Package ‘flows’

December 5, 2016

Type Package
Title Flow Selection and Analysis
Version 1.1.1
Date 2016-06-06
Description Selections on flow matrices, statistics on selected flows, map and graph visualisations.
License GPL-3
LazyData TRUE
Depends R (>= 2.10)
Imports reshape2, igraph, sp, graphics, stats
Suggests knitr
URL https://github.com/rCarto/flows
BugReports https://github.com/rCarto/flows/issues
VignetteBuilder knitr
Encoding UTF-8
RoxygenNote 5.0.1
NeedsCompilation no
Author Timothée Giraud [cre, aut],
  Laurent Beauguette [aut],
  Marianne Guérois [ctb]
Maintainer Timothée Giraud <timothee.giraud@ums-riate.fr>
Repository CRAN
Date/Publication 2016-12-05 18:28:47

R topics documented:

  compmat .................................................. 2
domflows .................................................. 3
firstflows .................................................. 4
Comparison of Two Matrices

Description

Compares two matrices of same dimension, with same column and row names.

Usage

```R
compmat(mat1, mat2, digits = 0)
```

Arguments

- `mat1`: A square matrix of flows.
- `mat2`: A square matrix of flows.
- `digits`: An integer indicating the number of decimal places to be used when printing the data.frame in the console (see `round`).

Value

A data.frame that provides statistics on differences between `mat1` and `mat2`: `absdiff` are the absolute differences and `reldiff` are the relative differences (in percent).

See Also

- `statmat`

Examples

```R
# Import data
data(nav)
myflows <- prepflows(mat = nav, i = "i", j = "j", fij = "fij")

# Remove the matrix diagonal
diag(myflows) <- 0

# Select the dominant flows (incoming flows criterion)
```
domflows <- domflows(mat = myflows, w = colSums(myflows), k = 1)
# Select the first flows
flowSel1 <- firstflows(mat = myflows, method = "nfirst", ties.method = "first",
                      k = 1)
# Select flows greater than 2000
flowSel2 <- firstflows(mat = myflows, method = "xfirst", k = 2000)
# Combine selections
flowSel <- myflows * flowSel1 * flowSel2 * flowSel3
# Compare flow matrices
compmat(mat1 = myflows, mat2 = flowSel, digits = 1)

---

**domflows**

*Dominant Flows Selection*

**Description**

Dominant flows selection.

**Usage**

domflows(mat, w, k)

**Arguments**

- **mat**: A square matrix of flows.
- **w**: A vector of units weights (sum of incoming flows, sum of outgoing flows...).
- **k**: A threshold (see 'Details').

**Details**

This function selects which flow (fij or fji) must be kept. If the ratio weight of destination (wj) / weight of origin (wi) is greater than k, then fij is selected and fji is not. This function can perform the second criterion of the Nystuen & Dacey’s dominants flows analysis.

As the output is a boolean matrix, use element-wise multiplication to get flows intensity.

**Value**

A boolean matrix of selected flows.

**References**


**See Also**

firstflows, firstflowsg, plotDomFlows, plotMapDomFlows
Examples

```r
# Import data
data(nav)
myflows <- prepflows(mat = nav, i = "i", j = "j", fij = "fij")

# Remove the matrix diagonal
diag(myflows) <- 0

# Select the dominant flows (incoming flows criterion)
flowSel <- domflows(mat = myflows, w = colSums(myflows), k = 1)
statmat(mat = myflows * flowSel, output = "none")
```

firstflows  

Flow Selection from Origins

Description
Flow selection from origins.

Usage

```
firstflows(mat, method = "nfirst", ties.method = "first", k)
```

Arguments

- `mat` A square matrix of flows.
- `method` A method of flow selection, one of "nfirst", "xfirst" or "xsumfirst":
  - `nfirst` selects the k first flows from origins,
  - `xfirst` selects flows greater than k,
  - `xsumfirst` selects as many flows as necessary for each origin so that their sum is at least equal to k. If k is not reached for one origin, all its flows are selected.
- `ties.method` In case of equality with "nfirst" method, use "random" or "first" (see rank).
- `k` Selection threshold.

Details
As the output is a boolean matrix, use element-wise multiplication to get flows intensity.

Value
A boolean matrix of selected flows.

See Also

`firstflowsg, domflows`
Examples

```r
# Import data
data(nav)
myflows <- prepflows(mat = nav, i = "i", j = "j", fij = "fij")

# Remove the matrix diagonal
diag(myflows) <- 0

# Select the 2 first flows of each origin
flowSel <- firstflows(mat = myflows, method = "nfirst", ties.method = "first", k = 2)
statmat(mat = myflows * flowSel, output = "none")

# Select flows greater than 2000
flowSel <- firstflows(mat = myflows, method = "xfirst", k = 2000)
statmat(mat = myflows * flowSel, output = "none")

# Select as many flows as necessary for each origin so that their sum is at least equal to 20000
flowSel <- firstflows(myflows, method = "xsumfirst", k = 20000)
statmat(mat = myflows * flowSel, output = "none")

# Select each flows that represent at least 10% of the outputs
myflowspct <- myflows / rowSums(myflows) * 100
flowSel <- firstflows(mat = myflowspct, method = "xfirst", k = 10)
statmat(mat = myflows * flowSel, output = "none")
```

firstflowsg  

Flow Selection Based on Global Criteria

Description

Flow selection based on global criteria.

Usage

```r
firstflowsg(mat, method = "nfirst", k, ties.method = "first")
```

Arguments

- **mat**: A square matrix of flows.
- **method**: A method of flow selection, one of "nfirst", "xfirst" or "xsumfirst":
  - nfirst selects the k first flows of the matrix,
  - xfirst selects flows greater than k,
  - xsumfirst selects as many flows as necessary so that their sum is at least equal to k.
- **k**: Selection threshold.
- **ties.method**: In case of equality with "nfirst" method, use "random" or "first" (see rank).
Details

As the output is a boolean matrix, use element-wise multiplication to get flows intensity.

Value

A boolean matrix of selected flows.

See Also

firstflows, domflows

Examples

# Import data
data(nav)
myflows <- prepflows(mat = nav, i = "i", j = "j", fij = "fij")

# Remove the matrix diagonal
diag(myflows) <- 0

# Select the 50 first flows of the matrix
flowSel <- firstflows(mat = myflows, method = "nfirst", ties.method = "first", k = 50)
statmat(mat = myflows * flowSel, output = "none")

# Select all flows greater than 2000
flowSel <- firstflows(mat = myflows, method = "xfirst", k = 2000)
statmat(mat = myflows * flowSel, output = "none")

# Select flows that represent at least 50% of the matrix flows
k50 <- sum(myflows)/2
flowSel <- firstflows(mat = myflows, method = "xsumfirst", k = 150000)
statmat(mat = myflows * flowSel, output = "none")
**GE**

*Grand Est Region*

**Description**

SpatialPolygonsDataFrame of the Grand Est region in France.

**References**

http://professionnels.ign.fr/geofla#tab-3

**Examples**

```r
###GE
data(nav)
sp::plot(GE, col = "#cc3e7", border = "grey50")
```

---

**nav**

*Commuters*

**Description**

Data on commuters between Urban Areas of the French Grand Est region in 2011. Fields:

- i: Code of the urban area of residence
- namei: Name of the urban area of residence
- wi: Total number of active occupied persons in the urban area of residence
- j: Code of the urban area of work
- namej: Name of the urban area of work
- wj: Total number of active occupied persons in the urban area of work
- fij: Number of commuters between i and j

**References**


**Examples**

```r
## nav
data(nav)
str(nav)
```
plotDomFlows

Dominant Flows Graph

Description

This function plots a dominant flows graph.

Usage

\[
\text{plotDomFlows(mat, legend.flows.pos = "topright", legend.flows.title = "Flows Intensity", legend.nodes.pos = "bottomright", legend.node.txt = c("Dominant", "Intermediary", "Dominated", "Size proportional\nto sum of inflows"), labels = FALSE)}
\]

Arguments

- `mat` A square matrix of dominant flows (see \texttt{domflows}).
- `legend.flows.pos` Position of the flows legend, one of "topleft", "top", "topright", "left", "right", "bottomleft", "bottom", "bottomright".
- `legend.flows.title` Title of the flows legend.
- `legend.nodes.pos` Position of the nodes legend, one of "topleft", "top", "topright", "left", "right", "bottomleft", "bottom", "bottomright".
- `legend.node.txt` Text of the nodes legend.
- `labels` A boolean, if TRUE, labels of dominant and intermediary nodes are plotted.

Note

As square matrices can easily be plot with \texttt{plot.igraph} or \texttt{gplot} functions from igraph and sna packages, we do not propose visualisation for other outputs.

See Also

\texttt{domflows, plotMapDomFlows}

Examples

\[
\begin{align*}
# \text{Import data}\\
data(nav)\\
myflows \leftarrow \text{prepflows(mat = nav, i = "i", j = "j", fij = "fij")}\\
# \text{Remove the matrix diagonal}\\
diag(myflows) \leftarrow \emptyset
\end{align*}
\]
# Select the dominant flows (incoming flows criterion)
flowsel1 <- domflows(mat = myflows, w = colSums(myflows), k = 1)
# Select the first flows
flowsel2 <- firstflows(mat = myflows, method = "nfirst", ties.method = "first", k = 1)

# Combine selections
flowsel <- myflows * flowsel1 * flowsel2

# Plot dominant flows graph
plotDomFlows(mat = flowsel, legend.flows.title = "Nb. of commuters")

---

**plotMapDomFlows**

**Dominant Flows Map**

**Description**

This function plots a dominant flows map.

**Usage**

```r
plotMapDomFlows(mat, spdf, spdfid, w, wid, wvar, wcex = 0.05,
legend.flows.pos = "topright", legend.flows.title = "flow intensity",
legend.nodes.pos = "topleft", legend.node.txt = c("Dominant", "Intermediary", "Dominated", "Size proportional to sum of inflows"),
add = FALSE)
```

**Arguments**

- **mat**: A square matrix of dominant flows (see `domflows`).
- **spdf**: A SpatialPolygonsDataFrame or a SpatialPointsDataFrame of units.
- **spdfid**: Name of the unique identifier variable in the spdf data.frame.
- **w**: A data.frame which contains the weight variable used to plot units sizes on the map.
- **wid**: Name of the unique identifier variable in w.
- **wvar**: Name of the weight variable in w.
- **wcex**: Share of the surface of the map occupied by circles (0.02 is 2%).
- **legend.flows.pos**: Position of the flows legend, one of "topleft", "top", "topright", "left", "right", "bottomleft", "bottom", "bottomright".
- **legend.flows.title**: Title of the flows legend.
- **legend.nodes.pos**: Position of the nodes legend, one of "topleft", "top", "topright", "left", "right", "bottomleft", "bottom", "bottomright".
- **legend.node.txt**: Text of the nodes legend.
- **add**: A boolean, if TRUE, add the layer to an existing plot.
See Also
domflows, plotDomFlows

Examples

```r
# Import data
data(nav)
myflows <- prepflows(mat = nav, i = "i", j = "j", fij = "fij")

# Remove the matrix diagonal
diag(myflows) <- 0

# Select the dominant flows (incoming flows criterion)
flowSel1 <- domflows(mat = myflows, w = colSums(myflows), k = 1)
# Select the first flows
flowSel2 <- firstflows(mat = myflows, method = "nfirst", ties.method = "first",
                        k = 1)

# Combine selections
flowSel <- myflows * flowSel1 * flowSel2

# Node weights
inflows <- data.frame(id = colnames(myflows), w = colSums(myflows))

# Plot dominant flows map
opar <- par(mar = c(0, 0, 2, 0))
sp::plot(GE, col = "#ceae7", border = NA)
plotMapDomFlows(mat = flowSel, spdf = UA, spdfid = "ID", w = inflows, wid = "id",
                wvar = "w", wcex = 0.05, add = TRUE,
                legend.flows.pos = "bottomleft",
                legend.flows.title = "Nb. of commuters")
title("Dominant Flows of Commuters")
mtext(text = "INSEE, 2011", side = 4, line = -1, adj = 0.01, cex = 0.8)
par(opar)
```

prepflows  Flows Preparation

Description

From a long format matrix to a wide format matrix.

Usage

prepflows(mat, i, j, fij)
Arguments

mat  A data.frame of flows between origins and destinations: long format matrix (origins, destinations, flows intensity).
i  A character giving the origin field name in mat.
j  A character giving the destination field name in mat.
fij  A character giving the flow field name in mat.

Value

A square matrix of flows. Diagonal can be filled or empty depending on data used.

Examples

# Import data
data(nav)
head(nav)
# Prepare data
myflows <- prepflows(mat = nav, i = "i", j = "j", fij = "fij")
myflows[1:5,1:5]

Description

This function provides various indicators and graphical outputs on a flow matrix.

Usage

statmat(mat, output = "all", verbose = TRUE)

Arguments

mat  A square matrix of flows.
output  Graphical output. Choices are "all" for all graphics, "none" to avoid any graphical output, "degree" for degree distribution, "wdegree" for weighted degree distribution, "lorenz" for Lorenz curve of link weights and "boxplot" for boxplot of link weights (see 'Details').
verbose  A boolean, if TRUE, returns statistics in the console.

Details

Graphical outputs concern outdegrees by default. If the matrix is transposed, outputs concern indegrees.
**Value**

The function returns a list of statistics and may plot graphics.

- **nblinks**: number of cells with values > 0
- **density**: number of links divided by number of possible links (also called gamma index by geographers), loops excluded
- **connectcomp**: number of connected components (isolates included, weakly connected: use of clusters where mode = "weak")
- **connectcompx**: number of connected components (isolates deleted, weakly connected: use of clusters where mode = "weak")
- **sizecomp**: a data.frame of connected components: size and sum of flows per component (isolates included)
- **compocomp**: a data.frame of connected components giving membership of units (isolates included)
- **degrees**: a data.frame of nodes degrees and weighted degrees
- **sumflows**: sum of flows
- **min**: minimum flow
- **Q1**: first quartile of flows
- **median**: median flow
- **Q3**: third quartile of flows
- **max**: maximum flow
- **mean**: mean flow
- **sd**: standard deviation of flows

**See Also**

compmat

**Examples**

```r
# Import data
data(nav)
myflows <- prepflows(mat = nav, i = "i", j = "j", fij = "fij")

# Get statistics and graphs about the matrix
mystats <- statmat(mat = myflows, output = "all", verbose = TRUE)

# Size of connected components
mystats$sizecomp

# Sum of flows
mystats$sumflows

# Plot Lorenz curve only
statmat(mat = myflows, output = "lorenz", verbose = FALSE)
```
# Statistics only
mystats <- statmat(mat = myflows, output = "none", verbose = FALSE)
str(mystats)

---

**Description**


**References**

http://professionnels.ign.fr/geofla#tab-3

**Examples**

```r
# UA
data(nav)
sp::plot(GE, col = "#ccea7", border = "grey50")
sp::plot(UA, col = "#940000", border = "white", add = TRUE)
```
Index

clusters, 12
compmat, 2, 12

domflows, 3, 4, 6, 8–10

firstflows, 3, 4, 6
firstflowsg, 3, 4, 5
flows, 6
flows-package (flows), 6

GE, 7

nav, 7

plot.igraph, 8
plotDomFlows, 3, 8, 10
plotMapDomFlows, 3, 8, 9
prepflows, 10

rank, 4, 5
round, 2

statmat, 2, 11

UA, 13