Package: ggiraphAlluvial (via r-universe)

May 11, 2026

Type Package

Title Interactive Alluvial Geoms for 'ggplot2' Using 'ggiraph'

Version 0.1.1

Description Provides interactive extensions of alluvial geoms from the 'ggalluvial' package for use with 'ggiraph'. The package enables tooltips, hover effects, and clickable elements for alluvial plots created with 'ggplot2', making it easier to explore categorical flow data in interactive visualizations.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports ggplot2, ggalluvial, ggiraph, rlang, tidyselect, dplyr

Suggests knitr, rmarkdown

VignetteBuilder knitr

Config/pak/sysreqs
libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev make libicu-dev libpng-dev libuv1-dev

Repository https://amaendle.r-universe.dev

Date/Publication 2026-03-31 10:46:52 UTC

RemoteUrl https://github.com/amaendle/ggiraphAlluvial

RemoteRef HEAD

RemoteSha c8dbdac6b3db6878b71f0ef2875b1891fead01cf

Contents

`geom_flow_interactive` *Flows between lodes or strata*

Description

This geometry is based on `ggalluvial::geom_flow()`. See the documentation for those functions for more details.

Usage

```
geom_flow_interactive(...)
```

Arguments

`...`	arguments passed to base function, plus any of the interactive_parameters.

See Also

`ggiraph::girafe()`

Examples

```
# add interactive contours to a ggplot -------
library(ggplot2)
library(ggiraph)
#...
#x <- girafe(ggobj = p)
if (interactive()) print(x)
```

`geom_stratum_interactive`
Strata at axes

Description

This geometry is based on `ggalluvial::geom_stratum()`. See the documentation for those functions for more details.

Usage

```
geom_stratum_interactive(...)
```

Arguments

`...`	arguments passed to base function, plus any of the interactive_parameters.

See Also

ggiraph::girafe()

Examples

```
# add interactive contours to a ggplot -------
library(ggplot2)
library(ggiraph)
#...
#x <- girafe(ggobj = p)
if (interactive()) print(x)
```

GeomInteractiveFlow *ggproto class for ggiraph*

Description

ggproto class for ggiraph.

geom_flow_interactive returns a layer that contains a GeomInteractiveFlow object. The GeomFlow object is responsible for rendering the data in the plot.

GeomInteractiveStratum *ggproto class for ggiraph*

Description

ggproto class for ggiraph.

geom_stratum_interactive returns a layer that contains a GeomInteractiveStratum object. The GeomStratum object is responsible for rendering the data in the plot.

`helper functions (previous/following stratum)` *Two helpers to select stratum connected to a flow*

Description

Two helpers to select stratum connected to a flow

Usage

```
stratum_to(x = .data, value = "stratum")

stratum_from(x, value = "stratum")
```

Arguments

x	usually the .data pronoun
value	="stratum", the value that should be returned, e.g. "stratum" or "x"

Value

value mapped to the left (from) or right (to) side of the alluvium.

Examples

```
# stratum_to(.data)
# stratum_from(.data)
```

`self-adjoin` *Adjoin a dataset to itself*

Description

ggproto class for ggiraph.

This function binds a dataset to itself along adjacent pairs of a ‘key‘ variable. It is invoked by [ggalluvial::geom_flow()] to convert data in lodes form to something similar to alluvia form.

In contrast to ggalluvial::self_adjoin it also works with unregular coloumn names, which is necessary when "data-id" is passed to extra_interactive_params in an interactive plot.

‘self_adjoin‘ invokes [‘dplyr::mutate-joins‘] functions in order to convert a dataset with measures along a discrete ‘key‘ variable into a dataset consisting of column bindings of these measures (by any ‘by‘ variables) along adjacent values of ‘key‘.

Usage

```
self_adjoin(
  data,
  key,
  by = NULL,
  link = NULL,
  keep.x = NULL,
  keep.y = NULL,
  suffix = c(".x", ".y")
)
```

Arguments

`data`	A data frame in lodes form (repeated measures data; see [‘alluvial-data‘]).
`key`	Column of ‘data‘ indicating sequential collection; handled as in [tidyr::spread()].
`by`	Character vector of variables to self-adjoin by; passed to [‘dplyr::mutate-joins‘] functions.
`link`	Character vector of variables to adjoin. Will be replaced by pairs of variables suffixed by ‘suffix‘.
`keep.x`, `keep.y`	Character vector of variables to associate with the first (respectively, second) copy of ‘data‘ after adjoining. These variables can overlap with each other but cannot overlap with ‘by‘ or ‘link‘.
`suffix`	Suffixes to add to the adjoined ‘link‘ variables; passed to [‘dplyr::mutate-joins‘] functions.

Examples

```
#
#
```

Index