Package 'flowmapper'

July 22, 2025

Title Draw Flows (Migration, Goods, Money, Information) on 'ggplot2' Plots

Version 0.1.4

Description Adds flow maps to 'ggplot2' plots. The flow maps consist of 'ggplot2' layers which visualize the nodes as circles and the bilateral flows between the nodes as bidirectional half-arrows.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports dplyr, ggplot2, tidyr, forcats, scales, purrr, sfheaders, sf, lifecycle

URL https://github.com/JohMast/flowmapper

BugReports https://github.com/JohMast/flowmapper/issues

Depends R (>= 4.1.0)

LazyData true

NeedsCompilation no

Author Johannes Mast [aut, cre] (ORCID: <https://orcid.org/0000-0001-6595-5834>)

Maintainer Johannes Mast <johannes.mast@dlr.de>

Repository CRAN

Date/Publication 2025-06-11 13:00:13 UTC

Contents

`add_flowmap` *Add a flow map to a ggplot*

Description

Add a flow map to a ggplot

Usage

```
add_flowmap(
  p,
  flowdat = NULL,
  od = NULL,
  nodes = NULL,
  outline_linewidth = 0.01,
  alpha = 0.8,
  nodes_alpha = 0.8,
  outline_col = "black",
  k_nodes = NULL,
  node_buffer_factor = 1.2,
  node_radius_factor = 1,
  edge_offset_factor = 1,
  node_fill_factor = NULL,
  edge_width_factor = 1.2,
  arrow_point_angle = 45,
  add_legend = "none",
  legend_nudge_x = 0,
  legend_nudge_y = 0,
  legend_col = "gray",
  legend_gradient = FALSE
)
```

Arguments

`p`	The plot to which the flowmap should be added.
`flowdat`	Input dataframe. See details below.
`od`	As an alternative to `flowdat`, dataframe with the origin-destination pairs and the flow between them. Must contain the columns o, d, value. `nodes` must be provided as well. See details below.
`nodes`	As an alternative to `flowdat`, a dataframe with the nodes of the network. Must contain the columns name, x, y. See details below.

`outline_linewidth`
: The linewidth of the outline of the arrows.

`alpha`
: Opacity of the edges.

`nodes_alpha`
: Opacity of the nodes.

`outline_col`
: Color of the outline of the edges.

`k_nodes`
: Number of clusters to group nodes into. If defined, nodes will be clustered hierarchically based on spatial proximity. By default, no clustering will be applied.

`node_buffer_factor`
: Controls the distance between the nodes and the edges (in multiple of the nodes' radii).

`node_radius_factor`
: Controls the size of the nodes.

`edge_offset_factor`
: Controls the distance between the parallel arrows.

`node_fill_factor`
: Controls the downscaling of the fill of the nodes (as to not outshine the edges).

`edge_width_factor`
: Controls the width of the edges.

`arrow_point_angle`
: Controls the pointiness of the edges.

`add_legend`
: Add a legend for width to the plot? Must be one of "none","bottom","top","left", or "right". (Experimental)

`legend_nudge_x`
: Adjusts the horizontal position of the legend in map units.

`legend_nudge_y`
: Adjusts the vertical position of the legend in map units.

`legend_col`
: If `add_legend`, sets a monotone color for the legend. By default is "gray".

`legend_gradient`
: If TRUE, the legend color will be a gradient from min to max flow. If FALSE, the legend will be a single color.

Details

The function requires as inputs a dataframe `flowdat` which contains for every combination of two nodes a and b the coordinates of these nodes as well as the intensity of flow between those nodes in both directions (a to b, b to a). The dataframe should have the following columns:

- **id_a:** The unique id of node a
- **id_b:** The unique id of node b
- **xa:** The x coordinate of node a
- **ya:** The y coordinate of node a
- **xb:** The x coordinate of node b
- **yb:** The y coordinate of node b
- **flow_ab:** The intensity of flow from node a to node b
- **flow_ba:** The intensity of flow from node b to node a

Alternatively, the function can take as input a dataframe od which contains the origin-destination pairs and the flow between them. The dataframe should have the following columns:

- **o:** The unique id of the origin node
- **d:** The unique id of the destination node
- **value:** The intensity of flow between the origin and destination

In this case, the function also requires a dataframe nodes which contains the coordinates of the nodes. The dataframe should have the following columns:

- **name:** The unique id of the node
- **x:** The x coordinate of the node
- **y:** The y coordinate of the node

The function will impose coord_equal() on the ggplot.

Inspired by flowmap.gl.

Value

The ggplot with an additional polygon layer for the flow arrows and an additional polygon layer for the nodes

Author(s)

Johannes Mast

Examples

```
testdata <-
data.frame(
 id_a = c("X1","X2","X3","X3","X1"),
 id_b = c("X8","X7","X1","X8","X7"),
 xa = c(2,14,10,10,2),
 ya = c(6,10,9,9,6),
 xb = c(10,4,2,10,4),
 yb = c(4,10,6,4,10),
 flow_ab = c(2,1,1,1,1),
 flow_ba = c(5,1,1,1,2)
)
library(ggplot2)
plot <- ggplot()
plot |> add_flowmap(testdata)
```

add_flowmap_list *Add a list of flow maps to a ggplot, creating a list of plots*

Description

[Experimental]

Usage

```
add_flowmap_list(
  p,
  flowdat = NULL,
  od = NULL,
  nodes = NULL,
  outline_linewidth = 0.01,
  alpha = 0.8,
  nodes_alpha = 0.8,
  outline_col = "black",
  k_nodes = NULL,
  node_buffer_factor = 1.2,
  node_radius_factor = 1,
  edge_offset_factor = 1,
  node_fill_factor = NULL,
  edge_width_factor = 1.2,
  arrow_point_angle = 45,
  add_legend = "none",
  legend_nudge_x = 0,
  legend_nudge_y = 0,
  legend_col = "gray",
  legend_gradient = FALSE
)
```

Arguments

p	The plot to which the flowmap should be added.
flowdat	A list of input dataframes. See details below.
od	As an alternative to flowdat, a list of dataframes with the origin-destination pairs and the flow between them. Must contain the columns o, d, value. nodes must be provided as well. See details below.
nodes	As an alternative to flowdat, a list of dataframes with the nodes of the network. Must contain the columns name, x, y. See details below.
outline_linewidth	The linewidth of the outline of the arrows.
alpha	Opacity of the edges.
nodes_alpha	Opacity of the nodes.

`outline_col` Color of the outline of the edges.

`k_nodes` Number of clusters to group nodes into. If defined, nodes will be clustered hierarchically based on spatial proximity. By default, no clustering will be applied.

`node_buffer_factor`
Controls the distance between the nodes and the edges (in multiple of the nodes' radii).

`node_radius_factor`
Controls the size of the nodes.

`edge_offset_factor`
Controls the distance between the parallel arrows.

`node_fill_factor`
Controls the downscaling of the fill of the nodes (as to not outshine the edges).

`edge_width_factor`
Controls the width of the edges.

`arrow_point_angle`
Controls the pointiness of the edges.

`add_legend` Add a legend for width to the plot? Must be one of "none","bottom","top","left", or "right". (Experimental)

`legend_nudge_x` Adjusts the horizontal position of the legend in map units.

`legend_nudge_y` Adjusts the vertical position of the legend in map units.

`legend_col` If `add_legend`, sets a monotone color for the legend. By default is "gray".

`legend_gradient`
If TRUE, the legend color will be a gradient from min to max flow. If FALSE, the legend will be a single color.

Details

This function creates a list of ggplot objects, each containing a flow map based on the provided data, which match in scales and are thus comparable. It is designed to work with a list of dataframes, where each dataframe represents for example a different day. The function prepares the data for plotting and then adds the flow map to the ggplot object. The function requires as inputs a list of input dataframes `flowdat` which contain for every combination of two nodes a and b the coordinates of these nodes as well as the intensity of flow between those nodes in both directions (a to b, b to a). The dataframe should have the following columns:

- **id_a:** The unique id of node a
- **id_b:** The unique id of node b
- **xa:** The x coordinate of node a
- **ya:** The y coordinate of node a
- **xb:** The x coordinate of node b
- **yb:** The y coordinate of node b
- **flow_ab:** The intensity of flow from node a to node b
- **flow_ba:** The intensity of flow from node b to node a

Alternatively, the function can take as input a list of dataframes `od` which contain the origin-destination pairs and the flow between them. The dataframe should have the following columns:

- **o:** The unique id of the origin node
- **d:** The unique id of the destination node
- **value:** The intensity of flow between the origin and destination

In this case, the function also requires a list of dataframes `nodes` which contain the coordinates of the nodes. The dataframe should have the following columns:

- **name:** The unique id of the node
- **x:** The x coordinate of the node
- **y:** The y coordinate of the node

The function will impose coord_equal() on the ggplot.

Inspired by flowmap.gl.

Value

A list of ggplots, each corresponding to the input ggplot with an additional polygon layer for the flow arrows and an additional polygon layer for the nodes. One output is generated for each input element (in `flowdat`, `od`, or `nodes`).

Author(s)

Johannes Mast

Examples

```
flowdatA <-
data.frame(
  id_a = c("X1","X2","X3","X3","X1"),
  id_b = c("X8","X7","X1","X8","X7"),
  xa = c(2,14,10,10,2),
  ya = c(6,10,9,9,6),
  xb = c(10,4,2,10,4),
  yb = c(4,10,6,4,10),
  flow_ab = c(2,1,1,1,1),
  flow_ba = c(8,1,1,1,2))

flowdatB <-
  data.frame(
    id_a = c("X1","X2","X3","X3","X1"),
    id_b = c("X8","X7","X1","X8","X7"),
    xa = c(2,14,10,10,2),
    ya = c(6,10,9,9,6),
    xb = c(10,4,2,10,4),
    yb = c(4,10,6,4,10),
    flow_ab = c(2,3,2,0.2,1),
    flow_ba = c(3,3,2,1,5))
```

```
flowdatC <-
  data.frame(
    id_a = c("X1","X2","X3","X3","X1"),
    id_b = c("X8","X7","X1","X8","X7"),
    xa = c(2,14,10,10,2),
    ya = c(6,10,9,9,6),
    xb = c(10,4,2,10,4),
    yb = c(4,10,6,4,10),
    flow_ab = c(1,1,2,1,1)/2,
    flow_ba = c(3,3,2,1,5)/3)


list_of_flowdats <- list(flowdatA, flowdatB, flowdatC)

library(ggplot2)
base_plot <-
  ggplot()+
  theme_bw()+
  coord_equal()

flowmap_plots <-
  base_plot |>
  add_flowmap_list(list_of_flowdats,
                   legend_gradient = TRUE,
                   add_legend = "bottom",
                   k_nodes = 3)
```

`cantons` *cantons*

Description

Geometries of Cantons of Switzerland. CRS is unassigned, but should be EPSG:3857.

Usage

```
cantons
```

Format

`cantons`**:**

A sf object with 26 rows and 2 columns:

NAME_1 Name of Canton

geometry polygon coordinates

Source

GADM database https://gadm.org/

`CH_migration_data` *CH_migration_data*

Description

Internal migrations between Cantons of Switzerland, 2011-2016.

Usage

```
CH_migration_data
```

Format

`CH_migration_data`**:**

A data frame with 325 rows and 8 columns:

id_a, id_b Names of Cantons A and B

flow_ab Number of migrations from A to B

flow_ba Number of migrations from B to A

xa,ya Longitude and latitude of the centroid of Canton A. Web-Mercator projection (EPSG: 3857)

xb,yb Longitude and latitude of the centroid of Canton B. Web-Mercator projection (EPSG: 3857)

Source

Federal Statistical Office of Switzerland, under OPEN-BY-ASK terms of use: https://www.bfs.admin.ch/bfs/de/home/statistiken/bevoelkerung/migration-integration/binnenwanderung.assetdetail.3222163.html

---

`flowmap_sf` *Export flow edges and nodes as simple features.*

Description

Export flow edges and nodes as simple features.

Usage

```
flowmap_sf(
  flowdat = NULL,
  od = NULL,
  nodes = NULL,
  k_nodes = NULL,
  node_buffer_factor = 1.2,
  node_fill_factor = NULL,
  node_radius_factor = 1,
  edge_offset_factor = 1,
  edge_width_factor = 1.2,
  arrow_point_angle = 45,
  crs = 4326
)
```

Arguments

`flowdat`	Input dataframe. See details below.
`od`	As an alternative to `flowdat`, dataframe with the origin-destination pairs and the flow between them. Must contain the columns o, d, value. `nodes` must be provided as well. See details below.
`nodes`	As an alternative to `flowdat`, a dataframe with the nodes of the network. Must contain the columns name, x, y. See details below.
`k_nodes`	Number of clusters to group nodes into. If defined, nodes will be clustered hierarchically based on spatial proximity. By default, no clustering will be applied.
`node_buffer_factor`	Controls the distance between the nodes and the edges (in multiple of the nodes' radii).
`node_fill_factor`	Controls the downscaling of the fill of the nodes (as to not outshine the edges).
`node_radius_factor`	Controls the size of the nodes.
`edge_offset_factor`	Controls the distance between the parallel arrows.
`edge_width_factor`	Controls the width of the edges.
`arrow_point_angle`	Controls the pointiness of the edges.
`crs`	The EPSG code for the coordinate reference system of the input data. Default is 4326 (WGS84).

Value

A list with two elements: edges and nodes. Each element is a sf object.

For the edges, contains the following columns:

- **orig:** The unique id of the origin node.
- **dest:** The unique id of the destination node.
- **flow:** The flow from a to b.

And for the nodes:

- **name:** The unique id of the node.
- **flowsum:** The sum of all flows involving that node.

Examples

```
testdata <-
data.frame(
 id_a = c("X1","X2","X3","X3","X1"),
 id_b = c("X8","X7","X1","X8","X7"),
 xa = c(2,14,10,10,2),
 ya = c(6,10,9,9,6),
 xb = c(10,4,2,10,4),
 yb = c(4,10,6,4,10),
 flow_ab = c(2,1,1,1,1),
 flow_ba = c(5,1,1,1,2))
sf_objects <- flowmap_sf(flowdat = testdata,crs=4326)
sf_edges <- sf_objects$edges
sf_nodes <- sf_objects$nodes
```

---

get_circle_coords *Helper function to create coordinates for circles of nodes*

Description

Helper function to create coordinates for circles of nodes

Usage

```
get_circle_coords(center = c(0, 0), r = 1, npoints = 25)
```

Arguments

center	center y and y coordinates
r	radius
npoints	number of points

Value

a dataframe with x and y coordinates of the circle

Author(s)

Johannes Mast, Credit to https://stackoverflow.com/a/6863490

hca_flowdat *Use hierarchical clustering to merge nodes based on proximity*

Description

Use hierarchical clustering to merge nodes based on proximity

Usage

```
hca_flowdat(flowdat, k = 20, return_cluster_assignment = FALSE)
```

Arguments

flowdat	The data containing flows from a to b, b to a, and the coordinates of a and b
k	The number of nodes to keep.
return_cluster_assignment	Instead of an updated flowdat, return a dataframe with the cluster assignment of each node.

Value

a dataframe of the same format as flowdat, but with some nodes (and their flows) merged. Note that this will in most cases contain some circular flows (a to a) even if the input flowdat did not.

prep_flowmap *prep_flowmap*

Description

prep_flowmap

Usage

```
prep_flowmap(
  flowdat = NULL,
  od = NULL,
  nodes = NULL,
  k_nodes = NULL,
  node_buffer_factor = 1.2,
  node_radius_factor = 1,
  edge_offset_factor = 1,
  node_fill_factor = NULL,
  edge_width_factor = 1.2,
  arrow_point_angle = 45
)
```

Arguments

flowdat	Input dataframe. See details below.
od	As an alternative to flowdat, dataframe with the origin-destination pairs and the flow between them. Must contain the columns o, d, value. nodes must be provided as well. See details below.
nodes	As an alternative to flowdat, a dataframe with the nodes of the network. Must contain the columns name, x, y. See details below.
k_nodes	Number of clusters to group nodes into. If defined, nodes will be clustered hierarchically based on spatial proximity. By default, no clustering will be applied.
node_buffer_factor	Controls the distance between the nodes and the edges (in multiple of the nodes' radii).
node_radius_factor	Controls the size of the nodes.
edge_offset_factor	Controls the distance between the parallel arrows.
node_fill_factor	Controls the downscaling of the fill of the nodes (as to not outshine the edges).
edge_width_factor	Controls the width of the edges.
arrow_point_angle	Controls the pointiness of the edges.

Value

A list with two dataframes: edges and nodes. The edges dataframe contains the coordinates of the edges, and the nodes dataframe contains the coordinates of the nodes. Additional list elements contain the maximum and minimum width of the arrows.

Examples

```
testdata <-
data.frame(
 id_a = c("X1","X2","X3","X3","X1"),
 id_b = c("X8","X7","X1","X8","X7"),
 xa = c(2,14,10,10,2),
 ya = c(6,10,9,9,6),
 xb = c(10,4,2,10,4),
 yb = c(4,10,6,4,10),
 flow_ab = c(2,1,1,1,1),
 flow_ba = c(5,1,1,1,2))
flowmapper:::prep_flowmap(testdata)
```

short_scale *Create short scale format for numbers in the legend*

Description

Create short scale format for numbers in the legend

Usage

```
short_scale(x, digits = 3)
```

Arguments

x	The number
digits	Significant digits

Author(s)

Johannes Mast, credit: https://stackoverflow.com/a/59086755

util_data_flow_to_flowdat *util_data_flow_to_flowdat*

Description

This function takes a flow data frame in long format and a data frame with the nodes coordinates and returns a flowdat data frame

Usage

```
util_data_flow_to_flowdat(nodes, flows)
```

Arguments

nodes	A data frame with the nodes of the network
flows	A data frame with the flow data

Details

Helper function to merge od data in long data and nodes to flowdat format

Value

A data frame with the flow data in flowdat format

Author(s)

Johannes Mast,

Examples

```
#nodes <- data.frame(name=c("a","b","c"),x=c(0,1,2),y=c(0,1,2))
#flow <- data.frame(o=c("a","b"),d=c("b","c"),value=c(1,2))
#util_data_flow_to_flowdat(nodes,flow)
```

Index