.var.title.txt = "Absolute Deviation\n(Income redistribution)",
                      col = c("#ff000050","#0000ff50"),legend.var2.pos = "n",
                      legend.var.style = "e", inches = 0.2)
# add EPT boundaries
plot(st_geometry(ept), add=TRUE)
# add a layout
layoutLayer(title = "General Deviation (reference: Grand Paris Metropole)",
            sources = "GEOFLA® 2015 v2.1, Apur, impots.gouv.fr",
            north = TRUE, scale = 5, tabtitle = TRUE, frame = FALSE, theme = "red.pal",
            author = "MTA")

mapmst  

Map Multiscalar Typology

Description

Map the multiscalar typology according to the three relative deviations (general: G, territorial: T and spatial: S). The elementary units are classified in eight classes according to their three relative positions and they are mapped with appropriate colors.

Usage

mapmst(x, var1, var2, ref, key, order, dist, mat, xid, threshold,
       superior = FALSE, colNA = "white", border = "grey80", lwd = 0.2,
       upborder = "black", uplwd = "1", add = FALSE)

Arguments

x
var1
var2
an sf object or a SpatialPolygonsDataFrame including var1 and var2.
name of the numerator variable in x.
name of the denominator variable in x.
ref ratio of reference; if missing, the ratio of reference is the one of the whole study area (sum(var1) / sum(var2)).
key aggregation key field for measuring the deviation (intermediate territorial level).
order contiguity order.
dist distance threshold defining the contiguity. The cartesian distance between units centroids is used by default; use mat to apply different metrics.
mat a distance matrix (road distance, travel time...) between x units. Row and column names must fit x identifiers. (optional)
xid identifier field in x (to be used for importing a personal distance matrix). Default to the first column.
threshold defined to build the typology (100 is considered as the average).
superior if TRUE, deviation values must be greater than threshold. If FALSE, deviation values must be lower than threshold.

colNA no data color
border color of the polygons border
lwd borders width
upborder color of the intermediate territorial level border
uplwd intermediate territorial level border width
add whether to add the layer to an existing plot (TRUE) or not (FALSE)

Value
A typology map with colours and a sf object including the ratio (var1/var2), the 3 relative deviations (G, T and S) and the resulting ordered typology (0 to 7).

Typology (which deviation is above/under the threshold):
• 0: none
• 1: G
• 2: T
• 3: G and T
• 4: S
• 5: G and S
• 6: T and S
• 7: G, T and S

Examples
library(sf)
library(cartography)
data("GrandParisMetropole")
# Map wealthiest territories
synthesis <- mapmst(x = com, var1 = "INC", var2 = "TH", key = "EPT",
                   order = 1, threshold = 125, superior = TRUE)
# add a layout and adapted legend
layoutLayer(title = "Multiscalar synthesis - Income per household 2013",
sources = "GEOFLA® 2015 v2.1, Apur, impots.gouv.fr",
north = TRUE, scale = 5, tabtitle = TRUE, frame = FALSE, theme = "red.pal",
author = "MTA"
100: Deviation average
G: Situation as compared to the global context (Grand Paris Area)
T: Situation as compared to the territorial context (EPT of belonging)
S: Situation as compared to the neighbourhood context (contiguity order 1)"
# add label territorial objects above 125% for all the deviations
labelLayer(x = synthesis[synthesis$mst == 7, ], txt = "LIBCOM", cex = 0.6,
halo = TRUE, overlap = FALSE)

# Map poorest territories
synthesis <- mapmst(x = com, var1 = "INC", var2 = "TH", key = "EPT",
order = 1, threshold = 75, superior = FALSE)
# add a layout
layoutLayer(title = "Multiscalar synthesis - Income per household 2013",
sources = "GEOFLA® 2015 v2.1, Apur, impots.gouv.fr",
north = TRUE, scale = 5, tabtitle = TRUE, frame = FALSE, theme = "red.pal",
author = "MTA"
100: Deviation average
G: Situation as compared to the global context (Grand Paris Area)
T: Situation as compared to the territorial context (EPT of belonging)
S: Situation as compared to the neighbourhood context (contiguity order 1)"

# add labels for territorial objects under 75 % for all the deviations
labelLayer(x = synthesis[synthesis$mst == 7, ], txt = "LIBCOM", cex = 0.6,
halo = TRUE)

---

**mas**  
*Multiscalar Absolute Synthesis*

**Description**  
This function sums the total amount of redistributions according to the three absolute deviations (global, territorial, spatial).

**Usage**  

```r
mas(x, xid, var1, var2, ref, key, order, dist, mat)
```

**Arguments**  

- `x`: an sf object or a SpatialPolygonsDataFrame including var1 and var2.
- `xid`: identifier field in x (to be used for importing a personal distance matrix). Default to the first column.
- `var1`: name of the numerator variable in x.
mst

Multiscalar Typology

Description

Compute a multiscalar typology according to the three relative deviations (general: G, territorial: T and spatial: S). The elementary units are classified in eight classes according to their three relative positions.

Usage

mst(x, var1, var2, ref, key, order, dist, mat, xid, threshold, superior = FALSE)

Arguments

x a sf object or a SpatialPolygonsDataFrame including var1 and var2.
var1 name of the numerator variable in x.
var2 name of the denominator variable in x.
ref ratio of reference; if missing, the ratio of reference is the one of the whole study area (sum(var1) / sum(var2)).
key aggregation key field for measuring the deviation (intermediate territorial level).
order contiguity order.
dist distance threshold defining the contiguity. The cartesian distance between units centroids is used by default (see st_distance); use mat to apply different metrics.
mat a distance matrix (road distance, travel time...) between x units. Row and column names must fit xid identifiers. (optional)

Value

A dataframe including the mass of numerator to redistribute to reach a perfect equilibrium according to the 3 contexts, expressed in numerator measure unit and as a share of the numerator mass.

Examples

data("GrandParisMetropole")
redistr <- mas(x = com, var1 = "INC", var2 = "TH", dist = 5000, key = "EPT")
redistr

8
mst

var2
name of the denominator variable in x.
ref
ratio of reference; if missing, the ratio of reference is the one of the whole study area (sum(var1) / sum(var2)).
key
aggregation key field for measuring the deviation (intermediate territorial level).
order
contiguity order.
dist
distance threshold defining the contiguity. The cartesian distance between units centroids is used by default (see st_distance); use mat to apply different metrics.
mat
a distance matrix (road distance, travel time...) between x units. Row and column names must fit xid identifiers. (optional)
distance threshold defining the contiguity. The cartesian distance between units centroids is used by default; use mat to apply different metrics.

a distance matrix (road distance, travel time...) between x units. Row and column names must fit x identifiers. (optional)

identifier field in x (to be used for importing a personal distance matrix). Default to the first column.

defined to build the typology (100 is considered as the average).

if TRUE, deviation values must be greater than threshold. If FALSE, deviation values must be lower than threshold.

A dataframe including the ratio (var1/var2), the 3 relative deviations (G, T and S) and the resulting ordered typology (0 to 7).

Typology (which deviation is above/under the threshold):

- 0: none
- 1: G
- 2: T
- 3: G and T
- 4: S
- 5: G and S
- 6: T and S
- 7: G, T and S

# load data
data("GrandParisMetropole")
# Multiscalar typology - Wealthiest territories
row.names(com) <- com$LIBCOM
synthesis <- mst(x = com, var1 = "INC", var2 = "TH", key = "EPT",
order = 1, threshold = 125, superior = TRUE)
# Territories above 125% for the 3 deviations
synthesis[synthesis$mst == 7, ]

# Multiscalar typology - Poorest territories
synthesis <- mst (x = com, var1 = "INC", var2 = "TH", key = "EPT",
order = 1, threshold = 75, superior = FALSE)
# Territories below 75 % for the three deviations
synthesis[synthesis$mst == 7, ]
MTA  Multiscalar Territorial Analysis

Description

Build multiscalar territorial analysis based on various contexts for a given ratio defined by a numerator and a denominator.

Main functions:

- `gdev`: general deviation of each territorial unit as regards to all the study area (or a reference value).
- `tdev`: territorial deviation of each territorial unit as regards to an intermediate territorial level of reference.
- `sdev`: spatial deviation of each territorial unit as regards to its geographical neighborhood.
- `mst`: multiscalar typology based on the three deviations.
- `mas`: multiscalar absolute synthesis, total amount of redistributions based on the three deviations.
- `mapmst`: multiscalar typology map.

References


sdev  Spatial Deviation

Description

Compute the deviation of each territorial unit as regards to its geographical neighborhood. Neighborhood is defined either by contiguity order, by a distance value or by a personal matrix (travel time...)

Usage

`sdev(x, var1, var2, type = "rel", xid, order, dist, mat)`
Arguments

**x**  an sf object or a SpatialPolygonsDataFrame including var1 and var2.

**var1**  name of the numerator variable in x.

**var2**  name of the denominator variable in x.

**type**  type of deviation; "rel" for relative deviation, "abs" for absolute deviation (see Details).

**xid**  identifier field in x (to be used for importing a personal distance matrix). Default to the first column of x. (optional)

**order**  contiguity order.

**dist**  distance threshold defining the contiguity. The cartesian distance between units centroids is used by default; use mat to apply different metrics.

**mat**  a distance matrix (road distance, travel time...) between x units. Row and column names must fit xid identifiers. (optional)

Details

The relative spatial deviation is the ratio between var1/var2 and var1/var2 in the specified neighborhood. Values greater than 100 indicate that the unit ratio is greater than the ratio in its neighborhood. Values lower than 100 indicate that the unit ratio is lower than the ratio in its neighborhood.

The absolute spatial deviation is the amount of numerator that could be moved to obtain the same ratio in all units of its neighborhood.

Value

A vector is returned.

Examples

```r
library(sf)
library(cartography)
# load data
data("GrandParisMetropole")

# compute absolute spatial deviation in a neighborhood defined by a contiguity order of 2.
com$sdevabs <- sdev(x = com, var1 = "INC", var2 = "TH", order =2, type = "abs")

#compute relative spatial deviation in a neighborhood defined within a distance of 5km between communes' centroids
com$sdevrel <- sdev(x = com, var1 = "INC", var2 = "TH", type = "rel", dist = 5000)

# compute absolute spatial deviation in a neighborhood defined within a road travel time of 10 minutes by car
com$scardevabs <- sdev(x = com, var1 = "INC", var2 = "TH", type = "abs", dist = 10, mat = cardist)

# compute relative spatial deviation in a neighborhood defined within a road travel time of 10 minutes by car
com$scardevrel <- sdev(x = com, var1 = "INC", var2 = "TH", type = "rel", dist = 10, mat = cardist)
```
# map deviations
# set graphical parameters
par(mar = c(0,0,1.2,0))
# set breaks
bks <- c(min(com$scardevrel),75,100,125,150,max(com$scardevrel))
# set colot palette
cols <- carto.pal(pal1 = "blue.pal", n1 = 2,
                   pal2 = "wine.pal", n2 = 3)
# plot a choropleth map of the relative spatial deviation
choroLayer(x = com, var = "scardevrel",
           legend.pos = "topleft",
           legend.title.txt = "Relative Deviation\n(100 = spatial average)",
           breaks = bks, border = NA,
           col = cols)
# add symbols proportional to the absolute spatial deviation
com$sign <- ifelse(test = com$scardevabs<0, yes = "negative", no = "positive")
propSymbolsTypoLayer(x = com, var = "scardevabs", var2 = "sign",
                      legend.var.pos = "left",
                      legend.var.title.txt = "Absolute Deviation\n(Income redistribution)",
                      col = c("#ff000050",="#0000ff50"),
                      legend.var2.values.order = c("positive", "negative"),
                      legend.var2.pos = "n",
                      legend.var.style = "e", inches = 0.2)
# add EPT boundaries
plot(st_geometry(ept), add=TRUE)
# add a layout
layoutLayer(title = "Spatial Deviation (neighborhood : 10 minutes by car)",
            sources = "GEOFLA® 2015 v2.1, Apur, impots.gouv.fr",
            north = TRUE, scale = 5, tabtitle = TRUE, frame = FALSE, theme = "red.pal",
            author = "MTA")

---

### tdev

**Territorial Deviation**

**Description**

Compute the deviation of each territorial unit as regards to an intermediate territorial level of reference.

**Usage**

```r
tdev(x, var1, var2, type = "rel", key)
```

**Arguments**

- `x`: a dataframe, an sf object or a SpatialPolygonsDataFrame including `var1` and `var2`, and an aggregation key field (territorial belonging).
- `var1`: name of the numerator variable in `x`.
- `var2`: name of the denominator variable in `x`. 

---
type  type of deviation; "rel" for relative deviation, "abs" for absolute deviation (see Details).
key  aggregation key field for measuring the deviation (intermediate territorial level).

Details

The relative territorial deviation is the ratio between var1/var2 and var1/var2 at the aggregated level. Values greater than 100 indicate that the unit ratio is greater than the ratio at the aggregated level. Values lower than 100 indicate that the unit ratio is lower than the ratio of the aggregated level.

The absolute territorial deviation is the amount of numerator that could be moved to obtain the ratio of the aggregated level on all belonging units.

Value

A vector is returned.

Examples

```r
library(sf)
library(cartography)
# load data
data("GrandParisMetropole")
# compute absolute territorial deviation (EPT level)
com$tdevabs <- tdev(x = com, var1 = "INC", var2 = "TH", type = "abs", key = "EPT")
# compute relative territorial deviation (EPT level)
com$tdevrel <- tdev(x = com, var1 = "INC", var2 = "TH", type = "rel", key = "EPT")

# map deviations
# set graphical parameters
par(mar = c(0,0,1.2,0))
# set breaks
bks <- c(min(com$tdevrel),75,100,125,150,max(com$tdevrel))
# set color palette
cols <- carto.pal(pal1 = "blue.pal", n1 = 2,
    pal2 = "wine.pal", n2 = 3)
# plot a choropleth map of the relative territorial deviation
choroLayer(x = com, var = "tdevrel",
    legend.pos = "topleft",
    breaks = bks, border = NA,
    legend.title.txt = "Relative Deviation\n(100 = territorial average)",
    col = cols)
# add symbols proportional to the absolute territorial deviation
com$sign <- ifelse(test = com$tdevabs < 0, yes = "negative", no = "positive")
propSymbolsTypoLayer(com, var = "tdevabs", var2 = "sign",
    legend.var.pos = "left", legend.values.rnd = -2,
    legend.var2.values.order = c("positive", "negative"),
    legend.var.title.txt = "Absolute Deviation\n(Income redistribution)",
    col = c("#ff000050","#0000ff50"), legend.var2.pos = "n",
    legend.var.style = "e", inches = 0.2)
```
# add EPT boundaries
plot(st_geometry(ept), add=TRUE)
# add a layout
layoutLayer(title = "Territorial Deviation (reference : EPT)",
            sources = "GEOFLA© 2015 v2.1, Apur, impots.gouv.fr",
            north = TRUE, scale = 5, tabtitle = TRUE, frame = FALSE, theme = "red.pal",
            author = "MTA")
Index

cardist, 2
com, 2, 2

ept, 3

gdev, 4, 10

mapmst, 5, 10
mas, 7, 10
mst, 8, 10
MTA, 10
MTA-package (MTA), 10

sdev, 10, 10
st_distance, 8

tdev, 10, 12