

# Package ‘MTA’

October 12, 2022

**Title** Multiscalar Territorial Analysis

**Version** 0.4.1

**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/>

**Depends** R (>= 3.5)

**Imports** methods, stats, graphics, igraph, sf

**Suggests** knitr, rmarkdown, ineq, png, mapsf

**VignetteBuilder** knitr

**Encoding** UTF-8

**RoxygenNote** 7.2.0

**NeedsCompilation** no

**Author** Ronan Ysebaert [aut],  
Nicolas Lambert [aut],  
Timothée Giraud [aut, cre]

**Maintainer** Timothée Giraud <[timothee.giraud@cnrs.fr](mailto:timothee.giraud@cnrs.fr)>

**Repository** CRAN

**Date/Publication** 2022-06-03 16:00:02 UTC

## R topics documented:

|                     |    |
|---------------------|----|
| bidev . . . . .     | 2  |
| cardist . . . . .   | 3  |
| com . . . . .       | 4  |
| ept . . . . .       | 4  |
| gdev . . . . .      | 5  |
| map_bidev . . . . . | 6  |
| map_mst . . . . .   | 8  |
| mas . . . . .       | 11 |

|                      |    |
|----------------------|----|
| mst . . . . .        | 12 |
| MTA . . . . .        | 13 |
| plot_bidev . . . . . | 14 |
| plot_mst . . . . .   | 16 |
| sdev . . . . .       | 17 |
| tdev . . . . .       | 19 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>21</b> |
|--------------|-----------|

---

|       |                                            |
|-------|--------------------------------------------|
| bidev | <i>Multiscalar Typology (2 deviations)</i> |
|-------|--------------------------------------------|

---

### Description

Compute a multiscalar typology according to two relative deviations relative.

### Usage

```
bidev(x, dev1, dev2, breaks = c(25, 50, 100), xid = NULL)
```

### Arguments

|        |                                                                                                                                                          |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| x      | a sf object including 2 deviations among gdev, tdev and sdev.                                                                                            |
| dev1   | column name of the first relative deviation in x.                                                                                                        |
| dev2   | column name of the second relative deviation in x.                                                                                                       |
| breaks | Distance to the index 100 (average of the context), in percentage. A vector of three values. Default c(25,50,100). 25 to indexes 80 and 125. 50 and 200. |
| xid    | identifier field in x. Default the first column.                                                                                                         |

### Value

a column in x including a bidev column delivering the result of the typology. Values are classified in 13 classes according to their respective position on the two selected deviations and their distance to the average.

bidev typology values :

- ZZ: Near the average for the two selected deviation
- A1: Above the average for dev1 and dev2, distance to the average : +
- A2: Above the average for dev1 and dev2, distance to the average : ++
- A3: Above the average for dev1 and dev2, distance to the average : +++
- B1: Above the average for dev1 and below for dev2, distance to the average : +
- B2: Above the average for dev1 and below for dev2, distance to the average : ++
- B3: Above the average for dev1 and below for dev2, distance to the average : +++
- C1: Below the average for dev1 and dev2, distance to the average : +
- C2: Below the average for dev1 and dev2, distance to the average : ++

- C3: Below the average for dev1 and dev2, distance to the average : +++
- D1: Below the average for dev1 and above for dev2, distance to the average : +
- D2: Below the average for dev1 and above for dev2, distance to the average : ++
- D3: Below the average for dev1 and above for dev2, distance to the average : +++

### Examples

```
# bidev synthesis on general and territorial deviation (income data)
# Load data
library(sf)
com <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "com", quiet = TRUE)

# Prerequisite - Compute 2 deviations
com$gdev <- gdev(x = com, var1 = "INC", var2 = "TH")
com$tdev <- tdev(x = com, var1 = "INC", var2 = "TH", key = "EPT")

# Compute bidev
com$bidev <- bidev(x = com, dev1 = "gdev", dev2 = "tdev")
```

---

cardist

*Time Distance Matrix Between Communes*

---

### 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>).  
Data (c) OpenStreetMap contributors, ODbL 1.0. <http://www.openstreetmap.org/copyright>  
Routes: OSRM. <http://project-osrm.org/>

### Examples

```
cardist <- read.table(system.file("cardist.txt", package = "MTA"), check.names = FALSE)
cardist <- as.matrix(cardist)
cardist[1:10, 1:10]
```

---

 com

*Grand Paris Metropole Communes Data*


---

### 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 departement

**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):

<https://www.impots.gouv.fr/statistiques>

Atelier parisien d'urbanisme, Grand Paris communal composition (2015-12-17):

<https://www.apur.org/fr/nos-travaux/composition-12-territoires-metropole-grand-paris>

### Examples

```
library(sf)
com <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "com", quiet = TRUE)
```

---

 ept

*Grand Paris Metropole EPTs*


---

### 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):

<https://www.apur.org/fr/nos-travaux/composition-12-territoires-metropole-grand-paris>

## Examples

```
library(sf)
ept <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "ept", quiet = TRUE)
```

---

`gdev` *General Deviation*

---

## Description

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

## Usage

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

## Arguments

|                   |                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>    | a data.frame or a sf object including var1 and var2.                                                                                               |
| <code>var1</code> | name of the numerator variable in x.                                                                                                               |
| <code>var2</code> | name of the denominator variable in x.                                                                                                             |
| <code>type</code> | type of deviation; "rel" for relative deviation, "abs" for absolute deviation (see Details).                                                       |
| <code>ref</code>  | ratio of reference; if missing, the ratio of reference is the one of the whole study area ( $\text{sum}(\text{var1}) / \text{sum}(\text{var2})$ ). |

## Details

The relative global deviation is the ratio between var1/var2 and ref ( $100 * (\text{var1} / \text{var2}) / \text{ref}$ ). 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. ( $(\text{var1} - (\text{ref} * \text{var2}))$ ).

## Value

A vector is returned.

**Examples**

```

# Load data
library(sf)
com <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "com", quiet = TRUE)
ept <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "ept", quiet = TRUE)

if(require(mapsf)){
# 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")

# relative deviation map
# set breaks
bks <- c(min(com$gdevrel), 50, 75, 100, 125, 150, max(com$gdevrel))
# plot a choropleth map of the relative global deviation
mf_map(x = com, var = "gdevrel", type = "choro", leg_pos = "topleft",
       leg_title = "Relative Deviation\n(100 = general average)",
       breaks = bks, border = NA,
       pal = c("#4575B4", "#91BFDB", "#E0F3F8", "#FEE090", "#FC8D59", "#D73027"))

# add EPT boundaries
mf_map(x = ept, col = NA, add = TRUE)

# layout
mf_layout(title = "General Deviation (reference: Grand Paris Metropole)",
          credits = paste0("Sources: GEOFLA® 2015 v2.1, Apur, impots.gouv.fr",
                           "\nMTA", packageVersion("MTA")),
          arrow = FALSE)

# absolute deviation map
com$sign <- ifelse(test = com$gdevabs < 0, yes = "Under-Income", no = "Over-Income")
mf_map(ept)

mf_map(x = com, var = c("gdevabs", "sign"), type = "prop_typo", inches = 0.2,
       leg_title = c("Absolute Deviation\n(Income redistribution, euros)",
                    "Redistribution direction"),
       leg_pos = c("topleft", "topright"), leg_val_rnd = -2,
       val_order = c("Under-Income", "Over-Income"),
       pal = c("#ff0000", "#0000ff"), add = TRUE)

# layout
mf_layout(title = "General Deviation (reference: Grand Paris Metropole)",
          credits = paste0("Sources: GEOFLA® 2015 v2.1, Apur, impots.gouv.fr",
                           "\nMTA", packageVersion("MTA")),
          arrow = FALSE)
}

```

**Description**

Compute the multiscalar typology (2 deviations) and propose colors for mapping the results.

**Usage**

```
map_bidev(x, dev1, dev2, breaks = c(25, 50, 100), xid = NULL)
```

**Arguments**

|        |                                                                                                                                                          |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| x      | A sf object including a variable resulting from the mst function.                                                                                        |
| dev1   | column name of the first relative deviation in x.                                                                                                        |
| dev2   | column name of the second relative deviation in x.                                                                                                       |
| breaks | Distance to the index 100 (average of the context), in percentage. A vector of three values. Default c(25,50,100). 25 to indexes 80 and 125. 50 and 200. |
| xid    | identifier field in x. Default the first column.                                                                                                         |

**Value**

A list including a ordered sf object for mapping mst column (geom) and a vector of suggested colors (cols).

bidev typology values :

- ZZ: Near the average for the two selected deviation
- A1: Above the average for dev1 and dev2, distance to the avarage : +
- A2: Above the average for dev1 and dev2, distance to the avarage : ++
- A3: Above the average for dev1 and dev2, distance to the avarage : +++
- B1: Above the average for dev1 and below for dev2, distance to the avarage : +
- B2: Above the average for dev1 and below for dev2, distance to the avarage : ++
- B3: Above the average for dev1 and below for dev2, distance to the avarage : +++
- C1: Below the average for dev1 and dev2, distance to the avarage : +
- C2: Below the average for dev1 and dev2, distance to the avarage : ++
- C3: Below the average for dev1 and dev2, distance to the avarage : +++
- D1: Below the average for dev1 and above for dev2, distance to the avarage : +
- D2: Below the average for dev1 and above for dev2, distance to the avarage : ++
- D3: Below the average for dev1 and above for dev2, distance to the avarage : +++

**Examples**

```
# Focus on exceptional values (50, 100 and 200 % above-under the average)
# Load data
library(sf)
com <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "com", quiet = TRUE)
ept <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "ept", quiet = TRUE)

# Prerequisite - Compute 2 deviations
```

```

com$gdev <- gdev(x = com, var1 = "INC", var2 = "TH")
com$tdev <- tdev(x = com, var1 = "INC", var2 = "TH", key = "EPT")

# Compute map_bidev
bidev <- map_bidev(x = com, dev1 = "gdev", dev2 = "tdev", breaks = c(50, 100, 200))

# Unlist resulting function
com <- bidev$geom
cols <- bidev$cols

#Visualization
# One side for the map, another for the plot
opar <- par(mfrow = c(1,2), mar = c(0,4,0,0))

if(require(mapsf)){
# Cartography
mf_map(x = com, var = "bidev", type = "typo", val_order = unique(com$bidev),
      border = "grey50", pal = cols, lwd = 0.2, leg_pos = "n")
mf_map(ept, col = NA, add = TRUE)

# Label territories in the C3 category
mf_label(x = com[com$bidev == "C3",], var = "LIBCOM", halo = TRUE)

mf_layout(title = "2-Deviations synthesis : general and territorial contexts",
  credits = paste0("Sources: GEOFLA® 2015 v2.1, Apur, impots.gouv.fr",
    "\nMTA", packageVersion("MTA")),
  arrow = FALSE)

# Add plot_bidev on the right side of the map
plot_bidev(x = com, dev1 = "gdev", dev2 = "tdev",
  dev1.lab = "General deviation (MGP Area)",
  dev2.lab = "Territorial deviation (EPT of belonging)",
  breaks = c(50, 100, 200),
  lib.var = "LIBCOM", lib.val = "Clichy-sous-Bois", cex.lab = 0.8)
par(opar)
}

```

---

map\_mst

---

*Map Multiscalar Typology (3 deviations)*


---

## Description

Compute the multiscalar typology (3 deviations) and propose labels and colours for mapping the results.

## Usage

```
map_mst(x, gdevrel, tdevrel, sdevrel, threshold, superior = FALSE, xid = NULL)
```



**Arguments**

|           |                                                                                                                    |
|-----------|--------------------------------------------------------------------------------------------------------------------|
| x         | a sf object or a dataframe including 3 pre-calculated deviations.                                                  |
| gdevrel   | name of the general relative deviation variable in x.                                                              |
| tdevrel   | name of the territorial relative deviation variable in x.                                                          |
| sdevrel   | name of the the spatial relative deviation variable in x.                                                          |
| 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. |
| xid       | identifier field in x. Default the first column.                                                                   |

**Value**

A list including a ordered sf object for mapping mst column (geom), a vector of suggested colors (cols) and a vector of adapted labels for the legend (leg\_labels).

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

- 0: none (mst value = 0)
- G: general only (mst value = 1)
- T: territorial only (mst value = 2)
- G-T: general and Territorial (mst value = 3)
- S: spatial only (mst value = 4)
- G-S: general and Spatial (mst value = 5)
- T-S: territorial and Spatial (mst value = 6)
- G-T-S: all deviations (mst value = 7)

**Examples**

```
# Load data
library(sf)
com <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "com", quiet = TRUE)
ept <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "ept", quiet = TRUE)

# Prerequisites - Compute the 3 deviations
com$gdev <- gdev(x = com, var1 = "INC", var2 = "TH")
com$tdev <- tdev(x = com, var1 = "INC", var2 = "TH", key = "EPT")
com$sdev <- sdev(x = com, var1 = "INC", var2 = "TH", order = 1)

#Example 1 - Wealthiest territorial units
# Compute map_mst
mst <- map_mst(x = com, gdevrel = "gdev", tdevrel = "tdev", sdevrel = "sdev",
threshold = 125, superior = TRUE)

# Unlist outputs of the function
com <- mst$geom
cols <- mst$cols
leg_val <- mst$leg_val
```

```

if(require(mapsf)){
# Cartography
mf_map(x = com, var = "mst", type = "typo", border = "white", lwd = 0.2,
      pal = cols, val_order = unique(com$mst), leg_pos = "n")
mf_map(ept, col = NA, border = "black", lwd = 1, add = TRUE)

mf_legend(type = "typo", val = leg_val, pal = cols,
         title = "Situation on General (G)\nTerritorial (T) and\nSpatial (S) contexts",
         pos = "topleft")

mf_label(x = com[com$mst == 7,], var = "LIBCOM",
        cex = 0.6, halo = TRUE, overlap = FALSE)

mf_layout(title = "3-Deviations synthesis : Territorial units above index 125",
         credits = paste0("Sources: GEOFLA® 2015 v2.1, Apur, impots.gouv.fr",
                          "\nMTA", packageVersion("MTA")),
         arrow = FALSE)
}

# Example 2 - Lagging territorial units
# Compute map_mst
mst <- map_mst(x = com, gdevrel = "gdev", tdevrel = "tdev", sdevrel = "sdev",
              threshold = 75, superior = FALSE)

# Unlist resulting function
com <- mst$geom
cols <- mst$cols
leg_val <- mst$leg_val

if(require(mapsf)){
# Cartography
mf_map(x = com, var = "mst", type = "typo", border = "white", lwd = 0.2,
      pal = cols, val_order = unique(com$mst), leg_pos = "n")
mf_map(ept, col = NA, border = "black", lwd = 1, add = TRUE)

mf_legend(type = "typo", val = leg_val, pal = cols,
         title = "Situation on General (G)\nTerritorial (T) and\nSpatial (S) contexts",
         pos = "topleft")

mf_label(x = com[com$mst == 7,], var = "LIBCOM",
        cex = 0.6, halo = TRUE, overlap = FALSE)

mf_layout(title = "3-Deviations synthesis : Territorial units below index 75",
         credits = paste0("Sources: GEOFLA® 2015 v2.1, Apur, impots.gouv.fr",
                          "\nMTA", packageVersion("MTA")),
         arrow = FALSE)
}

```

---

mas *Multiscalar Absolute Synthesis*

---

**Description**

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

**Usage**

```
mas(x, gdevabs, tdevabs, sdevabs, num)
```

**Arguments**

|         |                                                                                 |
|---------|---------------------------------------------------------------------------------|
| x       | a sf object or a dataframe including gdevabs, tdevabs, sdevabs and num columns. |
| gdevabs | name of the general absolute deviation variable in x.                           |
| tdevabs | name of the territorial absolute deviation variable in x.                       |
| sdevabs | name of the spatial absolute deviation variable in x.                           |
| num     | name of the numerator used for computing the 3 absolute deviations in x.        |

**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**

```
# Load data
library(sf)
com <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "com", quiet = TRUE)

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

# Territorial absolute deviation calculation
com$mdevabs <- tdev(x = com, var1 = "INC", var2 = "TH", type = "abs",
  key = "LIBEPT")

# Local absolute deviation calculation redistribution
com$ldevabs <- sdev(x = com, xid = "DEPCOM", var1 = "INC", var2 = "TH",
  order = 1, type = "abs")

# Compute the synthesis DataFrame (absolute deviations)
mas(x = com,
  gdevabs = "gdevabs",
  tdevabs = "mdevabs",
  sdevabs = "ldevabs",
  num = "INC")
```

---

`mst`*Multiscalar Typology (3 deviations)*

---

### 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, gdevrel, tdevrel, sdevrel, threshold, superior = FALSE)
```

### Arguments

|                        |                                                                                                                    |
|------------------------|--------------------------------------------------------------------------------------------------------------------|
| <code>x</code>         | a sf object or a dataframe including <code>gdev</code> , <code>tdev</code> and <code>sdev</code> columns.          |
| <code>gdevrel</code>   | name of the general relative deviation variable in <code>x</code> .                                                |
| <code>tdevrel</code>   | name of the territorial relative deviation variable in <code>x</code> .                                            |
| <code>sdevrel</code>   | name of the spatial relative deviation variable in <code>x</code> .                                                |
| <code>threshold</code> | defined to build the typology (100 is considered as the average).                                                  |
| <code>superior</code>  | if TRUE, deviation values must be greater than threshold. If FALSE, deviation values must be lower than threshold. |

### Value

a vector in `x` including the `mst` typology. Values are classified in 8 classes following their respective position above/under the threshold:

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

```
# Load data
library(sf)
com <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "com", quiet = TRUE)

# Prerequisite - Compute the 3 deviations
com$gdev <- gdev(x = com, var1 = "INC", var2 = "TH")
com$tdev <- tdev(x = com, var1 = "INC", var2 = "TH", key = "EPT")
com$sdev <- sdev(x = com, var1 = "INC", var2 = "TH", order = 1)

# Multiscalar typology - wealthiest territorial units
# Compute mst
com$mstW <- mst(x = com, gdevrel = "gdev", tdevrel = "tdev", sdevrel = "sdev",
               threshold = 125, superior = TRUE)

#Multiscalar typology - lagging territorial units
# Compute mst
com$mstP <- mst(x = com, gdevrel = "gdev", tdevrel = "tdev", sdevrel = "sdev",
               threshold = 75, superior = FALSE)
```

## 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. #'
- **bidev**: multiscalar typology based on 2 deviations.
- **mst**: multiscalar typology based on the 3 deviations.
- **mas**: multiscalar absolute synthesis, total amount of redistributions based on the three deviations.
- **map\_bidev**: creating bidev and parameters for producing a map based on it.
- **map\_mst**: creating mst and parameters for producing a map based on it.
- **plot\_bidev**: creating a plot for visualizing bidev results.
- **plot\_mst**: creating a plot adapted for visualizing mst results.

## References

GRASLAND C., YSEBAERT R., ZANIN C., LAMBERT N., Spatial disparities in Europe (Chapter 4) in GLOERSEN E., DUBOIS A. (coord.), 2007, Regional disparities and cohesion: What Strategies for the future?, DG-IPOL – European Parliament.

---

plot\_bidev

*Plot Multiscalar Typology (2 deviations)*


---

## Description

Vizualizing bidev and select some territorial units on it.

## Usage

```
plot_bidev(
  x,
  dev1,
  dev2,
  breaks = c(25, 50, 100),
  dev1.lab = NULL,
  dev2.lab = NULL,
  lib.var = NULL,
  lib.val = NULL,
  cex.lab = 1,
  cex.axis = 0.7,
  cex.pt = 0.5,
  cex.names = 0.8,
  pos.names = 4
)
```

## Arguments

|           |                                                                                                                                                         |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| x         | a sf object or a dataframe including 2 pre-calculated deviations.                                                                                       |
| dev1      | column name of the first relative deviation in x.                                                                                                       |
| dev2      | column name of the second relative deviation in x.                                                                                                      |
| breaks    | distance to the index 100 (average of the context), in percentage. A vector of three values. Defaut c(25,50,100). 25 to indexes 80 and 125. 50 and 200. |
| dev1.lab  | label to be put in x-axis of the scatter plot (default: NULL).                                                                                          |
| dev2.lab  | label to be put in y-axis of the scatter plot (default: NULL).                                                                                          |
| lib.var   | column name of x including territorial units name/code we want to display on the plot.                                                                  |
| lib.val   | a vector of territorial units included in lib.label we want to display on the plot.                                                                     |
| cex.lab   | size of the axis label text (default = 1).                                                                                                              |
| cex.axis  | size of the tick label numbers (default = 0.7).                                                                                                         |
| cex.pt    | size of the dot used for extract specific territorial units (default 0.5).                                                                              |
| cex.names | size of the territorial units labels if selected (default 0.8).                                                                                         |
| pos.names | position of territorial units labels (default 4, to the right).                                                                                         |

## Value

A scatter-plot displaying the 13 bidev categories, which are the synthesis of the position of territorial units according to 2 deviations and their respective distance to the average. X-Y axis are expressed in logarithm (25 being index 80).

bidev typology values :

- ZZ: Near the average for the two selected deviation, in grey
- A1: Above the average for dev1 and dev2, distance to the average : +, in light red
- A2: Above the average for dev1 and dev2, distance to the average : ++, in red
- A3: Above the average for dev1 and dev2, distance to the average : +++, in dark red
- B1: Above the average for dev1 and below for dev2, distance to the average : +, in light yellow
- B2: Above the average for dev1 and below for dev2, distance to the average : ++, in yellow
- B3: Above the average for dev1 and below for dev2, distance to the average : +++, in dark yellow
- C1: Below the average for dev1 and dev2, distance to the average : +, in light blue
- C2: Below the average for dev1 and dev2, distance to the average : ++, in blue
- C3: Below the average for dev1 and dev2, distance to the average : +++, in dark blue
- D1: Below the average for dev1 and above for dev2, distance to the average : +, in light green
- D2: Below the average for dev1 and above for dev2, distance to the average : ++, in green
- D3: Below the average for dev1 and above for dev2, distance to the average : +++, in dark green

## Examples

```
# Load data
library(sf)
com <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "com", quiet = TRUE)

# Prerequisite - Compute 2 deviations
com$gdev <- gdev(x = com, var1 = "INC", var2 = "TH")
com$tdev <- tdev(x = com, var1 = "INC", var2 = "TH", key = "EPT")

# EX1 standard breaks with four labels
plot_bidev(x = com,
           dev1 = "gdev",
           dev2 = "tdev",
           dev1.lab = "General deviation (MGP Area)",
           dev2.lab = "Territorial deviation (EPT of belonging)",
           lib.var = "LIBCOM",
           lib.val = c("Marolles-en-Brie", "Suresnes",
                     "Clichy-sous-Bois", "Les Lilas"))

# EX2, change breaks, enlarge breaks
plot_bidev(x = com,
           breaks = c(75, 150, 300),
           dev1 = "gdev",
           dev2 = "tdev",
```

```
dev1.lab = "General deviation (MGP Area)",
dev2.lab = "Territorial deviation (EPT of belonging)"
```

---

plot\_mst

*Plot Multiscalar Typology (3 deviations)*


---

## Description

Vizualizing mst for selected territorial units.

## Usage

```
plot_mst(
  x,
  gdevrel,
  tdevrel,
  sdevrel,
  legend.lab = NULL,
  lib.var,
  lib.val,
  cex.lab = 1,
  cex.axis = 0.8,
  cex.names = 0.8
)
```

## Arguments

|            |                                                                                                                          |
|------------|--------------------------------------------------------------------------------------------------------------------------|
| x          | a sf object or a dataframe including 3 pre-calculated deviations.                                                        |
| gdevrel    | name of the general relative deviation variable in x.                                                                    |
| tdevrel    | name of the territorial relative deviation variable in x.                                                                |
| sdevrel    | name of the the spatial relative deviation variable in x.                                                                |
| legend.lab | label for explaining the plot (default = "G: general, T: territorial, S: spatial (relative deviations, average = 100)"). |
| lib.var    | column name of x including territorial units name/code we want to display on the plot.                                   |
| lib.val    | a vector of territorial units included in lib.label we want to display on the plot.                                      |
| cex.lab    | size of the axis legend label text (default = 1).                                                                        |
| cex.axis   | size of the tick label numbers (default = 0.8).                                                                          |
| cex.names  | size of the territorial units labels if selected (default 0.8).                                                          |

## Value

A barplot displaying the position for selected territorial units on three territorial deviation. Y axis is expressed in logarithm (25 being index 80).



**Examples**

```
# Load data
library(sf)
com <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "com", quiet = TRUE)

# Prerequisite - Compute the 3 relative deviations
com$gdev <- gdev(x = com, var1 = "INC", var2 = "TH")
com$tdev <- tdev(x = com, var1 = "INC", var2 = "TH", key = "EPT")
com$sdev <- sdev(x = com, var1 = "INC", var2 = "TH", order = 1)

# Synthesis barplot (3 territorial units)
plot_mst(x = com, gdevrel = "gdev", tdevrel = "tdev", sdevrel = "sdev", lib.var = "LIBCOM",
         lib.val = c("Neuilly-sur-Seine", "Clichy-sous-Bois", "Les Lilas"))
```

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     | a sf object 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  $\text{var1}/\text{var2}$  and  $\text{var1}/\text{var2}$  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

```
# Load data
library(sf)
com <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "com", quiet = TRUE)
ept <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "ept", quiet = TRUE)
cardist <- read.table(system.file("cardist.txt", package = "MTA"), check.names = FALSE)
cardist <- as.matrix(cardist)

# compute absolute spatial deviation in a neighborhood defined by a contiguity
# order of 1.
com$sdevabs <- sdev(x = com, var1 = "INC", var2 = "TH", order = 1, 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)

if(require(mapsf)){
# relative deviation map
# set breaks
bks <- c(min(com$scardevrel), 75, 100, 125, 150, max(com$scardevrel))
# plot a choropleth map of the relative spatial deviation
mf_map(x = com, var = "scardevrel", type = "choro", leg_pos = "topleft",
       leg_title = "Relative Deviation\n(100 = spatial average)",
       breaks = bks, border = NA,
       pal = c("#91BFDB", "#E0F3F8", "#FEE090", "#FC8D59", "#D73027"))

# add EPT boundaries
mf_map(x = ept, col = NA, add = TRUE)

# layout
mf_layout(title = "Spatial Deviation (neighborhood : 10 minutes by car)",
          credits = paste0("Sources: GEOFLA® 2015 v2.1, Apur, impots.gouv.fr",
```

```

      "\nMTA", packageVersion("MTA")),
    arrow = FALSE)
}

```

---

tdev *Territorial Deviation*

---

## Description

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

## Usage

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

## Arguments

|      |                                                                                                          |
|------|----------------------------------------------------------------------------------------------------------|
| x    | a dataframe or a sf object 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

```

# Load data
library(sf)
com <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "com", quiet = TRUE)
ept <- st_read(system.file("metroparis.gpkg", package = "MTA"), layer = "ept", quiet = TRUE)

# compute absolute territorial deviation
com$tdevabs <- tdev(x = com, var1 = "INC", var2 = "TH", key = "EPT",
  type = "abs")

```

```

# compute relative territorial deviation
com$tdevrel <- tdev(x = com, var1 = "INC", var2 = "TH", key = "EPT",
                  type = "rel")
if(require(mapsf)){
# relative deviation map
# set breaks
bks <- c(min(com$tdevrel), 80, 91, 100, 110, 125, max(com$tdevrel))
# plot a choropleth map of the relative territorial deviation
mf_map(x = com, var = "tdevrel", type = "choro", leg_pos = "topleft",
        leg_title = "Relative Deviation\n(100 = general average)",
        breaks = bks, border = NA,
        pal = c("#4575B4", "#91BFDB", "#E0F3F8", "#FEE090", "#FC8D59", "#D73027"))

# add EPT boundaries
mf_map(x = ept, col = NA, add = TRUE)

# layout
mf_layout(title = "Territorial Deviation (reference: EPT of belonging)",
           credits = paste0("Sources: GEOFLA® 2015 v2.1, Apur, impots.gouv.fr",
                             "\nMTA", packageVersion("MTA")),
           arrow = FALSE)

# absolute deviation map
com$sign <- ifelse(test = com$tdevabs < 0, yes = "Under-Income", no = "Over-Income")
mf_map(ept)

mf_map(x = com, var = c("tdevabs", "sign"), type = "prop_typo", inches = 0.2,
        leg_title = c("Absolute Deviation\n(Income redistribution, euros)",
                      "Redistribution direction"),
        leg_pos = c("topleft", "topright"), leg_val_rnd = -2,
        val_order = c("Under-Income", "Over-Income"),
        pal = c("#ff0000", "#0000ff"), add = TRUE)

# layout
mf_layout(title = "Territorial Deviation (reference: EPT of belonging)",
           credits = paste0("Sources: GEOFLA® 2015 v2.1, Apur, impots.gouv.fr",
                             "\nMTA", packageVersion("MTA")),
           arrow = FALSE)
}

```

# Index

bidev, [2](#), [13](#)

cardist, [3](#)  
com, [3](#), [4](#)

ept, [4](#)

gdev, [5](#), [13](#)

map\_bidev, [6](#), [13](#)  
map\_mst, [8](#), [13](#)  
mas, [11](#), [13](#)  
mst, [12](#), [13](#)  
MTA, [13](#)

plot\_bidev, [13](#), [14](#)  
plot\_mst, [13](#), [16](#)

sdev, [13](#), [17](#)

tdev, [13](#), [19](#)