

# Package ‘sankey’

October 22, 2017

**Title** Illustrate the Flow of Information or Material

**Version** 1.0.2

**Author** Gábor Csárdi, January Weiner

**Maintainer** Gábor Csárdi <csardi.gabor@gmail.com>

**Description** Plots that illustrate the flow of information or material.

**License** GPL (>= 2)

**LazyData** true

**URL** <https://github.com/gaborcsardi/sankey#readme>

**BugReports** <https://github.com/gaborcsardi/sankey/issues>

**Suggests** covr, mockery, testthat

**Imports** simplegraph, utils, graphics, grDevices

**RoxygenNote** 6.0.1

**Encoding** UTF-8

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-10-22 16:45:58 UTC

## R topics documented:

make_sankey . . . . .	2
sankey . . . . .	4
<b>Index</b>	<b>5</b>

---

 make\_sankey

 Create an object that describes a sankey plot
 

---

### Description

Create an object that describes a sankey plot

### Usage

```
make_sankey(nodes = NULL, edges, y = c("optimal", "simple"),
            break_edges = FALSE, gravity = c("center", "top", "bottom"))
```

### Arguments

nodes	A data frame of nodes on the plot, and possibly their visual style. The first column must be the ids of the nodes. If this argument is NULL, then the ids of the nodes are determined from edges.
edges	A data frame of the edges. The first two columns must be node ids, and they define the edges. The rest of the columns contain the visual style of the edges.
y	How to calculate vertical coordinates of nodes, if they are not given in the input. <i>optimal</i> tries to minimize edge crossings, <i>simple</i> simply packs nodes in the order they are given, from bottom to top.
break_edges	Whether to plot each edge as two segments, or a single one. Sometimes two segment plots look better.
gravity	Whether to push the nodes to the top, to the bottom or to the center, within a column.

### Details

The node and edges data frames may contain columns that specify how the plot is created. All parameters have reasonable default values.

Current list of graphical parameters for nodes:

- col Node color.
- size Node size.
- x Horizontal coordinates of the center of the node.
- y Vertical coordinates of the center of the node.
- shape Shape of the node. Possible values: rectangle, point, invisible.
- lty Lite type, see par.
- srt How to rotate the label, see par.
- textcol Label color.
- label Label text. Defaults to node name.
- adjx Horizontal adjustment of the label. See adj in the par manual.

- `adjy` Vertical adjustment of the label. See `adj` in the `par` manual.
- `boxw` Width of the node boxes.
- `cex` Label size multiplication factor.
- `top` Vertical coordinate of the top of the node.
- `center` Vertical coordinate of the center of the node.
- `bottom` Vertical coordinate of the bottom of the node.
- `pos` Position of the text label, see `par`.
- `textx` Horizontal position of the text label.
- `texty` Vertical position of the text label.

Current list of graphical parameters for edges:

- `colorstyle` Whether to use a solid color (`col`), or gradient to plot the edges. The color of a gradient edges is between the colors of the nodes.
- `curvestyle` Edge style, `sin` for sinusoid curves, `line` for straight lines.
- `col` Edge color, for edges with solid colors.
- `weight` Edge weight. Determines the width of the edges.

## Value

A sankey object that can be plotted via the `sankey` function.

## Examples

```
## Function calls in the pkgsnap package:
edges <- read.table(stringsAsFactors = FALSE, textConnection(
"
      get_deps      get_description
      get_deps      parse_deps
      get_deps      %||%
      get_deps      drop_internal
get_description  pkg_from_filename
      parse_deps    str_trim
      cran_file     get_pkg_type
      cran_file     r_minor_version
download_urls    split_pkg_names_versions
download_urls    cran_file
      pkg_download  dir_exists
      pkg_download  download_urls
      pkg_download  filename_from_url
      pkg_download  try_download
      restore       pkg_download
      restore       drop_missing_deps
      restore       install_order
      restore       get_deps
split_pkg_names_versions  data_frame
"))
pkgsnap_sankey <- make_sankey(edges = edges)
sankey(pkgsnap_sankey)
```

```
## Some customization
nodes <- data.frame(
  stringsAsFactors = FALSE,
  id = c("snap", sort(unique(c(edges[,1], edges[,2]))))
)
nodes$col <- ifelse(nodes$id %in% c("snap", "restore"), "orange", "#2ca25f")
edges$colorstyle <- "gradient"

sankey(make_sankey(nodes, edges))
```

---

sankey

*Sankey Diagrams*


---

### Description

Sankey plots illustrate the flow of information or material.

Draw a sankey plot

### Usage

```
## S3 method for class 'sankey'
plot(x, ...)

sankey(x, mar = c(0, 5, 0, 5) + 0.2, ...)
```

### Arguments

x	The plot, created via <a href="#">make_sankey</a> .
...	Additional arguments, ignored currently.
mar	Margin of the plot, see mar in the par manual.

### Value

Nothing.

# Index

`make_sankey`, [2](#), [4](#)

`plot_sankey` (`sankey`), [4](#)

`sankey`, [3](#), [4](#)

`sankey-package` (`sankey`), [4](#)